

Net.Commerce pour NT, AIX et Solaris



Tutoriel East West Food Mart

Version 3.2

Net.Commerce pour NT, AIX et Solaris



Tutoriel East West Food Mart

Version 3.2

Remarque

Avant d'utiliser ce manuel et le produit associé, prenez connaissance des informations générales figurant à la section "Remarques".

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT. IBM DECLINE TOUTE RESPONSABILITE, EXPRESSE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE QUALITE MARCHANDE OU D'ADAPTATION A VOS BESOINS.

Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.ibm.fr> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

© Copyright IBM France, 1999. Tous droits réservés.

© Copyright International Business Machines Corporation 1999. Tous droits réservés.

Table des matières

Chapitre 1. Bienvenue !	1
Ce que vous allez apprendre	1
Connaissances requises	3
Comment imprimer ce tutoriel	3
Comment naviguer dans ce tutoriel	3
Présentation générale de East West Food Mart	4
Caractéristiques de East West Food Mart	5
Devanture de magasin dynamique	5
Affichage dynamique du prix	5
Utilisation de pages de catégorie pour l'affichage des produits	6
Chariot avec informations sur l'expédition	6
Possibilité, pour les clients visiteurs, de passer des commandes	7
Processus de commande simplifié	7
Adresse d'expédition unique	8
Liste d'achats en option	8
Comment modifier les modèles du tutoriel	8
Résolution des incidents	9
 Chapitre 2. Démarrage	11
Configuration du navigateur	11
Ouverture de l'Administrateur Net.Commerce	11
Installation de la base de données du tutoriel	12
 Chapitre 3. Entrée des données du site et du magasin à l'aide de l'Administrateur Net.Commerce	13
Définition du magasin	13
Définition d'une page d'accueil et d'une bannière	14
Entrée des taux de taxe	14
Spécification des transporteurs	15
Attribution des droits d'accès à la base de données	15
Création d'un groupe de contrôle d'accès	15
Désignation d'un administrateur de magasin	16
Affectation de l'administrateur de magasin au groupe de contrôle d'accès	16
Changement de votre mot de passe d'administrateur de site	17
Accès à la base de données en tant qu'administrateur de magasin	17
Entrée des informations relatives au magasin	18
Création de catégories de produits	18
Création des catégories principales	19
Création des sous-catégories	21
Définition des services d'expédition	22
Sélection des services d'expédition	23
Création de codes d'expédition de produit	23
Définition des détails d'un code d'expédition	24
Entrée des informations produit	24
Création du produit	25
Affectation d'un modèle de produit	25
Définition des articles et de leurs attributs	26
 Chapitre 4. Suppression de produits et de catégories	29
Marquage de produits pour suppression	29
Suppression de catégories	29
Suppression des produits préalablement marqués	30

Chapitre 5. Importation des données de produits et de catégories à l'aide de l'utilitaire d'importation globale	31
Chapitre 6. Création de la devanture du magasin	35
Structure de la devanture du magasin	36
Visualisation de la devanture du magasin	37
Visualisation de la page d'accueil	37
Création de la liste des catégories principales	38
Test de la liste de catégories	40
Création de la zone Montant total	41
Test de la zone Montant total	42
Création du cadre de navigation	43
Test du cadre de navigation	45
Comment permettre aux clients non référencés de passer des commandes	46
Test de la création d'entrée d'adresse	48
Chapitre 7. Personnalisation de la page Enregistrement	49
Visualisation de l'état actuel de la page Enregistrement	49
Etude de la macro d'enregistrement	50
Retrait des zones d'informations démographiques de la page Enregistrement	52
Activation de la macro d'enregistrement personnalisée	53
Test de la nouvelle macro d'enregistrement	54
Chapitre 8. Chiffrement des informations d'enregistrement	55
Chapitre 9. Création des pages produit	57
Ajout de la publicité et de la bannière	58
Création du tableau de produits	60
Début de la création du tableau de produits	60
Ajout de l'image et du nom du produit	62
Création de la liste de sélection	62
Ajout de la zone Quantité et de l'icône Liste d'achats	65
Examen du tableau de produits	66
Fin du tableau de produits	67
Création du bouton Ajouter au chariot	68
Examen du bouton Ajouter au chariot	70
Fin de la macro de produit	71
Test de la macro de produit	72
Chapitre 10. Activation de la zone Montant total	75
Calcul du prix total	75
Insertion du fichier d'inclusion dans la macro de produit	76
Examen et test de votre travail	77
Chapitre 11. Création du chariot	79
Visualisation de l'état actuel de la page Chariot	80
Etude de la macro du chariot	80
Etude de la fonction de création du tableau du chariot	80
Ajout de la zone Quantité et du bouton Retirer	83
Récapitulatif des modifications apportées	84
Activation de la macro de chariot personnalisée	84
Test de la nouvelle macro du chariot	85
Chapitre 12. Création d'une fonction non prioritaire	87
Etude des boutons Mettre à jour et Vider le chariot	88
Etude du bouton Mettre à jour	88

Etude du bouton Vider le chariot	89
Classes nécessaires à l'écriture de la fonction	90
Préparation de votre fonction non prioritaire	91
Ecriture de la fonction non prioritaire	92
Ecriture du code permettant de vider le chariot.	94
Ecriture du code de mise à jour des quantités de produits	96
Compilation de la fonction non prioritaire	97
Enregistrement de la fonction non prioritaire dans la base de données	98
Activation de votre nouvelle fonction.	98
Test de la nouvelle fonction non prioritaire	99
Chapitre 13. Etude du processus de commande	101
Affichage de la page Détails	101
Etude de la macro de commande.	102
Etude des fonctions de base	102
Etude de la fonction de création du tableau récapitulatif de la commande	104
Etude des fonctions de création du formulaire d'adresse d'expédition	105
Etude de la fonction de création du formulaire d'adresse réservé aux visiteurs	105
Etude de la fonction de création du formulaire d'adresse réservé aux clients référéncés	107
Etude de la fonction de création du tableau des options de paiement	110
Etude du bouton Valider la commande	111
Etude du bouton Annuler	113
Personnalisation d'une page d'erreur système	113
Suppression d'adresses temporaires	115
Félicitations !	115
Remarques	117

Chapitre 1. Bienvenue !

Bienvenue dans le Tutoriel de East West Food Mart !

Ce tutoriel s'adresse à l'administrateur qui sera chargé de la création et de la gestion d'un magasin Net.Commerce. Il explique en détail comment créer un magasin à partir d'une base de données DB2, à l'aide du Gestionnaire de site et du Gestionnaire de magasin de l'Administrateur, de l'utilitaire d'importation globale et de vos propres macros Net.Data personnalisées. Il décrit également comment utiliser JavaScript pour ajouter des caractéristiques uniques à votre magasin et écrire votre propre fonction non prioritaire personnalisée afin de modifier la méthode de traitement des informations par le système.

Vous allez recréer une version personnalisée de "East West Food Mart", l'un des modèles de magasin fournis avec Net.Commerce, à partir d'une version élémentaire de la base de données, des fichiers HTML, des fichiers macro Net.Data et des fichiers CCP de ce magasin.

Remarques :

- Pour accéder à ce magasin, les clients doivent utiliser un navigateur qui prend en charge le niveau de JavaScript utilisé. C'est notamment le cas de Netscape Communicator version 4.0 ou ultérieure. Tenez-en compte si, après avoir suivi ce tutoriel, vous choisissez d'utiliser East West Food Mart comme point de départ pour créer votre propre magasin.
- Ce tutoriel ne peut fonctionner qu'avec une base de données DB2.

Avant de commencer, lisez les sections suivantes :

- Ce que vous allez apprendre
- Connaissances requises
- Comment imprimer ce tutoriel
- Comment naviguer dans ce tutoriel
- Présentation générale de East West Food Mart
- Caractéristiques de East West Food Mart
- Comment modifier les modèles du tutoriel
- Résolution des incidents
- Base de données Net.Commerce
- Modèles de magasins
- Tables de base de données
- HTML
- macros Net.Data
- DB2 Universal Database
- Fonctions non prioritaires

Ce que vous allez apprendre

Dans ce tutoriel, vous allez apprendre à effectuer les opérations suivantes :

- Utiliser l'Administrateur Net.Commerce pour :
 - créer un enregistrement de magasin et définir une page d'accueil, la devise et les taux de taxe utilisés ;

- entrer les informations relatives aux produits, aux catégories et aux services d'expédition proposés dans le magasin.
- Recourir à l'utilitaire de nettoyage de la base de données pour :
 - supprimer des produits de la base de données ;
 - supprimer des adresses temporaires de client de la base de données.
- Employer l'utilitaire d'importation globale pour importer dans la base de données des informations sur tous les produits et toutes les catégories, à l'aide d'un fichier texte.
- Créer les pages de magasin suivantes :
 - des pages de catégories ;
 - un cadre de navigation permettant aux clients de s'enregistrer, de visualiser le chariot et la liste d'achats (pour les clients référencés) ou de passer une commande ;
 - des pages de produit comportant :
 - une liste de tous les produits appartenant à la catégorie sélectionnée,
 - des listes de sélection qui affichent de manière dynamique les attributs disponibles pour chaque produit,
 - des icônes permettant d'ajouter individuellement des produits à la liste d'achats (disponibles uniquement pour les clients référencés),
 - un bouton **Ajouter au chariot** pour ajouter plusieurs produits et les quantités correspondantes dans le chariot.
 - une page Chariot comportant des zones **Quantité** indiquant les quantités choisies pour chaque article du chariot.
- Personnaliser les pages suivantes :
 - la page Enregistrement, pour éviter aux clients de remplir la partie "informations démographiques",
 - une page d'erreur système, pour qu'elle affiche le nom du magasin et son logo.
- Apprendre à créer une page permettant de passer une commande et comportant :
 - des informations sur l'expédition, les détails de la commande et les données relatives au paiement,
 - un formulaire d'adresse d'expédition qui, selon que le client est visiteur ou référencé, comporte des zones vierges (le client doit alors les compléter) ou contenant déjà les données appropriées.
- Ajouter les caractéristiques suivantes au magasin :
 - une zone d'affichage de prix, automatiquement mise à jour, qui indique le prix total de tous les articles placés dans le chariot,
 - la possibilité, pour les visiteurs, de passer des commandes sans s'enregistrer.
- Créer une fonction non prioritaire qui élargit les fonctionnalités de l'instruction `OrderItemUpdate` pour mettre à jour les quantités des produits dans le chariot ou pour retirer tous les articles du chariot, en fonction du choix du client.
- Administrateur Net.Commerce
- Utilitaire de nettoyage de la base de données
- Utilitaire d'importation globale
- Macros Net.Data
- Fonctions non prioritaires
- Pages de magasin

- Catégories
- Produits
- Expéditions
- Instructions

Connaissances requises

Pour créer et personnaliser les pages du magasin, vous devez connaître les principes de programmation dans les langages suivants :

- HTML (Hypertext Markup Language), pour créer et modifier la présentation des pages de magasin,
- SQL (Structured Query Language), pour écrire des requêtes de base de données dans les macros Net.Data,
- JavaScript, pour ajouter des caractéristiques uniques aux pages de magasin.

En outre, pour créer une fonction non prioritaire, vous devez posséder de solides connaissances du langage de programmation C++.

- HTML
- macros Net.Data
- SQL
- Fonctions non prioritaires

Comment imprimer ce tutoriel

Ce tutoriel est disponible en deux formats imprimables : PostScript et PDF. Pour imprimer un fichier PostScript, vous devez disposer d'une imprimante capable d'imprimer des documents de ce type. Pour imprimer un fichier PDF, vous pouvez utiliser Adobe Acrobat Reader. Si vous ne disposez pas de ce logiciel, vous pouvez le télécharger gratuitement à partir du site Web d'Adobe, à l'adresse suivante :

<http://www.adobe.com>

Les fichiers `ewtut.pdf` et `ewtut.ps` se trouvent dans le répertoire `docs` du CD-ROM et dans le répertoire suivant :

WIN \IBM\netcommerce3\<langue>\books

AIX /usr/lpp/NetCommerce3/<langue>/books

SUN /opt/IBMnetc/NetCommerce3/<langue>/books

Comment naviguer dans ce tutoriel

Ce tutoriel se scinde en plusieurs sections, chacune étant généralement constituée d'une série d'étapes. Respectez l'ordre des sections tel qu'il est présenté dans la table des matières. Utilisez les flèches situées dans le bas de chaque page pour aller à l'étape suivante (flèche vers la droite) ou revenir à la précédente (flèche vers la gauche). Vous avez parfois la possibilité de sauter une section.

Présentation générale de East West Food Mart

East West Food Mart est un exemple de scénario de vente aux particuliers. Ce magasin vend divers produits alimentaires tels que primeurs, boissons, surgelés et viennoiserie. Il est d'une conception plus simple que les autres modèles de magasins Net.Commerce. Certaines instructions étant volontairement évitées, sa structure de navigation est à la fois plus simple et plus directe, si bien que les clients peuvent exécuter rapidement des activités. La plupart des macros utilisent le langage JavaScript pour accentuer l'aspect dynamique des pages du magasin et permettre l'exécution simultanée de plusieurs fonctions.

Pour mieux comprendre la structure de navigation simplifiée de ce magasin, comparez les schémas suivants, qui illustrent le parcours type d'un client dans le centre commercial Le Métropole (un modèle de centre commercial fourni par Net.Commerce) et dans East West Food Mart. Le flux d'instructions Net.Commerce est indiqué sur chaque schéma. Les flèches en pointillés symbolisent les chemins optionnels, tandis que celles en traits continus indiquent les trajets obligatoires.

Schéma n°1 :

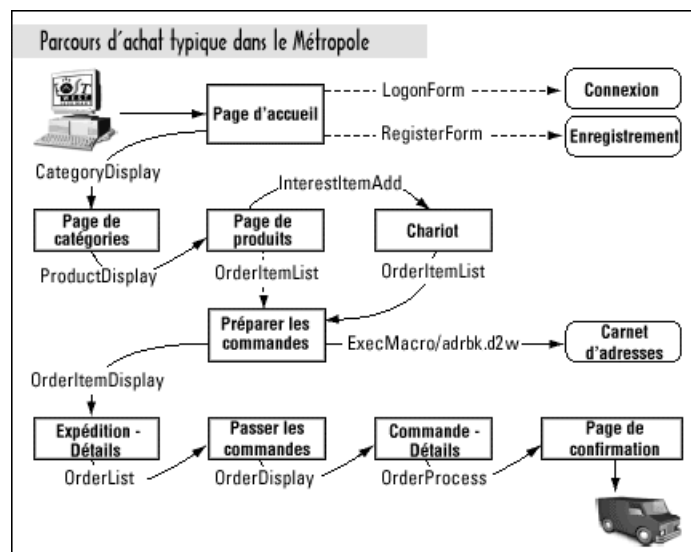
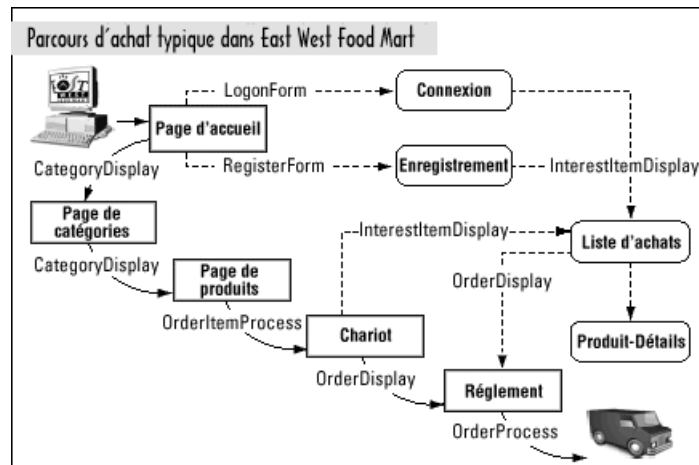


Schéma n°2 :



Notez que le trajet illustré sur le deuxième schéma est à la fois plus court et plus simple que celui du premier. En effet, dans le premier parcours, toutes les instructions disponibles sont utilisées, contrairement au deuxième parcours dans lequel certaines instructions ont été évitées afin de simplifier le processus d'achat.

Caractéristiques de East West Food Mart

Voici les principales caractéristiques du magasin East West Food Mart :

- Devanture de magasin dynamique
- Affichage dynamique du prix
- Utilisation de pages de catégorie pour l’affichage des produits
- Chariot avec informations sur l’expédition
- Possibilité, pour les clients visiteurs, de passer des commandes
- Processus de commande simplifié
- Adresse d’expédition unique
- Liste d’achats en option

Devanture de magasin dynamique

La devanture du magasin est constituée de plusieurs cadres HTML, l'un d'eux affichant une liste de catégories générée dynamiquement. Le client peut accéder directement au catalogue à partir de cette liste.

Techniques de personnalisation

- La devanture est composée d'un ensemble de cadres juxtaposés.
- Une macro de catégories Net.Data est appelée à partir de l'ensemble de cadres pour afficher dynamiquement les catégories de niveau supérieur du magasin.

Affichage dynamique du prix

Le prix total des articles se trouvant dans le chariot est affiché et mis à jour à chaque modification du contenu du chariot. La zone d'affichage de ce prix est visible à partir de n'importe quelle page du magasin.

Technique de personnalisation

On utilise un code JavaScript pour calculer et mettre à jour dynamiquement la zone d'affichage du prix.

Utilisation de pages de catégorie pour l'affichage des produits

Tous les produits appartenant à la même catégorie sont répertoriés sur une même page, accompagnés d'une brève description qui permet de les identifier. A partir de cette page, le client peut :

- préciser les quantités souhaitées ;
- sélectionner les attributs (le cas échéant) ;
- ajouter simultanément plusieurs articles dans le chariot ;
- ajouter des articles à la liste d'achats (si le client est référencé).

Remarque : Comme chaque page contient plusieurs produits, on peut apparenter ce type de pages aux pages de catégorie. Cependant, nous continuerons à les appeler "pages de produit" dans ce tutoriel pour éviter de les confondre avec les pages de catégorie qui font référence à des groupes de catégories et non à des groupes de produits.

Avantages

- Les produits en vente dans notre magasin étant très courants, il n'y a pas lieu de créer une page de produit individuelle pour chaque produit.
- Les clients peuvent effectuer leurs achats plus rapidement, car on leur offre la possibilité d'ajouter plusieurs articles au chariot en une seule opération.
- Les produits apparentés étant regroupés sur une seule page, les clients passent moins de temps à parcourir le catalogue.

Techniques de personnalisation

- L'instruction `CategoryDisplay` est utilisée à la place de l'instruction `ProductDisplay` pour appeler les macros de catégorie servant à l'affichage des produits. Les noms des fichiers macro sont stockés dans la table `CATEGORY`.
- Des constructions et des boucles `Net.Data` sont utilisées pour générer des listes de sélection contenant les attributs de chaque produit.
- Le bouton **Ajouter au chariot** intègre l'instruction `OrderItemProcess`, qui insère plusieurs produits ainsi que leurs quantités respectives dans la table de base de données `SHIPTO`.
- Une nouvelle fonction non prioritaire est créée pour combiner différentes quantités d'un même article en un seul enregistrement de la table `SHIPTO`. On évite ainsi d'accumuler dans cette table les enregistrements comportant les mêmes valeurs dans toutes leurs colonnes à l'exception de la colonne `STQUANT` (quantité).

Chariot avec informations sur l'expédition

Le chariot présente les avantages suivants :

- Il dresse la liste de tous les articles sélectionnés par le client.
- Il indique la quantité choisie pour chaque article (cette donnée étant tirée automatiquement de la page produit où ce choix a été opéré).
- Le client peut à tout moment modifier la quantité de plusieurs articles placés dans le chariot.
- Le client dispose d'un bouton lui permettant de vider le chariot.

Avantages

- Tous les articles du chariot sont considérés comme faisant partie d'une seule et même commande. Ils feront l'objet d'un seul règlement.
- Le client sait que tous les articles placés dans le chariot seront livrés en même temps, au même endroit.

Techniques de personnalisation

- La table SHIPTO est utilisée à la place de la table SHOPPINGS pour enregistrer les produits sélectionnés. De cette manière, les quantités choisies sont stockées et affichées dans le chariot.
- L'utilisation de JavaScript associé à une fonction non prioritaire personnalisée permet d'étendre les possibilités de l'instruction `OrderItemUpdate` de façon que le client puisse à la fois mettre à jour les quantités de plusieurs produits dans la table SHIPTO et supprimer tous les articles de cette table.

Possibilité, pour les clients visiteurs, de passer des commandes

Lorsqu'un client accède pour la première fois à un magasin, il doit généralement préciser ses coordonnées (nom, adresse de livraison, etc.) avant de pouvoir sélectionner des articles et les placer dans son chariot. East West Food Mart est conçu différemment : un nouveau client peut sélectionner des articles et les placer dans le chariot dès son entrée dans le magasin, puis préciser ses coordonnées lorsqu'il a terminé ses achats.

Avantage

Les clients peuvent remplir en toute tranquillité leur chariot sans se préoccuper de l'adresse de livraison des articles choisis.

Technique de personnalisation

Une entrée de carnet d'adresses temporaire est automatiquement créée (à l'aide de l'instruction `AddressAdd`) lorsqu'un client accède au magasin. Si le client s'enregistre ou se connecte ensuite, cette entrée est mise à jour avec l'adresse qui est fournie à ce moment-là. Sinon, sa mise à jour a lieu au moment où le client fournit une adresse pour pouvoir valider sa commande.

Processus de commande simplifié

Toutes les activités liées à la préparation d'une commande ont lieu sur une même page, y compris la vérification des détails de la commande, la spécification d'une adresse de livraison et la validation des données relatives à la carte de paiement. Dans le cas d'un client référencé, le formulaire d'adresse d'expédition comporte déjà les données appropriées, tandis qu'il doit être complété dans le cas d'un client visiteur.

Avantage

Le temps de navigation est considérablement réduit.

Techniques de personnalisation

- Toutes les fonctions et sections "HTML Report" requises pour l'affichage des détails de la commande, de l'adresse de livraison et des options de paiement sont regroupées dans une seule macro.

- Le bouton **Valider la commande** utilise un code JavaScript pour traiter les informations relatives à la commande. Si le client n'est pas référencé, une nouvelle entrée de carnet d'adresses est créée sur la base des informations fournies par le client.
- Réservée à l'usage du commerçant, la colonne SAFIELD2 de la table SHADDR est utilisée pour déterminer si le client est référencé, en fonction de quoi on affiche la version appropriée du formulaire d'adresse d'expédition sur la page de commande (c'est-à-dire une version complétée dans le cas d'un client référencé, ou une version vierge dans le cas d'un nouveau client).

Adresse d'expédition unique

Les clients ne peuvent indiquer qu'une seule adresse de livraison pour leurs commandes. Si un client est référencé, cette adresse est celle qu'il a précisée lors de son enregistrement. Dans le cas contraire, le client doit l'indiquer avant de valider sa commande.

Avantage

En raison de la nature des produits proposés (produits alimentaires), il est peu probable que les clients souhaitent les offrir. Il n'y a donc pas lieu de préciser plusieurs adresses, ni de gérer un carnet d'adresses personnel, ce qui simplifie le processus de règlement.

Technique de personnalisation

On utilise une technique identique à celle permettant aux clients non référencés d'ajouter des articles dans leur chariot avant même de préciser une adresse de livraison : chaque fois qu'un client accède à la page d'accueil, une entrée de carnet d'adresses temporaire est créée. Si le client s'enregistre ou se connecte ensuite, cette entrée est mise à jour. Sinon, sa mise à jour a lieu au moment où le client valide sa commande.

Liste d'achats en option

Chaque client référencé a la possibilité de gérer une liste d'articles qu'il achète régulièrement. Ce peut être le cas de denrées alimentaires à renouveler toutes les semaines. Les articles peuvent être ajoutés à cette liste à partir de n'importe quelle page de produit. Ils y resteront jusqu'à ce que le client décide de les supprimer.

Avantage

Lors de ses prochaines visites, le client pourra se connecter et afficher directement cette liste pour ajouter tout ou partie de son contenu au chariot, sans avoir à rechercher les produits voulus dans le catalogue.

Technique de personnalisation

On a recours à la table SHOPPINGS pour stocker le contenu de la liste d'achats.

Comment modifier les modèles du tutoriel

Ce tutoriel s'accompagne de modèles (fichiers personnalisables) que vous allez personnaliser afin de créer des macros Net.Data et des fichiers HTML pour votre magasin. Tous les modèles sont identifiés par le préfixe tmp_ (par exemple, tmp_shopcart.d2w).

Sauf indication contraire, les modèles se trouvent dans les répertoires suivants :

Fichiers HTML (.htm ou .html) :

```
WIN \ibm\netcommerce3\html\<langue>\tutorial\  
AIX /usr/lpp/NetCommerce3/html/<langue>/tutorial/  
SUN /opt/IBMnetc/NetCommerce3/html/<langue>/tutorial/
```

Macros (.d2w) et fichiers d'inclusion (.inc):

```
WIN \ibm\netcommerce3\macro\<langue>\tutorial\  
AIX /usr/lpp/NetCommerce3/macro/<langue>/tutorial/  
SUN /opt/IBMnetc/NetCommerce3/macro/<langue>/tutorial/
```

Pour vous aider à identifier les endroits où vous devez insérer ou modifier des données dans les modèles, des blocs de commentaire marquant le début et la fin des modifications ont été créés. Ils se présentent comme suit :

```
<!-->  
<!-- Début du code -->  
<!-->  
Insérez votre code ici.  
<!-->  
<!-- Fin du code -->  
<!-->
```

Entrez votre code entre les deux blocs de commentaire, à l'emplacement représenté ci-dessus par la mention **Insérez votre code ici**.

Remarque : Supprimez les blocs de commentaire une fois le code inséré dans le modèle concerné pour que la macro s'exécute normalement.

Dans certains cas, l'insertion de lignes de code entre deux blocs de commentaire s'étend sur plusieurs étapes du tutoriel. Pour vous permettre de distinguer facilement le code déjà inséré de celui qui est à ajouter, le nouveau code apparaît en gras dans les instructions du tutoriel. Par exemple :

```
<!-->  
<!-- Début du code -->  
<!-->  
<TR>  
<TD><A HREF="/cgi-bin/ncommerce3/ExecMacro/t_welcome.d2w/report"  
TARGET="main">  
<IMG SRC="/grocery/bb_log.gif" BORDER=0></A></TD>  
<TD><A HREF="/cgi-bin/ncommerce3/RegisterForm" NAME="register"  
TARGET="main">  
<IMG SRC="/grocery/bb_reg.gif" WIDTH=60 HEIGHT=18 BORDER=0></A></TD>  
</TR>  
<!-->  
<!-- Fin du code -->  
<!-->
```

Après avoir ajouté ou modifié le code requis dans un modèle, vous devrez renommer ce dernier en utilisant le préfixe **t_** pour indiquer qu'il a été traité.

Résolution des incidents

Si, après avoir été modifié conformément aux instructions d'une section de ce tutoriel, un modèle ne s'affiche pas ou ne fonctionne pas correctement dans le magasin, commencez par vérifier que le code source a été entré correctement.

Si l'incident persiste, sachez que vous disposez d'une version finale de chaque modèle (stockée dans le même répertoire que les modèles à modifier). Toutes les versions finales sont identifiées par le préfixe f_. Pour remplacer un fichier en cours de test par sa version finale, supprimez-le et faites une copie de la version finale en lui donnant le nom prévu pour le fichier à tester.

Chapitre 2. Démarrage

Dans cette section, vous allez effectuer les étapes suivantes afin de préparer la création de votre magasin :

- Configuration du navigateur
- Ouverture de l'Administrateur Net.Commerce
- Installation de la base de données

Configuration du navigateur

Vous devez commencer par définir certains paramètres du navigateur pour optimiser les résultats obtenus.

1. Lancez Netscape Communicator version 4.04.
2. Assurez-vous que la fenêtre du navigateur est à sa taille maximale.
3. Dans le menu **Edition**, sélectionnez **Préférences...** pour afficher la boîte de dialogue correspondante.
4. Dans la liste **Catégorie**, cliquez sur **Aspect**.
5. Cliquez sur **Polices**.
6. Sélectionnez **Times New Roman 12 points** comme **Police vectorielle**.
7. Sélectionnez **Courier 10 points** comme **Police à chasse fixe**.
8. Dans la liste **Catégorie**, cliquez sur **Avancées**.
9. Sélectionnez les options **Activer Java** et **Activer JavaScript**.
10. Sélectionnez le bouton d'option **Accepter tous les cookies**.
11. Cochez la case **M'avertir avant d'accepter un cookie**.
12. Cliquez sur **Avancées** puis sur **Cache**.
13. Videz les caches en mémoire et sur disque, et vérifiez que les zones **Cache en mémoire** et **Cache sur disque** sont à zéro.
14. Sous l'option **Le document du cache est comparé à celui sur le réseau**, sélectionnez **Une fois par session**.
15. Cliquez sur **OK** pour enregistrer les paramètres.

Ouverture de l'Administrateur Net.Commerce

Pour ouvrir l'Administrateur Net.Commerce, procédez comme suit :

1. Entrez l'URL suivant sur la ligne de commande de votre navigateur :
http://<nom_hôte>/ncadmin
ou, pour Domino Webserver, tapez : http://<nom_hôte>/ncadmin/login
où <nom_hôte> est le nom que vous avez affecté à votre serveur lors de l'installation de Net.Commerce.
2. Répondez par l'affirmative aux éventuels messages relatifs au certificat de votre site.
3. Si une fenêtre s'affiche en incrustation, acceptez le cookie de session en cliquant sur **OK**. Le serveur Net.Commerce a défini un ID de session temporaire qui est à présent renvoyé à votre navigateur dans un cookie Netscape signé numériquement, afin d'identifier votre session utilisateur. Votre navigateur renverra ce cookie au serveur chaque fois qu'il le sollicitera.
4. Lorsque la page de connexion de l'Administrateur Net.Commerce apparaît, entrez ncadmin, à la fois comme nom d'utilisateur et comme mot de passe, puis

cliquez sur **Connexion**.

Cet ID et ce mot de passe ont été automatiquement créés lors de l'installation de la base de données du tutoriel, ce qui vous permet, en tant qu'administrateur de site, d'accéder à l'Administrateur Net.Commerce et de configurer le magasin.

5. Si une autre fenêtre Netscape apparaît, indiquant que votre cookie de session a été mis à jour en fonction de votre nom d'utilisateur (ncadmin), acceptez le nouveau cookie en cliquant sur **OK**. La page d'accueil de l'Administrateur Net.Commerce s'affiche.
6. Ajoutez cette page à votre liste de signets.

Installation de la base de données du tutoriel

Pour créer la base de données du tutoriel, dans laquelle seront stockées les données de votre version de East West Food Mart, procédez comme suit :

1. Dans la barre des tâches de l'Administrateur Net.Commerce, cliquez sur **Modèles et tutoriel**. Le formulaire correspondant apparaît.
2. Dans la liste déroulante **Sélection du modèle/tutoriel**, sélectionnez **Tutoriel East West Food Mart**. Le nom TUTORIEL est affecté à la base de données et apparaît dans la zone **Nom BD**.
3. Cliquez sur **Installer**. Une fenêtre s'affiche, indiquant la progression de l'installation.
4. Une fois la base de données installée, suivez les instructions à l'écran pour redémarrer le serveur Net.Commerce.

Vous êtes à présent connecté à la base de données du tutoriel.

Remarque : N'utilisez pas cette base de données pour créer votre propre magasin ; elle est exclusivement réservée à une utilisation dans le cadre du tutoriel.

Chapitre 3. Entrée des données du site et du magasin à l'aide de l'Administrateur Net.Commerce

Actuellement, la base de données du tutoriel ne contient aucune donnée sur le magasin ou les produits. Pour entrer ces informations, vous allez utiliser le Gestionnaire de site et le Gestionnaire de magasin de l'Administrateur Net.Commerce.

Dans un magasin indépendant comme celui que vous allez créer, l'administrateur de site peut soit gérer lui-même le site et les données stockées, soit désigner un administrateur de magasin pour entrer et gérer les données du magasin. Tout au long de ce tutoriel, vous aurez le rôle d'administrateur de site. Vous serez donc chargé de la configuration du site et de la création du contenu du magasin. Vous allez également désigner un administrateur de magasin chargé de gérer le magasin, puis accéder au Gestionnaire de magasin en tant qu'administrateur de magasin. Vous aurez donc l'occasion de tenir deux rôles tour à tour : celui d'un administrateur de site et celui d'un administrateur de magasin.

Ce que vous allez faire

Vous allez d'abord mettre en place et configurer le site en tant qu'administrateur de site, puis entrer les informations sur le magasin et les produits en tant qu'administrateur de magasin. Les étapes correspondantes sont les suivantes :

Administration de site :

1. Définition du magasin
2. Spécification de la page d'accueil
3. Saisie des taux de taxe
4. Spécification des transporteurs
5. Affectation des accès à la base de données
6. Changement de votre mot de passe d'administrateur de site

Administration de magasin :

1. Accès à la base de données en tant qu'administrateur de magasin
2. Saisie d'informations supplémentaires sur le magasin
3. Création de catégories de produits
4. Définition des services d'expédition
5. Entrée des informations sur les produits

Définition du magasin

En premier lieu, vous devez définir le magasin en créant un enregistrement de magasin qui comportera des informations générales sur le magasin, c'est-à-dire son nom (enseigne commerciale), le nom de la société (raison sociale) et les coordonnées de la personne à contacter.

1. Dans la barre des tâches de l'Administrateur Net.Commerce, cliquez sur **Gestionnaire de site**, puis sur **Enregistrements de magasin**. Le formulaire correspondant apparaît.
2. Dans la zone **Nom du magasin**, entrez East West Food Mart.

3. Dans les zones appropriées, entrez le nom et le numéro de téléphone d'une société fictive, ainsi que les nom, prénom et numéro de téléphone de la personne désignée comme contact. Les zones qui apparaissent en gras sont obligatoires.
4. Cliquez sur **Sauvegarder**.
5. Cliquez sur **OK** dans la fenêtre de confirmation. Les informations sont stockées comme nouvel enregistrement de magasin dans la table MERCHANT (profil de commerçant) de la base de données. Assurez-vous qu'un message de confirmation apparaît dans la zone d'affichage.

A compter de maintenant, si les informations que vous venez d'entrer doivent être modifiées, vous devrez procéder à cette opération sur le formulaire "Informations magasin et commerçant" du Gestionnaire de magasin. C'est généralement l'administrateur de magasin qui, au besoin, se charge d'apporter ces modifications.

Remarque : Si vous étiez en train de créer un centre commercial, il vous faudrait maintenant répéter cette procédure pour chaque autre magasin.

Définition d'une page d'accueil et d'une bannière

Vous allez à présent définir une page d'accueil et une bannière qui formeront la devanture de votre magasin.

1. Dans la barre des tâches, cliquez sur **Informations du centre commercial** pour afficher le formulaire correspondant.
2. Dans la zone **Page d'accueil**, entrez les informations suivantes :
 \tutorial\index.html
  /tutorial/index.html
Nous reparlerons de ce fichier plus loin.
3. Dans la zone **En-tête**, entrez les informations suivantes :
 \tutorial\banner.html
  /tutorial/banner.html
Cette image constituera la bannière affichée en haut de la page d'accueil.
4. Cliquez sur **Sauvegarder**. Vos sélections sont enregistrées dans la table MALL de la base de données. Assurez-vous qu'un message de confirmation apparaît dans la zone d'affichage.

Entrée des taux de taxe

Vous devez maintenant préciser le taux de taxe à ajouter au prix des produits vendus.

1. Dans la zone **Taux de taxe 1** du formulaire Informations du centre commercial, remplacez la valeur actuelle par 8.00.
Pour passer rapidement d'une zone à l'autre, utilisez la touche de tabulation.
2. Si plusieurs taxes sont applicables (par exemple, une taxe gouvernementale plus une taxe locale), entrez le pourcentage de la deuxième taxe dans la zone **Taux de taxe 2**. Entrez ici la valeur 7.00.
3. Cliquez sur **Sauvegarder**. Ces informations sont ajoutées à la table MALL de la base de données. Assurez-vous qu'un message de confirmation apparaît dans la zone d'affichage.

Spécification des transporteurs

L'administrateur de site crée et tient à jour une liste générale de toutes les sociétés de transport et des types de service proposés aux clients dans ce domaine.

L'administrateur de magasin peut alors sélectionner dans cette liste les sociétés et les types de services qu'il souhaite associer à son magasin. Vous allez maintenant créer cette liste, puis, dans une étape ultérieure, vous choisirez un transporteur pour le magasin.

1. Dans la barre des tâches, cliquez sur **Transporteurs** pour afficher le formulaire correspondant.
2. Dans la liste déroulante **Sélection**, cliquez sur **Ajouter un nouveau transporteur/mode d'expédition**.
3. Dans la zone **Transporteur**, entrez **Livraison express**. Il s'agit du nom d'une société qui offre le service d'expédition souhaité.
4. Dans la zone **Mode d'expédition**, entrez **24 heures** pour définir la classe du service.
5. Cliquez sur **Sauvegarder**. Les informations sont enregistrées dans la table de base de données SHIPMODE (Mode d'expédition du commerçant).
Assurez-vous qu'un message de confirmation apparaît dans la zone d'affichage.

Un seul transporteur est actuellement disponible pour votre magasin. Pour proposer un choix de plusieurs transporteurs, l'administrateur devrait répéter les étapes 2 à 5 pour chaque nouveau transporteur.

Attribution des droits d'accès à la base de données

Pour protéger les données du magasin, vous devez réglementer l'accès à la base de données de celui-ci. En tant qu'administrateur de site, vous avez automatiquement accès aux données du site et aux données du magasin. Toutefois, pour vous aider à gérer les données du magasin, vous allez désigner un administrateur de magasin, qui n'aura accès qu'aux données du magasin.

Ce que vous allez faire

Dans cette section, vous allez effectuer les opérations suivantes :

1. Créer un groupe de contrôle d'accès et l'affecter au magasin.
2. Désigner un administrateur de magasin.
3. Affecter l'administrateur de magasin au groupe de contrôle d'accès.

Création d'un groupe de contrôle d'accès

Avant de désigner un administrateur de magasin, vous devez créer un groupe de contrôle d'accès et l'affecter à votre magasin. Un groupe de contrôle d'accès est fourni avec un ensemble d'instructions qui définissent les droits d'accès dont bénéficient les membres de ce groupe pour une catégorie d'informations spécifique dans la base de données. Les groupes que vous créez peuvent être entièrement nouveaux ou basés sur des rôles prédéfinis.

Dans cette section, vous allez créer un groupe de contrôle d'accès basé sur le rôle d'administrateur de magasin. Ce rôle intègre toutes les instructions requises pour que les personnes concernées puissent accéder aux formulaires du Gestionnaire de magasin et gérer les données du magasin.

1. Dans la barre des tâches, cliquez sur **Groupes d'accès** pour afficher le formulaire d'affectation des groupes d'accès.
2. Dans la liste déroulante **Nom du groupe d'accès (Magasin)**, sélectionnez **Administrateurs de magasin (East West Food Mart)**.
3. Dans la zone **Nom du groupe d'accès**, supprimez le nom du groupe (Administrateurs de magasin) et entrez Admin East West Food Mart.
4. Dans la liste déroulante **Magasin**, sélectionnez **East West Food Mart**.
5. Dans la zone **Description**, tapez Administrateur de magasin pour East West Food Mart.
6. Cliquez sur **Mettre à jour**. Vérifiez qu'un message de confirmation apparaît. Vous venez de créer un groupe d'accès et de l'affecter au magasin.

Désignation d'un administrateur de magasin

Maintenant que vous avez créé un groupe de contrôle d'accès et que vous l'avez affecté au magasin, vous devez désigner un administrateur de magasin, ici Martin Durand.

1. Dans la barre des tâches, cliquez sur **Contrôle des accès** pour afficher le formulaire correspondant.
2. Dans la zone **ID de l'administrateur**, tapez mdurand.
3. Entrez le mot de passe durand dans les deux zones prévues à cet effet (la deuxième saisie sert à confirmer la première).
4. Dans la liste déroulante **Titre**, sélectionnez **M**.
5. Dans la zone **Nom**, indiquez Durand.
6. Dans la zone **Prénom**, indiquez Martin.
7. Cliquez sur **Sauvegarder**.
8. Cliquez sur **OK** dans la fenêtre de confirmation. Ces différentes informations constituent un nouvel enregistrement dont une partie est stockée dans la table SHOPPER de la base de données, et l'autre, dans la table ACCTRL.
 - Contrôle des accès
 - Rôles prédéfinis des contrôles d'accès
 - Administrateurs de site et administrateurs de magasin

Affectation de l'administrateur de magasin au groupe de contrôle d'accès

Vous allez affecter votre nouvel administrateur de magasin, Martin Durand, au groupe de contrôle d'accès créé précédemment. Il aura ainsi le pouvoir de gérer les données du magasin.

1. Dans la barre des tâches, cliquez sur **Affectation des accès** pour afficher le formulaire Administrateur - Affectation de groupes d'accès.
2. Dans la liste déroulante **Tous les groupes** située au bas du formulaire, sélectionnez **Admin East West Food Mart(East West Food Mart)**.
3. Cliquez sur **Ajouter des groupes**. Le nom du magasin doit apparaître dans la liste déroulante **Groupes affectés à l'Administrateur**.
4. Cliquez sur **Sauvegarder** pour mettre à jour la base de données. Vérifiez qu'un message de confirmation apparaît. Martin Durand a désormais l'autorisation de gérer les données du magasin à l'aide du Gestionnaire de magasin.

Changement de votre mot de passe d'administrateur de site

Votre mot de passe d'administrateur de site est actuellement celui qui a été défini par défaut lors de l'installation de la base de données du tutoriel. Vous allez le remplacer par un mot de passe de votre choix, afin d'empêcher toute personne non autorisée d'accéder aux données du site ayant trait au magasin.

1. Dans la barre des tâches, cliquez sur **Contrôle des accès** pour revenir au formulaire correspondant.
2. Cliquez sur **Effacer** pour effacer le contenu de toutes les zones du formulaire.
3. Cliquez sur **Rechercher**. Une liste de tous les administrateurs du magasin apparaît dans la zone d'affichage située au bas du formulaire.
4. Faites défiler cette liste pour faire apparaître les ID de connexion des administrateurs définis, c'est-à-dire vous-même (ncadmin) et Martin Durand (mdurand).
5. Dans la zone d'affichage, sélectionnez votre ID en cliquant sur **ncadmin**, sous la rubrique **ID de connexion Administrateur**.
6. Remplacez votre mot de passe d'administrateur de site. Pour cela, entrez un nouveau mot de passe de votre choix dans les deux zones prévues à cet effet (la seconde saisie confirme la première).
7. Notez en lieu sûr votre nouveau mot de passe ; il vous sera utile tout au long de ce tutoriel.
8. Dans les zones appropriées, indiquez vos nom et prénom. Vous pouvez également renseigner d'autres zones pour vous identifier.
9. Cliquez sur **Sauvegarder**.
10. Cliquez sur **OK** dans la fenêtre de confirmation. Les informations sont enregistrées dans les tables de base de données SHOPPER (Profil du client) et ACCTRL (Contrôle des accès).

Accès à la base de données en tant qu'administrateur de magasin

Vous allez utiliser le composant Gestionnaire de magasin de l'Administrateur Net.Commerce pour entrer toutes les données relatives à votre magasin. Comme nous l'avons vu auparavant, la gestion des données du magasin peut être effectuée soit par l'administrateur de site, soit par l'administrateur de magasin. Ici, vous allez entrer les données en tant qu'administrateur du magasin, Martin Durand. Vous aurez ainsi l'occasion de voir comment la gestion des données peut être répartie entre un administrateur de magasin et un administrateur de site. Cette structure administrative s'avère particulièrement utile dans le cas d'un centre commercial en ligne contenant plusieurs magasins.

Pour accéder à l'Administrateur Net.Commerce en prenant le rôle de l'administrateur de magasin Martin Durand, procédez comme suit :

1. Fermez votre navigateur et rouvrez-le.
2. Ouvrez l'Administrateur Net.Commerce à l'aide du signet que vous avez ajouté lors d'une précédente étape.
3. Si un message vous propose d'affecter un cookie de session au navigateur, acceptez-le en cliquant sur **OK** dans la fenêtre correspondante.
4. Lorsque vous y êtes invité, entrez le nom d'utilisateur et le mot de passe que vous avez affectés à l'administrateur de magasin (nom d'utilisateur : mdurand, mot de passe : durand).
5. Cliquez sur **Connexion**.

6. Si vous y êtes invité, acceptez le nouveau cookie créé pour l'ID de session mdurand.

A présent, vous avez accès au Gestionnaire de magasin de l'Administrateur Net.Commerce, mais pas au Gestionnaire de site. Plus tard, si vous devez mettre à jour les données du site, vous devrez fermer le navigateur et rouvrir l'Administrateur Net.Commerce à l'aide de votre ID de connexion et de votre mot de passe d'administrateur de site.

Entrée des informations relatives au magasin

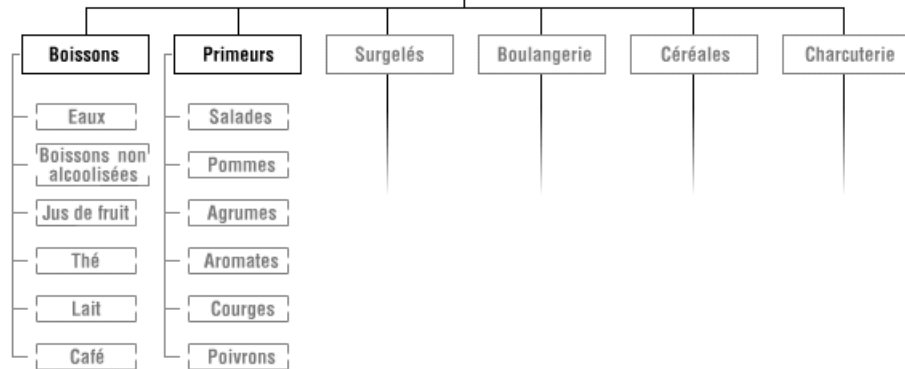
En règle générale, la première activité d'un administrateur de magasin consiste à entrer certaines informations complémentaires sur le magasin, telles que l'unité monétaire (devise) dans laquelle les prix seront indiqués, l'adresse postale de la société dont dépend le magasin et les coordonnées de la personne désignée comme interlocuteur (contact).

1. Dans la barre des tâches, cliquez sur **Gestionnaire de magasin**, puis sur **Informations sur le magasin**. Le formulaire Informations magasin et commerçant s'affiche.
Il est lié au formulaire Enregistrements de magasin du Gestionnaire de site, de sorte que certaines de ses zones comportent déjà les informations que vous avez indiquées précédemment.
2. Dans la zone **Devise**, entrez FRF (Franc français).
3. Faites défiler le formulaire. Dans les zones appropriées de la section **Informations sur l'entreprise du commerçant**, entrez l'adresse (rue), la ville, l'état et le pays où est située l'entreprise. Ces zones sont obligatoires. Complétez les autres zones de votre choix.
4. Cliquez sur **Sauvegarder** pour mettre à jour l'enregistrement de magasin dans la table MERCHANT. Vérifiez qu'un message de confirmation apparaît.
 - Informations sur le magasin

Création de catégories de produits

A l'issue de la configuration de votre magasin et de la définition des services d'expédition, vous pouvez créer les catégories de produits. Elles faciliteront la navigation dans le magasin d'alimentation et permettront aux clients de trouver rapidement les articles désirés. Ces catégories correspondent en quelque sorte aux panneaux d'identification suspendus au-dessus des rayons dans les grandes surfaces.

Dans East West Food Mart, les catégories ne sont divisées qu'en deux niveaux. Dans un souci de simplification, seules deux catégories principales seront étudiées et comporteront des sous-catégories et des produits : Boissons et Primeurs. D'autres catégories ont été prévues, mais elles sont vides et ne seront pas utilisées dans le cadre de ce tutoriel. La structure hiérarchique du magasin est la suivante :



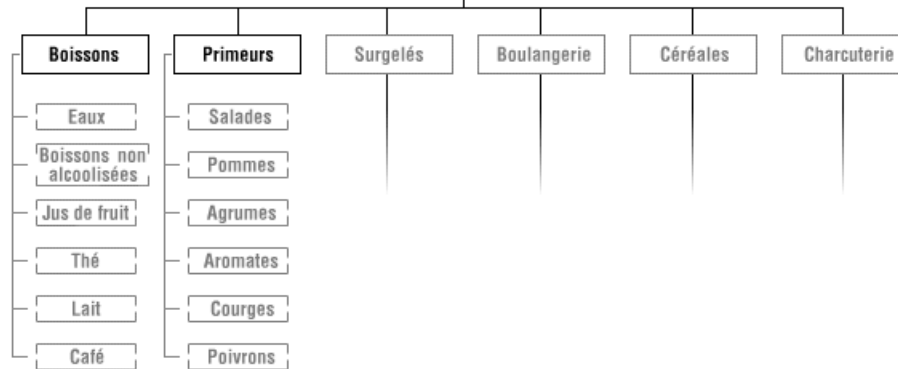
Ce que vous allez faire

Dans cette section, vous allez effectuer les opérations suivantes :

1. Créer deux des catégories de niveau supérieur, à savoir Boissons et Primeurs.
 2. Ajouter la sous-catégorie Boissons non alcoolisées dans la catégorie parent Boissons.
 3. Pour chaque catégorie ou sous-catégorie créée, inclure les informations suivantes :
 - Nom de la catégorie.
 - Numéro d'ordre (position de la catégorie sur la page par rapport aux autres catégories de même niveau).
 - Image miniature.
 - Modèle (fichier macro Net.Data) qui servira à l'affichage de la catégorie. Vous indiquerez un fichier modèle pour les sous-catégories, et un pour les produits appartenant à chaque sous-catégorie.
- Catégories
 - macros Net.Data

Création des catégories principales

Nous allons voir ici comment créer les catégories principales (ou “catégories de niveau supérieur”), intitulées Boissons et Primeurs. Ces catégories, à l’instar de toutes les catégories de niveau supérieur, apparaissent directement sous le nom du magasin, comme ci-après :



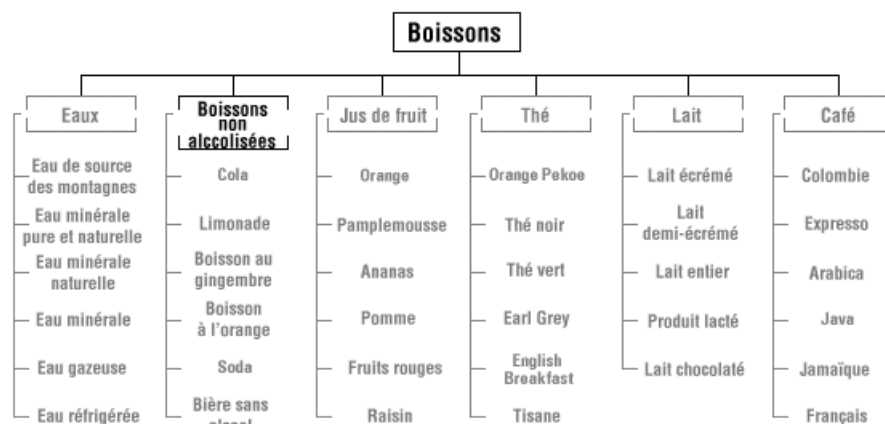
1. Dans la barre des tâches, cliquez sur **Catégories de produits** pour afficher le formulaire correspondant.
2. Vérifiez que **East West Food Mart** est sélectionné dans la liste déroulante **Sélection du magasin**.
Comme vous avez accédé au Gestionnaire de magasin en tant qu'administrateur de magasin, seul East West Food Mart est disponible dans la liste (souvenez-vous que Martin Durand est autorisé à accéder uniquement aux données de ce magasin). C'est dans la zone située sous la liste de sélection que vous allez créer une arborescence de catégories correspondant à la structure de navigation au sein de votre magasin. Elle est actuellement vide et ne contient que le nom du magasin en caractères italiques rouges. Le losange situé en regard indique qu'aucune catégorie enfant n'est encore définie.
3. Dans la barre d'actions située au bas de l'écran, cliquez sur **Ajouter** pour afficher le formulaire Ajouter une catégorie.
4. Dans la zone **Nom**, entrez Boissons.
5. Vérifiez que la zone **Publier** contient la valeur **Oui**. De cette manière, la catégorie sera affichée dès l'ouverture du magasin.
6. Dans la zone **Numéro d'ordre**, entrez 1 pour que la catégorie apparaisse en première position sur la page.
7. Dans la zone **Image miniature**, indiquez c1_n.gif. Il s'agit de l'image miniature qui sera affichée dans le magasin. Le serveur Web recherchera l'image dans le répertoire \html\fr_FR\grocery\.
8. Cliquez sur **Sauvegarder**. Les informations sont enregistrées dans les tables CATEGORY et CGRYREL (relations de catégories) de la base de données. Vous retrouvez ensuite l'arborescence des catégories. Notez que Boissons apparaît maintenant comme catégorie enfant de East West Food Mart.
9. Vous devez à présent spécifier le modèle (macro Net.Data) qui affichera la catégorie. Cliquez sur **Boissons** dans l'arborescence des catégories. Ce nom apparaît en caractères italiques rouges pour indiquer qu'il est sélectionné.
10. Dans la barre d'actions au bas de l'écran, cliquez sur **Modèle** pour afficher le formulaire Affectation de modèle de catégorie.
11. Dans la liste déroulante **Groupe de clients**, sélectionnez **Aucun (modèle par défaut)**.
Avec cette option, le modèle sera valable pour tous les clients. Si vous

décidez de créer des groupes de clients, vous pourriez affecter un modèle différent à chacun d'eux. De cette manière, la présentation de la catégorie serait fonction du groupe dont le client est membre.

12. Dans la zone **Modèle**, entrez `t_catalog.d2w`. C'est le nom de la macro Net.Data dont nous étudierons le contenu plus tard.
13. Cliquez sur **Sauvegarder** pour insérer un nouvel enregistrement dans la table CATESGP (modèle de catégorie pour groupe de clients) de la base de données. Assurez-vous qu'un message de confirmation apparaît dans la zone d'affichage.
14. Cliquez sur **Retourner**, dans le bas de l'écran, pour revenir à l'arborescence des catégories.
15. Dans l'arborescence, cliquez sur **East West Food Mart**.
16. Répétez les étapes 3 à 14 pour créer, sous le nom du magasin, une nouvelle catégorie principale intitulée Primeurs. Donnez-lui les caractéristiques suivantes :
 - **Nom de la catégorie** : Primeurs
 - **Publier** : Oui
 - **Numéro d'ordre** : 2
 - **Image miniature** : `c2_n.gif`
 - **Modèle** : `t_catalog.d2w` (N'oubliez pas de cliquer sur **Primeurs** dans l'arborescence des catégories avant d'affecter le modèle.)
- Catégories
- Groupes de clients
- Table CATEGORY
- Table CGRYREL
- Table CATESGP

Création des sous-catégories

Nous allons voir ici comment créer la sous-catégorie Boissons non alcoolisées, sous la catégorie parent Boissons :



1. Dans l'arborescence des catégories, cliquez sur **Boissons**. Cliquez ensuite sur **Ajouter** dans la barre d'actions située au bas de l'écran.
2. Dans la zone **Nom**, entrez Boissons non alcoolisées.

3. Vérifiez que la zone **Publier** contient la valeur **Oui**.
4. Dans la zone **Numéro d'ordre**, tapez 2 pour indiquer que la sous-catégorie Boissons non alcoolisées apparaîtra en deuxième position sur la page.
5. Dans la zone **Description**, entrez une brève description de la catégorie.
6. Dans la zone **Image miniature**, indiquez `clso_n.gif`.
7. Dans la zone **Zone de personnalisation 1**, tapez `clso_h.gif`.
Cette zone de personnalisation peut être utilisée pour stocker des informations auxquelles ne correspond aucune colonne prédéfinie dans les tables CATEGORY et CGRYREL de la base de données. Nous avons choisi ici de stocker une image bannière pour la page de produits. Cette bannière affichera tous les produits appartenant à la catégorie Boissons non alcoolisées.
8. Cliquez sur **Sauvegarder**.
Les informations sont enregistrées dans les tables CATEGORY et CGRYREL de la base de données. Vous revenez alors à l'arborescence. Les catégories Boissons et Primeurs doivent apparaître au même niveau sous la racine, et la sous-catégorie Boissons non alcoolisées doit apparaître sous Boissons.
9. Il faut à présent définir le modèle (macro Net.Data) qui servira à afficher la sous-catégorie Boissons non alcoolisées. Cliquez sur **Boissons non alcoolisées** dans l'arborescence.
10. Dans la barre d'actions au bas de l'écran, cliquez sur **Modèle** pour afficher le formulaire Affectation de modèle de catégorie.
11. Dans la liste déroulante **Groupe de clients**, sélectionnez **Aucun (modèle par défaut)**.
12. Dans la zone **Modèle**, entrez `t_toc_item.d2w`. Il s'agit du nom d'une macro que vous allez créer par la suite.
13. Cliquez sur **Sauvegarder** pour insérer un nouvel enregistrement dans la table CATESGP de la base de données. Assurez-vous qu'un message de confirmation s'affiche.
14. Cliquez sur **Retourner** pour revenir à l'arborescence. Le numéro de référence de chaque catégorie est généré en interne par le système ; il figure entre parenthèses dans l'arborescence, à côté de chaque nom de catégorie.
15. Notez le numéro de référence des catégories Boissons, Primeurs et Boissons non alcoolisées, car vous en aurez besoin ensuite pour lier les produits à leur catégorie parent.

Définition des services d'expédition

Vous allez maintenant choisir les transporteurs que votre magasin utilisera et définir le mode de livraison de chaque produit. Ici, vous allez affecter un code d'expédition à un produit que vous ne créerez qu'à l'étape suivante. Toutefois, les administrateurs peuvent choisir de créer d'abord les produits, puis les codes d'expédition, et d'affecter ces derniers à chaque produit. L'ordre dans lequel vous effectuez les opérations à partir du Gestionnaire de magasin dépend avant tout de vos besoins et des informations dont vous disposez.

Ce que vous allez faire

Dans cette section, vous allez effectuer les opérations suivantes :

1. Sélectionner les services d'expédition pour le magasin.
2. Créer des codes d'expédition pour les produits.
3. Définir les détails des codes d'expédition (tels que les tranches et les taux) et associer ces codes aux modes d'expédition.

Sélection des services d'expédition

Pour commencer, vous allez sélectionner les services d'expédition utilisés avec East West Food Mart, en procédant comme suit :

1. Dans la barre des tâches, cliquez sur **Services d'expédition** pour afficher le formulaire correspondant.
2. Vérifiez que **East West Food Mart** est sélectionné dans la liste déroulante **Sélection du magasin**.
3. Vérifiez que l'option **Ajouter un nouveau mode d'expédition** est sélectionnée dans la liste déroulante **Mode d'expédition pris en charge**.
4. Vérifiez que l'option **Livraison express - 24 heures** est sélectionnée dans la liste déroulante **Transporteur & Mode d'expédition**.
5. Dans les zones appropriées, indiquez les dates de début et de fin de la période de disponibilité du transporteur pour le magasin. Pour chaque date, utilisez le format `aaaa-mm-dd` ; par exemple, 1998-01-01.
Remarque : Veillez à ce que la date de début soit antérieure ou égale à la date en cours.
6. Cliquez sur **Sauvegarder**. Les informations sont enregistrées comme nouvel enregistrement dans la table MSHIPMODE (Mode d'expédition du commerçant). Le nouveau transporteur doit maintenant apparaître dans la liste **Mode d'expédition pris en charge** ; il sera disponible lorsque vous définirez les détails d'expédition.

Le magasin East West Food Mart n'utilise qu'un transporteur et un seul mode d'expédition. Si l'administrateur de site avait défini d'autres transporteurs, vous pourriez, en tant qu'administrateur du magasin, les choisir tous ou n'en sélectionner que certains.

Création de codes d'expédition de produit

Vous allez à présent créer un code d'expédition qui, associé à chaque produit, déterminera son mode de livraison. Un code d'expédition est un nom ou un numéro qui identifie un ensemble spécifique de détails relatifs à l'expédition, par exemple le transporteur utilisé et le mode de calcul des frais de port.

1. Dans la barre des tâches, cliquez sur **Codes produit** pour afficher le formulaire Codes d'expédition produit.
2. Dans la liste **Code pris en charge**, sélectionnez **Ajouter un nouveau code**.
3. Dans la zone **Code d'expédition produit**, entrez le code de votre choix. Dans cet exemple, entrez FD01 (pour Livraison express - Type 1).
4. Dans la liste déroulante **Méthode de calcul**, sélectionnez **(Q1) Quantité - Tranche - Total**.
Cette méthode calcule les frais de port pour l'ensemble de la commande ; elle applique une somme forfaitaire qui varie suivant la tranche dans laquelle se situe la quantité d'articles commandés. Par exemple, 5,00 FRF pour une commande comprenant de 1 à 5 articles, 10,00 FRF pour une commande de 6 à 10 articles. La livraison d'une commande comprenant 7 articles sera donc facturée 10,00 FRF.
5. Cliquez sur **Sauvegarder** pour enregistrer le code d'expédition dans la liste **Code pris en charge**. Ce code est également stocké comme enregistrement dans la table de base de données PRSPCODE (code d'expédition produit). Assurez-vous qu'un message de confirmation apparaît dans la zone d'affichage.

Un administrateur de magasin peut créer autant de codes d'expédition produit que nécessaire. Pour que notre magasin reste aussi simple que possible, nous avons choisi de n'en créer qu'un. Pour créer des codes supplémentaires, l'administrateur aurait simplement à répéter les étapes 2 à 5 en spécifiant à chaque fois les informations appropriées.

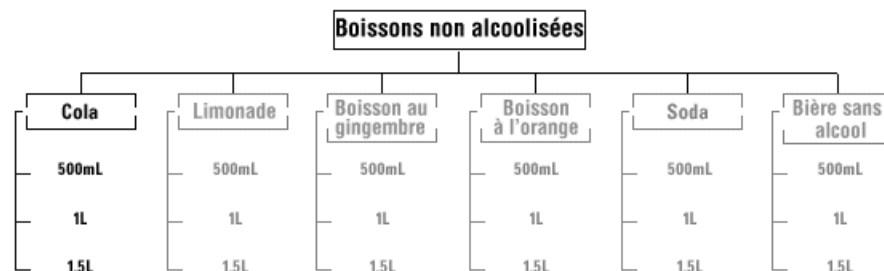
Définition des détails d'un code d'expédition

Vous devez maintenant préciser les modes d'expédition à utiliser pour livrer tous les produits liés au code d'expédition que vous venez de créer. Vous définirez également les tranches requises pour les méthodes de calcul.

1. Dans la barre des tâches, cliquez sur **Expéditions - Détails** pour afficher le formulaire correspondant.
2. Dans la liste déroulante **Code d'expédition produit**, sélectionnez **FD01 (Q1) Quantité - Tranche - Total**.
3. Dans la liste déroulante **Mode d'expédition**, sélectionnez **Livraison express - 24 heures**.
4. Entrez 0 dans la zone **Tranche : borne inférieure** et 21 dans la zone **Tranche : borne supérieure**, pour appliquer le coût choisi aux livraisons ne dépassant pas 20 articles. (Pour la borne supérieure, vous devez indiquer la valeur immédiatement supérieure à celle souhaitée.)
5. Dans la zone **Total (coût total)**, entrez 5,00.
6. Cliquez sur **Créer** pour enregistrer ces informations comme nouvel enregistrement de la table SHIPPING de la base de données. Assurez-vous qu'un message de confirmation apparaît dans la zone d'affichage.
7. Répétez les étapes 4 à 6 pour associer une autre tranche de calcul au code d'expédition FDO1. Utilisez les données suivantes :
 - **Borne inférieure/supérieure** : 21 à 501 (frais de port facturés pour une commande de 21 à 500 articles)
 - **Total (coût total)** : 2.00

Entrée des informations produit

East West Food Mart propose un certain nombre de produits dans chaque sous-catégorie. Parmi ces produits, certains existent en plusieurs tailles. Dans cette section, vous allez ajouter le produit Cola et les différentes contenances disponibles, sous la forme d'articles, sous la sous-catégorie parent Boissons non alcoolisées, comme le montre l'illustration ci-après :



Les informations à entrer seront les suivantes :

- Numéro du produit et référence de chacun des articles associés
- Référence de la catégorie parent
- Code d'expédition
- Nom du produit
- Image miniature
- Articles et leurs attributs, c'est-à-dire les tailles et les prix correspondants
- Modèle (macro Net.Data) pour l'affichage des informations détaillées sur le produit
- Informations produit
- Produits et articles
- macros Net.Data

Création du produit

Pour créer le produit Cola, procédez comme suit :

1. Dans la barre des tâches, cliquez sur **Informations produit** pour afficher le formulaire Informations produit/article.
2. Vérifiez que **East West Food Mart** est sélectionné dans la liste déroulante **Sélection du magasin**.
3. Dans la liste **Entrée pour**, sélectionnez **Produit**.
4. Dans la zone **Numéro du produit ou référence de l'article**, entrez 110. Notez que la zone **Référence produit/article** reste vide. Une fois que vous avez créé et enregistré le produit dans la base de données, le système lui affecte un numéro et complète cette zone.
5. Dans la zone **Référence de la catégorie parent**, entrez le numéro de la catégorie Boissons alcoolisées, celle-ci étant la catégorie parent de notre produit. (Si vous avez oublié ce numéro, cliquez sur **Sélection du parent** pour l'afficher. Fermez ensuite la fenêtre pour revenir au formulaire Informations produit/article.)
6. Dans la liste déroulante **Publier**, sélectionnez **Oui** pour que ce produit soit proposé dans le magasin.
7. Dans la zone **Nom/Description abrégée du produit**, tapez Cola.
8. Dans la zone **Image miniature**, entrez soc111a.gif.
9. Dans la liste déroulante **Code d'expédition produit**, sélectionnez **FD01**.
10. Cliquez sur **Sauvegarder**. Les informations sont sauvegardées sous la forme d'un nouvel enregistrement de la table PRODUCT de la base de données. Vérifiez qu'un message de confirmation apparaît.

Affectation d'un modèle de produit

Vous devez à présent choisir un modèle (macro Net.Data) qui servira à l'affichage, dans le magasin, des informations détaillées sur le produit Cola.

1. Dans la barre des tâches, cliquez sur **Modèles** pour afficher le formulaire Affectation d'un modèle de produit.
2. Dans la liste **Groupe de clients**, sélectionnez **Aucun (modèle par défaut)**.
3. Dans la zone **Modèle**, entrez t_itemdsp.d2w. C'est cette macro qui servira à créer une page de produit.
4. Cliquez sur **Sauvegarder** pour insérer un nouvel enregistrement dans la table PRODSGP (modèle de produit pour groupe de clients) de la base de données. Vérifiez qu'un message de confirmation apparaît.

Définition des articles et de leurs attributs

Vous pouvez maintenant définir les articles correspondant au produit cola, ainsi que les attributs associés. Ce produit est disponible en trois contenances : 500 ml, 1 l et 1,5 l.

1. Cliquez sur **Informations sur le produit** pour revenir au formulaire Informations produit/article.
2. Cliquez sur **Mettre à jour le formulaire** pour que toutes les zones soient renseignées.
3. Notez la valeur de la zone **Référence produit/article**, qui vous servira pour lier les articles à leur produit parent.
4. Dans la liste **Entrée pour**, sélectionnez **Article**.
5. Dans la zone **Numéro du produit ou référence de l'article**, entrez 110-S.
6. Effacez le contenu de la zone **Référence produit/article**. Lorsque vous aurez créé l'article, un nouveau numéro sera automatiquement généré et inséré dans cette zone.
7. Dans la zone **Référence de la catégorie parent**, entrez le numéro noté à l'étape 3.
8. Cliquez sur **Sauvegarder** avant de poursuivre. Les informations sont stockées dans un nouvel enregistrement de la table PRODUCT. Vérifiez qu'un message de confirmation apparaît.
9. Dans la barre des tâches, cliquez sur **Attributs** pour afficher le formulaire Attributs d'article. Le numéro de référence d'article doit déjà être indiqué.
10. Sous la rubrique **Créer/retirer un attribut**, entrez Contenance dans la zone **Nom d'attribut**.
11. En regard de la zone, cliquez sur **Créer**.
12. Sous la rubrique **Définir l'attribut**, entrez 500 ml dans la zone **Valeur d'attribut**.
13. En regard de la zone, cliquez sur **Mettre à jour**. Les informations sont stockées dans un nouvel enregistrement réparti sur les tables de base de données PRODATR (Attribut de produit) et PRODDSTATR (Attribut distinctif de produit). Vérifiez qu'un message de confirmation apparaît.
14. Vous devez maintenant définir le prix de cet article. Dans la barre des tâches, cliquez sur **Prix** pour afficher le formulaire Informations prix.
15. Dans la zone **Prix**, entrez 6,00 et vérifiez que la zone **Devise** est renseignée. (Cette dernière donnée est automatiquement copiée à partir du formulaire Informations magasin et commerçant.)
16. Pour indiquer qu'il s'agit du prix normal, tapez 0 dans la zone **Priorité**. (Vous pourriez spécifier d'autres prix, applicables en période de soldes ; dans ce cas, il faudrait leur affecter une valeur de priorité plus élevée.)
17. Pour que ce prix s'applique à tous les clients, sélectionnez **Aucun** dans la liste **Groupe de clients**.
18. Entrez 01/12/99 comme date d'expiration du prix. N'indiquez pas de date de prise d'effet. Pour celle-ci, le système insérera automatiquement la date en cours dans la base de données.
19. Cliquez sur **Sauvegarder**. Les informations sont stockées dans un nouvel enregistrement de la table PRODPRCS (Prix produit). Assurez-vous qu'un message de confirmation s'affiche.
20. Dans la barre des tâches, sélectionnez **Informations produit** pour revenir au formulaire Informations produit/article.

21. Répétez les étapes 4 à 20 pour créer deux articles supplémentaires associés au produit Cola. Pour ce faire, utilisez les données indiquées ci-après. Il n'est pas utile de redéfinir à chaque fois l'attribut Contenance (étapes 10 et 11). En revanche, vous devez préciser sa valeur.

- **Numéro de référence de l'article** : 110-M
- **Contenance** : 1 l
- **Prix** : 12.00
- **Numéro de référence de l'article** : 110-L
- **Contenance** : 1,5 l
- **Prix** : 18.00

Chapitre 4. Suppression de produits et de catégories

Au cours des étapes précédentes, vous avez défini deux catégories et un produit dans la base de données. Vous avez donc une idée du temps qu'il faudrait si vous deviez répéter ces étapes pour des centaines ou des milliers de produits. Il existe heureusement une méthode nettement plus rapide et efficace, qui consiste à importer en une seule opération toutes les données relatives aux produits et aux catégories dans la base de données du magasin. Une fois les informations dans la base de données, les formulaires de l'Administrateur Net.Commerce sont parfaitement adaptés à la mise à jour de catégories et de produits individuels, dès que cela est nécessaire.

Vous allez maintenant "nettoyer" la base de données, c'est-à-dire supprimer les informations précédemment entrées, pour pouvoir y importer ensuite les données d'un ensemble de catégories et de produits à l'aide de l'utilitaire d'importation globale.

Ce que vous allez faire

Dans cette section, vous allez effectuer les opérations suivantes :

1. Marquer le produit Cola en vue de sa suppression.
2. Supprimer les catégories Boissons, Primeurs et Boissons non alcoolisées.
3. Supprimer le produit Cola préalablement marqué.

Marquage de produits pour suppression

Vous allez marquer le produit Cola en vue de le supprimer. (Vous verrez plus tard comment supprimer définitivement l'enregistrement de ce produit à l'aide de l'utilitaire de nettoyage de la base de données.)

1. Dans la barre des tâches, cliquez sur **Informations produit** pour afficher le formulaire Informations produit/article.
2. Cliquez sur **Effacer** pour effacer le contenu de toutes les zones du formulaire.
3. Dans la liste **Entrée pour**, sélectionnez **Produit**.
4. Cliquez sur **Rechercher**. Le produit apparaît dans la zone d'affichage.
5. Dans cette zone, cliquez sur le numéro de produit **110**. Les zones du formulaire doivent afficher les informations relatives au produit sélectionné.
Dans ce tutoriel, vous n'avez ajouté qu'un produit à la base de données. Toutefois, lorsque vous ajoutez plusieurs produits et que vous souhaitez en supprimer une partie uniquement, vous pouvez entrer des informations dans les zones du formulaire pour identifier les produits à supprimer. Il suffit ensuite de cliquer sur **Rechercher** pour obtenir la liste des produits répondant aux critères de recherche entrés.
6. Dans la liste déroulante **Publier**, sélectionnez **Marquer pour suppression**.
7. Cliquez sur **Sauvegarder**. L'enregistrement du produit est mis à jour dans la table PRODUCT. Vérifiez qu'un message de confirmation apparaît dans la zone d'affichage.

Suppression de catégories

Vous pouvez désormais supprimer les catégories Primeurs, Boissons et Boissons non alcoolisées.

1. Sur la barre des tâches, cliquez sur **Catégories de produit** pour afficher l'arborescence des catégories.
2. Cliquez sur le triangle situé en regard de **East West Food Mart** pour développer l'arborescence.
3. Cliquez sur le triangle situé en regard de **Boissons** pour développer cette catégorie.
4. Cliquez sur **Boissons non alcoolisées**.
5. Cliquez sur **Supprimer**. Un message de confirmation s'affiche.
6. Cliquez sur **OK** pour rendre effective la suppression. Assurez-vous qu'un message de confirmation apparaît dans la zone d'affichage.
7. Répétez les étapes 4 à 6 pour les catégories Boissons et Primeurs.
8. Fermez la fenêtre du navigateur.

Suppression des produits préalablement marqués

L'utilitaire de nettoyage de la base de données Net.Commerce permet de mettre de l'ordre dans la base de données en effectuant des activités telles que l'élimination de produits préalablement marqués pour suppression.

Pour supprimer définitivement le produit Cola de la table de base de données PRODUCT, procédez comme suit :

1. Veillez à utiliser la machine sur laquelle Net.Commerce est installé.
2. A partir d'une ligne de commande quelconque, accédez au répertoire :

WIN \ibm\netcommerce3\bin\

AIX /usr/lpp/NetCommerce3/bin/

SUN /opt/IBMnetc/NetCommerce3/bin/

3. Tapez ce qui suit :

```
ncclean -table product -db tutorial
```

Tous les produits marqués pour suppression sont alors retirés de la base de données du tutoriel (en particulier de la table PRODUCT). Vous pouvez connaître toutes les activités enregistrées par l'utilitaire en affichant le fichier journal dbclog.txt, situé dans le répertoire en cours.

Chapitre 5. Importation des données de produits et de catégories à l'aide de l'utilitaire d'importation globale

Créer individuellement des produits et des catégories à l'aide des formulaires du Gestionnaire de magasin peut vite se révéler fastidieux. Pour aller plus vite, vous pouvez recourir à l'utilitaire d'importation globale. Utilisé conjointement avec un seul fichier texte, cet utilitaire permet d'importer dans la base de données toutes les informations relatives à un grand nombre de produits et de catégories.

1. Préparez un fichier d'importation.

Il s'agit d'un fichier texte dans lequel doivent être insérées les instructions et les données nécessaires au remplissage des tables appropriées de la base de données. Pour vous faire gagner du temps, ce tutoriel s'accompagne d'un fichier d'importation contenant toutes les instructions et données relatives aux produits et aux catégories du magasin East West Food Mart. Il a pour nom `tutorial.in` et se trouve dans le répertoire suivant :

WIN \ibm\netcommerce3\nc_schema\db2\tutorial\

AIX /usr/lpp/NetCommerce3/nc_schema/db2/tutorial/

SUN /opt/IBMnetc/NetCommerce3/nc_schema/db2/tutorial/

2. Ouvrez le fichier `tutorial.in` à l'aide d'un éditeur de texte.

3. Examinez le contenu du fichier.

Chaque ligne commençant par le signe dièse (#) suivi d'un nom d'instruction (par exemple, #PRODUCT) représente une transaction provoquant l'insertion d'une ligne de données dans une table de la base de données. Le nom de cette table est spécifié, ainsi que les valeurs à insérer dans chacune de ses colonnes. Par défaut, le séparateur de colonnes est un point-virgule. Si une colonne doit rester à blanc, il convient néanmoins de marquer son emplacement en insérant le séparateur correspondant. S'il existe une valeur par défaut pour cette colonne, elle sera insérée automatiquement par l'utilitaire. Plusieurs transactions commençant par le même nom d'instruction représentent chacune un enregistrement différent à insérer dans la même table. Toutes les références (par exemple, celles des produits et des catégories) sont générées automatiquement par l'utilitaire.

4. Etudions maintenant quelques transactions de ce fichier d'importation pour identifier les valeurs qui seront insérées et pour déterminer dans quelles colonnes de la base de données elles viendront se placer. Ces transactions contiennent certaines des données que vous avez eu l'occasion d'entrer manuellement à l'aide du Gestionnaire de magasin.

#STORE;East West Food Mart

Cette transaction, située en haut du fichier, indique le nom du magasin pour lequel les données produits et catégories sont créées. Le nom du magasin doit correspondre à celui indiqué dans la colonne MESTNAME de la table MERCHANT. Il s'agit ici de East West Food Mart. (Vous avez entré ce nom à l'aide du formulaire Enregistrements de magasin.)

#CATEGORY;Boissons non

alcoolisées;;;c1so_n.gif;1;c1so_h.gif;ad2.gif;Boissons;2

Cette transaction remplit les tables CATEGORY et CGRYREL (relations entre catégories) en y insérant les données suivantes :

Boissons non alcoolisées : s'insère dans la colonne CGNAME (nom de la catégorie).

c1so_n.gif : s'insère dans la colonne CGTHMB (nom de fichier de l'image miniature).

1 : s'insère dans la colonne CGPUB (la valeur 1 indique que la catégorie sera accessible au public).

c1so_h.gif : s'insère dans la colonne CGFIELD1 (il s'agit d'une zone définie par le commerçant qui est utilisée ici pour désigner le fichier d'image taille réelle représentant la bannière de la page produit).

ad2.gif : s'insère dans la colonne CGFIELD2 (il s'agit d'une zone définie par le commerçant qui est utilisée ici pour désigner le fichier image ad2.gif).

Boissons : nom de la catégorie parent.

2 : s'insère dans la colonne CRSEQNBR (numéro d'ordre de la catégorie) de la table CGRYREL.

#CATESGP;Boissons non alcoolisées;t_toc.d2w;;;;

Cette transaction remplit la table CATESGP (modèle de catégorie pour groupe de clients) en y insérant les données suivantes :

Boissons non alcoolisées : nom de la catégorie.

t_toc.d2w : s'insère dans la colonne CSDISPLAY (macro qui affichera la catégorie).

#PRODUCT;110;;Cola;;;soc1110a.gif;1;;;;;FD01;;;

Cette transaction remplit la table PRODUCT des données suivantes :

110 : s'insère dans la colonne PRNBR (numéro de produit ou référence article).

Cola : s'insère dans la colonne PRDESC (nom du produit).

soc1110a.gif : s'insère dans la colonne PRTHMB (fichier d'image miniature).

1 : s'insère dans la colonne PRPUB (la valeur 1 indique que le produit sera proposé au public).

FD01 : code d'expédition produit tel qu'il est défini dans la colonne PSCODE de la table PRSPCODE.

#PRODDSTATR;110;Contenance

Cette transaction remplit la table PRODDSTATR (attribut distinctif de produit) en y insérant les données suivantes :

110 : numéro du produit tel qu'il est défini dans la table PRODUCT.

Contenance : s'insère dans la colonne PDNAME (nom de l'attribut).

#PRODATR;110-S;Contenance;500 ml

Cette transaction remplit la table PRODATR (attribut de produit) en y insérant les données suivantes :

110-S : numéro du produit ou référence de l'article tels qu'ils sont définis dans la table PRODUCT.

Contenance : s'insère dans la colonne PANAME (partie "nom" de la paire nom-valeur).

500 ml : s'insère dans la colonne PAVAL (partie "valeur" de la paire nom-valeur).

#PRODPRCS;110-S;;6.00;FRF

Cette transaction remplit la table PRODPRCS (prix de produit) en y insérant les données suivantes :

110-S : numéro du produit ou référence de l'article tels qu'ils sont définis dans la table PRODUCT.

6.00 : s'insère dans la colonne PPPRC (prix de l'article).

FRF : s'insère dans la colonne PPCUR (devise dans laquelle est exprimé le prix de l'article, conformément à ce qui est défini dans la table MERCHANT).

#PRODSGP;110;;t_itemdsp.d2w;

Cette transaction remplit la table PRODSGP (modèle de produit pour groupe de clients) en y insérant les données suivantes :

110 :numéro du produit ou référence de l'article tels qu'ils sont définis dans la table PRODUCT.

t_itemdsp.d2w :s'insère dans la colonne PSDISPLAY (macro qui affichera la catégorie).

#CGPRREL;Boissons non alcoolisées;110;1

Cette transaction remplit la tableCGPRREL (relation entre catégorie et produit) en y insérant les données suivantes :

Boissons non alcoolisées :nom de la catégorie.

110 : numéro du produit tel qu'il est défini dans la table PRODUCT.

1 : s'insère dans la colonne CRSEQNBR (numéro d'ordre de la catégorie).

5. Fermez le fichier. Vous allez maintenant exécuter l'utilitaire d'importation globale pour remplir la base de données avec les données contenues dans le fichier d'importation tutorial.in.
6. A partir d'une ligne de commande sur la machine sur laquelle Net.Commerce est installée, accédez au répertoire :

WIN \ibm\netcommerce3\bin\

AIX /usr/lpp/NetCommerce3/bin/

SUN /opt/IBMnetc/NetCommerce3/bin/

7. Entrez la commande suivante sur une seule ligne :

WIN massimpt -infile \ibm\netcommerce3\nc_schema\db2\tutorial\tutorial.in
-db tutorial -log \ibm\netcommerce3\nc_schema\db2\tutorial\tutmsg.log

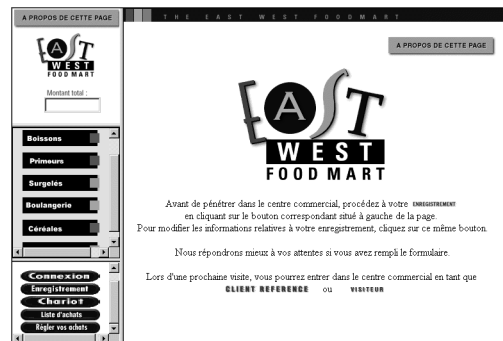
AIX massimpt -infile
/usr/lpp/NetCommerce3/nc_schema/db2/tutorial/tutorial.in
-db tutorial -log
/usr/lpp/NetCommerce3/nc_schema/db2/tutorial/tutmsg.log

SUN massimpt - infile
/opt/IBMnetc/NetCommerce3/nc_schema/db2/tutorial/tutorial.in
-db tutorial -log
/opt/IBMnetc/NetCommerce3/nc_schema/db2/tutorial/tutmsg.log

Vous pouvez afficher le contenu du fichier tutmsg.log et consulter l'enregistrement répertoriant toutes les transactions ayant abouti. Toute erreur survenue pendant l'importation des données est également consignée dans ce fichier journal.

Chapitre 6. Création de la devanture du magasin

Vous êtes prêt à créer la devanture du magasin East West Food Mart. C'est la première page que les clients voient s'afficher lorsqu'ils accèdent au magasin. Elle leur souhaite la bienvenue et les invite à s'enregistrer dans le magasin. Elle contient également des liens hypertexte aux principales catégories de produits du magasin et aux fonctions essentielles telles que l'enregistrement, la liste d'achats, le chariot et le règlement des achats.



Contrairement à la plupart des devantures de magasin, qui sont des fichiers HTML statiques, celle-ci contient dans un de ses cadres une macro de catégorie affichant dynamiquement la liste des catégories de niveau supérieur du magasin, chacune de ces catégories proposant un lien aux sous-catégories correspondantes.

Ce que vous allez faire

Dans cette section, vous allez effectuer les opérations suivantes :

- Etudier la structure de la devanture du magasin.
- Voir dans quel état se trouve la devanture actuellement.
- Visualiser la page d'accueil.
- Créer une macro de catégorie simple pour afficher les catégories de niveau supérieur du magasin.
- Créer la zone permettant de toujours afficher le prix total des articles contenus dans le chariot.
- Créer le cadre de navigation.
- Donner la possibilité aux clients de passer des commandes sans s'enregistrer.

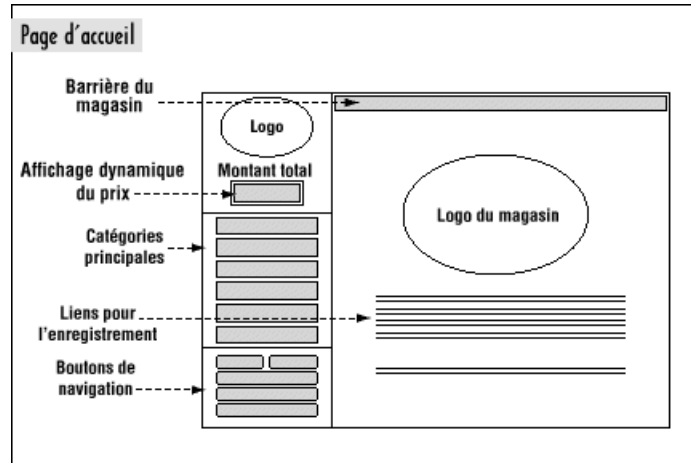
Pour passer cette section

Si vous choisissez de ne pas suivre cette section, vous devez effectuer les opérations suivantes :

1. Effacez le fichier `t_navmid.d2w` et faites une copie de `f_navmid.d2w` sous le nom `t_navmid.d2w`.
2. Effacez le fichier `t_navtop.html` et faites une copie de `f_navtop.html` sous le nom `t_navtop.html`.
3. Effacez le fichier `t_navbot.html` et faites une copie de `f_navbot.html` sous le nom `t_navbot.html`.
4. Effacez le fichier `index.html` et faites une copie de `f_index.html` sous le nom `index.html`.
5. Allez à la section Personnalisation de la page d'enregistrement.

Structure de la devanture du magasin

La page d'accueil du magasin East West Food Mart est constituée des cinq cadres HTML suivants :



Les trois cadres de gauche sont toujours affichés, si bien que le client bénéficie d'un accès permanent aux fonctions principales ainsi qu'aux catégories de niveau supérieur. Le contenu du cadre principal occupant la partie droite de la page varie selon la fonction qui a été demandée par le client. Les fichiers constituant la devanture du magasin sont les suivants :

index.html

Ce fichier constitue la structure de la devanture. Il définit la taille et l'emplacement des différents cadres sur la page et spécifie les fichiers à charger dans chacun d'eux.

t_home.html

Contenu de la page d'accueil, dans le cadre principal de droite. Il contient un fichier GIF animé représentant le logo du magasin et des liens aux fonctions d'enregistrement dans le magasin, de mise à jour de l'enregistrement et de connexion. Ce cadre est remplacé par une nouvelle page chaque fois que le client accède à une autre fonction du magasin.

t_banner.html

Contient l'image (fichier GIF) de la bannière disposée horizontalement, dans la partie supérieure du cadre principal. (La bordure du cadre réservé à la bannière est masquée.)

t_navtop.html

Contenu du cadre supérieur gauche. Il contient une zone affichant en permanence le prix total des articles placés dans le chariot.

t_navmid.d2w

Contenu du cadre intermédiaire gauche. Il s'agit d'une macro qui interroge la base de données pour afficher les catégories de niveau supérieur du magasin (l'affichage est dynamique et reflète en permanence la structure du magasin). Chaque catégorie est un lien aux sous-catégories correspondantes.

t_navbot.html

Contenu du cadre de navigation. Il est constitué d'images GIF qui établissent chacune un lien à l'une des cinq fonctions principales du

magasin : connexion (pour les clients référencés), enregistrement (pour les nouveaux clients), chariot, liste d'achats et règlement des achats.

Visualisation de la devanture du magasin

Voyons à quoi ressemble actuellement la devanture du magasin :

1. Assurez-vous que le navigateur dans lequel l'Administrateur Net.Commerce était affiché est fermé.
2. Ouvrez une nouvelle fenêtre de navigateur et entrez l'URL suivant :
`http://<nom_hôte>/tutorial/index.html`
où <nom_hôte> est le nom complet qualifié de votre serveur.
3. Si vous y êtes invité, acceptez le cookie de session temporaire qui vous est affecté en cliquant sur **OK** dans la fenêtre. Le serveur a été sollicité et a affecté un ID de session temporaire qui est maintenant renvoyé à votre navigateur dans un cookie Netscape signé numériquement afin d'identifier votre session utilisateur.
Remarque : Cette fenêtre ne s'affiche que si le client a configuré son navigateur pour avoir le choix d'accepter ou de refuser les cookies. Lors d'une précédente étape de ce tutoriel, il vous a été demandé de configurer de la sorte votre navigateur, cela dans le seul but de vous montrer à quels moments des cookies sont envoyés par le serveur pour identifier les sessions des clients. La devanture apparaît. Les trois cadres de gauche sont vides. C'est à vous de créer leur contenu.
4. Ajoutez cette page à votre liste de signets.

Visualisation de la page d'accueil

Voyons de plus près la page d'accueil affichée dans le cadre principal de la devanture du magasin. Cette page comporte des liens permettant aux clients de s'enregistrer, ou de se connecter pour ceux qui sont déjà référencés. Les clients peuvent parcourir le magasin sans s'enregistrer. (Vous verrez plus tard comment un client peut effectuer des achats dans le magasin sans être référencé.) Toutefois, l'enregistrement présente des avantages : les clients référencés n'ont pas besoin d'entrer leur adresse de livraison à chaque commande passée, ils peuvent constituer et tenir à jour une liste d'achats personnelle et ont accès à toutes les promotions.

Etudions les liens de la page d'accueil.

1. Placez le curseur sur le premier lien, **ENREGISTREMENT**, juste en dessous du logo du magasin. Observez l'instruction affichée dans la barre d'état, en bas de la fenêtre :
`/cgi-bin/ncommerce3/RegisterForm`
Si le client est référencé, l'instruction RegisterForm affiche la page d'enregistrement qui lui correspond ; sinon, elle affiche une page d'enregistrement vierge.
2. Placez le curseur sur le lien **CONNEXION**. Observez l'instruction correspondante (affichée sur une seule ligne) :
`/cgi-bin/ncommerce3/LogonForm?url=/cgi-bin/ncommerce3/ExecMacro/tutorial/t_welcome1.d2w/report`
L'instruction LogonForm affiche une page de connexion à partir de laquelle un client référencé peut entrer un ID utilisateur et un mot de passe pour se connecter au magasin. Lorsque la connexion est établie, une page de bienvenue (créée par la macro t_welcome1.d2w) s'affiche. Elle contient un lien à

la promotion du jour. Cette macro transfère également, dans le chariot associé au nom d'utilisateur spécifié lors de la connexion, tous les articles que le client a ajouté à son chariot avant de se connecter.

Création de la liste des catégories principales

Vous allez créer une macro simple qui affichera dynamiquement les catégories de niveau supérieur dans le cadre intermédiaire gauche de la devanture. Elle peut être créée à l'aide d'un éditeur de texte classique ou à l'aide de l'Editeur de modèles. Dans le présent tutoriel, vous allez faire appel à un éditeur de texte. Pour vous permettre de démarrer, un modèle de macro Net.Data est à votre disposition. Il comporte déjà les marques de bloc Net.Data appropriées. Dans chaque bloc, vous allez ajouter les instructions HTML, SQL et Net.Data nécessaires à la mise en oeuvre de la macro.

1. Pour consulter une présentation des macros Net.Data, reportez-vous à la rubrique Macros Net.Data.
2. Assurez-vous que vous avez pris connaissance des instructions relatives à l'édition des modèles du tutoriel.
3. A l'aide d'un éditeur de texte, ouvrez le modèle de tutoriel tmp_navmid.d2w.
4. Examinez le contenu du fichier. Il comporte uniquement des sections Net.Data vides : une section Define, pour la définition de toutes les variables internes à utiliser dans la macro ; une section Function Definition, qui servira à interroger la base de données Net.Commerce en vue d'obtenir les informations relatives aux catégories appropriées et à afficher les lignes extraites ; une section HTML Report pour afficher les informations dans un cadre de page Web.
5. Vous devez d'abord définir le nom de la macro. Entrez l'instruction suivante dans le bloc %DEFINE (elle apparaît en gras ci-après) :

```
%DEFINE
{
  <!-- Début du code -->
  <!-- Début du code -->
  <!-- Début du code -->
  macroName="t_navmid.d2w"
  <!-- Début du code -->
  <!-- Fin du code -->
  <!-- Fin du code --> < ?XPP:break>
%}
```

6. Ensuite, vous devez créer une fonction qui exécutera une requête SQL pour extraire de la base de données toutes les catégories de niveau supérieur. Entrez l'instruction SQL suivante dans le bloc %FUNCTION :

```
%FUNCTION(DTW_ODBC) getCategoryList()
{
  <!-- Début du code -->
  <!-- Début du code -->
  <!-- Début du code -->
  select cgrfnbr, cgname, cgthmb, cgmenbr
  from category, cgryrel
  where crpcgnbr is null and
  cgrfnbr=crccgnbr and cgpub=1
  <!-- Fin du code -->
  <!-- Fin du code -->
  <!-- Fin du code -->
  %REPORT
  {
```

Cette instruction SQL extrait les données suivantes des tables CATEGORY et CGRYREL (relations entre catégories), pour toutes les catégories de produits qui ont pour parent la catégorie racine et qui pourront être affichées par les clients : référence de la catégorie, nom de la catégorie, image miniature et référence du commerçant. (Lorsque les données des catégories ont été importées à l'aide de l'utilitaire d'importation globale, ce dernier a automatiquement affecté une valeur de type null à la catégorie racine (nom du magasin).)

7. Vous devez maintenant définir comment afficher chaque ligne extraite des tables de la base de données. Pour ce faire, vous allez créer un bloc %REPORT contenant des codes HTML et des variables pour référencer les données extraites par la requête. Entrez le code HTML suivant dans le bloc %REPORT placé à la suite de l'instruction SQL :

```
%REPORT
{
  %ROW
  {
    <!-- Début du code -->
    <!-- Début du code -->
    <TR>
    <TD>
    <A HREF="/cgi-bin/ncommerce3/CategoryDisplay?
    cgrfnbr=$(V_cgrfnbr)&cgmenbr=$(V_cgmenbr)"
    target="main">
    <IMG ALIGN=CENTER SRC="/grocery/$(V_cgthmb)" BORDER=0
    ALT="$(V_cgname)" WIDTH=120 HEIGHT=25></A>
    </TD>
    </TR>
    <!-- Fin du code -->
    <!-- Fin du code -->
  }
}
```

Remarque : L'instruction /cgi-bin/ncommerce3/CategoryDisplay?cgrfnbr=\$(V_cgrfnbr)&cgmenbr=\$(V_cgmenbr) (entre les marques <A HREF> et) ne doit contenir aucun espace, car cela empêche son exécution. Vous le voyez ici sur trois lignes pour des raisons de présentation.

Pour chaque ligne extraite par la requête, ce fichier HTML affichera l'image miniature de la catégorie concernée, avec un lien à ses enfants (sous-catégories). L'image est affichée à l'aide de la variable Net.Data \$(V_cgthmb), qui est automatiquement remplacée par le contenu de la colonne CGTHMB de la table CATEGORY. Le serveur Web trouve le fichier source de l'image dans le répertoire suivant :

WIN \ibm\netcommerce3\html\<langue>\grocery\

AIX /usr/lpp/NetCommerce3/html/<langue>/grocery/

SUN /opt/IBMnetc/NetCommerce3/html/<langue>/grocery/

Le lien contient l'instruction CategoryDisplay. Celle-ci définit une activité qui permet de rechercher la référence du commerçant et celle de la catégorie dans la colonne CSDISPLAY de la table CATESGP (modèle de catégorie pour

groupe de clients). Ces informations ont été transmises à l'instruction, afin qu'elle puisse extraire le nom du fichier macro à exécuter. Dans ce cas, la macro sera `t_catalog.d2w`.

8. Pour finaliser le modèle, vous devez créer la section HTML Report. C'est elle qui présentera les données dans le navigateur Web du client. Le code HTML approprié figure déjà dans notre modèle. Il ne vous reste qu'à ajouter une référence à la fonction `getCategoryList()` que vous venez de définir, afin que celle-ci soit appelée lors du chargement de la section HTML Report. Dans la section `%HTML_REPORT`, entrez l'appel de fonction suivant :

```
%HTML_REPORT
{
  <HTML>
  ; <HEAD>
  <TITLE>East West Food Mart</TITLE>
  </HEAD>
  <BODY BGCOLOR=#F7DEB5 BORDERCOLOR=#F7DEB5 TEXT=#000000
  LINK=#399FF VLINK=#A7A7A7>
  <TABLE BORDER=0 CELLPADDING=0 CELLSPACING=3 ALIGN=LEFT>
  <!--
  & rbl; <!-- Début du code -->
  <!--
  @getCategoryList()
  <!--
  <!-- Fin du code -->
  <!-- --> ;
  < /TABLE>
  </BODY>
  </HTML>
  %}
```

9. Supprimez du fichier tous les blocs de commentaire marquant le début et la fin des portions de code que vous avez insérées.
10. Enregistrez le fichier sous le nom `t_navmid.d2w`.
11. Cliquez sur **Oui** pour remplacer le fichier existant.
12. Fermez le fichier. La création de la macro `t_navmid.d2w` est à présent terminée.

Test de la liste de catégories

Voyons comment se présente maintenant la devanture du magasin avec la macro de catégorie que vous venez de créer.

1. Revenez à la fenêtre du navigateur dans laquelle le magasin est affiché.
2. Cliquez sur le bouton **Recharger** (Reload) du navigateur. La liste de catégories doit apparaître dynamiquement dans le cadre intermédiaire gauche.
3. Placez le curseur sur le premier lien de catégorie, **Boissons**. Observez l'instruction affichée dans la barre d'état, en bas de la fenêtre :

```
/cgi-bin/ncommerce3/CategoryDisplay?cgrfnbr=XX&cgmenbr=XX
```

Lorsque le client clique sur ce lien, l'instruction `CategoryDisplay` exécute la macro affichant les sous-catégories de la catégorie parent Boissons. Pour déterminer la macro à exécuter, elle se base sur la valeur des paramètres `cgrfnbr` (référence de catégorie) et `cgmenbr` (référence du commerçant). Ces deux références apparaissent ici sous forme de `XX`, mais dans votre navigateur, vous devez voir s'afficher les références que vous avez entrées à l'aide de l'utilitaire d'importation globale.

Remarque : Si l'instruction comporte des espaces (même entre les

paramètres), vous devez les supprimer en éditant la macro (t_navmid.d2w). Sinon, cette instruction ne pourra pas s'exécuter.

4. Examinez les autres liens. Leur fonctionnement est identique, à ceci près qu'ils utilisent une référence de catégorie différente.

Résolution des incidents

Si la liste des catégories ne s'affiche pas correctement, vérifiez le code source et apportez les corrections nécessaires. Si l'affichage demeure incorrect, vous disposez d'une version définitive de la macro, dans le même répertoire que celle que vous venez de modifier. Pour remplacer la macro de catégorie par la version définitive, effacez le fichier t_navmid.d2w et faites une copie de f_navmid.d2w sous le nom t_navmid.d2w. Rechargez ensuite le cadre dans le magasin.

Création de la zone Montant total

Vous allez ajouter une zone dans laquelle s'affichera le montant total des articles placés dans le chariot. Ce montant sera mis à jour chaque fois que le client modifiera le contenu du chariot. Il apparaîtra dans le cadre supérieur gauche de la devanture du magasin. Pour créer cette zone, vous allez éditer un modèle du tutoriel présentant déjà la structure HTML appropriée. Vous y ajouterez du code HTML et écrirez une fonction JavaScript permettant d'afficher le prix total dans une zone d'entrée de formulaire.

1. Pour consulter une présentation des macros Net.Data, reportez-vous à la rubrique Macros Net.Data.
2. Assurez-vous que vous avez pris connaissance des instructions relatives à l'édition des modèles du tutoriel.
3. A l'aide d'un éditeur de texte, ouvrez le modèle de tutoriel tmp_navtop.html.
4. Etudiez le contenu du fichier. Vous y trouverez les marques <SCRIPT> et </SCRIPT> de la fonction JavaScript que vous allez créer, ainsi que la structure d'un tableau qui, actuellement, comporte une seule ligne contenant le logo du magasin East West Food Mart.
5. Vous allez ajouter une ligne au tableau, qui contiendra la zone d'entrée prévue pour l'affichage du prix total. Entrez le code HTML suivant entre les deux derniers blocs de commentaire :

```
<CENTER>
<TABLE BORDER=0 CELLPADDING=0 CELLSPACING=3>
<TR>
<TD ALIGN="CENTER" colspan=2>
<IMG SRC="/grocery/hm_logosmall.gif"
ALT="East West Food Mart" WIDTH=93 HEIGHT=74></TD>
</TR>
<!--<td colspan=2>
<div> Début du code -->
<!--<td colspan=2>
<TR>
<TD WIDTH=25></TD>
<TD><BR><IMG SRC="/grocery/totalamt.gif"></TD>
</TR>
<TR>
<TD WIDTH=25></TD>
<TD>
<FORM NAME="totalForm">
<INPUT NAME="totalAmount" TYPE="TEXT" SIZE=12
onFocus="this.blur();">
```

```

</FORM>
</TD>
</TR>
<!-->
<!--      Fin du code      -->
<!--> ;
</TABLE>
</CENTER>
</BODY>
& rbl;</HTML>

```

Vous venez de créer un formulaire intitulé `totalForm` contenant une zone accessible en lecture seule, `totalAmount`. (C'est l'événement JavaScript `onFocus="this.blur();" qui rend la zone accessible en lecture seule.`) Une image GIF comportant le texte `Montant total` est utilisé comme intitulé de la zone.

- Maintenant, vous allez créer une fonction JavaScript qui permettra de mettre à jour la zone avec le prix total des articles placés dans le chariot. Entre les marques JavaScript `<SCRIPT>` et `</SCRIPT>`, tapez le code ci-après :

```

<HTML>
<HEAD><TITLE>East West Food Mart</TITLE></HEAD>
<BODY BGCOLOR=#FFFFFF TEXT=#000000 LINK=#0000FF VLINK=#F00FF0
ALINK=#FF0000>
<SCRIPT LANGUAGE="JavaScript">
<!-->
<!--      Début du code      -->
<!-->
function refreshTotal(aNumber) {
    document.totalForm.totalAmount.value = ' FRF' + aNumber
}
<!-->
<!--      Fin du code      -->
<!--> ;
</SCRIPT>

```

Vous venez d'écrire une fonction JavaScript appelée `refreshTotal` qui mettra à jour la valeur de la zone d'entrée (`totalAmount`) dans le formulaire (`totalForm`) au moyen de la valeur de `aNumber`.

La fonction reçoit la valeur pour `aNumber` de la macro du chariot (`t_shptodsp.d2w`), des macros de produit (`t_toc_item.d2w` ou `t_toc.d2w`) et de la macro de la liste d'achats (`t_shopcart.d2w`) dès que le client ajoute des articles au chariot, en retire, ou modifie les quantités. Vous apprendrez plus tard, dans ce tutoriel, comment appeler la fonction à partir de ces macros et lui transmettre la valeur de `aNumber`.

- Supprimez du fichier tous les blocs de commentaire marquant le début et la fin des portions de code que vous avez insérées.
- Enregistrez le fichier sous le nom `t_navtop.html`.
- Cliquez sur **Oui** pour remplacer le fichier existant.
- Fermez le fichier. La création de la zone **Montant total** est à présent terminée.

Test de la zone Montant total

Voyons maintenant comment la devanture du magasin intègre la zone **Montant total**. Rechargez la page du magasin affichée dans le navigateur. Le cadre supérieur gauche contient maintenant notre nouvelle zone.

Cette zone est vide puisque le système ne dispose pas encore d'un moyen de calculer le prix total des articles placés dans le chariot et de transmettre le résultat à la fonction `refreshTotal()`, qui insère ensuite la valeur dans la zone. Vous apprendrez comment effectuer cette opération plus tard, dans ce tutoriel, lorsque vous aborderez la création de pages produit.

Résolution des incidents

Si la zone ne s'affiche pas correctement, vérifiez le code et apportez les corrections nécessaires. Si l'affichage demeure incorrect, vous disposez d'une version définitive du fichier HTML, dans le même répertoire que le fichier que vous venez de modifier. Pour remplacer le fichier HTML par la version définitive, effacez le fichier `t_navtop.html` et faites une copie de `f_navtop.html` sous le nom `t_navtop.html`. Rechargez ensuite le cadre dans le magasin.

Création du cadre de navigation

Vous allez maintenant ajouter des liens au cadre inférieur gauche du magasin de sorte que le client puisse effectuer les opérations ci-après tout au long de son parcours.

- S'enregistrer (s'il s'agit d'un nouveau client)
 - Se connecter (si le client est référencé)
 - Afficher le contenu du chariot
 - Afficher la liste d'achats (si le client est référencé)
 - Passer une commande
1. Pour consulter une présentation des macros `Net.Data`, reportez-vous à la rubrique `Macros Net.Data`.
 2. Assurez-vous que vous avez pris connaissance des instructions relatives à l'édition des modèles du tutoriel.
 3. À l'aide d'un éditeur de texte, ouvrez le modèle de tutoriel `tmp_navbot.html`.
 4. Vous devrez d'abord créer le lien **Connexion** en ajoutant le code HTML suivant (partie en gras) :

```
<HTML>
<HEAD><TITLE>East West Food Mart</TITLE></HEAD>
<BODY BACKGROUND="/grocery/backwhit.gif" TEXT=#000000 LINK=#0000FF
VLINK=#F00FF0 ALINK=#FF0000>
<CENTER>
<BR>
<TABLE BORDER=0 CELLPADDING=1 CELLSPACING=0 COLS=2>

<!--<td> Début du code -->
<!--<td> Fin du code -->
<tr>
<td nowrap><a href="/cgi-bin/ncommerce3/LogonForm?url=/cgi-bin/ncommerce3/ExecMacro/tutorial/t_welcome1.d2w/report"
TARGET="main">
</a>
<td>
<!--<td> Fin du code -->
<!--<td> Fin du code -->
```

Remarque : Il est très important que l'instruction ci-dessus (entre les marques `<A HREF>` et `</A>`) ne contienne aucun espace. Sinon, son exécution sera impossible.

Vous venez d'ajouter une image munie d'un lien qui appelle la même instruction que le lien **CONNEXION** figurant dans le cadre principal, à savoir LogonForm. Ce lien mène à une page de connexion invitant le client à entrer son nom d'utilisateur et son mot de passe. Une fois connecté, le client verra s'afficher une page de bienvenue comportant un lien aux produits en promotion.

5. Ajoutez maintenant le lien **Enregistrement** à la suite du lien **Connexion** en insérant le code suivant :

```
<!-->
<!-- & rbl; Début du code -->
<!-->
<TR>
<TD><A HREF="/cgi-bin/ncommerce3/LogonForm?
url=/cgi-bin/ncommerce3/ExecMacro/tutorial/t_welcome1.d2w/report"
TARGET="main">
; <IMG SRC="/grocery/bb_log.gif" BORDER=0></A>
<A HREF="/cgi-bin/ncommerce3/RegisterForm" NAME="register"
TARGET="main">
<IMG SRC="/grocery/bb_reg.gif" WIDTH=60 HEIGHT=18
BORDER=0></A></TD>
</TR>
<!-->
<!-- Fin du code -->
<!--> ;
```

Vous venez d'ajouter une autre image munie d'un lien qui appelle la même instruction que le lien **ENREGISTREMENT** figurant dans le cadre principal, à savoir RegisterForm. L'instruction RegisterForm affichera un formulaire d'enregistrement vierge pour les clients non référencés, et le formulaire Mise à jour de l'enregistrement pour les clients référencés.

6. Vous allez maintenant ajouter le lien **Chariot** qui provoquera l'affichage de la page Chariot. Pour ce faire, ajoutez le code HTML suivant à la suite du lien **Enregistrement** :

```
<TD><A HREF="/cgi-bin/ncommerce3/RegisterForm" NAME="register"
TARGET="main">
<IMG SRC="/grocery/bb_reg.gif" WIDTH=60 HEIGHT=18
BORDER=0></A></TD>
</TR>
<TR>
<TD COLSPAN=2><A NAME="basket" HREF="/cgi-
bin/ncommerce3/OrderItemDisplay"
TARGET="main">
<IMG SRC="/grocery/bb_cart.gif" WIDTH=124 HEIGHT=18
BORDER=0></A></TD>
</TR>
<!-->
<!-- Fin du code -->
<!--> ;
```

Vous venez d'ajouter une nouvelle image munie d'un lien qui appelle l'instruction OrderItemDisplay pour afficher la page Chariot.

7. Maintenant, ajoutez le lien **Liste d'achats** à la suite du lien **Chariot** :

```
<TD COLSPAN=2><A NAME="basket" HREF="/cgi-
bin/ncommerce3/OrderItemDisplay"
TARGET="main">
<IMG SRC="/grocery/bb_cart.gif" WIDTH=124 HEIGHT=18
BORDER=0></A></TD>
```

```

</TR>
<TR>
<TD COLSPAN=2><A NAME="register"
HREF="/cgi-bin/ncommerce3/InterestItemDisplay"
TARGET="main">
<IMG SRC="/grocery/bb_list.gif" WIDTH=124 HEIGHT=18
BORDER=0></A></TD>
</TR>
<!-->
<!-- Fin du code -->
<!-->

```

Lorsqu'un utilisateur cliquera sur ce lien, l'instruction InterestItemDisplay sera appelée pour afficher la page Liste d'achats.

8. Enfin, ajoutez le lien **Régler vos achats** pour afficher la page Commande - Détails, à partir de laquelle le client pourra passer sa commande. Pour ce faire, ajoutez le code HTML ci-après, à la suite du lien **Liste d'achats** :

```

<TR>
<TD COLSPAN=2><A NAME="register"
HREF="/cgi-bin/ncommerce3/InterestItemDisplay"
TARGET="main">
<IMG SRC="/grocery/bb_list.gif" WIDTH=124 HEIGHT=18
BORDER=0></A></TD>
</TR>
<TR>
<TD COLSPAN=2><A NAME="register"
HREF="/cgi-bin/ncommerce3/OrderDisplay?status=P"
TARGET="main">
<IMG SRC="/grocery/bb_chek.gif" WIDTH=124 HEIGHT=18
BORDER=0></A></TD>
</TR>
<!-->
<!-- Fin du code -->
<!--> ;

```

Ce lien appelle l'instruction OrderDisplay qui déclenche l'affichage de la page Détails. L'état de la commande P est transmis en tant que paramètre avec l'instruction, celle-ci affichant alors une commande dont l'état est "en attente".

9. Supprimez du fichier tous les blocs de commentaire marquant le début et la fin des portions de code que vous avez insérées.
10. Enregistrez le fichier sous le nom t_navbot.html.
11. Cliquez sur **Oui** pour remplacer le fichier existant.
12. Fermez le fichier. La création du cadre de navigation est à présent terminée.

Test du cadre de navigation

Voyons maintenant comment les boutons de navigation s'intègrent dans le cadre inférieur gauche du magasin.

1. Rechargez le magasin dans le navigateur. Les liens devraient apparaître sous la forme de boutons dans le cadre inférieur gauche.
2. Placez le curseur sur chaque bouton pour afficher les instructions appelées. Elles apparaissent sur la barre d'état située en bas de la fenêtre.

Remarque : Si une instruction comporte des espaces (même entre les paramètres), vous devez les supprimer du fichier HTML. Sinon, cette instruction ne pourra pas s'exécuter.

3. Cliquez sur chaque bouton pour voir la page obtenue. Ne vous enregistrez pas encore dans le magasin. Vous aurez l'occasion de le faire plus tard.

Résolution des incidents

Si le cadre de navigation ne s'affiche pas ou ne fonctionne pas correctement, vérifiez le code source et apportez les corrections nécessaires. Si le résultat n'est toujours pas correct, vous disposez d'une version définitive du fichier HTML, dans le même répertoire que le fichier que vous venez de modifier. Pour remplacer le cadre de navigation par la version définitive, effacez le fichier `t_navbot.html` et faites une copie de `f_navbot.html` sous le nom `t_navbot.html`. Rechargez ensuite le cadre dans le magasin.

Comment permettre aux clients non référencés de passer des commandes

Un client qui n'aurait pas procédé à son enregistrement est censé pouvoir effectuer des achats dans le magasin. Toutefois, le système exige qu'il indique une adresse avant de pouvoir placer des articles dans son chariot. Tel que le magasin est configuré actuellement, le seul moyen pour un client de fournir cette adresse est de s'enregistrer.

Nous allons voir ici comment donner la possibilité à un client non référencé de remplir son chariot. A cet effet, vous devez écrire une macro qui crée automatiquement un enregistrement d'adresse dès que le client accède au magasin. Ainsi, sans même qu'il le sache, le client dispose d'une d'adresse grâce à laquelle il peut commencer à naviguer dans le magasin et choisir des articles sans devoir s'enregistrer au préalable. Si le client s'enregistre ou se connecte ensuite, l'adresse est mise à jour avec les informations fournies lors de cette étape ; sinon, elle est mise à jour lorsque le client fournit une adresse de livraison pour la commande. Pour vous faire gagner du temps, la macro `Net.Data` est déjà créée et prête à l'emploi. Vous allez examiner son contenu, puis faire en sorte qu'elle soit appelée à partir du fichier HTML regroupant les cadres du magasin.

1. Pour consulter une présentation des macros `Net.Data`, reportez-vous à la rubrique `Macros Net.Data`.
2. Assurez-vous que vous avez pris connaissance des instructions relatives à l'édition des modèles du tutoriel.
3. Ouvrez la macro `t_temp.d2w`, dont le rôle est de créer l'enregistrement d'adresse.
4. Examinez le code. Cette macro vérifie s'il existe une adresse correspondant à l'ID de session (`SESSION_ID`) du client, dans la base de données. En effet, cet enregistrement n'est présent que si le client est actuellement connecté ou déjà référencé.

S'il n'existe aucune adresse pour le client, la macro insère un enregistrement "fictif" dans la table `SHADDR`, à l'aide de l'instruction suivante :

```
AddressAdd?sashnbr=$(shopper_rn)&sanick=$(SESSION_ID)
&saname=empty&saaddr1=
empty&sacity=empty&sastate=empty&sacntry=empty&
sazipc=empty&safield2=T
```

Ce nouvel enregistrement d'adresse est constitué d'un alias (le `SESSION_ID`), de la référence du client (attribuée en interne par le système) et de la valeur `T` pour signaler qu'il s'agit d'une entrée temporaire (cette valeur étant insérée dans la colonne `SAFIELD2`, réservée à l'usage du commerçant). L'instruction insère la chaîne de texte "empty" (vide) dans toutes les autres colonnes. En effet, ces colonnes doivent obligatoirement contenir une valeur. Si le client

s'enregistre ultérieurement, un nouvel enregistrement composé de son nom d'utilisateur et de son adresse est inséré dans cette table. La colonne SAFIELD2 de ce nouvel enregistrement reste vide, tandis que l'enregistrement initialement créé pour le client est marqué comme "temporaire". Au terme de l'exécution de cette instruction, ou dans le cas où le client considéré est déjà référencé et possède donc un enregistrement d'adresse, la macro affiche la liste des catégories principales dans le cadre intermédiaire gauche du magasin.

5. Fermez le fichier.
6. Il faut que cette macro soit appelée chaque fois qu'un client accède au magasin, de sorte que s'il n'est ni référencé ni connecté, un enregistrement d'adresse temporaire soit créé pour permettre à ce client d'effectuer ses achats. La meilleure façon de mettre en oeuvre ce mécanisme est d'appeler la macro à partir du fichier HTML regroupant les cadres du magasin, puisqu'il s'agit du premier fichier chargé. Ouvrez index.html.

7. Recherchez la ligne provoquant le chargement de la liste des catégories dans le cadre intermédiaire gauche :

```
<FRAME NAME="mid" SRC="/cgi-bin/ncommerce3/ExecMacro/tutorial/t_navmid.d2w/report"
MARGINWIDTH=0 SCROLLING="AUTO">
```

8. Modifiez cette ligne de la façon suivante de sorte qu'elle provoque désormais le chargement d'un fichier appelé t_address.html :

```
<FRAME NAME="mid"
SRC="/tutorial/t_address.html"
MARGINWIDTH=0 SCROLLING="AUTO">
```

9. Enregistrez les modifications et fermez le fichier.

10. Ouvrez et consultez le fichier t_address.html. Son contenu est le suivant :

```
<HTML>
<HEAD><TITLE>Initialisation des paramètres de la session en cours.
Veuillez patienter...</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function LoadImage ()
{
window.location = "https://" + window.location.host +
"/cgi-bin/ncommerce3/ExecMacro/tutorial/t_temp.d2w/report" + "";
}
</SCRIPT>
</HEAD>
<BODY BGCOLOR=#FFFFFF TEXT=#000000 LINK=#0000FF VLINK=#F00FF0
ALINK=#FF0000>
<CENTER>
<BR>
<TABLE BORDER=0 CELLSPACING=1 CELLPADDING=1 WIDTH=200>
<TR ALIGN="CENTER"><TD><IMG SRC="/grocery/wait.gif"></TD></TR>
<TR><TD><BR><BR></TD></TR>
<TR ALIGN="CENTER"><TD><IMG SRC="/grocery/hm_logosmall.gif"
onLoad=LoadImage();></TD></TR>
</TABLE>
& rbl; </CENTER>
</BODY>
</HTML>
```

La première partie de ce fichier, comprise entre les marques <HEAD> et </HEAD>, utilise JavaScript pour exécuter la macro t_temp.d2w. Le corps du

fichier (BODY) affiche quant à lui le logo du magasin avec la mention “Veuillez patienter...” pour que l’écran ne reste pas vide pendant l’exécution de la macro `t_temp.d2w`.

11. Fermez le fichier.
12. Voyons maintenant ce qui va se passer lorsque le client accédera au magasin :
 - a. Le fichier `index.html` charge tous les cadres de la devanture du magasin comme d’habitude. Toutefois, au lieu de charger la liste de catégories (`t_navmid.d2w`) dans le cadre situé dans la partie centrale gauche, il charge `t_address.html`.
 - b. `T_address.html` affiche le message “Veuillez patienter” pendant l’exécution de la macro `t_temp.d2w`.
 - c. `T_temp.d2w` vérifie si le client est enregistré ou connecté. Si tel n’est pas le cas, elle crée un enregistrement d’adresse temporaire dans la base de données.
 - d. La liste de catégories s’affiche dans le cadre intermédiaire gauche.

Test de la création d’entrée d’adresse

Etudions comment l’entrée du carnet d’adresses est automatiquement créée lorsque le client accède au magasin. Comme vous n’avez pas encore procédé à votre enregistrement dans le magasin, la base de données ne contient pas d’entrée de carnet d’adresses correspondant à votre “SESSION_ID”. Par conséquent, une adresse “fictive” sera créée par la macro `t_temp.d2w`.

1. Rechargez le magasin dans le navigateur.
2. Maintenant, examinez attentivement le cadre intermédiaire gauche. Le logo du magasin doit apparaître pendant quelques instants, accompagné du message “Veuillez patienter...” (sous forme de texte et d’image), puis la liste des catégories s’affiche. C’est pendant les secondes qui ont précédé l’affichage de cette liste que votre enregistrement d’adresse temporaire a été créé dans la base de données.

Résolution des incidents

Si le cadre intermédiaire gauche ne se charge pas correctement, vérifiez le code. Si l’affichage demeure incorrect, vous disposez d’une version définitive du fichier `index.html`, dans le même répertoire que le fichier que vous avez modifié. Pour remplacer ce dernier par la version définitive, effacez le fichier `index.html` et faites une copie de `f_index.html` en `index.html`. Ensuite, rechargez le magasin.

Chapitre 7. Personnalisation de la page Enregistrement

Vous allez maintenant personnaliser la page Enregistrement du magasin. Elle a déjà été personnalisée de façon à comporter un fond blanc et des images du magasin. Les commerçants personnalisent généralement les macros par défaut Net.Commerce, s'ils les utilisent pour leur propre magasin. La page Enregistrement comporte actuellement des zones destinées à la saisie des informations suivantes sur le client :

- Informations personnelles : ID utilisateur de connexion, mot de passe, titre, nom et entreprise.
- Informations sur le contact : adresse complète, numéros de téléphone et de télécopie, adresse électronique (e-mail) ou URL.
- Informations démographiques : sexe, situation de famille, employeur, nombre de personnes dans le foyer, revenu annuel.

Une fois que le client se sera enregistré et aura passé une commande, les informations entrées sur ce formulaire seront utilisées comme adresse d'expédition par défaut. Vous verrez cela plus tard lorsque nous examinerons le processus de commande.

Ce que vous allez faire

Dans cette étape, vous effectuerez les opérations suivantes :

- Afficher la page Enregistrement telle qu'elle se présente actuellement et vous enregistrer.
- Supprimer de la page Enregistrement les zones réservées aux informations démographiques (cela a déjà été fait pour la page Mise à jour de l'enregistrement).
- Activer les modifications apportées en affectant la macro personnalisée à l'activité qui lui correspond.
- Tester les modifications dans le navigateur.

Pour passer cette section

Si vous choisissez de ne pas suivre cette section, vous devez effectuer les opérations suivantes :

1. Effacez le fichier `tmp_msregform.d2w` et faites une copie de `f_msregform.d2w` sous le nom `tmp_msregform.d2w`.
2. Connectez-vous au magasin en procédant comme indiqué à la section Visualisation de la page Enregistrement.
3. Passez à la section Chiffrement des informations d'enregistrement.

Visualisation de l'état actuel de la page Enregistrement

Vous allez tout d'abord visualiser la page Enregistrement telle qu'elle apparaît actuellement aux clients, puis vous en profiterez pour vous enregistrer dans le magasin.

1. Dans le cadre de navigation, cliquez sur **Enregistrement**. Un formulaire d'enregistrement vierge s'affiche.
2. Faites défiler le formulaire. Celui-ci comporte trois sections : Informations personnelles, Informations sur le contact et Informations démographiques.

3. A présent, enregistrez-vous. Complétez les zones obligatoires (en gras), dont le prénom et le titre. Veillez à entrer un ID de connexion et un mot de passe différents de ceux que vous utilisez en tant qu'administrateur de site ou de magasin.
4. Notez votre ID de connexion et votre mot de passe.
5. Cliquez sur le bouton **Enregistrement** dans le bas du formulaire. Une fenêtre s'affiche, indiquant que le serveur a mis à jour votre cookie de session temporaire en fonction de l'ID de connexion avec lequel vous vous êtes enregistré.
6. Cliquez sur **OK** pour accepter le nouveau cookie de session. Une page de confirmation s'affiche.

Etude de la macro d'enregistrement

Examinons à présent la macro qui crée cette page.

1. Pour consulter une présentation des macros Net.Data, reportez-vous à la rubrique Macros Net.Data.
2. Assurez-vous que vous avez pris connaissance des instructions relatives à l'édition des modèles du tutoriel.
3. A l'aide d'un éditeur de texte, ouvrez le modèle de tutoriel tmp_msregform.d2w.
4. Examinez la section HTML Report :

```
%{ ***** HTML Report
Section ***** %}
%HTML_REPORT
{
<HTML>
<HEAD>
<META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
<TITLE>Enregistrement - Informations</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF TEXT=#000000 LINK=#399FF VLINK=#A7A7A7>
<FORM METHOD="POST" ACTION="/cgi-bin/ncommerce3/RegisterNew">
<INPUT TYPE="HIDDEN" Name="url" VALUE="/cgi-bin/ncommerce3
/ExecMacro/tutorial/t_msregform.d2w/input">
<INPUT TYPE="HIDDEN" Name="mode" VALUE="new" >
<INPUT TYPE="HIDDEN" Name="cancel_url" VALUE=url_in >
<IMG SRC="/grocery/h_regis.gif">
<TABLE BORDER=0 WIDTH=600>
<TR>
<P><TD>Pour vous faire connaître en tant que nouveau client,
remplissez le formulaire ci-dessous.
Toutes les zones en <B>gras</B>sont obligatoires.
Toutes ces informations sont confidentielles et seront conservées
par les propriétaires
du centre commercial et des magasins. Elles ne seront en aucun cas
divulguées
à des fins de démarchage. </TD></P>
</TR>
</TABLE>
<BR>
%INCLUDE "/tutorial/t_msregform1.inc"
%INCLUDE "/tutorial/t_msregform2.inc"
%INCLUDE "/tutorial/t_msregform3.inc"
%INCLUDE "/tutorial/t_msregform4.inc"
```

```

<TABLE BORDER=0 WIDTH=600>
<TR>
<TD WIDTH=460 VALIGN="MIDDLE"><FONT
SIZE=+1><B>Enregistrement</B></FONT></TD>
</TR>
<P><TD>Vous êtes prêt à faire valider votre enregistrement.
    Pour cela, cliquez sur le bouton Enregistrement.
    Pour annuler les modifications, cliquez sur le bouton
Annuler.</TD></P>
</TABLE>
<TABLE BORDER=0 CELLPADDING=10 ALIGN=CENTER>
<TR><TD><INPUT TYPE="SUBMIT" VALUE="Enregistrement "></TD>
</FORM>
<TD>
%IF (url_in == "")
<A HREF="/tutorial/t_home.html"><IMG SRC="/grocery/g_cancel.gif"
BORDER=0></A>
%ELSE
<A HREF="%(url_in)"><IMG SRC="/grocery/g_cancel.gif"
BORDER=0></A>
%ENDIF
</TD></TR>
</TABLE>
</BODY>
</HTML>
%}

```

Dans cette section, les opérations suivantes sont effectuées :

1. Affichage du titre de la page et d'une introduction.
2. Importation d'un fichier d'inclusion appelé `msregform1.inc`, qui affecte les variables `Net.Data` utilisées pour les intitulés des zones du formulaire. A chaque zone est affectée la valeur de la colonne correspondante dans la table `SHADDR` (Carnet d'adresses du client), `SHOPPER` (Profil du client) ou `SHOPDEM` (Informations démographiques).
3. Importation du fichier `msregform2.inc`, qui contient le code HTML affichant la section **Informations personnelles** du formulaire.
4. Importation du fichier `msregform3.inc`, qui contient le code HTML affichant la section **Informations sur le contact** du formulaire.
5. Importation du fichier `msregform4.inc`, qui contient le code HTML affichant la section **Informations démographiques** du formulaire.
Le contenu de chaque section du formulaire d'enregistrement est stocké dans des fichiers "include" distincts. Vous pouvez ainsi exclure une section en supprimant simplement l'instruction d'inclusion qui appelle le fichier correspondant. Cette technique de personnalisation a été mise en oeuvre pour simplifier le contenu de la macro.
6. Affichage des boutons **Enregistrement** et **Annuler**. Lorsque le client clique sur le bouton **Enregistrement**, l'URL ACTION du formulaire appelle l'instruction `RegisterNew`, qui crée pour ce client un nouvel enregistrement dans la base de données. Une fois l'exécution de l'instruction terminée, un deuxième URL est appelé. Ce dernier est défini dans une zone d'entrée cachée appelée `url` (voir ci-après) :

```

<INPUT TYPE="HIDDEN" Name="url" VALUE="/cgi-bin/ncommerce3
/ExecMacro/tutorial/t_msregform.d2w/input">

```

Cet URL appelle l'instruction `ExecMaco` afin d'exécuter la section HTML Input de

la macro, qui extrait le nom du client de la table SHADDR et l'affiche accompagné d'un message de confirmation.

Retrait des zones d'informations démographiques de la page Enregistrement

Vous allez maintenant éditer la macro d'enregistrement pour retirer les zones réservées à l'affichage des informations démographiques. En général, vous pouvez retirer d'une macro les zones qui ne sont pas requises par le système Net.Commerce ni par une fonction non prioritaire. Les zones obligatoires correspondent aux colonnes des tables de la base de données qui sont définies comme non nulles (NOT NULL) dans la base de données. Dans cette macro, les zones de ce type sont SHLOGID (ID de connexion du client), SHLPSPWD (mot de passe du client) et SHLPSPWDVER (mot de passe du client pour vérification).

Ainsi, lorsque le client s'enregistre, l'instruction RegisterNew appelle l'activité de traitement AUDIT_REG pour qu'elle vérifie les zones suivantes du formulaire : SALNAME, SADDR1, SACITY, SASTATE, SACNTRY et SAZIPC. Vous ne pouvez donc pas exclure ces zones sans remplacer la fonction non prioritaire affectée à l'activité de traitement AUDIT_REG. Dans le cas des autres zones, vous pouvez masquer ou supprimer les marques HTML qui servent à les afficher sur le formulaire d'enregistrement.

1. La macro tmp_msregform.d2w étant ouverte, recherchez les quatre instructions suivantes, qui importent différentes sections du formulaire sous forme de fichiers d'inclusion :

```
%INCLUDE "/tutorial/t_msregform1.inc"
%INCLUDE "/tutorial/t_msregform2.inc"
%INCLUDE "/tutorial/t_msregform3.inc"
%INCLUDE "/tutorial/t_msregform4.inc"
```

2. Supprimez la dernière instruction, qui importe la section Informations démographiques dans le formulaire. Il ne doit rester que les trois premières instructions :

```
%INCLUDE "/tutorial/t_msregform1.inc"
%INCLUDE "/tutorial/t_msregform2.inc"
%INCLUDE "/tutorial/t_msregform3.inc"
```

Au lieu de supprimer cette instruction, vous pouvez la mettre en commentaire à l'aide des marques HTML (<!-- et -->). Ainsi, vous pourrez la réintégrer rapidement si, par la suite, vous décidez de faire figurer les zones d'informations démographiques sur le formulaire.

3. Enregistrez le fichier sous le nom t_msregform.d2w.
4. Cliquez sur **Oui** pour remplacer le fichier existant.
5. Fermez le fichier.

Remarque : Normalement, vous devriez aussi exclure les zones d'informations démographiques de la macro t_msregupd.d2w, chargée d'afficher la page Mise à jour de l'enregistrement. Cependant, pour vous faire gagner du temps, cette opération a déjà été effectuée.

Activation de la macro d'enregistrement personnalisée

Le système Net.Commerce utilise actuellement le modèle tmp_msregform.d2w pour afficher un formulaire d'enregistrement lorsqu'un client souhaite s'enregistrer dans le magasin. Pour indiquer au système qu'il doit commencer à utiliser la macro t_msregform.d2w personnalisée, vous devez affecter celle-ci à l'activité d'affichage REG_NEW, qui sert à afficher un nouveau formulaire d'enregistrement. Chaque fois que vous remplacez un fichier macro ou une fonction non prioritaire qui implémente une activité, vous devez affecter le nom du nouveau fichier à cette activité.

1. Ouvrez une nouvelle fenêtre de navigateur.
2. Ouvrez l'Administrateur Net.Commerce à l'aide du signet que vous avez ajouté précédemment.
3. Lorsque vous y êtes invité, entrez votre nom d'utilisateur et votre mot de passe d'administrateur de site. N'oubliez pas d'entrer le nouveau mot de passe que vous avez défini à l'étape Changement de votre mot de passe. Cliquez ensuite sur **Connexion**. L'affectation de macros personnalisées et de fonctions non prioritaires aux activités associées est réservée aux administrateurs de site.
4. Si vous y êtes invité, acceptez le cookie de session qui est mis à jour. La page d'accueil de l'Administrateur Net.Commerce s'affiche.
5. Sur la barre des tâches, cliquez sur **Gestionnaire de site**, puis sur **Gestion des activités**. Le formulaire Gestion des activités s'affiche.
6. Dans la liste **Sélection du type d'activité**, sélectionnez VIEW.
Les activités d'affichage servent à afficher les pages que les clients voient normalement lorsqu'ils effectuent leurs achats. La liste des activités de type VIEW (affichage) s'affiche dans le cadre inférieur.
7. Faites défiler cette liste et cliquez sur **REG_NEW**.
Immédiatement, les informations décrivant l'activité apparaissent dans les zones du formulaire. Puisque vous créez le site Web d'un seul magasin, vous n'avez pas besoin de changer la portée de l'activité. Si vous étiez en train de créer un centre commercial, vous pourriez très bien affecter votre nouveau fichier macro à certains magasins uniquement.
8. Sur la barre des tâches, cliquez sur **Affectation d'activité**.
9. Cliquez sur le bouton **Macro** dans le bas du cadre. Le formulaire Affectation d'une macro s'affiche.
10. Dans la zone **Nom du fichier de macros**, remplacez le nom de la macro tmp_msregform.d2w par t_msregform.d2w.
Remarque : La liste **Sélection centre/magasin** affiche la mention **Centre commercial** car votre macro a été affectée à l'activité en tant que macro par défaut, au niveau du centre commercial. Considérez votre magasin comme un centre commercial comportant un seul magasin.
11. Cliquez sur **Sauvegarder** pour mettre à jour l'enregistrement correspondant dans la table MACROS de la base de données.
12. Vérifiez qu'un message de confirmation apparaît.
Désormais, votre macro personnalisée sera appelée chaque fois qu'un client s'enregistrera dans le magasin.
13. Fermez la fenêtre du navigateur.

Test de la nouvelle macro d'enregistrement

Testons les modifications apportées à la macro d'enregistrement. Pour cela, vous devez accéder au magasin comme si vous étiez un autre client qui souhaite s'enregistrer.

1. Fermez la fenêtre du navigateur dans laquelle le magasin est actuellement affiché.
2. Ouvrez une nouvelle fenêtre de navigateur et sélectionnez le magasin dans la liste de signets.
3. Le cas échéant, acceptez le nouveau cookie de session temporaire qui vous est affecté.
4. Lorsque la devanture du magasin apparaît, cliquez sur **Enregistrement**, dans le cadre de navigation. Un formulaire d'enregistrement vierge s'affiche.
5. Faites défiler le formulaire jusqu'en bas. Notez que les zones réservées aux informations démographiques ont disparu.
6. Puisque vous vous êtes déjà enregistré dans le magasin, vous allez à présent vous connecter. Dans le cadre de navigation, cliquez sur **Connexion**. La page Connexion s'affiche.
7. Dans les zones appropriées, entrez le nom d'utilisateur et le mot de passe que vous avez indiqués lors de votre enregistrement (c'est-à-dire à l'étape Visualisation de l'état actuel de la page Enregistrement).
8. Cliquez sur **Connexion**.
9. Cliquez sur **OK** dans la fenêtre Netscape pour accepter le nouveau cookie de session. Une page de bienvenue apparaît. Notez que votre nom de client référencé y est mentionné.
10. Fermez le navigateur.

Résolution des incidents

Si la macro d'enregistrement ne s'affiche pas correctement, vérifiez le code source et apportez les corrections nécessaires. Si l'affichage demeure incorrect, vous disposez d'une version définitive de la macro, dans le même répertoire que celle que vous venez de modifier. Pour remplacer la macro d'enregistrement par la version définitive, effacez le fichier `t_msregform.d2wet` faites une copie de `f_msregform.d2w` sous le nom `t_msregform.d2w`, puis répétez les étapes 4 à 10.

Chapitre 8. Chiffrement des informations d'enregistrement

Les instructions Net.Commerce qui transfèrent des informations personnelles sur un client (par exemple, son adresse et son mot de passe) du navigateur au serveur et inversement doivent être utilisées conjointement avec le protocole SSL (Secure Sockets Layer). Ce dernier chiffre les informations pour éviter qu'elles ne soient accessibles à des personnes non autorisées.

Dans cette étape, vous allez affecter au protocole SSL les instructions utilisées pour afficher, ajouter et mettre à jour les informations d'enregistrement des clients.

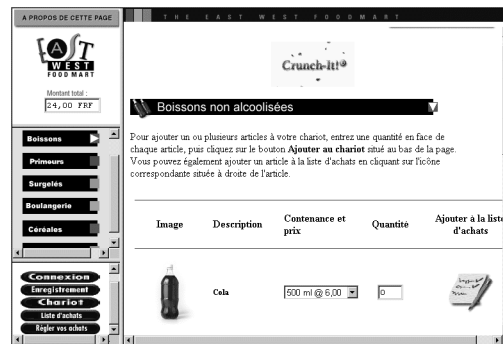
1. Ouvrez une nouvelle fenêtre de navigateur.
2. Sélectionnez Administrateur Net.Commerce dans votre liste de signets.
3. Lorsque vous y êtes invité, entrez votre nom d'utilisateur et votre mot de passe d'administrateur de site, puis cliquez sur **Connexion**. Seuls les administrateurs de site peuvent affecter des instructions au protocole SSL.
4. Si vous y êtes invité, acceptez le cookie de session qui est mis à jour. La page d'accueil de l'Administrateur Net.Commerce s'affiche.
5. Sur la barre des tâches, cliquez sur **Gestionnaire de site**, puis sur **Sécurisation des instructions**. Le formulaire Affectation d'instruction sécurisée s'affiche.
6. Dans la liste déroulante **Magasin**, sélectionnez **East West Food Mart**.
7. Dans la liste **Instructions**, sélectionnez les instructions suivantes (maintenez enfoncée la touche Majuscule lors de la sélection de chaque instruction) :
 - RegisterForm
 - RegisterNew
 - RegisterUpdate
8. Cliquez sur le bouton **Ajouter à la liste** pour ajouter les instructions à la liste **Instructions affectées (SSL)(Authentification)**. Elles sont alors retirées de la liste **Instructions**.
9. Pour affecter le protocole SSL aux instructions, cochez la case **SSL activé**.
10. Cliquez sur **Mettre à jour**. Vérifiez qu'un message de confirmation apparaît. Désormais, lorsqu'un client s'enregistrera ou mettra à jour ses informations d'enregistrement, toutes les données échangées à cet effet seront chiffrées.
11. Fermez la fenêtre du navigateur.

Remarque : Par défaut, le système Net.Commerce affecte automatiquement les instructions d'enregistrement (Register) au protocole SSL. Cette procédure vous a conduit à les réaffecter dans le seul but de vous montrer comment sécuriser une instruction quelle qu'elle soit.

Chapitre 9. Création des pages produit

Vous allez maintenant créer les pages produit du magasin. Elles seront conçues de telle sorte que tous les produits appartenant à la même catégorie parent figurent sur une page produit spécifique. Les produits en vente dans notre magasin étant très courants, il n'y a pas lieu de les décrire en détail et, par conséquent, il est inutile de créer une page individuelle pour présenter chaque produit. En outre, le client passe moins de temps à parcourir le magasin.

Certains produits, tels que l'eau en bouteille et les boissons non alcoolisées, sont disponibles en différentes contenances vendues à des prix différents. Cela vous conduit à créer deux types de pages produit : une pour les produits disponibles en plusieurs contenances et une autre pour les produits qui ne le sont pas. Cette dernière page a déjà été créée pour vous, car elle est de nature plus simple. Vous allez créer les pages destinées aux produits existant en plusieurs contenances. Elles seront présentées au client lorsqu'il cliquera sur **Eaux** ou sur **Boissons non alcoolisées**, sous la catégorie principale Boissons. Ces pages auront l'apparence suivante :



A partir de cette page, le client pourra :

- voir une publicité animée des produits affichés ;
- indiquer la quantité souhaitée pour chaque article de son choix ;
- sélectionner une contenance (petite, moyenne ou grande) ;
- ajouter simultanément plusieurs articles dans le chariot ;
- ajouter des articles un par un à la liste d'achats (si le client est référencé).

Ce que vous allez faire

Dans cette section, vous allez créer une macro Net.Data pour l'affichage des pages produit. Vous exécuterez en particulier les opérations suivantes :

- Insertion de la publicité animée dans le haut de la page.
- Création d'un tableau répertoriant les produits et comportant :
 - des listes de sélection présentant dynamiquement les attributs disponibles pour chaque produit,
 - des zones de texte permettant d'indiquer les quantités,
 - des icônes permettant d'ajouter individuellement les produits à la liste d'achats.
- Ajout du bouton **Ajouter au chariot**, qui permettra au client d'ajouter plusieurs produits et les quantités correspondantes à son chariot.
- Test des pages résultantes dans le navigateur.

Pour passer cette section

La macro que vous allez créer est complexe et fait largement appel à JavaScript. Si vous choisissez de ne pas suivre cette section, vous devez effectuer les opérations suivantes :

1. Effacez le fichier \macro\<langue>\category\t_toc_item.d2w et faites une copie de \macro\<langue>\category\fl_toc_item.d2w sous le nom \macro\<langue>\category\t_toc_item.d2w.
2. Ajoutez des produits au chariot et à la liste d'achats en procédant comme indiqué à la section Test de la macro de produit.
3. Passez à la section Activation de la zone Montant total.

Ajout de la publicité et de la bannière

Vous allez maintenant commencer à créer la macro de produit t_toc_item.d2w, qui affichera les produits et leurs différentes contenances. Pour vous aider à démarrer, un modèle a été créé à votre intention. En premier lieu, vous ajouterez à ce modèle la publicité animée qui s'affichera dans le haut des pages produit, puis vous écrirez quelques fonctions permettant d'extraire les données indispensables à d'autres parties de la macro.

1. A l'aide d'un éditeur de texte, ouvrez le modèle de tutoriel tmp_toc_item.d2w dans le répertoire suivant :

```
WIN \ibm\netcommerce3\macro\<langue>\category\
```

```
AIX /usr/lpp/NetCommerce3/macro/<langue>/category/
```

```
SUN /opt/IBMnetc/NetCommerce3/macro/<langue>/category/
```

2. Examinez le contenu de ce fichier. Une structure est déjà en place.
3. Vous allez à présent écrire une fonction qui affichera une image animée de la catégorie parent dans le haut de la page. Le nom du fichier GIF correspondant est stocké dans la zone CGFIELD2 (réservée à l'usage du commerçant) de la table CATEGORY. Ce type d'image n'est pas disponible pour toutes les catégories. Pour celles auxquelles aucune image publicitaire n'est associée, la page produit correspondante comportera une zone vide à l'emplacement prévu pour cette image. Dans la section Function Definition de la macro, ajoutez la fonction suivante :

```
%{ ***** Function Definition Section
***** %}
<!-- Début du code -->
<!-->
%FUNCTION(DTW_ODBC) display_ad()
{
    select cgfield2
    from category
    where cgrfnbr=$(cgrfnbr)
    and cgmenbr=$(cgmenbr)
    and cgpub=1
    and cgfield2 is not NULL
    %REPORT {
        <CENTER>
        <TABLE BORDER=0 CELLPADDING=0 CELLSPACING=0>
        <TR>
            <TD><IMG SRC="/grocery/$(V_cgfield2)"
            ALT="Publicité"></TD>
        </TR>
```

```

        <BR>
    </TABLE>
</CENTER>
<BR>
%}
%MESSAGE {
    100: { %} :continue
%}
%}
<!-->
<!--      Fin du code      -->
<!-->

```

Vous venez d'écrire une fonction `display_ad()` qui consultera la colonne `CGFIELD2` de la table `CATEGORY` pour extraire le nom du fichier image correspondant à la catégorie sélectionnée par le client, cette catégorie étant identifiée à la fois par sa référence et par la référence du magasin chez le commerçant. S'il existe un fichier image pour la catégorie, la section Report l'affichera dans le haut de la page.

4. Vous allez maintenant écrire une fonction qui va extraire la référence d'adresse attribuée au client. Cette référence sera utilisée comme paramètre fourni à la fonction du bouton **Ajouter au chariot**, que vous créerez plus tard. Ajoutez la fonction ci-après à la suite de la fonction `display_ad()` :

```

%FUNCTION(DTW_ODBC) find_addr_rn()
{
    select sarfnbr
    from shaddr, shopper
    where (shlogid='$(SESSION_ID)')
    and (shrfnbr=sashnbr)
    and (sanick=shlogid)
    %REPORT {
        @DTW_ASSIGN(addr_rn, V_sarfnbr)
    %}
<!-->
<!--      Fin du code      -->
<!-->

```

Cette fonction exécute une requête SQL pour extraire la référence d'adresse du client des tables `SHADDR` (Carnet d'adresses du client) et `SHOPPER` (Profil du client). La section Report affecte la variable `addr_rn` à la valeur extraite. Cette variable sera référencée ultérieurement dans la macro.

5. Vous devez maintenant écrire une fonction qui affichera le nom de la catégorie sélectionnée dans une bannière, sous la publicité. Ajoutez la fonction ci-après à la suite de la fonction `find_addr_rn()` :

```

%FUNCTION(DTW_ODBC) get_subcat_name()
{
    select cgfield1, cgrfnbr
    from category
    where cgrfnbr=$(cgrfnbr)
    and cgmenbr=$(cgmenbr)
    and cgpub=1
    %REPORT {
        @DTW_ASSIGN(category_name, V_cgrfnbr)
        <IMG SRC="/grocery/$(V_cgfield1)"
        ALT="Intitulé de la catégorie"
        ALIGN="CENTER">
        <BR>
    %}
}

```

```

%}
%MESSAGE {
  100:{ %} :continue
%}
%}
<!-->
<!-- Fin du code -->
<!-->

```

Cette fonction consulte la colonne CGFIELD1 de la table CATEGORY pour obtenir le nom du fichier image associé à la catégorie et la référence de cette dernière. La section Report affiche l'image dans le haut de la page et affecte la variable category_name à la référence de catégorie extraite. Ce nom de variable sera utilisé ultérieurement dans la macro.

- Enregistrez le fichier sous le nom t_toc_item.d2w.
- Cliquez sur **Oui** pour remplacer le fichier existant.

Création du tableau de produits

Vous êtes prêt à créer le tableau de présentation des produits. Il comportera une ligne par produit, constituée des informations suivantes :

- Une image miniature du produit
- Le nom du produit
- Une liste de sélection présentant les différentes variantes (contenances) disponibles, ainsi que les prix correspondants
- Une zone d'entrée de texte permettant de spécifier la quantité souhaitée
- Une icône permettant d'ajouter le produit sélectionné à la liste d'achats

Le tableau aura l'apparence suivante :

Pages de produits

Publicité

CATÉGORIE

Introduction

Image	Description	Contenance et prix	Quantité	Ajouter à la liste d'achats
Image miniature	Nom du produit	▼		Icône
Image miniature	Nom du produit	▼		Icône
Image miniature	Nom du produit	▼		Icône

Ajouter au chariot

Le nombre de lignes du tableau dépendra du nombre de produits répertoriés dans la base de données, pour la catégorie parent considérée.

Début de la création du tableau de produits

Commençons à ajouter le code de construction du tableau de produits.

- En premier lieu, vous devez extraire de la base de données toutes les données de produits et d'attributs correspondant à la catégorie sélectionnée par le client.

Ajoutez la fonction ci-après à la suite de la fonction `get_subcat_name()` :

```
%FUNCTION(DTW_ODBC) build_prod_list()
{
  select prrfnbr, prprfnbr, prsdsc, prnbr, prthmb,
  ppprc, paname, paval, mmrfnbr, mmmenbr
  from product, prodprcs, prodatr, cgprrel, mshipmode
  where ppprnbr=prrfnbr
  and paprnbr=prrfnbr
  and cpcgnbr=$(category_name)
  and cpprnbr=prprfnbr
  and mmmenbr=prmenbr
  <!-->
  <!--      Fin du code      -->
  <!-->
```

Pour chaque produit appartenant à la catégorie sélectionnée par le client, ces instructions SQL extraient la référence du produit, la référence produit du parent, la description, le numéro du produit, l'image miniature, le prix unitaire, les noms d'attribut, les valeurs d'attribut, la référence du mode d'expédition du commerçant et la référence du commerçant. Ces données sont extraites des tables PRODUCT, PRODPRCS (Prix du produit), PRODATR (Attribut de produit), CGPRREL (Relation entre produit et catégorie) et MSHIPMODE (Mode d'expédition du commerçant). La variable `category_name` (référence de catégorie affectée par la fonction `get_subcat_name()`) permet à l'utilisateur de s'assurer que les produits corrects sont extraits.

2. Vous allez maintenant ajouter des instructions expliquant au client comment ajouter des produits à son chariot, puis créer les en-têtes de colonne du tableau. Ajoutez le code suivant après la requête SQL :

```
%FUNCTION(DTW_ODBC) build_prod_list()
{
  select prrfnbr, prprfnbr, prsdsc, prnbr, prthmb,
  ppprc, paname, paval, mmrfnbr, mmmenbr
  from product, prodprcs, prodatr, cgprrel, mshipmode
  where ppprnbr=prrfnbr
  and paprnbr=prrfnbr
  and cpcgnbr=$(category_name)
  and cpprnbr=prprfnbr
  and mmmenbr=prmenbr
  %REPORT {
    <TABLE BORDER=0 CELLPADDING=5 COLS=5>
    <P>Pour ajouter un ou plusieurs articles à votre chariot,
    entrez une quantité pour chacun d'eux, puis cliquez sur le bouton
    <B>Ajouter au chariot</B> figurant dans le bas de la page.
    Vous pouvez également ajouter un article à votre
    liste d'achats en cliquant sur l'icône qui lui est associée.
    <TR><TD HEIGHT=5 COLSPAN=5><IMG SRC="/grocery/widespa2.gif"
    WIDTH=750></TD></TR>
    <TR>
      <TD ALIGN="CENTER"><B>Image</B></TD>
      <TD ALIGN="LEFT"><B>Description</B></TD>
      <TD ALIGN="LEFT"><B>Contenance et prix</B></TD>
      <TD ALIGN="CENTER"><B>Quantité</B></TD>
      <TD ALIGN="CENTER"><B>Ajouter à la liste d'achats</B></TD>
    </TR>
```

3. Enregistrez votre travail (sous le nom `t_toc_item.d2w`).

Ajout de l'image et du nom du produit

Vous devez à présent commencer à définir la manière dont chaque produit extrait par la requête s'affichera dans le tableau. Vous allez ajouter l'image miniature du produit dans la première colonne, et le nom du produit dans la deuxième colonne.

1. Ajoutez le code suivant à la section Report de la fonction `build_prod_list()` :

```
<TR>
  <TD ALIGN="CENTER"><B>Image</B></TD>
  <TD ALIGN="LEFT"><B>Description</B></TD>
  <TD ALIGN="LEFT"><B>Contenance et prix</B></TD>
  <TD ALIGN="CENTER"><B>Quantité</B></TD>
  <TD ALIGN="CENTER"><B>Ajouter à la liste d'achats</B></TD>
</TR>
%ROW {
<TR><TD HEIGHT=5 COLSPAN=5<IMG SRC="/grocery/widespa2.gif"
WIDTH=750></TD>
<TR><TD><IMG SRC="/grocery/${V_prthmb}" WIDTH=100 HEIGHT=100></TD>
<TD><B><FONT SIZE=-1>${V_prsdesc}</FONT></B></TD>
<!-- _____ ->
<!-- Fin du code ->
<!-- _____ ->
```

Pour chaque ligne de données extraite par la requête, ce code HTML crée deux lignes dans le tableau. La première ligne est une barre noire s'étendant sur toute la largeur du tableau. La deuxième ligne utilise des variables de base de données pour afficher l'image miniature du produit dans la première colonne, et le nom du produit dans la deuxième colonne.

2. Enregistrez le travail effectué jusqu'ici.

Création de la liste de sélection

Vous êtes maintenant prêt à créer la liste de sélection dans la troisième colonne du tableau de produits. Cette opération est plus complexe que celles que vous avez réalisées jusqu'ici.

1. Pour comprendre comment vous allez procéder, examinez d'abord les données qui seront extraites par la fonction `build_prod_list()`. Vous trouverez ci-dessous une liste partielle des données extraite par la fonction pour la catégorie Eaux :

PRRFNBR	PRPRFNBR	PRSDISC	PRNBR	PRTHMB	PPPRC	PANAME	PAVAL
2	1	Eau de source	100-S	wac1102a.gif	6.00 FRF	Contenance	500 ml
3	1	Eau de source	100-M	wac1102a.gif	12.00 FRF	Contenance	1 l
4	1	Eau de source	100-L	wac1102a.gif	18.00 FRF	Contenance	1.5 l
6	5	Eau minérale	101-S	wac1101a.gif	6.00 FRF	Contenance	500 ml
7	5	Eau minérale	101-M	wac1101a.gif	12.00 FRF	Contenance	1 l
8	5	Eau minérale	101-L	wac1101a.gif	18.00 FRF	Contenance	1.5 l
10	9	Eau minérale naturelle	102-S	wac1100a.gif	6.00 FRF	Contenance	500 ml
11	9	Eau minérale naturelle	102-M	wac1100a.gif	12.00 FRF	Contenance	1 l

12	9	Eau minérale naturelle	102-L	wac1100a.gif	18.00 FRF	Contenance	1.5 l
----	---	------------------------	-------	--------------	-----------	------------	-------

Les en-têtes sont ceux des colonnes de la base de données d'où sont extraites les données. La requête extrait une ligne pour chaque attribut des produits appartenant à cette catégorie. La création des listes de sélection exige d'être soigneusement planifiée, car, pour un même produit, les différentes combinaisons contenance/prix doivent toutes apparaître dans la même liste. Le résultat obtenu ressemblera à l'exemple ci-dessous :

wac1102a.gif	Eau de source	500 ml @ FRF6.00
wac1101a.gif	Eau minérale	500 ml @ FRF6.00 1 l @ FRF12.00 1.5 l @ FRF18.00
wac1100a.gif	Eau minérale naturelle	500 ml @ FRF6.00 1 l @ FRF12.00 1.5 l @ FRF18.00

Pour créer le format de tableau ci-dessus, vous devez définir une variable "fanion" après le traitement de la première ligne de la table de résultats de la requête, et utiliser cette variable pour empêcher la création d'autres listes de sélection tant que tous les attributs du produit en cours n'ont pas été traités et qu'un nouveau produit n'a pas été détecté.

- Définissons cette variable fanion. Dans la section Define de la macro, ajoutez la définition de variable suivante :

```
%{ ***** Define Section ***** %}
%DEFINE {
productcounter = "0"
selectcounter = "1"
<!-->
<!-- Début du code -->
<!-->
product_flag = ""
<!-->
<!-- F in du code -->
<!-->
%}
```

Vous venez de définir une variable appelée `product_flag` et de lui donner une valeur nulle (chaîne vide). Elle sera utilisée ultérieurement pour déterminer dans quelles circonstances créer une nouvelle liste de sélection. Les autres variables définies seront abordées plus tard.

- Ajoutez le code HTML suivant au bloc ROW de la fonction `build_prod_list()` :

```
%ROW {
<TR><TD HEIGHT=5 COLSPAN=5><IMG SRC="/grocery/widespa2.gif"
WIDTH=750></TD>
<TR><TD><IMG SRC="/grocery/$(V_prthmb)" WIDTH=100 HEIGHT=100></TD>
<TD><B><FONT SIZE=-1>$(V_prsdsc)</FONT></B></TD>
<TD><SELECT NAME="select">
<OPTION VALUE="$(V_prrfnbr)">$(V_paval) @ $(V_ppprc) $(CURRENCY)
<!-->
<!-- Fin du code -->
<!-->
```

Vous venez d'ajouter des marques HTML identifiant le début d'une liste de sélection. Pour l'instant, cette liste se compose d'une seule option d'attribut. Vous n'avez pas encore inséré la marque de fermeture (`</SELECT>`) de la liste, car il reste à ajouter les deux autres attributs du produit en cours de traitement. Si vous fermiez cet élément maintenant, la prochaine ligne traitée par la requête

figurerait dans une liste de sélection distincte. Vous souhaitez au contraire que tous les attributs du produit en cours s'affichent dans la même liste de sélection. Pour cela, un critère est nécessaire afin de déterminer si la prochaine ligne à traiter est un attribut du même produit ou celui d'un nouveau produit. Puisque tous les attributs d'un produit partagent le même nom, vous pouvez utiliser le nom du produit comme qualificatif. Vous allez donc affecter la variable de base de données `V_prsdesc` à la variable `product_flag`, que vous avez définie au départ comme une chaîne vide.

Ajoutez l'instruction `Net.Data` suivante :

```
%ROW {
  <TR><TD HEIGHT=5 COLSPAN=5><IMG SRC="/grocery/widespa2.gif"
  WIDTH=550 ></TD>
  <TR><TD><IMG SRC="/grocery/${V_prthmb}" WIDTH=100 HEIGHT=100></TD>
  <TD><B><FONT SIZE=-1>${V_prsdesc}</FONT></B></TD>
  @DTW_ASSIGN(product_flag, V_prsdesc)
  <TD><SELECT NAME="select">
  <OPTION VALUE="${V_prrfnbr}">${V_paval} @ ${V_ppprc} $(CURRENCY)
  <!-- Fin du code -->
  <!-- Fin du code -->
  <!-- Fin du code -->
```

A cet endroit du bloc `ROW`, la variable `product_flag` reçoit comme valeur le nom du produit en cours de traitement.

4. Une instruction `IF` doit maintenant être ajoutée pour éviter la création, dans le tableau de produits, d'une ligne supplémentaire pour chaque autre attribut portant le même nom que le produit actuellement en cours de traitement.

Ajoutez les instructions suivantes avant et après le code `HTML` :

```
%ROW {
  %IF(product_flag != V_prsdesc)
  <TR><TD HEIGHT=5 COLSPAN=5><IMG SRC="/grocery/widespa2.gif"
  WIDTH=750></TD>
  <TR><TD><IMG SRC="/grocery/${V_prthmb}" WIDTH=100 HEIGHT=100></TD>
  <TD><B><FONT SIZE=-1>${V_prsdesc}</FONT></B></TD>
  @DTW_ASSIGN(product_flag, V_prsdesc)
  <TD><SELECT NAME="select">
  %ENDIF
  <OPTION VALUE="${V_prrfnbr}">${V_paval} @ ${V_ppprc} $(CURRENCY)
  %}
  <!-- Fin du code -->
  <!-- Fin du code -->
  <!-- Fin du code -->
```

Remarque : N'oubliez pas d'ajouter la marque de fin `%}` comme indiqué ci-dessus.

Dès lors, si la ligne en cours de traitement contient le même nom de produit que la dernière ligne traitée, le code `HTML` situé entre l'instruction `%IF` et l'instruction `%ENDIF` sera ignoré et une autre option d'attribut sera créée pour la liste de sélection.

5. Vous devez maintenant définir comment la liste de sélection se terminera lorsque toutes les lignes contenant le même nom de produit auront été traitées et qu'une ligne contenant un nouveau nom de produit sera détectée. Ajoutez le code `HTML` suivant au début du bloc `ROW` :

```
%ROW {
  %IF((product_flag != V_prsdesc) && (${ROW_NUM} != "1"))
  </SELECT></TD>
  %ENDIF
  %IF(product_flag != V_prsdesc)
```

```

<TR><TD HEIGHT=5 COLSPAN=5><IMG SRC="/grocery/widespa2.gif"
WIDTH=550 ></TD>
<TR><TD><IMG SRC="/grocery/$(V_prthmb)" WIDTH=100 HEIGHT=100></TD>
<TD><B><FONT SIZE=-1>$(V_prsdesc)</FONT></B></TD>
@DTW_ASSIGN(product_flag, V_prsdesc)
<TD><SELECT NAME="select">
%ENDIF
<OPTION VALUE="$(V_prrfnbr)">$(V_paval) @ $(V_ppprc) $(CURRENCY)
%}
<!-->
<!-- Fin du code -->
<!-->

```

Ce nouveau bloc d'instructions IF indique que lorsqu'une ligne contenant un nouveau nom de produit est détectée (ce nom étant alors différent de la valeur actuellement associée à la variable `product_flag`), les informations incluses entre cette instruction `%IF` et l'instruction `%ENDIF` seront traitées. Elles mettront fin à la liste de sélection. La condition `$(ROW_NUM) != "1"` est la suivante : si le numéro de la ligne en cours de traitement n'est pas égal à 1, il faut traiter les informations comprises entre les instructions `%IF` et `%ENDIF`. Cette condition est nécessaire pour la première ligne extraite par la requête, car dans ce cas cette section est ignorée et le traitement se poursuit avec le second bloc d'instructions IF.

6. Enregistrez votre travail.

Ajout de la zone Quantité et de l'icône Liste d'achats

Vous allez à présent ajouter les deux dernières colonnes du tableau de produits : la zone de texte permettant d'indiquer une quantité et l'icône permettant d'ajouter le produit concerné à la liste d'achats.

1. Ajoutez le code HTML suivant au bloc ROW de la section HTML Report :

```

%ROW {
%IF((product_flag != V_prsdesc) && ($(ROW_NUM) != "1"))
</SELECT></TD>
<TD ALIGN="CENTER"><INPUT TYPE="TEXT" NAME="textbox" VALUE=0 SIZE=3
MAXLENGTH=3
onFocus="this.select()"></TD>
<TD ALIGN="CENTER"><A HREF="#"
onClick=shopc_add("$(productcounter)") onMouseOver="changeStatus();
return true">
<IMG SRC="/grocery/list1.gif" HEIGHT=64 WIDTH=75 BORDER=0></A></TD>
</TR>
%ENDIF
%IF(product_flag != V_prsdesc)
@DTW_ADD(productcounter, "1", "1", productcounter)
<TR><TD HEIGHT=5 COLSPAN=5><IMG SRC="/grocery/widespa2.gif"
WIDTH=750></TD>
<TR><TD><IMG SRC="/grocery/$(V_prthmb)" WIDTH=100 HEIGHT=100></TD>
<TD><B><FONT SIZE=-1>$(V_prsdesc)</FONT></B></TD>
@DTW_ASSIGN(product_flag, V_prsdesc)
<TD><SELECT NAME="select">
%ENDIF
<OPTION VALUE="$(V_prrfnbr)">$(V_paval) @ $(V_ppprc) $(CURRENCY)
%}
<!-->
<!-- Fin du code -->

```

<!-->

Dans la quatrième colonne, vous venez d'ajouter une zone d'entrée de formulaire HTML permettant la saisie de la quantité souhaitée pour un article.

Dans la cinquième colonne, vous avez ajouté l'image (ou icône) de la liste d'achats. Le comportement de cette image dépend de l'action entreprise à l'aide de la souris. Si le client place simplement le pointeur de la souris sur l'image, une fonction JavaScript est appelée et affiche un message décrivant la fonction de l'icône, sur la barre d'état. Si le client clique sur l'image, le produit sélectionné est ajouté à la liste d'achats, selon le déroulement suivant :

- a. La fonction `shopc_add()` de la section HTML de la macro est appelée, et la valeur du paramètre `productcounter` lui est transmise. Le paramètre `productcounter` permet de répertorier le nombre de produits différents affichés sur la page. Chaque fois qu'une liste de sélection est créée, un numéro identifiant un nouveau produit lui est affecté.
- b. La fonction utilise le paramètre `productcounter` pour localiser la ligne contenant le produit à ajouter à la liste d'achats.
- c. Elle détermine quelle est la valeur sélectionnée dans la liste de sélection à l'aide du numéro d'index, puis extrait la référence de produit correspondante.
- d. Ensuite, la fonction crée un URL contenant l'instruction `InterestItemAdd` pour ajouter le produit sélectionné à la liste d'achats.
- e. A l'issue de l'exécution de l'instruction `InterestItemAdd`, la fonction `shopc_add()` appelle un autre URL contenant l'instruction `CategoryDisplay` pour réafficher la page de produit en cours.
- f. Lorsque cette page réapparaît, un message de confirmation s'affiche sur la barre d'état du navigateur. (Une fonction JavaScript est appelée à cet effet dans la section HTML Report.)

2. Enregistrez votre travail.

Examen du tableau de produits

Voyons comment notre tableau de produits sera créé. Pour cela, étudions le mode de traitement de chaque ligne de la table de résultats de la requête :

1. **Ligne 1** : Le bloc consécutif à la première instruction IF est ignoré, car la condition n'est pas vérifiée (le numéro de cette ligne est effectivement égal à 1).
2. La condition énoncée par la deuxième instruction IF est vérifiée : la variable `product_flag` n'est pas égale à la valeur de `V_prsdesc`. (Cette variable est pour l'instant une chaîne vide, affectée lors du chargement de la macro.) Voici ce qui se produit :
 - a. La barre horizontale s'affiche dans une nouvelle ligne du tableau de produits.
 - b. L'image miniature et le nom du produit s'affichent dans la première et la deuxième cellule de la ligne suivante.
 - c. Le début d'un formulaire HTML est défini.
 - d. La valeur du nom du produit en cours (`V_prsdesc`) est affectée à la variable `product_flag`.
 - e. La liste de sélection commence à la troisième cellule.
 - f. L'attribut de contenance et le prix du produit sont combinés dans une chaîne pour former la première option de la liste de sélection.

3. **Ligne 2** : La fin du bloc ROW est atteinte et on entame alors le traitement de la deuxième ligne. Celui-ci reprend au début de la section ROW.
4. La ligne 2 étant toujours un attribut du même produit, le nom de produit qu'elle contient est égal à la valeur de V_prsdesc. Par conséquent, aucune des conditions énoncées par les instructions IF n'est vérifiée, si bien que seule l'instruction située en dehors de ces instructions est exécutée : l'attribut de contenance et le prix du produit sont combinés dans une chaîne pour former la deuxième option de la liste de sélection.
5. **Ligne 3** : La fin du bloc ROW est atteinte et on entame alors le traitement de la troisième ligne. Celui-ci reprend au début de la section ROW.
6. La ligne 3 contenant le troisième attribut du produit, son traitement est identique à celui de la ligne 2. L'attribut et le prix correspondant sont insérés dans la liste de sélection comme troisième option.
7. **Ligne 4** : La fin du bloc ROW est atteinte et on entame alors le traitement de la quatrième ligne. Celui-ci reprend au début de la section ROW.
8. La ligne 4 contenant un nouveau produit avec un nouveau nom, la condition énoncée par la première instruction IF est maintenant vérifiée. Voici ce qui se produit :
 - a. La liste de sélection est terminée (refermée).
 - b. La zone Quantité est affichée.
 - c. L'icône Liste d'achats est affichée.
9. Le traitement se poursuit jusqu'à la deuxième instruction IF, dont la condition est également vérifiée. Le contenu de ce bloc est traité comme décrit à l'étape 2 du traitement de la **Ligne 1**.
10. **Ligne 5** : La fin du bloc ROW est atteinte et on entame alors le traitement de la cinquième ligne, qui est identique à celui de la ligne 2.
11. Toutes les autres lignes sont traitées de manière analogue, par rebouclages successifs sur les étapes appropriées.

Fin du tableau de produits

Lorsque la dernière ligne de la table de résultats de la requête est traitée, aucune autre ligne ne peut être appelée pour terminer la liste de sélection. Il faut donc ajouter du code pour terminer la ligne en affichant la zone Quantité et l'icône Liste d'achats, puis pour terminer (fermer) le tableau.

1. Ajoutez le code HTML suivant après le bloc ROW :


```
%ROW {
%IF((product_flag != V_prsdesc) && ($(ROW_NUM) != "1"))
  </SELECT></TD>
  <TD ALIGN="CENTER"><INPUT TYPE="TEXT" NAME="textbox" VALUE=0
SIZE=3 MAXLENGTH=3
onFocus="this.select()"></TD>
  <TD ALIGN="CENTER"><A HREF="#"
onClick=shopc_add("$(productcounter)")
onMouseOver="changeStatus(); return true">
  <IMG SRC="/grocery/list1.gif" HEIGHT=64 WIDTH=75
BORDER=0></A></TD>
</TR>
%ENDIF
%IF(product_flag != V_prsdesc)
  @DTW_ADD(productcounter, "1", "1", productcounter)
  <TR><TD HEIGHT=5 COLSPAN=5><IMG SRC="/grocery/widespa2.gif"
WIDTH=750></TD>
```

```

        </TR><TD><IMG SRC="/grocery/$(V_prthmb)" WIDTH=100
HEIGHT=100></TD>
        <TD><B><FONT SIZE=-1>$(V_prsdesc)</FONT></B></TD>
        @DTW_ASSIGN(product_flag, V_prsdesc)
        <TD><SELECT NAME="select">
%ENDIF
<OPTION VALUE="$(V_prrfnbr)">$(V_paval) @ $(V_ppprc) $(CURRENCY)
%}
</SELECT></TD>
<TD ALIGN="CENTER"><INPUT TYPE="TEXT" NAME="textbox" VALUE=0 SIZE=3
MAXLENGTH=3
onFocus="this.select()"></TD>
<TD ALIGN="CENTER"><A HREF="#"
onClick=shopc_add("$(productcounter)") onMouseOver="changeStatus();
return true">
<IMG SRC="/grocery/list1.gif" HEIGHT=64 WIDTH=75 BORDER=0></A></TD>
</TR>
<TR><TD HEIGHT=5 COLSPAN=5><IMG SRC="/grocery/widespa2.gif"
WIDTH=750></TD></TR>
</TABLE>
<!-->
<!-- Fin du code -->
<!-->

```

Ce code HTML est semblable à celui du premier bloc IF. Il met simplement fin à la liste de sélection, crée la zone Quantité, affiche l'icône Liste d'achats et insère une nouvelle ligne contenant une barre noire sur toute la largeur du tableau. Le tableau est ensuite refermé.

2. Enregistrez votre travail.

Création du bouton Ajouter au chariot

Vous voici prêt à créer le bouton **Ajouter au chariot** dans votre macro de produit. Lorsque le client cliquera sur ce bouton, tous les articles préalablement sélectionnés seront ajoutés au chariot. La fonction de ce bouton est complexe et nécessite du code JavaScript pour pouvoir exécuter plusieurs actions simultanément. Il fonctionnera comme un bouton de validation ("Envoi") de formulaire HTML. Puisqu'il s'agit d'un élément de formulaire, vous devez créer un formulaire HTML et définir un URL ACTION qui traitera le contenu de ce formulaire lorsque le client cliquera sur le bouton. Vous avez déjà créé deux éléments de formulaire : la liste de sélection des attributs de produit et la zone d'entrée de texte permettant d'indiquer une quantité. Les valeurs précisées par le client à l'aide de ces éléments seront également traitées par notre bouton de validation.

1. Ajoutez le code HTML suivant à la section Report de la fonction build_prod_list() :
- ```

%ROW {
%IF((product_flag != V_prsdesc) && ($(ROW_NUM) != "1"))
</SELECT></TD>
<SCRIPT LANGUAGE="JavaScript">
for (i=1; i < (parseInt('$(selectcounter)')); i++) {
document.writeln("<INPUT TYPE=\"HIDDEN\" NAME=\"attrib\" + i +
\"\"\" + \"
VALUE=0>")
}
</SCRIPT>
@DTW_ASSIGN(selectcounter, "1")
<TD ALIGN="CENTER"><INPUT TYPE="TEXT" NAME="textbox" VALUE=0

```

```

SIZE=3
 MAXLENGTH=3
 onFocus="this.select()"></TD>
 <TD ALIGN="CENTER"><A HREF="#"
 onClick=shopc_add("${productcounter}")
onMouseOver="changeStatus();
 return true">
 <IMG SRC="/grocery/list1.gif" HEIGHT=64 WIDTH=75
BORDER=0></TD>
</TR>
%ENDIF
%IF(product_flag != V_prsdesc)
 @DTW_ADD(productcounter, "1", "1", productcounter)
 <TR><TD HEIGHT=5 COLSPAN=5><IMG SRC="/grocery/widespa2.gif"
WIDTH=600 ></TD>
 <TR><TD></TD>
 <TD>${V_prsdesc}</TD>
 <FORM METHOD="POST" ACTION="/cgi-bin/ncommerce3/OrderItemProcess"
 NAME="list">
 @DTW_ASSIGN(product_flag, V_prsdesc)
 <TD><SELECT NAME="select">
%ENDIF
<OPTION VALUE="${V_prrfnbr}">${V_paval} @ ${V_pprc} $(CURRENCY)
@DTW_ADD(selectcounter, "1", "1", selectcounter)
%}
</SELECT></TD>
<SCRIPT LANGUAGE="JavaScript">
for (i=1; i < (parseInt('${selectcounter}')); i++) {
 document.writeln("<INPUT TYPE=\"HIDDEN\" NAME=\"attrib\" + i + \"\"
+ "
 VALUE=0>")
}
</SCRIPT>
@DTW_ASSIGN(selectcounter, "1")
<TD ALIGN="CENTER"><INPUT TYPE="TEXT" NAME="textbox" VALUE=0 SIZE=3
 MAXLENGTH=3
onFocus="this.select()"></TD>
<TD ALIGN="CENTER"><A HREF="#"
onClick=shopc_add("${productcounter}") onMouseOver="changeStatus();
 return true">
</TD>
</TR>
<TR><TD HEIGHT=5 COLSPAN=5><IMG SRC="/grocery/widespa2.gif"
WIDTH=600></TD></TR>
</TABLE>
<INPUT TYPE="HIDDEN" NAME="addr_rn" VALUE="${addr_rn}">
<INPUT TYPE="HIDDEN" NAME="merchant_rn" VALUE="${V_mmmenbr}">
<INPUT TYPE="HIDDEN" NAME="shpmthd" VALUE="${V_mmrfnbr}">
<INPUT TYPE="HIDDEN" NAME="url" VALUE="/cgi-
bin/ncommerce3/CategoryDisplay?
cgrfnbr=${cgrfnbr}&cgmenbr=${cgmenbr}&success1=true">

& lt;CENTER><INPUT TYPE="SUBMIT" NAME="addtocart"
VALUE="Ajouter au chariot"
onClick="update_prodef(this.form)"></CENTER>
</FORM>
%}

```

```
<!--> <!-- Fin du code -->
<!-->
```

**Remarque :** Ne laissez aucun espace dans l'instruction `/cgi-bin/ncommerce3/CategoryDisplay?cgrfnbr=$(cgrfnbr)&cgmenbr=$(cgmenbr)&success1=true` (valeur de l'url de la zone d'entrée cachée).

2. Enregistrez votre travail.

## Examen du bouton Ajouter au chariot

Examinons le code que vous avez ajouté pour le bouton **Ajouter au chariot**. Vous venez de créer un formulaire comportant un bouton de validation ("Envoi") qui déclenche les opérations suivantes lorsque le client clique dessus :

1. La fonction `update_prodef()` de la section HTML Report de la macro est appelée, et les valeurs des autres éléments du formulaire (la zone d'entrée de texte et la liste de sélection) lui sont transmises.
2. La fonction `update_prodef()` crée une chaîne de paires nom-valeur correspondant chacune à un article présenté sur la page. Pour chaque article sélectionné par le client, elle affecte la référence de produit correspondante à la partie nom, et la quantité indiquée dans la boîte de texte à la partie valeur. Pour tous les autres articles (ceux que le client n'a pas sélectionnés), elle donne à la partie nom la référence de produit correspondante et affecte la quantité 0 à la partie valeur. Ces paires nom-valeur remplacent celles de chaque attribut dans les listes de sélection. Vous avez ajouté, après chaque liste de sélection, une fonction JavaScript qui crée les emplacements réservés initiaux (`name=attrib1 value=0, name=attrib2 value=0, name=attrib3 value=0`) pour ces attributs.
3. La fonction exécute ensuite l'URL ACTION du formulaire, qui appelle l'instruction `OrderItemProcess`. Celle-ci insère dans la table SHIPTO un enregistrement par quantité de produits sélectionnée par le client :
  - a. Elle reçoit les paramètres suivants de la macro de produit :
    - La référence du produit et la quantité demandée, le tout sous forme de paire nom-valeur, pour chaque article présenté sur la page produit (paramètre obligatoire).
    - La référence d'adresse du client (paramètre obligatoire pour une adresse d'expédition).
    - La référence du mode d'expédition du commerçant et celle du commerçant (paramètres obligatoires pour le calcul des frais de port lorsque le client passe ensuite sa commande).
  - b. Elle insère la référence du mode d'expédition du commerçant et celle du commerçant dans la table SHIPTO.
  - c. Pour chaque paire nom-valeur fournie, constituée d'une référence de produit et d'une quantité, elle exécute les étapes suivantes pour insérer un enregistrement par produit dans la table SHIPTO :
    - 1) Elle ne tient pas compte des quantités égales à 0 (articles non sélectionnés par le client).
    - 2) Elle appelle des activités de traitement pour vérifier que la quantité demandée est en stock et pour extraire le prix actuel du produit ou de l'article.
    - 3) Elle crée une association entre le produit et l'adresse d'expédition fournie.
    - 4) Elle appelle l'activité de traitement `EXT_SHIPTO_PROC`. Par défaut, celle-ci n'effectue aucun traitement supplémentaire ; elle appelle

simplement la fonction non prioritaire par défaut DoNothingNoArgs, qui ne fait rien. Cette dernière a été remplacée par une fonction appelée NCApiExtendShiptoProcess, qui regroupe en une seule ligne les quantités d'articles identiques dans la table SHIPTO. Normalement, la table SHIPTO contient un enregistrement distinct pour chaque quantité d'articles différente spécifiée par le client, même si tous les autres attributs des articles sont identiques. En somme, si le client ajoute une première fois deux articles identiques au chariot, et ensuite trois autres, la table SHIPTO comportera deux enregistrements : un enregistrement avec la quantité 2 et l'autre avec la quantité 3. La fonction NCApiExtendShiptoProcess combine ces deux enregistrements dans la table SHIPTO afin qu'ils ne forment plus qu'un seul enregistrement contenant la quantité 5.

- 5) Lorsque l'instruction OrderItemProcess aboutit, elle appelle l'URL caché contenant l'instruction CategoryDisplay afin de réafficher la page produit en cours. Pour identifier la macro appropriée à exécuter, cette instruction utilise la référence de catégorie et la référence de commerçant qui lui sont fournies.
- 6) Lorsque la page apparaît, la barre d'état située dans le bas affiche un message de confirmation indiquant que les produits ont bien été ajoutés au chariot. (Une fonction JavaScript est appelée à cet effet dans la section HTML Report de la macro.)

---

## Fin de la macro de produit

Pour terminer votre macro de produit, vous devez appeler les fonctions que vous avez écrites afin de créer les éléments des pages produit : la publicité et la bannière, le tableau de produits et le bouton Ajouter au chariot.

1. Ajoutez les appels de fonction suivants dans la section HTML Report :

```
%{ ***** HTML Report Section ***** %}
%HTML_REPORT
{
<HTML>
<HEAD>
</HEAD>
<BODY BGCOLOR=#FFFFFF TEXT=#000000 LINK=#399FF VLINK=#A7A7A7>
<!-- Début du code -->
<!--
 @display_ad()
 @find_addr_rn()
 @get_subcat_name()
 @build_prod_list()
-->
<!-- Fin du code -->
<!--
```

Ce code indique l'ordre d'exécution des différentes fonctions de construction de la page produit.

2. Supprimez du fichier tous les blocs de commentaire marquant le début et la fin des portions de code que vous avez insérées.
3. Enregistrez le fichier sous le nom t\_toc\_item.d2w.
4. Ne fermez pas le fichier, car vous allez bientôt y revenir. La macro de produit est à présent terminée.

---

## Test de la macro de produit

Voyons comment vos pages produit se présenteront dans le magasin. Pour cela, vous allez accéder à une page produit, sélectionner quelques articles et les ajouter au chariot ainsi qu'à votre liste d'achats.

1. Assurez-vous que la fenêtre du navigateur affichant l'Administrateur Net.Commerce est fermée.
2. Ouvrez une nouvelle fenêtre de navigateur.
3. Accédez au magasin à partir de votre liste de signets.
4. Si vous y êtes invité, acceptez le cookie de session temporaire qui vous est affecté.
5. Vous devez d'abord vous connecter au magasin en tant que client référencé.
6. A partir de la page d'accueil, cliquez sur **Connexion**.
7. Sur la page **Connexion**, entrez le nom d'utilisateur et le mot de passe que vous avez déjà utilisés pour vous connecter au magasin, puis cliquez sur **Connexion**.
8. Acceptez le nouveau cookie de session qui vous est affecté et qui tient compte de votre nom d'utilisateur référencé.
9. Dans la liste de catégories, cliquez sur **Boissons**.
10. Sur la page de sous-catégories obtenue, placez le curseur sur **Boissons non alcoolisées**. Prenez note de l'URL qui s'affiche sur la barre d'état, dans le bas de la fenêtre. Les produits qui appartiennent à ces sous-catégories étant affichés dans une table et non sur des pages distinctes, ce qui correspond au format fourni par une macro de catégorie, ils sont affichés à l'aide de la même instruction que celle qui appelle les macros de catégorie : `CategoryDisplay`.
11. Cliquez sur **Boissons non alcoolisées**. Une page produit s'affiche.
12. Si la page ne s'affiche pas correctement, ou si vous obtenez des messages d'erreur à cette étape ou lors des étapes suivantes, reportez-vous à la rubrique ci-dessous **Résolution des incidents**.
13. En regard de l'un des produits, choisissez une contenance dans la liste de sélection.
14. Entrez la quantité 3.
15. Sur la ligne d'un autre produit, sélectionnez une contenance.
16. Entrez la quantité 4.
17. Dans le bas de la page, cliquez sur **Ajouter au chariot**. Les deux produits sont ajoutés au chariot et la page est régénérée.
18. Notez le message de confirmation qui s'affiche sur la barre d'état, dans le bas de la fenêtre du navigateur.
19. A droite des deux produits ajoutés au chariot, cliquez sur l'icône **Ajouter à la liste d'achats** (qui représente un bloc-notes et un crayon). Ces produits sont également ajoutés à votre liste d'achats personnelle.
20. Notez qu'un message de confirmation s'affiche sur la barre d'état.
21. Vous remarquerez que la zone **Montant total**, dans le cadre supérieur gauche du magasin, n'affiche pas le prix total des articles se trouvant actuellement dans votre chariot. Vous activerez cette zone à la section suivante.
22. Ajoutez deux autres produits à votre chariot.
23. A présent, prenons un moment pour examiner votre liste d'achats. Dans le cadre de navigation, cliquez sur **Liste d'achats**. Les articles que vous venez d'y ajouter figurent sur la page.  
La liste d'achats précise les articles que vous avez sélectionnés à partir de la

page produit. Lors de vos visites ultérieures du magasin, vous pourrez ajouter des articles à votre chariot en les sélectionnant directement à partir de cette liste, au lieu de parcourir à chaque fois les pages de catégorie et de produits. Votre parcours sera ainsi plus efficace, car vous pourrez vous servir de cette liste pour sélectionner rapidement les articles que vous achetez régulièrement, tels que les provisions de la semaine : lait, pain, oeufs, par exemple. Ces articles resteront dans la liste jusqu'à ce que vous les en retiriez. Seuls les clients référencés peuvent disposer d'une liste d'achats personnelle.

### Résolution des incidents

- Si vous obtenez un message d'erreur lorsque vous cliquez sur le bouton **Ajouter au chariot**, vérifiez dans le fichier source (`t_toc_item.d2w`) qu'il n'y a pas d'espace dans l'instruction `CategoryDisplay`, ni dans ses paramètres. S'il y en a, supprimez-les et renouvelez les étapes 9 à 16.
- S'il se produit d'autres erreurs, vous avez peut-être mal entré le code dans la macro. Une version définitive de celle-ci est disponible dans le même répertoire que celui où est stocké votre fichier modifié. Pour remplacer la macro de produit par la version définitive, effacez le fichier `t_toc_item.d2w` et faites une copie de `f1_toc_item.d2w` sous le nom `t_toc_item.d2w`. Renouvelez ensuite les étapes 9 à 22.



---

## Chapitre 10. Activation de la zone Montant total

Vous devez à présent activer la zone **Montant total**, dans le cadre supérieur gauche du magasin, pour qu'elle présente le prix total de tous les produits que vous venez d'ajouter à votre chariot.

### Ce que vous allez faire

Dans cette section, vous allez effectuer les opérations suivantes :

1. Calculer le prix total des articles présents dans le chariot.
2. Mettre à jour la zone Montant total pour qu'elle affiche ce prix.

### Pour passer cette section

Si vous choisissez de ne pas suivre cette section, vous devez effectuer les opérations suivantes :

1. Effacez le fichier `t_totalAmount.in` et faites une copie de `f_totalAmount.inc` sous le nom `t_totalAmount.inc`.
2. Effacez le fichier `\macro\<langue>\category\t_toc_item.d2wet` faites une copie de `\macro\<langue>\category\f2_toc_item.d2w` sous le nom `\macro\<langue>\category\t_toc_item.d2w`.
3. Passez à l'étape Création du chariot.

---

## Calcul du prix total

Tout d'abord, vous allez calculer le prix total de tous les articles présents dans le chariot.

1. Assurez-vous que vous avez pris connaissance des informations relatives à l'édition des modèles du tutoriel.
2. A l'aide d'un éditeur de texte, ouvrez le fichier d'inclusion du tutoriel `tmp_totalAmount.inc`:
3. Etudiez le contenu du fichier. Le nom de la fonction que vous allez écrire est déjà défini dans la section Function Definition (`do_totalAmount()`). Maintenant, vous allez ajouter des instructions SQL permettant d'extraire le prix total de l'ensemble des articles présents dans le chariot, comme suit :

```
%{ ***** Function Definition Section
***** %}
%FUNCTION(DTW_ODBC) do_totalAmount1()
{
<!-->
<!-- Début du code -->
<!--> & gt;
 select distinct sum(stprice*stquant) as current_total
 from shipto, shopper
 where ststat='P'
 and stshnbr=shrfnbr
 and shlogid='${SESSION_ID}'
<!-->
<!-- Fin du code -->
<!--> ;
```

Cette requête SQL extrait de la table SHIPTO (utilisée pour stocker le contenu du chariot) le prix total de chaque article en multipliant son prix unitaire par la quantité demandée. Elle fait ensuite la somme des résultats pour obtenir le

montant total de tous les articles présents dans cette table. Une colonne `current_total` est créée pour recevoir le résultat final du calcul.

- Vous devez alors affecter un nom de variable à la colonne `current_total`. Cette variable est référencée par les trois macros (chariot, produit et liste d'achats) qui doivent mettre à jour la zone **Montant total**. Ajoutez la section Report suivante :

```
%FUNCTION(DTW_ODBC) do_totalAmount1()
{
<!-->
<!-- Début du code -->
<!-->
select distinct sum(stprice*stquant) as current_total
from shipto, shopper
where ststat='P'
and stshnbr=shrfnbr
and shlogid='$(SESSION_ID)'
%REPORT {
%ROW {
%IF (V_current_total == NULL)
@DTW_ASSIGN(running_total, "0")
%ELSE
; @DTW_ASSIGN(running_total, V_current_total)
%ENDIF
%}
%}
<!-->
<!-- Fin du code -->
<!--> ;
%MESSAGE {
100: { Erreur SQL : $(SQL_CODE) %} :continue
default: { Erreur SQL : $(SQL_CODE) %} :continue
%}
%}
```

Cette section exécute l'opération suivante : si la valeur de la colonne `current_total` est NULL (aucune valeur extraite), une variable Net.Data appelée `running_total` est définie et la valeur 0 lui est affectée ; sinon, la variable `running_total` est définie et la valeur de la colonne `current_total` lui est affectée. La première condition est vérifiée si le chariot ne contient aucun article.

- Supprimez du fichier tous les blocs de commentaire marquant le début et la fin des portions de code que vous avez insérées.
- Enregistrez le fichier sous le nom `t_totalAmount.inc`.
- Cliquez sur **Oui** pour remplacer le fichier existant.
- Fermez le fichier.

---

## Insertion du fichier d'inclusion dans la macro de produit

Vous devez maintenant insérer (ou plutôt appeler) le fichier d'inclusion `t_totalAmount.inc` dans la macro de produit. Ce fichier sera partagé avec les autres macros qui doivent également mettre à jour la zone **Montant total**, à savoir : la macro du chariot et celle de la liste d'achats. L'utilisation d'un fichier d'inclusion est un moyen simple d'insérer le même code dans plusieurs macros.

- Revenez dans la macro de produit `t_toc_item.d2w`, qui se trouve dans le répertoire suivant :

**WIN** \ibm\netcommerce3\macro\<langue>\category\

**AIX** /usr/lpp/NetCommerce3/macro/<langue>/category/

**SUN** /opt/IBMnetc/NetCommerce3/macro/<langue>/category/

2. Ajoutez l'instruction suivante à la section Include de cette macro :

```
%{ ***** Include Section ***** %}
%INCLUDE "/ncsample/currency.inc"
%INCLUDE "/tutorial/t_totalAmount.inc"
```

Cette instruction importe le code du fichier `t_totalAmount.inc` dans la macro.

3. Il faut maintenant appeler la fonction `do_totalAmount1()` codée dans ce fichier d'inclusion, pour qu'elle calcule la valeur de la variable `running_total`. Puis il faut transmettre cette valeur à la fonction `refreshTotal()`, dans le cadre supérieur gauche du magasin. Dans la section HTML Report, ajoutez le code suivant après les appels de fonction :

```
%{ ***** HTML Report Section ***** %}
%HTML_REPORT
{
<HTML>
 <HEAD>
</HEAD>
<BODY BGCOLOR=#FFFFFF TEXT=#000000 LINK=#399FF VLINK=#A7A7A7>
 @display_ad()
 @find_addr_rn()
 ; @get_subcat_name()
 @build_prod_list()
 @do_totalAmount1()
 <SCRIPT LANGUAGE="JavaScript">
 parent.frames[0].refreshTotal("${running_total}");
 </SCRIPT>
```

La première instruction appelle la fonction `do_totalAmount1()` dans le fichier d'inclusion, pour qu'elle calcule le prix en cours. La fonction JavaScript qui suit appelle la fonction `refreshTotal()` dans le cadre supérieur gauche en référant le numéro d'index du cadre correspondant à `t_navtop.html` et lui transmet la valeur de la variable `running_total`.

4. Enregistrez votre travail et fermez le fichier.

---

## Examen et test de votre travail

Voyons, étape par étape, comment le système calcule le prix total des articles présents dans le chariot, puis met à jour la zone **Montant total** en conséquence. Nous testerons ensuite le fonctionnement de cette zone dans le magasin.

1. Une fois qu'un client a ajouté des articles dans le chariot à partir d'une page produit et que celle-ci a été réaffichée, la fonction `do_totalAmount1()` du fichier d'inclusion est appelée.
2. La fonction `do_totalAmount1()` calcule le nouveau prix total (stocké dans la variable `running_total`) de tous les articles présents dans le chariot (table SHIPTO).
3. La dernière fonction JavaScript que vous avez ajoutée à la macro de produit appelle la fonction `refreshTotal()` dans le cadre supérieur gauche et lui transmet la valeur de la variable `running_total`.
4. La fonction `refreshTotal()` met à jour la zone **Montant total** pour y faire apparaître la nouvelle valeur reçue. Chaque fois que la macro de produit est appelée, la zone **Montant total** est régénérée.

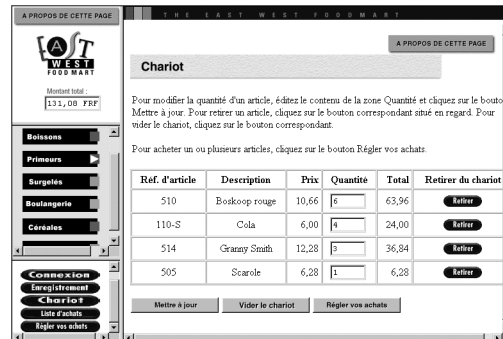
5. A présent, testons le travail effectué. Retournez au magasin et allez jusqu'à la page produit **Boissons non alcoolisées**.
6. Sélectionnez un produit, entrez une quantité et cliquez sur **Ajouter au chariot**.
7. Observez la zone **Montant total**. Elle doit maintenant afficher le montant total de tous les articles que vous avez ajoutés à votre chariot.

#### **Résolution des incidents**

Si la mise à jour de la zone **Montant total** ne se déroule pas correctement, vérifiez le code que vous avez ajouté dans le fichier d'inclusion et dans la macro de produit. Si l'affichage demeure incorrect, vous disposez d'une version définitive des fichiers, dans le même répertoire que ceux que vous venez de modifier. Pour remplacer la macro de produit par la version définitive, effacez le fichier `t_toc_item.d2w` et faites une copie de `f2_toc_item.d2w` sous le nom `t_toc_item.d2w`. Pour remplacer le fichier d'inclusion, effacez le fichier `t_totalAmount.inc` et faites une copie de `f_totalAmount.inc` en `t_totalAmount.inc`. Renouvelez ensuite les étapes 5 à 7.

## Chapitre 11. Création du chariot

Vous allez maintenant créer la page qui, dans votre magasin, représentera le chariot du client. Elle ressemblera à l'illustration ci-dessous :



Le chariot présente les avantages suivants :

- Il dresse la liste de tous les articles sélectionnés par le client.
- Il indique la quantité choisie pour chaque article (cette donnée étant tirée automatiquement de la page produit où ce choix a été opéré).
- Le client peut à tout moment modifier la quantité des articles placés dans le chariot.
- Le client peut retirer des articles de son chariot.
- Tous les articles du chariot sont considérés comme faisant partie d'une seule et même commande. Ils feront l'objet d'un seul règlement.

Si un client s'enregistre après avoir ajouté des articles au chariot, il ne perd pas pour autant le contenu de ce dernier : les articles sélectionnés sont automatiquement transférés de son ID temporaire au nouvel ID de connexion indiqué lors de son enregistrement.

### Ce que vous allez faire

Dans cette section, vous allez effectuer les opérations suivantes :

- Afficher la page du chariot telle qu'elle se présente actuellement.
- Etudier la macro de création de cette page.
- Ajouter la zone **Quantité** à la macro pour afficher dynamiquement la quantité de chaque article présent dans le chariot.
- Ajouter le bouton **Retirer** à la macro pour supprimer des articles du chariot.
- Activer les modifications apportées à la macro.
- Tester le travail effectué.

### Pour passer cette section

Si vous choisissez de ne pas suivre cette section, vous devez effectuer les opérations suivantes :

1. Effacez le fichier tmp\_shptodsp.d2w et faites une copie de f\_shptodsp.d2w en tmp\_shptodsp.d2w.
  2. Passez à la section Création d'une fonction non prioritaire.
- Chariot

---

## Visualisation de l'état actuel de la page Chariot

Avant de commencer à modifier le modèle de la page chariot, voyons à quoi ressemble cette page pour l'instant.

1. Revenez à la fenêtre du navigateur dans laquelle le magasin est affiché.
2. Dans le cadre de navigation, cliquez sur **Chariot**. La page Chariot s'affiche.
3. Vous remarquerez que le tableau présentant le contenu du chariot ne comporte pas encore de colonnes permettant de modifier la quantité des articles et de supprimer ceux-ci. Vous allez bientôt les ajouter.

---

## Etude de la macro du chariot

Avant de commencer à ajouter la zone Quantité et le bouton Retirer, vous allez examiner le code existant pour comprendre comment la macro fonctionne.

1. A l'aide d'un éditeur de texte, ouvrez le modèle de tutoriel tmp\_shptodsp.d2w.
2. Examinez tout d'abord la section HTML Report de la macro. Faites défiler le code jusqu'à cette section, qui commence ainsi :

```
%{ ***** HTML Report Section ***** %}
%HTML_REPORT
{
<HTML>
 <HEAD><TITLE>Révision du contenu du chariot</TITLE></HEAD>
 @do_totalAmount1()
 <SCRIPT LANGUAGE="JavaScript">
 parent.frames[0].refreshTotal("${running_total}");
 </SCRIPT>
 @display_table()
```

La fonction `do_totalAmount1()` et le code JavaScript qui lui fait suite se chargent de mettre à jour la zone **Montant total**, dans le cadre supérieur gauche, pour y afficher le prix total des articles présents dans le chariot. Ce code est identique à celui que vous avez ajouté à la macro de produit, dans la section précédente. La procédure d'ajout à la présente macro est la même.

La dernière fonction, `display_table()`, extrait le contenu du chariot et l'affiche dans un tableau HTML.

## Etude de la fonction de création du tableau du chariot

Nous allons à présent examiner la fonction `display_table()`, qui crée le tableau de la page du chariot. Ce tableau ressemblera à celui qui est illustré ci-après :



Pour retirer un article, cliquez sur "Retirer" en regard de cet article.

Pour vider le chariot, cliquez sur le bouton Vider le chariot.

<P> Une fois prêt à acheter un ou plusieurs articles, cliquez sur le bouton Régler vos achats.

```
</TD>
</TR>
</TABLE>
<FORM ACTION="/cgi-bin/ncommerce3/OrderItemUpdate"
NAME="shiptolist">
<TABLE BORDER=1 WIDTH=600 CELLPADDING=4 CELLSPACING=1>
<TR>
<TD ALIGN="CENTER">Référence d'article</TD>
<TD ALIGN="CENTER">Description</TD>
<TD ALIGN="RIGHT">Prix</TD>
<TD ALIGN="RIGHT">Total</TD>
</TR>
%ROW {
@DTW_ADD(productcounter, "1", "1", productcounter)
<TR>
<TD ALIGN="CENTER">$(V_prnbr)</TD>
<TD ALIGN="CENTER">$(V_prsdesc)</TD>
<TD ALIGN="RIGHT">$(V_stprice) $(CURRENCY)</TD>
<TD ALIGN="RIGHT">$(V_format_total) $(CURRENCY)</TD>
</TR>
%}
< ?XPP:break> <INPUT TYPE="HIDDEN" NAME="addr_rn"
VALUE=$(V_stsanbr)>
<INPUT TYPE="HIDDEN" NAME="quantity" VALUE=$(V_stquant)>
<INPUT TYPE="HIDDEN" NAME="product_rn" VALUE=$(V_prrfnbr)>
<INPUT TYPE="HIDDEN" NAME="shipto_rn" VALUE=$(V_strfnbr)>
<INPUT TYPE="HIDDEN" NAME="shipmode_rn" VALUE=$(V_smrfnbr)>
</TABLE>

</FORM>
</BODY> %}
```

Cette section Report commence par afficher le titre de la page et un paragraphe décrivant comment utiliser le chariot. Puis elle définit un formulaire pour les éléments qui apparaîtront dans le tableau et affiche les en-têtes des colonnes de ce dernier.

Pour chaque ligne extraite par la requête de la fonction, le bloc ROW affiche, sur une ligne distincte du tableau, la référence de l'article, le nom du produit, le prix unitaire et le prix total (hors taxes et frais de port). Un numéro est affecté à la variable productcounter chaque fois qu'une ligne est affichée dans le tableau, pour assurer le suivi des produits. Cette variable sera référencée ultérieurement dans la macro.

Des paramètres d'entrée cachés sont ensuite créés. Ils utilisent les données suivantes, qui proviennent de la dernière ligne traitée par la requête : référence d'adresse du client, quantité, référence du produit, référence d'expédition

(shipto), référence du mode d'expédition. Ils seront transmis aux instructions qui sont appelées lorsque le client clique sur l'un des boutons **Mettre à jour** ou **Vider le chariot**. Ce mécanisme fait l'objet d'une description détaillée dans la section suivante du tutoriel.

---

## Ajout de la zone **Quantité** et du bouton **Retirer**

Ajoutons maintenant la zone **Quantité** et le bouton **Retirer** au tableau du chariot.

1. Vous allez d'abord insérer les en-têtes de colonne correspondant à ces données. Ajoutez le code HTML suivant au tableau :

```
%REPORT {
 <BODY BGCOLOR=#FFFFFF TEXT=#000000 LINK=#0000FF VLINK=#F00FF0
 ALINK=#FF0000>

 <TABLE WIDTH=600 BORDER=0>
 <TR>
 <TD>
 <P>Pour modifier la quantité d'un article, éditez le contenu de la
zone
 Quantité et cliquez sur le bouton Mettre à jour.

 Pour retirer un article, cliquez sur "Retirer" en regard de cet
article.
 Pour vider le chariot, cliquez sur le bouton Vider le chariot.
 <P> Une fois prêt à acheter un ou plusieurs articles, cliquez sur
le bouton Régler vos achats.
 </TD>
 </TR>
 </TABLE>
 <FORM ACTION="/cgi-bin/ncommerce3/OrderItemUpdate"
NAME="shiptolist">
 <TABLE BORDER=1 WIDTH=600 CELLPADDING=4 CELLSPACING=1>
 <TR>
 <TD ALIGN="CENTER">Référence d'article</TD>
 <TD ALIGN="CENTER">Description</TD>
 <TD ALIGN="RIGHT">Prix</TD>
 <TD ALIGN="CENTER">Quantité</TD>
 <TD ALIGN="RIGHT">Total</TD>
 <TD ALIGN="CENTER">Retirer du chariot</TD>
 </TR>
```

2. Insérez à présent deux nouvelles colonnes dans le tableau, destinées respectivement à la zone d'entrée de texte **Quantité** et au lien (ou bouton) **Retirer** :

```
%ROW {
 @DTW_ADD(productcounter, "1", "1", productcounter)
 <TR>
 <TD ALIGN="CENTER">$(V_prnbr)</TD>
 <TD ALIGN="CENTER">$(V_prsdesc)</TD>
 <TD ALIGN="RIGHT">$(V_stprice) $(CURRENCY)</TD>
 <TD ALIGN="CENTER">
 <INPUT TYPE="HIDDEN" NAME="refnum" VALUE=$(V_strfnbr)>
 <INPUT TYPE="TEXT" NAME="number" VALUE=$(V_stquant) SIZE=5>
 </TD>
 <TD ALIGN="RIGHT">$(V_format_total) $(CURRENCY)</TD>
 <TD ALIGN="CENTER"><A HREF="/cgi-bin/ncommerce3/OrderItemDelete?
```

```

 shipto_rn=$(V_strfnbr)&url=/cgi-
bin/ncommerce3/OrderItemDisplay">
 <I MG SRC="/grocery/bb_remv.gif" BORDER=0></TD>
 </TR>
%}

```

**Remarque :** Ne laissez aucun espace dans l'instruction `/cgi-bin/ncommerce3/OrderItemDelete?shipto_rn=$(V_strfnbr)&url=/cgi-bin/ncommerce3/OrderItemDisplay` utilisée pour le bouton **Retirer** (c'est-à-dire entre les marques `<A HREF>` et `</A>`).

3. Enregistrez le fichier sous le nom `t_shptodsp.d2w`.
4. Cliquez sur **Oui** pour remplacer le fichier existant. Vous avez terminé l'écriture de la macro `t_shptodsp.d2w`.

## Récapitulatif des modifications apportées

Résumons ce que vous venez d'ajouter à la macro du chariot :

Vous avez tout d'abord ajouté une nouvelle colonne dans laquelle viendra s'afficher la quantité pour chaque ligne extraite par la requête. Cette donnée s'affichera dans une zone d'entrée de texte de formulaire HTML. La valeur (VALUE) de cette zone est le contenu de la colonne STQUANT de la table SHIPTO, et son nom (NAME) est number. Vous avez également ajouté pour cette colonne une zone d'entrée cachée, dont le nom est refnum et la valeur `V_strfnbr` (référence d'expédition). Les valeurs de ces deux zones (quantité et référence d'expédition) seront utilisées ultérieurement pour créer une chaîne de paires nom-valeur, pour tous les produits présents dans le tableau du chariot. Ce mécanisme sera décrit plus en détail dans une étape ultérieure.

Vous avez également ajouté une autre colonne qui contient un lien permettant de retirer un article du chariot. Ce lien contient l'instruction `OrderItemDelete`, qui supprime de la table SHIPTO l'enregistrement correspondant à l'article considéré. Une fois son exécution terminée, cette instruction appelle à son tour l'instruction `OrderItemDisplay` pour réafficher la page du chariot.

---

## Activation de la macro de chariot personnalisée

Le système Net.Commerce utilise actuellement le modèle du tutoriel `tmp_shptodsp.d2w` pour afficher le chariot. Pour remplacer ce modèle par votre version personnalisée `t_shptodsp.d2w`, vous devez affecter celle-ci à l'activité d'affichage associée, SHIPTO\_ASSOC. Pour cela, vous devez retourner à la fonction Gestion des activités de l'Administrateur Net.Commerce.

1. Fermez la fenêtre du navigateur dans laquelle le magasin est affiché. Cela met fin à votre session de connexion en tant que client référencé.
2. Ouvrez une nouvelle fenêtre de navigateur et sélectionnez Administrateur Net.Commerce dans votre liste de signets.
3. Lorsque vous y êtes invité, entrez votre nom d'utilisateur et votre mot de passe d'administrateur de site, puis cliquez sur **Connexion**. La page d'accueil de l'Administrateur Net.Commerce s'affiche.
4. Sur la barre des tâches, cliquez sur **Gestionnaire de site**, puis sur **Gestion des activités**. Le formulaire Gestion des activités s'affiche.
5. Dans la liste **Sélection du type d'activité**, sélectionnez VIEW. La liste des activités de type VIEW (affichage) s'affiche dans le cadre inférieur.

6. Dans le cadre inférieur, cliquez sur **SHIPTO\_ASSOC**. Immédiatement, les informations décrivant l'activité apparaissent dans les zones du formulaire.
7. Sur la barre des tâches, cliquez sur **Affectation d'activité**.
8. Cliquez sur le bouton **Macro** dans le bas du cadre. Le formulaire Affectation d'une macro s'affiche.
9. Dans la zone **Nom du fichier de macros**, remplacez `tmp_shptodsp.d2w` par `t_shptodsp.d2w`.
10. Cliquez sur **Sauvegarder** pour mettre à jour l'enregistrement correspondant dans la table MACROS de la base de données. Vérifiez qu'un message confirmant que la nouvelle macro a bien été affectée à l'activité sélectionnée apparaît.  
Désormais, votre macro personnalisée est activée et sera exécutée chaque fois qu'un client accédera à la page du chariot.
11. Fermez la fenêtre du navigateur. Cela met fin à votre session de connexion en tant qu'administrateur de site.

---

## Test de la nouvelle macro du chariot

Vous allez maintenant voir l'effet produit, dans le magasin, par les modifications apportées à la macro du chariot.

1. Ouvrez une nouvelle fenêtre de navigateur.
2. Accédez au magasin à l'aide de votre signet.
3. Si vous y êtes invité, acceptez le cookie de session temporaire qui vous est affecté.
4. En premier lieu, vous devez vous reconnecter avec le même nom de client référencé pour pouvoir accéder à votre chariot.
5. A partir de la page d'accueil, cliquez sur **Connexion**. Entrez vos nom et mot de passe d'utilisateur référencé, puis cliquez sur **Connexion**.
6. Acceptez le cookie de session mis à jour.
7. Dans le cadre de navigation, cliquez sur le bouton **Chariot**. Le tableau de la page Chariot doit maintenant comporter les deux colonnes réservées respectivement aux zones Quantité et aux boutons **Retirer**. Les produits que vous avez placés auparavant dans le chariot doivent également figurer dans ce tableau.
8. Cliquez sur l'un des boutons **Retirer** pour tester son fonctionnement. L'article correspondant doit normalement être supprimé du chariot.
9. Changez la quantité de deux articles du chariot, puis cliquez sur le bouton **Mettre à jour**. Rien ne se produit.
10. Essayez de supprimer tous les articles du chariot en cliquant sur le bouton **Vider le chariot**. Dans ce cas également, vous n'obtenez aucun résultat.

### Résolution des incidents

Si la macro du chariot ne s'affiche pas ou ne fonctionne pas correctement, vérifiez à nouveau le code source. Si les difficultés persistent, vous disposez d'une version définitive de la macro, dans le même répertoire que celui où est stocké votre fichier. Pour remplacer la macro du chariot par la version définitive, effacez le fichier `t_shptodsp.d2w` et faites une copie de `f_shptodsp.d2w` sous le nom `t_shptodsp.d2w`, puis répétez les étapes 7 à 10.



---

## Chapitre 12. Création d'une fonction non prioritaire

Dans cette section, vous allez créer une fonction non prioritaire qui servira, sur la page du chariot, à activer les boutons **Mettre à jour** et **Vider le chariot** comme suit :

- Lorsque le client modifie la quantité pour un ou plusieurs produits placés dans le chariot et qu'il clique ensuite sur le bouton **Mettre à jour**, la fonction met à jour les quantités indiquées dans le chariot.
- Lorsque le client clique sur le bouton **Vider le chariot**, la fonction retire tous les produits du chariot.

### Ce que vous allez faire

Dans cette section, vous allez effectuer les opérations suivantes :

1. Etudier le code associé aux boutons **Mettre à jour** et **Vider le chariot**, dans la macro du chariot.
2. Apprendre les classes de base que vous utiliserez pour écrire la fonction.
3. Planifier l'écriture de la fonction.
4. Ecrire la fonction.
5. Compiler la fonction.
6. Enregistrer la fonction.
7. Activer la fonction.
8. Tester la fonction.

### Pour passer cette section

Une version finale de la fonction que l'on se propose de créer ici est à votre disposition, si vous décidez de ne pas suivre les exercices de cette section. Pour l'activer dans le magasin, vous devez effectuer les opérations suivantes :

1. A partir de la fenêtre de commande DB2, connectez-vous à la base de données du tutoriel en tapant la commande suivante :  
`connect to tutorial`
2. Enregistrez la fonction non prioritaire dans la base de données en tapant l'instruction suivante sur une seule ligne :  

```
WIN insert into ofs (refnum, dll_name, vendor, product, name,
version, description) values (5001, 'custof.dll', 'IBM' , 'NC',
'NCApiExtendShiptoUpdate', 1.0, NULL)

AIX insert into ofs (refnum, dll_name, vendor, product, name,
version, description) values (5001, 'libcustof.a', 'IBM' , 'NC',
'NCApiExtendShiptoUpdate', 1.0, NULL)

SUN insert into ofs (refnum, dll_name, vendor, product, name,
version, description) values (5001, 'libcustof.so', 'IBM' , 'NC',
'NCApiExtendShiptoUpdate', 1.0, NULL)
```
3. Activez la fonction en tapant l'instruction suivante sur une seule ligne :  
`update task_mer_of set of_rn = 5001 where task_rn = 2004 and merchant_rn is NULL`
4. Tapez ensuite :  
`insert into of_task (of_rn, task_rn) values (5001, 2004)`
5. Affichez le **Panneau de configuration** et redémarrez le serveur Net.Commerce.
6. Reportez-vous à Etude du processus de commande.

---

## Etude des boutons Mettre à jour et Vider le chariot

Avant d'écrire la fonction non prioritaire, vous devez comprendre le rôle des deux boutons suivants de la page du chariot, cette fonction leur étant destinée :

- Le bouton Mettre à jour régénère le contenu de la page du chariot afin de prendre en compte les nouvelles quantités de produits spécifiées par le client.
- Le bouton Vider le chariot retire tous les produits du chariot.

Ces boutons contiennent l'instruction qui appellera votre fonction, ainsi que les paramètres qui lui seront transmis.

### Etude du bouton Mettre à jour

Etudions le code du bouton **Mettre à jour**. Localisé dans la section HTML Report de la macro du chariot, le code source se présente comme suit :

```


```

Lorsque le client change la quantité d'un ou de plusieurs articles dans le chariot et qu'il clique sur le bouton **Mettre à jour**, la fonction JavaScript `update_shipto()` (dans la section HTML Report) est appelée. Le code source de la fonction `update_shipto()` est le suivant :

```
function update_shipto() {
 var pcounter = parseInt(`${productcounter}`)
 var tempValue = ""
 var url1 = "/cgi-bin/ncommerce3/OrderItemUpdate?addr_rn="
 var url2 = "&url=/cgi-bin/ncommerce3/OrderItemDisplay
 &success3=true"
 var address_ref = document.forms[0].addr_rn.value;
 var shipto_ref = document.forms[0].shipto_rn.value;
 var quantity = document.forms[0].quantity.value;
 var shipmode_ref = document.forms[0].shipmode_rn.value;

 for(i=0; i < pcounter; i++) {
 if(pcounter == 1) {
 if(document.forms[0].number.value <= 0) {
 alert("Entrez une quantité correcte") return }
 tempValue += "&" + document.forms[0].refnum.value + "="
 + document.forms[0].number.value
 }
 else {
 if(document.forms[0].number[i].value <= 0) {
 alert("Entrez une quantité correcte") return }
 tempValue += "&" + document.forms[0].refnum[i].value + "="
 + document.forms[0].number[i].value
 }
 }
 var new_url = "https://" + window.location.host + url1 +
 address_ref + "&shipto_rn=" + shipto_ref + "&quantity=" +
 quantity + "&shipmode_rn=" + shipmode_ref + tempValue + url2
 window.location.href = new_url
}
```

La fonction `update_shipto()` exécute les opérations suivantes :

1. Contrôle de validité de la saisie. Si le client entre une quantité incorrecte (par exemple, un nombre négatif) et qu'il clique sur le bouton **Mettre à jour**, une boîte d'alerte JavaScript s'affiche pour l'informer de l'erreur commise. La mise à jour est alors interdite.
2. Définition de variables qui seront utilisées par la fonction.
3. Construction d'un URL constitué des paramètres suivants :
  - a. La variable `Url1`, qui contient l'instruction `OrderItemUpdate` et les paramètres indispensables à celle-ci : référence de l'adresse, référence d'expédition ("shipto"), quantité et référence du mode d'expédition. Les valeurs de ces paramètres sont tirées du dernier produit dans la table. (On utilise le dernier produit parce que les paramètres d'entrée cachés sont définis après le traitement de la dernière ligne dans la requête SQL.)
  - b. La variable `tempValue`, qui stocke dans la table une chaîne de paires nom-valeur correspondant à la quantité et à la référence d'expédition (shipto) de chaque produit. La fonction extrait les valeurs de ces paires en référençant la valeur des zones d'entrée de texte `refnum` et `number`.
  - c. La variable `Url2`, qui est un URL redirectionnel contenant l'instruction `OrderItemDisplay`.
4. Exécution de l'URL comme suit :
  - a. L'instruction `OrderItemUpdate` met à jour la table `SHIPTO` pour y insérer la quantité du dernier produit dans la table. A ce stade, l'instruction exécute une mise à jour "fictive", car le client n'a pas forcément changé la quantité du dernier produit dans la table ; il a très bien pu changer celle d'un ou de plusieurs autres produits. Aussi, la "véritable" mise à jour intervient seulement lorsque l'instruction appelle l'activité de traitement `EXT_SHIPTO_UPD`.
  - b. L'instruction `OrderItemUpdate` appelle l'activité de traitement `EXT_SHIPTO_UPD`. Par défaut, cette activité ne fait rien, mais vous allez la personnaliser de sorte qu'elle puisse mettre à jour la table `SHIPTO` pour y inclure les quantités d'un ou de plusieurs produits changées par le client. Normalement, l'instruction `OrderItemUpdate` peut mettre à jour la quantité d'un seul produit.
  - c. Lorsque la mise à jour aboutit, l'URL redirectionnel `url2` est appelé et exécute l'instruction `OrderItemDisplay` pour réafficher la page du chariot avec les quantités modifiées.
  - d. Une fois la page affichée, JavaScript fait apparaître un message de confirmation dans la barre d'état, indiquant que le chariot intègre désormais les nouvelles quantités.

## Etude du bouton Vider le chariot

Examinons le bouton **Vider le chariot**. Son code source, dans la macro du chariot, se présente comme suit :

```


```

Lorsque le client clique sur le bouton, la fonction JavaScript `empty_shipto(form)` (dans la section HTML Report) est appelée. Son code source est le suivant :

```
function empty_shipto(form) {
 Ret = confirm("Souhaitez-vous réellement supprimer tous les articles
du chariot ?")
 if(!Ret)
 return
 var pcounter = parseInt('${productcounter}')
```

```

var tempValue = ""
var url1 = "/cgi-bin/ncommerce3/OrderItemUpdate?addr_rn="
var url2 = "&empty=1&url=/cgi-bin/ncommerce3/OrderItemDisplay"
var address_ref = document.forms[0].addr_rn.value;
var shipto_ref = document.forms[0].shipto_rn.value;
var quantity = document.forms[0].quantity.value;
var shipmode_ref = document.forms[0].shipmode_rn.value;
for(i=0; i < pcounter; i++) {
 if(pcounter == 1)
 tempValue += "&" + document.forms[0].refnum.value + "="
 + document.forms[0].number.value
 else
 tempValue += "&" + document.forms[0].refnum[i].value + "="
 + document.forms[0].number[i].value
}
var new_url = "https://" + window.location.host + url1 + address_ref
+
 "&shipto_rn=" + shipto_ref + "&quantity=" + quantity +
 "&shipmode_rn=" +
 shipmode_ref + tempValue + url2
window.location.href = new_url
}

```

La fonction `empty_shipto(form)` est similaire à la fonction du bouton Mettre à jour. Elle exécute les opérations suivantes :

1. Elle affiche une boîte d'alerte JavaScript, qui demande au client de confirmer qu'il souhaite effectivement supprimer tous les articles du chariot.
2. Après que le client a cliqué sur **OK**, elle définit les variables dont elle aura besoin.
3. Elle construit un URL constitué des mêmes paramètres que ceux de l'URL de la fonction `update_shipto()`, avec en plus la paire nom-valeur, `empty=1`.
4. Elle exécute l'URL comme suit :
  - a. L'instruction `OrderItemUpdate` opère une mise à jour "fictive" sur la table SHIPTO (en utilisant les paramètres d'entrée pour le dernier produit dans le chariot).
  - b. L'instruction `OrderItemUpdate` appelle l'activité de traitement `EXT_SHIPTO_UPD`. Vous allez également la personnaliser pour qu'elle retire tous les produits du chariot. Vous utiliserez le paramètre `empty=1` pour effectuer la suppression des produits.
  - c. Une fois les produits supprimés, la page du chariot est affichée avec un message indiquant qu'il ne reste plus d'articles dans le chariot.

---

## Classes nécessaires à l'écriture de la fonction

Net.Commerce fournit un ensemble de classes C++ destinées à l'écriture de fonctions non prioritaires. Selon la nature de la fonction, vous aurez besoin de tout ou partie de ces classes. Cette étape du tutoriel présente les classes que vous utiliserez pour écrire votre fonction non prioritaire.

Toutes les fonctions non prioritaires sont des classes dérivées de la classe `NC_OverridableFunction`. Elles communiquent avec Net.Commerce à travers trois classes qui sont transmises, comme paramètres, à la méthode `Process` de la classe :

```
bool Process(const HttpRequest& Req, HttpResponse& Res, NC_Environment& Env)
```

Voici la description de ces paramètres :

- `HttpRequest Req` comporte toutes les informations contenues dans une requête HTTP, y compris les paramètres d'entrée qui sont transmis à la fonction par l'instruction appelante.
- `HttpResponse Res` contient toutes les informations nécessaires au renvoi d'un document HTTP complet au navigateur, y compris les paramètres que la fonction non prioritaire renvoie. Normalement, les fonctions non prioritaires qui implémentent des activités de traitement (process) ne renvoient pas de documents HTTP.
- `NC_Environment Env` contient des informations sur l'environnement `Net.Commerce`, notamment les paramètres explicites d'entrée et de sortie de la fonction non prioritaire, tels que définis par son activité.

Vous devrez également faire usage de la classe `User`. Celle-ci contient les informations nécessaires à l'identification de chaque client ou utilisateur auprès du système `Net.Commerce`. Elle contient également certaines données de base permettant de le contacter et de le classer dans une catégorie. Elle utilise et référence la table `SHOPPER` (profils de client) de la base de données. Vous vous servirez de cette classe pour extraire les données relatives au profil du client en cours.

---

## Préparation de votre fonction non prioritaire

Pour préparer l'écriture de votre fonction non prioritaire, voici ce que vous devez faire :

1. Déterminez le but à atteindre dans votre logique commerciale.  
Vous voulez écrire une fonction non prioritaire qui, selon que le client clique sur le bouton **Mettre à jour** ou **Vider le chariot**, met à jour la page du chariot pour prendre en compte les quantités modifiées pour un ou plusieurs produits, ou retire tous les produits du chariot.
2. Déterminez quelle activité sera utilisée pour implémenter cette logique.  
En étudiant le code des boutons **Mettre à jour** et **Vider le chariot** dans la macro du chariot, vous avez pu constater que l'instruction `OrderItemUpdate` est appelée chaque fois que le client clique sur l'un ou l'autre de ces boutons. Par défaut, cette instruction met à jour la table `SHIPTO` pour y stocker la quantité d'un seul produit. Avant de prendre fin, elle appelle l'activité de traitement `EXT_SHIPTO_UPD`. C'est cette activité que vous utiliserez pour implémenter votre logique. Les commerçants peuvent l'utiliser librement pour entreprendre tout traitement complémentaire juste avant l'aboutissement de l'instruction `OrderItemUpdate`.
3. Déterminez le rôle joué par la fonction non prioritaire actuellement associée à l'activité.  
Pour l'instant, c'est la fonction non prioritaire `DoNothingNoArgs` qui implémente l'activité de traitement `EXT_SHIPTO_UPD`. Cette fonction n'effectue aucun traitement.
4. Identifiez les paramètres d'entrée que votre fonction recevra de `Net.Commerce`, et les paramètres de sortie qu'elle renverra au système.  
Comme cela est défini par l'activité de traitement `EXT_SHIPTO_UPD`, votre fonction non prioritaire recevra la référence de la sous-commande en tant que paramètre d'entrée explicite. Celui-ci sera transmis à travers la classe `NC_Environment`. La fonction recevra également tous les paramètres transmis avec l'instruction `OrderItemUpdate` via le paramètre `HttpRequest`. `HttpRequest` contiendra les éléments suivants :

- La variable `Ur11`, qui contient les paramètres transmis avec l'instruction `OrderItemUpdate` (référence de l'adresse, référence d'expédition "shipto", quantité et référence du mode d'expédition).
- La variable `tempValue`, qui contient une chaîne de paires nom-valeur correspondant à la quantité et à la référence d'expédition (shipto) de chaque produit dans le chariot.
- La variable `Ur12`, qui est un URL redirectionnel contenant l'instruction `OrderItemDisplay`.
- Si le client clique sur le bouton **Vider le chariot**, la paire nom-valeur `empty=1`.

Puisque votre fonction implémentera une activité de traitement, elle ne renverra pas de paramètres au système.

5. Planifiez le mode de traitement des conditions d'exception que pourra rencontrer votre fonction.

Vous pouvez :

- définir une activité d'exception pour traiter l'erreur,
- laisser le système traiter l'erreur,
- laisser la fonction traiter l'erreur.

Pour la fonction que vous allez créer dans le cadre de ce tutoriel, vous opterez pour la deuxième solution (laisser le système traiter l'erreur). Pour ce faire, vous devez renvoyer la valeur `False` à l'instruction appelante (`OrderItemUpdate`). Si la valeur `False` est renvoyée lorsque l'instruction vérifie les paramètres d'entrée, le gestionnaire d'instructions appelle la page d'erreur système `/ncerror/cmdinc.html`, qui signale que la syntaxe de l'instruction appelante est incorrecte ou que les paramètres requis par le système n'ont pas été transmis. Si la valeur `False` est renvoyée lors du traitement de l'instruction, le gestionnaire d'instructions appelle la page d'erreur système `/ncerror/cmdexe.html`, qui signale que l'instruction appelante n'a pas pu s'exécuter.

---

## Ecriture de la fonction non prioritaire

Vous voici prêt à écrire la fonction non prioritaire qui mettra à jour les quantités indiquées pour un ou plusieurs produits sur la page du chariot, ou qui retirera tous les produits du chariot. Votre code exécutera les opérations suivantes :

1. Il extrait la référence du client en cours.
2. Si la paire nom-valeur `empty=1` est transmise en tant que paramètre de la fonction (lorsque le client clique sur le bouton **Vider le chariot**), il extrait toutes les références d'expédition de la table `SHIPTO` pour le client considéré et les supprime.
3. Autrement (lorsque le client clique sur le bouton **Mettre à jour**), pour chaque paire nom-valeur transmise comme paramètre et dans laquelle le nom est un entier, il traite ce nom comme une référence d'expédition (shipto) et la valeur correspondante comme une quantité. Ensuite, il met à jour la table `SHIPTO` avec les quantités associées à chaque référence d'expédition, pour le client considéré.

Vous devez écrire votre fonction en tant que classe C++. Le code personnalisable de la classe est déjà écrit.

1. Ouvrez le fichier `template.cpp` dans le répertoire suivant :

 `\ibm\netcommerce3\adt\samples\tutorial`

**AIX** /usr/lpp/NetCommerce3/adt/samples/tutorial

**SUN** /opt/IBMnetc/NetCommerce3/adt/samples/tutorial

2. Examinez le contenu du fichier. La première classe, NCApiExtendShiptoProcess, a été créée pour le bouton **Ajouter au chariot**. A la section Création du chariot, vous avez appris que cette fonction combine en une seule ligne les quantités d'articles identiques dans la table SHIPTO.
3. Allez à la fin de la classe NCApiExtendShiptoProcess. C'est à cet endroit que commence le code personnalisable de la classe que vous allez écrire. Actuellement, la classe a pour nom MyOverridableFunction.
4. Pour la renommer, remplacez toutes les occurrences de MyOverridableFunction par NCApiExtendShiptoUpdate.
5. Commencez à écrire votre fonction. Tout le code doit être entré dans la méthode Process de la classe. Comme nous l'avons vu précédemment, cette méthode se charge de tous les traitements pour votre classe. Par exemple, elle manipule les objets et accède à la base de données. Elle extrait toutes les données dont elle a besoin via les classes HttpRequest Req, HttpResponse Res et NC\_Environment Env, qui sont reçues comme paramètres d'entrée.

En premier lieu, vous devez extraire la référence du client en cours. Ajoutez le code suivant à la méthode Process :

```
public:
// The actual processing happens here
virtual bool Process (const HttpRequest& Req, HttpResponse& Res,
NC_Environment& Env)
{
// Set the error handler
String ErrTaskName (_TaskName);
ErrTaskName << "_ERR";
error << "ERROR: ";
//-----//
//-- Début du code --//
//-----//
User* user = (User*) Env.Seek (NC_Environment::_VAR_Shopper);
// Contrôler si aboutissement
if (user == NULL)
{
error.nls (&_ERR_CANT_LOAD_SHOPPER) << endl;
return false;
}
String shopperRefNum = user->getValue(User::_COL_REF_NUM);
//-----//
//-- Fin du code --//
//-----//
```

Après la définition d'une activité d'exception destinée à la gestion des erreurs rencontrées par la fonction, votre code est exécuté comme suit : la méthode Process appelle d'abord la fonction Seek(NC\_Environment::\_VAR\_SHOPPER), dans l'instance de classe NC\_ENVIRONMENT Env, pour extraire un pointeur à destination d'une classe User. (La classe User contient des informations sur le client en cours.) Elle vérifie ensuite qu'un utilisateur a été extrait. Si ce n'est pas le cas, elle renvoie une erreur au système. Elle stocke la valeur extraite du pointeur de la classe User à l'aide de la fonction getValue(User::\_COL\_REF\_NUM) dans la chaîne shopperRefNum.

6. Enregistrez votre travail.

---

## Ecriture du code permettant de vider le chariot

Vous voici prêt à écrire le code permettant de supprimer tous les produits du chariot lorsque le client clique sur le bouton **Vider le chariot**. Pour cela, vous devez d'abord créer une condition visant à déterminer si la fonction doit vider le chariot ou mettre à jour celui-ci pour refléter les quantités de produits entrées par le client. Vous utiliserez comme critère la valeur du paramètre `empty`. Celui-ci est stocké dans `HttpRequest Req`. S'il est égal à "1", le chariot sera vidé ; dans le cas contraire, les quantités de produits seront mises à jour.

1. Pour extraire la valeur du paramètre `empty`, ajoutez le code suivant :

```
//-----//
// Début du code -//
//-----//
User* user = (User*) Env.Seek (NC_Environment::_VAR_Shopper);
// Contrôler si aboutissement
if (user == NULL)
{
 error.nls (&_ERR_CANT_LOAD_SHOPPER) << endl;
 return false;
}
String shopperRefNum = user->getValue(User::_COL_REF_NUM);
String emptyVal = Req.getNVPs().Get("empty");
String stmt;
//-----//
// Fin du code -//
//-----//
```

La méthode `Process` appelle maintenant la fonction `getNVPs()` pour extraire la classe `NameValuePairMap` stockée dans `HttpResponse Req`. Elle appelle ensuite la fonction `Get("empty")` pour obtenir de l'instance de classe `NameValuePairMap` la valeur de la paire nom-valeur contenant le nom `empty`. Puis elle déclare la chaîne `stmt` pour le stockage des instructions SQL que vous allez écrire à l'étape suivante.

2. Si la valeur de `empty` est égale à "1", le chariot sera vidé de son contenu. Ce processus comprend deux phases : d'abord extraire toutes les références d'expédition de la table `SHIPTO`, puis les supprimer. Commençons par ajouter le code visant à extraire les références d'expédition :

```
if (emptyVal == "1")
{ // The table has to be cleared
 long* shiptoRefNums;
 // Count the number of entries that need to be cleared
 stmt << "SELECT COUNT(strfnbr) FROM shipto WHERE stshnbr = ";
 stmt << shopperRefNum << " AND ststat = 'P'";
 SQL Sql(*(DataBaseManager::GetCurrentDataBase()), stmt);
 Row SqlRow;
 if (Sql.getNextRow(SqlRow) != ERR_DB_NO_ERROR)
 {
 Sql.ReportError();
 error <& lt; "getting count of products to clear" << endl;
 return false;
 }
 long count = 0;
 SqlRow.getCol(1).getVal(count);
 if (count > 0)
 {
 //-----//
 }
}
```

```

//-- Début du code --//
//-----//
shiptoRefNums = new long[count];
stmt.Clean();
stmt << "SELECT strfnbr FROM shipto";
stmt << " WHERE stshnbr = " << shopperRefNum << " AND ststat =
'P'";

SQL Sql12(*(DataBaseManager::GetCurrentDataBase()), stmt);
Row SqlRow2;
long temp;
int i = 0;
while (Sql12.getNextRow(SqlRow2) == ERR_DB_NO_ERROR)
{
 SqlRow2.getCol(1).getVal(temp);
 & rbl;shiptoRefNums[i++] = temp;
}
if (i == 0)
{
 Sql1.ReportError();
 error << " Collecting shiptoRefNums for deletion" << endl;
 delete[] shiptoRefNums;
 return false;
}
//-----//
//-- Fin du code --//
//-----//

```

Après que le nombre de références d'expédition pour le client en cours dans la table SHIPTO a été déterminé et qu'il s'avère que cette valeur est supérieure à zéro (sinon, une erreur est renvoyée), la portion de code que vous avez ajoutée est ensuite exécutée comme suit : la méthode Process réserve de l'espace pour le stockage de toutes les références d'expédition à supprimer. Ensuite, elle appelle la fonction Clean() dans l'instance de classe String stmt pour effacer le contenu de la variable stmt. Elle compose ensuite les instructions SQL visant à extraire de la table SHIPTO toutes les références d'expédition du client en cours, puis elle les transmet au constructeur de la classe SQL. La classe SQL exécute la requête et stocke le résultat dans l'instance Sql12. L'instance SqlRow2 de la classe Row extrait le résultat. La fonction getNextRow récupère les lignes une à une jusqu'à ce qu'il n'en reste plus à extraire (cette condition étant signalée par le renvoi d'une erreur).

Pour chaque ligne extraite, la valeur de la première colonne est isolée et stockée dans le tableau shiptoRefNums. Un contrôle est effectué pour permettre de vérifier qu'au moins une ligne est extraite.

3. Il vous faut maintenant supprimer les références d'expédition qui ont été extraites de la table SHIPTO. Ajoutez le code suivant :

```

if (i == 0)
{
 Sql1.ReportError();
 error << " Collecting shiptoRefNums for deletion" << endl;
 delete[] shiptoRefNums;
 return false;
}
for (int j=0; j<i; j++)
{
 stmt.Clean();
 stmt << "DELETE FROM shipto WHERE strfnbr = " << shiptoRefNums[j];

```

```

SQL Sql13(*(DataBaseManager::GetCurrentDataBase()), stmt);
if (Sql13.getSQLrc() != ERR_DB_NO_ERROR)
{
 Sql13.ReportError();
 error << "deleting products from shipto" << endl;
 delete[] shiptoRefNums;
 return false;
}
}
delete[] shiptoRefNums;
//-----//
//-- Fin du code --//
//-----//

```

Pour chaque référence d'expédition contenue dans le tableau, la méthode Process appelle d'abord la fonction Clean() pour effacer stmt. Elle compose ensuite une nouvelle requête SQL dans stmt pour supprimer la référence d'expédition en cours de la table SHIPTO. Une fois la requête exécutée, son résultat est stocké dans l'instance Sql13. (Généralement, chaque fois que vous voulez exécuter une requête SQL, une nouvelle instance SQL doit être créée pour stocker le résultat.) Un contrôle est effectué pour permettre de vérifier qu'au moins une ligne est extraite.

Une fois que toutes les références d'expédition ont été supprimées, l'espace précédemment réservé est libéré et rendu au système.

4. Enregistrez votre travail.

---

## Ecriture du code de mise à jour des quantités de produits

Vous allez maintenant écrire l'autre partie du code de votre fonction. Lorsque le client clique sur le bouton **Mettre à jour**, cette partie du code doit provoquer la régénération du contenu du chariot afin que celui-ci reflète les nouvelles quantités de produits spécifiées par ce client. Pour cela, vous allez énoncer la condition suivante : si la valeur du paramètre empty transmis à la fonction est différente de "1", le chariot doit être mis à jour pour refléter les nouvelles quantités de chaque produit, d'après les paires nom-valeur qui correspondent à des associations référence d'expédition-quantité. Ces paires nom-valeur sont stockées dans la variable tempValue de HttpRequest Req. Pour qu'elles soient extraites de HttpRequest Req, toutes les paires dans lesquelles le nom est un entier seront traitées comme des références d'expédition.

1. Ajoutez le code suivant :

```

 }
}
else
{
 //-----//
 //-- Début du code --//
 //-----//
 NVPHash::Iterator I = Req.getNVPs().getIterator();
 I.Begin();
 while (*I != NULL)
 {
 if ((*I)->getName().IsLong() == true)
 {
 stmt.Clean();
 stmt << "UPDATE shipto SET stquant = " << ((*I)->getValue());
 stmt << " WHERE stshnbr = " << shopperRefNum;

```

```

 stmt << " AND strfnbr = " << ((*I)->getName());
 stmt << " AND ststat ='P'";
 SQL Sql14(*(DataBaseManager::GetCurrentDataBase()), stmt);
 if (Sql14.getSQLrc() != ERR_DB_NO_ERROR)
 {
 Sql14.ReportError();
 error << "Updating product quantities" << endl;
 return false;
 }
 }
 ++I; }
//-----//
//-- Fin du code --//
//-----//
}
return true;

```

Un itérateur I est d'abord défini pour extraire les paires nom-valeur stockées dans NameValuePairMap, qui se trouve dans HttpRequest Req. Ensuite, l'itérateur est établi au début de la liste.

Pour chaque paire nom-valeur stockée dans l'itérateur I, si le nom est un entier, une instruction SQL est composée pour insérer dans la table SHIPTO la quantité stockée dans la partie "valeur" de la paire nom-valeur. Une fois la requête exécutée, son résultat est stocké dans l'instance Sql14. Un contrôle est effectué pour permettre de vérifier qu'au moins une ligne est extraite.

La méthode Process renvoie la valeur "true" lorsque l'exécution s'est déroulée correctement.

2. Enregistrez votre travail.
3. Vous avez terminé l'écriture de la classe de fonction NCApiExtendShiptoUpdate. Avant de passer à l'étape suivante, examinez la portion de code d'enregistrement qui fait suite à cette classe :

```

const ClassName
NCApiExtendShiptoUpdate::_STR_ThisClass("NCApiExtendShiptoUpdate");
static bool X2 =
NC_ApiManager::GetUniqueInstance().RegisterApi("IBM", "NC",
"NCApiExtendShiptoUpdate", 1.0, new NCApiExtendShiptoUpdate);

```

Ce code instancie automatiquement la classe et l'enregistre sous le nom que vous lui avez donné, afin que le système puisse localiser la fonction au moment de l'exécution. Notez que le nom (X2) de la variable statique booléenne doit être différent pour chaque fonction non prioritaire compilée dans une même DLL ou bibliothèque partagée.

---

## Compilation de la fonction non prioritaire

Vous êtes maintenant prêt à compiler le fichier CPP. Le tableau suivant offre un aperçu de l'objet créé, du compilateur requis et de l'emplacement du modèle et de la version finale. Pour plus de simplicité, la version finale de l'objet a été créée et copiée dans le répertoire correct.

Plateforme et objet créé	Compilateur requis	Emplacement du modèle de fichier makefile	Version finale
 DLL	Microsoft Visual C++ version 4.2	\ibm\netcommerce3 \adt\samples\templates	\ibm\netcommerce3 \bin\custof.dll

Plateforme et objet créé	Compilateur requis	Emplacement du modèle de fichier makefile	Version finale
 bibliothèque partagée	txlC++ version 4.1.x	/usr/lpp/NetCommerce3 /adt/samples/templates	/usr/lpp/NetCommerce3 /bin/libcustof.a
 bibliothèque partagée	Sun Workshop Compiler C++ version 4.2	/opt/IBMnetc/NetCommerce3 /adt/samples/templates	/opt/IBMnetc/NetCommerce3 /bin/libcustof.so

## Enregistrement de la fonction non prioritaire dans la base de données

Vous pouvez maintenant enregistrer (inscrire) votre nouvelle fonction non prioritaire dans la base de données, afin que le système puisse y avoir accès au moment de l'exécution.

1. A partir de la fenêtre de commande DB2, connectez-vous à la base de données du tutoriel en tapant la commande suivante :

```
db2 connect to tutorial
```

2. Entrez la commande suivante sur une seule ligne :

```
insert into ofs (refnum, dll_name, vendor, product, name,
version, description) values (5001, '<nomfichier>', 'IBM' , 'NC',
'NCapiExtendShiptoUpdate', 1.0, NULL)
```

en remplaçant <nomfichier> par le nom du fichier compilé pour votre plateforme :

 custof.dll

 libcustof.a

 libcustof.so

## Activation de votre nouvelle fonction

Pour activer votre nouvelle fonction non prioritaire, c'est-à-dire la mettre en service dans votre magasin, vous devez l'affecter à l'activité prévue, EXT\_SHIPTO\_UPD, qui remplacera la fonction par défaut, DoNothingNoArgs, actuellement affectée à cette activité.

1. Fermez la fenêtre du navigateur dans laquelle le magasin est affiché.
2. Ouvrez une nouvelle fenêtre de navigateur et accédez à l'Administrateur Net.Commerce.
3. Connectez-vous avec vos nom d'utilisateur et mot de passe d'administrateur de site.
4. Sur la barre des tâches, cliquez sur **Gestionnaire de site**, puis sur **Gestion des activités**. Le formulaire Gestion des activités s'affiche.
5. Dans la liste **Sélection du type d'activité**, choisissez **PROCESS** (traitement). La liste de toutes les activités de traitement disponibles s'affiche dans la zone prévue à cet effet.
6. Dans cette zone, cliquez sur **EXT\_SHIPTO\_UPD**. (Vous devrez peut-être faire défiler la liste pour localiser cette activité). Immédiatement, les informations décrivant l'activité apparaissent dans les zones du formulaire.
7. Sur la barre des tâches, cliquez sur **Affectation d'activité**. Le formulaire Affectation de fonction non prioritaire s'affiche.

8. Dans la liste **Magasin — Affectation de fonction non prioritaire(Produit)(Fournisseur)(Version)**, sélectionnez **Centre commercial — DoNothingNoArgs(NC)(IBM)(1.0)**.
9. Dans la liste **Fonctions non prioritaires**, sélectionnez **NCApiExtendShiptoUpdate(NC)(IBM)(1.0)**.
10. Cliquez sur **Mettre à jour**. Un message s'affiche, confirmant que votre nouvelle fonction a bien été affectée à l'activité, pour le magasin East West Food Mart (au niveau centre commercial).
11. Fermez la fenêtre du navigateur.
12. Affichez le **Panneau de configuration** et redémarrez le serveur Net.Commerce.  
La fonction non prioritaire est à présent activée pour le magasin East West Food Mart.

---

## Test de la nouvelle fonction non prioritaire

Procédons au test de notre nouvelle fonction dans le magasin pour vérifier qu'elle se comporte comme prévu.

1. Ouvrez une nouvelle fenêtre de navigateur et accédez au magasin.
2. Connectez-vous en tant que client référencé (avec le nom et le mot de passe correspondants).
3. Dans le cadre de navigation, cliquez sur **Chariot** pour afficher la page du chariot et son contenu.
4. Commençons par tester le fonctionnement de la partie de la fonction chargée de mettre à jour la table SHIPTO conformément aux nouvelles quantités spécifiées pour un ou plusieurs produits. Changez les quantités des deux premiers produits placés dans le chariot en modifiant le contenu des zones **Quantité** correspondantes.
5. Cliquez sur le bouton **Mettre à jour**. La page du chariot est régénérée et reflète désormais les nouvelles quantités que vous venez d'indiquer.
6. Testons maintenant la partie de la fonction dédiée à la suppression de tous les produits de la table SHIPTO. Cliquez sur le bouton **Vider le chariot**.
7. Cliquez sur **OK** pour confirmer la suppression. Tous les produits sont retirés du chariot et une nouvelle page s'affiche, indiquant que le chariot est maintenant vide.
8. Notez que la zone **Montant total**, dans le cadre supérieur gauche indique désormais \$0.
9. La section suivante du tutoriel consistant à passer une commande, le chariot ne doit pas rester vide. Par conséquent, ajoutez-y deux produits de votre choix avant d'aborder cette section.



---

## Chapitre 13. Etude du processus de commande

Voyons maintenant comment le processus de commande a été simplifié pour le magasin East West Food Mart. Lorsque le client est prêt à passer sa commande, il clique sur le bouton **Régler vos achats**, soit à partir du cadre de navigation, soit à partir de la page Chariot. Tous les articles se trouvant dans le chariot à ce moment-là sont transférés sur la page Commande - Détails. A partir de cette page, le client pourra :

- visualiser les informations détaillées sur les articles de la commande, y compris l'adresse d'expédition, les quantités, les prix totaux, les frais de port et les diverses taxes ;
- entrer une adresse d'expédition (pour les visiteurs) ou mettre à jour l'adresse d'expédition si elle est différente de celle spécifiée au cours de l'enregistrement (pour les clients référencés) ;
- faire valider les informations relatives à sa carte de paiement.

Le client a déjà spécifié les quantités de chaque article de la commande sur la page Produit ou Chariot. Les quantités sont tout simplement reportées sur cette page.

### Ce que vous allez faire

Dans cette étape, vous effectuerez les opérations suivantes :

1. Afficher la page Détails.
2. Etudier la macro de commande afin de voir comment elle a été personnalisée pour :
  - afficher et traiter les informations relatives à l'expédition, les détails de la commande et les données ayant trait au paiement, le tout sur la même page ;
  - afficher, sur le formulaire d'adresse d'expédition, des zones vierges (à compléter par le client lorsque celui-ci est un visiteur) ou comportant les données de l'adresse d'expédition (lorsque le client est déjà enregistré auprès du magasin).

### Pour passer cette section

Si vous décidez de ne pas suivre cette section, allez directement à Personnalisation d'une page d'erreur système.

---

## Affichage de la page Détails

Voyons comment cette page se présentera pour les clients.

1. A partir de la page Chariot, cliquez sur **Régler vos achats**. Vous obtenez la page Détails. Elle se compose de quatre éléments principaux : le tableau des détails de la commande, le formulaire d'adresse d'expédition, le tableau des options de paiement et le bouton **Valider la commande**.
2. Notez que votre nom d'utilisateur figure dans la colonne **Expédier à** du tableau détaillant la commande, tandis que l'adresse que vous avez indiquée lors de l'enregistrement est affichée dans le formulaire d'adresse d'expédition.

---

## Etude de la macro de commande

Examinons le contenu de la macro de création de la page Commande - Détails, t\_orderdspp.d2w, pour voir comment sont générées les quatre principales parties qui la composent. Nous verrons également comment faire en sorte que la partie formulaire d'adresse d'expédition se présente différemment selon que le client est référencé dans le magasin ou simple visiteur.

1. Assurez-vous que vous avez pris connaissance des instructions relatives à l'édition des modèles du tutoriel.
2. A l'aide d'un éditeur de texte, ouvrez la macro t\_orderdspp.d2w.
3. Examinez la section HTML Report de la macro. Elle commence par appeler les fonctions suivantes :

```
%{ ***** HTML Report Section ***** %}
%HTML_REPORT
{
<HTML>
<HEAD>
<META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
<TITLE>Commande - Détails</TITLE>
</HEAD>
 @getShopperInfo()
 @get MerchantInfo()
 @priceSubtotal()
 %IF (isTempShopper == "T")
 @displayTempShopper()
 %ELSE
 @displayRegisteredShopper()
 %ENDIF
```

Dans les étapes à venir, vous allez étudier chaque fonction en détail.

## Etude des fonctions de base

Examinons le code source des deux premières fonctions appelées à partir de la section HTML Report de la macro de commande. Ces fonctions ont pour tâche d'extraire des informations générales sur le client qui passe la commande.

1. Observez le code source de la première fonction, getShopperInfo() :

```
%{ ***** Function Definition Section ***** %}
%FUNCTION(DTW_ODBC) getShopperInfo()
{
 select sarfnbr, shrfnbr, sanick, salname, safield2
 from shaddr, shopper
 where (sashnbr=shrfnbr)
 and (shlogid='${SESSION_ID}')
 and (sanick=shlogid)
 and (saadrflg='P')
 %REPORT {
 %ROW {
 @DTW_ASSIGN(address_rn, V_sarfnbr)
 @DTW_ASSIGN(shopper_rn, V_shrfnbr)
 @DTW_ASSIGN(nick, V_sanick)
 @DTW_ASSIGN(salname, V_salname)
 @DTW_ASSIGN(isTempShopper, V_safield2)
 }
 }
}
```

%}

Cette fonction exécute une requête SQL standard qui permet d'obtenir des informations sur le client, à savoir la référence de l'adresse, la référence du client, son alias et son nom ainsi que la valeur de la colonne SAFIELD2, à partir des tables SHADDR et SHOPPER,.

La colonne SAFIELD2 a une importance toute particulière ici, car c'est d'après son contenu que l'on déterminera si le client est référencé ou non.

- Si le client n'est pas référencé, la valeur T lui sera affectée. Au cours d'une précédente étape, vous avez précisé qu'une entrée de carnet d'adresses temporaire devait être créée à chaque accès d'un client non référencé au magasin. Cette entrée est constituée du SESSION\_ID attribué aléatoirement au navigateur pour l'alias, d'un numéro attribué aléatoirement pour la référence du client, de la lettre T pour la colonne SAFIELD2 pour signaler que le client est un visiteur, et de la chaîne "empty" pour toutes les autres colonnes obligatoires de la table SHADDR. La valeur T sera donc encore présente dans la colonne SAFIELD2 pour les visiteurs, puisqu'ils n'ont pas mis à jour l'entrée en s'enregistrant.
- Si le client s'est enregistré, une nouvelle entrée de carnet d'adresses a nécessairement été créée. Elle est constituée de l'adresse spécifiée par le client lors de son enregistrement et de son nouvel ID de connexion pour l'alias (colonne SANICK). Fait plus important encore, aucune valeur n'est insérée dans la colonne SAFIELD2 (cette colonne étant réservée à l'usage du commerçant, elle n'est pas mise à jour automatiquement).

Dans la section Report de cette fonction, un nom de variable Net.Data est affecté à chaque valeur de colonne extraite par la requête. Ces noms de variable seront référencés plus loin dans la macro. Notez que la valeur de la colonne SAFIELD2 reçoit le nom de variable isTempShopper.

2. Passons à la fonction suivante, getMerchantInfo. Son code source est le suivant :

```
%FUNCTION(DTW_ODBC) getMerchantInfo()
{
 select mestname, merfnbr
 from order_pend, merchant
 where orrfnbr=$(order_rn)
 and ormenbr=merfnbr
 %REPORT {
 %ROW {
 <BODY BGCOLOR=#FFFFFF TEXT=#000000 LINK=#0000FF VLINK=#008484
ALINK=#FF0000>
 <P></P>
 <TABLE BORDER=0>
 <TR>
 <TD>Voici le récapitulatif de votre commande avec les prix
totaux,
 les frais de port et les taxes.
Vous pouvez supprimer
certains articles
 de votre commande, en visualiser les détails et entrer les
informations
 relatives à votre moyen de paiement.</TD>
 </TR>
 <TR>
 <TD COLSPAN=4><HR WIDTH=550 ALIGN="LEFT"></TD>
 </TR>
 </TABLE>
```

```

@DTW_ASSIGN(V_merrf, V_merfnbr)
%}
%}

```

Cette fonction extrait de la table MERCHANT le nom du magasin et la référence du commerçant. La section Report affiche la bannière de la page ainsi qu'un texte d'introduction ; elle stocke la valeur extraite de la colonne MERFNBR (référence du commerçant) dans la variable V\_merrf. Cette variable sera utilisée comme paramètre par les instructions associées au bouton **Valider la commande**. Ce mécanisme sera décrit plus en détail dans une étape ultérieure.

## Etude de la fonction de création du tableau récapitulatif de la commande

Examinons la fonction suivante, priceSubtotal(), qui sert à créer un tableau récapitulant la commande du client :

```

%FUNCTION(DTW_ODBC) priceSubtotal()
{
 select distinct sanick, oyprtot as sub_total,
 oyshtot as ship_charge, oytax1 as sales_tax,
 oysanbr, (oyprtot+oyshtot+oytax1) as total_price,
 (select sum(stquant) from shipto, merchant
 where sarfnbr=stsanbr
 and merfnbr=stmenbr
 and ststat='P'
 and stmenbr=$(merchant_rn)) as items
 from orderpay, shaddr, shopper
 where shlogid=$(SESSION_ID)
 and oyornbr=$(order_rn)
 and sashnbr=shrfnbr
 and sarfnbr=oysanbr
 order by sanick
 %REPORT {
 <TABLE WIDTH=550>
 <TR>
 <TD>Les prix, les frais de port et les taxes applicables à
chaque expédition
 figurent ci-dessous.</TD>
 </TR>
 </TABLE>

 <TABLE BORDER=1 CELLSPACING=1 CELLPADDING=1 WIDTH=550>
 <TR>
 <TD ALIGN="CENTER">Expédier à</TD>
 <TD ALIGN="CENTER">Sous-total</TD>
 <TD ALIGN="CENTER">Frais de port</TD>
 <TD ALIGN="CENTER">Taxes</TD>
 <TD ALIGN="CENTER">Total</TD>
 </TR>
 %ROW {
 <TR>
 %IF (isTempShopper == "T")
 <TD ALIGN="CENTER">Visiteur</TD>
 %ELSE
 <TD ALIGN="CENTER">$(V_sanick)</TD>
 %ENDIF
 </TR>
 }
 }
}

```

```

<TD ALIGN="CENTER">$(V_sub_total)</TD>
<TD ALIGN="CENTER">$(V_ship_charge)</TD>
<TD ALIGN="CENTER">$(V_sales_tax)</TD>
<TD ALIGN="CENTER">$(V_total_price)</TD>
%}
</TABLE>
%}

```

A partir de la base de données, cette fonction extrait tous les éléments nécessaires à l'indication du sous-total, des frais de port, des taxes et du prix total, pour chaque article de la commande.

La section Report présente les résultats de la requête dans un tableau. Notez que la première colonne, qui précise à qui expédier la commande, contient l'alias du client. Elle a été personnalisée de façon à afficher la mention *Visiteur* dans le cas d'un client visiteur, ou l'alias du client lorsque celui-ci est référencé. On utilise pour cela la valeur de la variable *isTempShopper*. Si cette valeur est T, la mention *Visiteur* sera affichée ; sinon, c'est le contenu de la colonne *SANICK* de la table *SHADDR* qui sera présenté ici. On évite ainsi d'afficher le *SESSION\_ID* qui est attribué par le système aux clients visiteurs, cette donnée n'ayant pas d'intérêt pour eux.

## Etude des fonctions de création du formulaire d'adresse d'expédition

Voyons comment la partie formulaire d'adresse d'expédition est créée. Ce formulaire est vierge dans le cas d'un client visiteur, tandis qu'il comporte une adresse dans le cas d'un client référencé. La valeur de *isTempShopper* permet de déterminer si le client est référencé ou non. A chacun de ces deux cas de figure correspond une fonction déterminée, que l'on appellera si la condition correspondante est vérifiée.

Etudions comment la fonction appropriée est appelée à partir de la section HTML Report :

```

%IF (isTempShopper == "T")
 @displayTempShopper()
%ELSE
 @displayRegisteredShopper() %ENDIF

```

Ces instructions énoncent que si la valeur de *isTempShopper* est T (ce qui indique que le client est un visiteur auquel on a associé une entrée de carnet d'adresses temporaire), alors il faut exécuter la fonction *displayTempShopper()* ; sinon, la fonction *displayRegisteredShopper()* doit être exécutée.

Nous allons maintenant étudier chacune de ces fonctions en détail.

## Etude de la fonction de création du formulaire d'adresse réservé aux visiteurs

Etudiez d'abord la fonction *displayTempShopper*. Elle crée un formulaire vierge, prêt à être rempli par le client visiteur.

Son code source est le suivant :

```

%FUNCTION(DTW_ODBC) displayTempShopper()
{
 select sarfnbr, shrfnbr, sanick, satitle, salname, safname, samname,
saaddr1,
 saaddr2, sacity, sastate, sacntry, sazipc

```

```

from shaddr, shopper
where (sashnbr=shrfnbr)
and (shlogid='$(SESSION_ID)')
and (sanick=shlogid)
and (saadrflg='P')
%REPORT {
 %IF (V_satitle == "M.")
 @DTW_ASSIGN(sel1, "SELECTED")
 %ELIF (V_satitle == "Mme")
 @DTW_ASSIGN(sel2, "SELECTED")
 %ELIF (V_satitle == "Mlle")
 @DTW_ASSIGN(sel3, "SELECTED")
 %ELIF (V_satitle == "Dr")
 @DTW_ASSIGN(sel4, "SELECTED")
 %ENDIF
 <FORM>
 <TABLE WIDTH=600 CELLPADDING=10 BORDER=0 COLS=4>
 <TR>
 <TD COLSPAN=4><HR WIDTH=550 ALIGN="LEFT"></TD>
 </TR>
 <TR>
 <TD ALIGN="RIGHT">Titre</TD>
 <TD>
 <SELECT NAME="satitle">
 <OPTION $(sel1) VALUE='M.'>M.
 <OPTION $(sel2) VALUE='Mme'>Mme
 <OPTION $(sel3) VALUE='Mlle'>Mlle
 <OPTION $(sel4) VALUE='Dr'>Dr
 </SELECT>
 </TD>
 <TD ALIGN="RIGHT">Nom</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="salname" VALUE="" SIZE="22"
 MAXLENGTH="30"></TD>
 </TR>
 <TR>
 <TD ALIGN="RIGHT">Prénom</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="safname" VALUE="" SIZE="22"
 MAXLENGTH="30"></TD>
 <TD ALIGN="RIGHT">Initiale ou deuxième
prénom</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="samname" VALUE="" SIZE="22"
 MAXLENGTH="30"></TD>
 </TR>
 <TR>
 <TD ALIGN="RIGHT">Adresse (ligne
1)</TD>
 <TD COLSPAN=3><INPUT TYPE="TEXT" NAME="saaddr1" VALUE="" SIZE="64"
 MAXLENGTH="50"></TD>
 </TR>
 <TR>
 <TD ALIGN="RIGHT">Adresse (ligne 2)</TD>
 <TD COLSPAN=3><INPUT TYPE="TEXT" NAME="saaddr2" VALUE="" SIZE="64"
 MAXLENGTH="50"></TD>
 </TR>
 <TR>
 <TD ALIGN="RIGHT">Ville</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="sacity" VALUE="" SIZE="22"
 MAXLENGTH="30"></TD>

```

```

 <TD ALIGN="RIGHT">Département/Province</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="sastate" VALUE="" SIZE="22"
 MAXLENGTH="20"></TD>
 </TR>
 <TR>
 <TD ALIGN="RIGHT">Code postal</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="sazipc" VALUE="" SIZE="22"
 MAXLENGTH="20"></TD>
 <TD ALIGN="RIGHT">Pays</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="sacntry" VALUE="" SIZE="22"
 MAXLENGTH="30"></TD>
 </TR>
</TABLE>
%}

```

Cette fonction extrait toutes les données nécessaires au formulaire d'adresse d'expédition, à savoir la référence de l'adresse du client, l'alias, le titre, le nom, le prénom, l'initiale ou le deuxième prénom, la rue, la ville, le département ou la province, le code postal et le pays, des tables SHOPPER et SHADDR. La section Report contient les marques HTML pour l'affichage et la présentation de ces données. Notez que la première partie de cette section détermine, d'après la valeur de la colonne SATITLE (titre du client), la sélection par défaut dans la liste **Titre** du formulaire.

Pour afficher les valeurs extraites de la base de données dans les zones du formulaire, il serait logique d'utiliser des références aux colonnes appropriées des tables concernées, mais à la place, on constate que des chaînes vides (" ") sont affectées à chaque zone. En effet, dans le cas d'un client visiteur, toutes les colonnes à l'exception de SASHNBR et SANICK contiennent la chaîne de texte empty. Pour éviter que la chaîne de texte empty n'apparaisse dans toutes les zones, on leur attribue une chaîne vide (" ") comme valeur.

## Etude de la fonction de création du formulaire d'adresse réservé aux clients référencés

Examinons l'autre fonction, `displayRegisteredShopper()`, chargée de créer un formulaire d'adresse d'expédition pour les clients référencés.

Son code source est le suivant :

```

%FUNCTION(DTW_ODBC) displayRegisteredShopper()
{
 select sarfnbr, shrfnbr, sanick, satitle, salname, safname, samname,
saaddr1,
saaddr2, sacity, sastate, sacntry, sazipc
from shaddr, shopper
where (sashnbr=shrfnbr)
and (shlogid='${SESSION_ID}')
and (sanick=shlogid)
and (saadrflg='P')
%REPORT {
 %IF (V_satitle == "M.")
 @DTW_ASSIGN(sel1, "SELECTED")
 %ELIF (V_satitle == "Mme")
 @DTW_ASSIGN(sel2, "SELECTED")
 %ELIF (V_satitle == "Mlle")

```

```

 @DTW_ASSIGN(sel3, "SELECTED")
<!-- %ELIF (V_satitle == "Dr") >rencés
<!-- @DTW_ASSIGN(sel4, "SELECTED") >if" border="0" width="58"
height="29">
 %ENDIF
 <FORM>
 <TABLE WIDTH=600 CELLPADDING=10 BORDER=0 COLS=4>
 <TR>
 <TD COLSPAN=4><HR WIDTH=550 ALIGN="LEFT"></TD>
 </TR>
 <TR>
 <TD COLSPAN=4>Pour modifier l'adresse indiquée ci-dessous,
cliquez
 sur le bouton "Enregistrement" situé à gauche.</TD>
 <TR>
 <TD ALIGN="RIGHT">Titre</TD>
 <TD>
 <SELECT NAME="satitle" onFocus="this.blur();">
 <OPTION $(sel1) VALUE='M.'>M.
 <OPTION $(sel2) VALUE='Mme'>Mme
 <OPTION $(sel3) VALUE='Mlle'>Mlle
<!-- <OPTION $(sel4) VALUE='Dr'>Dr >="0" width="58" height="29">
 </SELECT>
 </TD>
 <TD ALIGN="RIGHT">Nom</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="salname"
VALUE="$(V_salname)" SIZE="22"
 MAXLENGTH="30" onFocus="this.blur();"></TD>
 </TR>
 <TR>
 <TD ALIGN="RIGHT">Prénom</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="safname"
VALUE="$(V_safname)" SIZE="22"
 MAXLENGTH="30" onFocus="this.blur();"></TD>
 <TD ALIGN="RIGHT">Initiale ou deuxième
prénom</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="samname"
VALUE="$(V_samname)" SIZE="22"
 MAXLENGTH="30" onFocus="this.blur();"></TD>
 </TR>
 <TR>
 <TD ALIGN="RIGHT">Adresse (ligne
1)</TD>
 <TD COLSPAN=3><INPUT TYPE="TEXT" NAME="saaddr1"
VALUE="$(V_saaddr1)" SIZE="64"
 MAXLENGTH="50" onFocus="this.blur();"></TD>
 </TR>
 <TR>
 <TD ALIGN="RIGHT">Adresse (ligne 2)</TD>
 <TD COLSPAN=3><INPUT TYPE="TEXT" NAME="saaddr2"
VALUE="$(V_saaddr2)" SIZE="64"
 MAXLENGTH="50" onFocus="this.blur();"></TD>
 </TR>
 <TR>
 <TD ALIGN="RIGHT">Ville</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="sacity" VALUE="$(V_sacity)"

```

```

SIZE="22"
 MAXLENGTH="30" onFocus="this.blur();"></TD>
 <TD ALIGN="RIGHT">Département/Province</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="sastate"
VALUE="$ (V_sastate)" SIZE="22"
 MAXLENGTH="20" onFocus="this.blur();"></TD>
</TR>
<TR>
 <TD ALIGN="RIGHT">Code postal</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="sazipc" VALUE="$ (V_sazipc)"
SIZE="22"
 MAXLENGTH="20" onFocus="this.blur();"></TD>
 <TD ALIGN="RIGHT">Pays</TD>
 <TD COLSPAN=1><INPUT TYPE="TEXT" NAME="sacntry"
VALUE="$ (V_sacntry)" SIZE="22"
 MAXLENGTH="30" onFocus="this.blur();"></TD>
</TR>
</TABLE>
%}

```

En termes de contenu, cette fonction est très proche de la fonction `displayTempShopper()`. Elle diffère cependant de cette dernière en ce qu'elle extrait et affiche les données d'adresse spécifiées par le client lors de son enregistrement. Comme la fonction `displayTempShopper()`, elle commence par une requête portant sur les tables `SHOPPER` et `SHADDR` afin d'extraire toutes les données relatives à l'adresse à insérer dans les zones du formulaire.

Toutefois, la section `Report` est quelque peu différente. Puisque la base de données contient cette fois l'adresse du client (alors que dans le cas d'un client visiteur, elle comprend seulement une entrée temporaire dont les colonnes contiennent pour la plupart la chaîne de texte "empty" en guise de valeur), elles peuvent être affichées au moyen de variables `Net.Data` qui référencent les colonnes appropriées de la base de données.

Notez également que la section `Report` commence par proposer au client de cliquer sur le bouton **Enregistrement** s'il veut modifier une quelconque information dans ses coordonnées (auquel cas le formulaire d'enregistrement lui est présenté). À l'aide de ce formulaire, le client a toute liberté de modifier l'adresse. Son carnet d'adresses est ensuite mis à jour (via l'instruction `RegisterUpdate`) pour refléter les modifications apportées. Si, au lieu d'utiliser ce formulaire, le client rectifiait les informations directement sur la page `Commande - Détails`, l'instruction `AddressProcess` serait mise à contribution pour traiter les changements. Cependant, plutôt que de mettre à jour l'enregistrement existant, cette instruction créerait une nouvelle entrée dans le carnet d'adresses. (Il n'existe pas d'instruction `Address` pour mettre à jour une entrée.) Aussi, pour éviter de créer un nouvel enregistrement de carnet d'adresses dans la table `SHADDR` chaque fois que le client modifie une information d'adresse sur la page `Commande - Détails`, on l'invite à utiliser la page `Mise à jour de l'enregistrement`. Par ailleurs, pour empêcher le client de modifier les informations à partir du formulaire d'adresse d'expédition, la section `Report` insère l'événement JavaScript `onFocus="this.blur();"` dans chaque zone afin d'y interdire toute écriture. Ainsi, le curseur disparaît si le client tente d'entrer du texte dans une zone du formulaire.

## Etude de la fonction de création du tableau des options de paiement

Vous savez maintenant comment sont créés les deux premiers éléments de la page Détails de la commande (le tableau récapitulatif de la commande et le formulaire d'adresse d'expédition). Examinons maintenant le code du troisième élément, le tableau présentant les options de paiement par carte de crédit.

Situé dans la section HTML Report, ce code se présente comme suit :

```
<TABLE WIDTH=600>
 <TR>
 <TD>
 & rbl; <P>Entrez ci-dessous les informations relatives à votre
carte de paiement
 et cliquez sur le bouton Valider la commande.
 Si vous ne souhaitez pas entrer les informations relatives au
paiement
 à ce moment, cliquez sur le bouton Annuler pour revenir à la
liste d'achats.</P>
 </TD>
 </TR>
</TABLE>
<TABLE BORDER=1 WIDTH=600>
 <TR>
 <TD ALIGN="CENTER">Carte de crédit</TD>
 <TD ALIGN="CENTER">Numéro</TD>
 <TD ALIGN="CENTER">Mois
 d'expiration</TD>
 <TD ALIGN="CENTER">Année
 d'expiration</TD>
 </TR>
 <TR>
 <TD ALIGN="CENTER" VALIGN="MIDDLE">
 <SELECT NAME="cctype" SIZE=1>
 <OPTION SELECTED VALUE="AMEX">American Express</OPTION>
 <OPTION VALUE="MAST">MasterCard</OPTION>
 <OPTION VALUE="VISA">VISA</OPTION>
 </SELECT>
 </TD>
 ; <TD ALIGN="CENTER" VALIGN="MIDDLE">
 <INPUT TYPE=TEXT SIZE=15 MAXLENGTH=256 NAME="ccnum"
VALUE="$(ccname)">
 </TD>
 <TD ALIGN="CENTER" VALIGN="MIDDLE">
 <SELECT NAME="ccxmonth" SIZE=1>
 <OPTION SELECTED VALUE="1"> 01 </OPTION>
 <OPTION VALUE="2">02</OPTION>
 <OPTION VALUE="3">03</OPTION>
 <OPTION VALUE="4">04</OPTION>
 <OPTION VALUE="5">05</OPTION>
 <OPTION VALUE="6">06</OPTION>
 <OPTION VALUE="7">07</OPTION>
 <OPTION VALUE="8">08</OPTION>
 <OPTION VALUE="9">09</OPTION>
 <OPTION VALUE="10">10</OPTION>
 <OPTION VALUE="11">11</OPTION>
 <OPTION VALUE="12">12</OPTION>
 </SELECT></TD>
 <TD ALIGN="CENTER">
```

```

<SELECT NAME="ccxyear" SIZE=1>
 <OPTION SELECTED VALUE="1997">1997</OPTION>
 <OPTION VALUE="1998">1998</OPTION>
 <OPTION VALUE="1999">1999</OPTION>
 <OPTION VALUE="2000">2000</OPTION>
 <OPTION VALUE="2001">2001</OPTION>
 <OPTION VALUE="2002">2002</OPTION>
 <OPTION VALUE="2003">2003</OPTION>
</SELECT></TD>
</TR>
</TABLE> ;
</FORM>
<P>

```

Cette section HTML construit simplement un tableau comportant des valeurs fixes.

## Etude du bouton Valider la commande

Penchons-nous sur le bouton **Valider la commande** qui fait suite au tableau des options de paiement. Le code source affichant ce bouton est le suivant :

```

<TABLE BORDER=0>
 <TR>
 <TD COLSPAN=4><HR WIDTH=550 ALIGN="LEFT"></TD>
 </TR>
 <TR>
 <TD>

 <IMG SRC="/grocery/g_subord.gif" WIDTH=138 HEIGHT=24

 </TD>
 </TR>
</TABLE>

```

Cette section HTML affiche l'image (fichier GIF) du bouton avec un ancrage prévu pour un lien hypertexte. Le lien contient l'événement JavaScript onClick qui appelle la fonction JavaScript sendURL. Cette dernière remplace l'ancrage par un URL qui assure la validation de la commande, des informations d'expédition et des informations de paiement. L'URL diffère selon que le client est référencé ou non. Le code source de la fonction sendURL est le suivant :

```

function sendURL(thisAnchor) {
 var tempValue = '${isTempShopper}'
 var newUrl = ""
 var url = ""
 if(tempValue == "T") {
 newUrl = "/cgi-bin/ncommerce3/AddressUpdate?"
 newUrl += "sarfnbr=$(address_rn)"
 newUrl += "&salname=" + document.forms[0].salname.value
 newUrl += "&satitle=" + document.forms[0].satitle.options
 newUrl += "[document.forms[0].satitle.selectedIndex].value"
 newUrl += "&safname=" + document.forms[0].safname.value
 newUrl += "&samname=" + document.forms[0].samname.value
 newUrl += "&saaddr1=" + document.forms[0].saaddr1.value
 newUrl += "&saaddr2=" + document.forms[0].saaddr2.value
 newUrl += "&sacity=" + document.forms[0].sacity.value
 newUrl += "&sastate=" + document.forms[0].sastate.value
 newUrl += "&sazipc=" + document.forms[0].sazipc.value
 newUrl += "&sacntry=" + document.forms[0].sacntry.value
 newUrl += "&sashnbr=$(shopper_rn)"
 newUrl += "&sanick=$(nick)"
 newUrl += "&mode=change"
 }
}

```

```

 newUrl += "&sascontyp=S"
 newUrl += "&saadrflg=P"
 newUrl += "&safield2=T"
 url = "&url=%2Fcgi-bin%2Fcommerce3%2F%3B0rderProcess%3Forder_rn%3D$(order_rn)"
 url += "%26merchant_rn%3D$(merchant_rn)"
 url += "%26ccnum%3D" + document.forms[0].ccnum.value
 url += "%26cctype%3D" + document.forms[0].cctype.options
 [document.forms[0].cctype.selectedIndex].value
 url += "%26ccxmonth%3D" + document.forms[0].ccxmonth.options
 [document.forms[0].ccxmonth.selectedIndex].value
 url += "%26ccxyear%3D" + document.forms[0].ccxyear.options
 [document.forms[0].ccxyear.selectedIndex].value
 newUrl += url
 newUrl = replaceString(newUrl, " ", "+")
 } else {
 newUrl = "/cgi-bin/ncommerce3/OrderProcess?order_rn=$(order_rn)"
 url = "&merchant_rn=$(merchant_rn)"
 url += "&ccnum=" + document.forms[0].ccnum.value
 url += "&cctype=" + document.forms[0].cctype.options
 [document.forms[0].cctype.selectedIndex].value
 url += "&ccxmonth=" + document.forms[0].ccxmonth.options
 [document.forms[0].ccxmonth.selectedIndex].value
 url += "&ccxyear=" + document.forms[0].ccxyear.options
 [document.forms[0].ccxyear.selectedIndex].value
 newUrl += url
 newUrl = replaceString(newUrl, " ", "+")
 }
 thisAnchor.href = newUrl
 return newUrl
}

```

Passons en revue chaque section de cette fonction. Elle commence par la déclaration de trois variables locales (qui seront utilisées au sein même de la fonction). La valeur de `isTempShopper` est maintenant stockée dans une variable appelée `tempValue`. Deux autres variables, `url` et `newUrl`, sont déclarées avec chacune une chaîne vide pour valeur.

Le déroulement du reste de la fonction dépend de la valeur de `tempValue`.

- Si `tempValue` contient la valeur T (ce qui est le cas si le client est un visiteur), la fonction construit un URL qui consiste en deux sous-URL. Le premier URL contient l'instruction `AddressUpdate` et une série de paires nom-valeur qui correspondent chacune à une zone du formulaire d'adresse d'expédition que le client a rempli. Cette instruction reçoit également la référence d'adresse, la référence du client, son alias et le fanion P (pour indiquer qu'il s'agit d'un enregistrement d'adresse permanent). L'instruction `AddressUpdate` utilise tous ces paramètres pour insérer une nouvelle entrée de carnet d'adresses dans la table `SHADDR`.

Une fois que l'instruction `AddressUpdate` s'est exécutée normalement, la fonction crée un deuxième URL constitué de l'instruction `OrderProcess` et d'une série de paires nom-valeur correspondant respectivement aux informations suivantes : référence de la commande, référence du commerçant, numéro de carte de crédit, type de carte de crédit, mois d'expiration, année d'expiration (ces quatre derniers éléments ayant été fournis par le client lorsqu'il a rempli le formulaire de paiement). L'instruction `OrderProcess` traite les informations de la commande.

Une fois que son exécution a abouti, la macro `t_orderacpt.d2w` est à son tour exécutée pour afficher un message d'acceptation de la commande et le numéro de référence de celle-ci.

- Si `tempValue` ne contient pas la valeur `T` (ce qui est le cas si le client est déjà référencé), la fonction construit un URL constitué uniquement de l'instruction `OrderProcess`. L'instruction `AddressUpdate` n'est pas nécessaire, car le client possède déjà une entrée de carnet d'adresses valide. (Appeler cette instruction aurait seulement pour effet de créer un double de l'entrée dont nous n'aurions pas l'utilité.) Les données extraites de la table `SHADDR` détermineront à quelle adresse expédier la commande du client.

## Etude du bouton Annuler

Examinons le code générant le bouton **Annuler**.

```
<TD WIDTH=200 ALIGN="RIGHT">
 <A HREF="#" onClick='window.location.href="/cgi-
bin/ncommerce3/OrderItemDisplay"'>

</TD>
</TR>
</TABLE>
```

Cette section HTML affiche le bouton avec un lien contenant l'instruction `OrderItemDisplay` pour afficher la page Chariot.

---

## Personnalisation d'une page d'erreur système

Vous allez maintenant personnaliser une page HTML qui s'affiche lorsqu'il se produit une erreur empêchant la poursuite du traitement. La plupart des pages d'erreur système sont des fichiers HTML, ce qui simplifie leur personnalisation. Les pages d'erreur du magasin que vous créez à l'aide de ce tutoriel sont déjà personnalisées, à l'exception du fichier `cmdinc.html`, qui est celui sur lequel vous allez travailler. Ce fichier affiche un message signalant une erreur de structure dans une instruction. Il précise que l'instruction n'a pas pu s'exécuter en raison d'une erreur de syntaxe ou parce que des paramètres requis par le système n'ont pas été transmis.

1. Faites une copie de sauvegarde de tous les fichiers se trouvant dans le répertoire suivant :

**WIN** \ibm\netcommerce3\html\<langue>\ncerror

**AIX** /usr/lpp/NetCommerce3/html/<langue>/ncerror

**SUN** /opt/IBNnetc/NetCommerce3/html/<langue>/ncerror

Toutes les pages HTML d'erreur système doivent se trouver dans ce répertoire afin que le serveur puisse y accéder. Pour éviter de remplacer les pages d'erreur système par défaut, il convient d'en effectuer une copie de sauvegarde.

2. Copiez tous les fichiers du répertoire `\html\<langue>\tutorial\error` dans le répertoire `\html\<langue>\ncerror`.
3. Assurez-vous que vous avez pris connaissance des instructions relatives à l'édition des modèles du tutoriel.
4. A l'aide d'un éditeur de texte, ouvrez le fichier `cmdinc.html` à partir du répertoire `\ncerror`. Etudiez-en le contenu. Il affiche une image générique et un message indiquant que l'exécution d'une instruction a échoué.

5. Voyons comment le message se présente actuellement dans le magasin. Pour qu'il s'affiche, vous devez provoquer l'échec d'une instruction. Aussi, vous allez supprimer des paramètres indispensables dans une instruction afin qu'elle ne puisse s'exécuter correctement. A l'aide d'un éditeur de texte, ouvrez le fichier t\_navmid.d2w à partir du répertoire \tutorial. Il s'agit de la macro qui génère la liste des catégories principales dans le cadre intermédiaire gauche de la devanture.
6. Repérez la ligne qui contient l'instruction appelée lorsque le client clique sur l'une des catégories proposées :  

```
<A HREF="/cgi-bin/ncommerce3/CategoryDisplay?cgrfnbr=$(V_cgrfnbr)
&cgmenbr=$(V_cgmenbr)" target="main">
```
7. Retirez les paramètres de cette instruction de façon à conserver seulement la partie suivante :  

```
<A HREF="/cgi-bin/ncommerce3/CategoryDisplay"
target="main">
```

Les paramètres que vous venez de supprimer, cgrfnbr (référence de catégorie) et cgmenbr (référence du commerçant), sont requis par le système pour localiser la macro d'affichage des sous-catégories de la catégorie sélectionnée. Sans ces paramètres, l'instruction ne peut plus s'exécuter normalement.
8. Testons l'instruction. Dans le magasin, cliquez sur l'une des catégories répertoriées dans la liste. La page d'erreur système s'affiche.
9. Vous allez maintenant personnaliser la page du magasin East West Food Mart en modifiant le texte, en remplaçant l'image par le logo du magasin et en ajoutant un fond blanc à la page. Modifiez le fichier cmdinc.html comme suit :  

```
<HTML>
<HEAD><TITLE>Erreur Centre commercial/Magasin</TITLE></HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#399FFF"
VLINK="#A7A7A7">

<CENTER>
& 1t;IMG SRC="/grocery/hm_logosmall.gif">

Des incidents se sont produits au niveau du magasin East West
Food Mart.

Veuillez relancer la procédure ultérieurement.<P>
```
10. Enregistrez le fichier sous le même nom. **Remarque** : Toutes les pages de message d'erreur système que vous personnalisez doivent être enregistrées sous leur nom d'origine.
11. Voyons maintenant comment se présente notre page ainsi modifiée. Retournez au navigateur affichant le message d'erreur. Cliquez sur une catégorie dans la liste. Votre page d'erreur système personnalisée s'affiche.
12. Remplacez les fichiers du répertoire ncerror par ceux que vous avez sauvegardés à l'étape 1.
13. Rétablissez les paramètres de l'instruction CategoryDisplay dans la macro t\_navmid.d2w.

Vous avez terminé la personnalisation d'une page d'erreur système. Toutes les autres pages d'erreur système peuvent être personnalisées selon le même principe.

---

## Suppression d'adresses temporaires

Vous vous êtes déjà servi de l'utilitaire de nettoyage de la base de données pour supprimer tous les produits ayant été marqués pour suppression. Vous allez maintenant l'utiliser pour accomplir une autre activité de nettoyage : la suppression des enregistrements d'adresses temporaires.

Comme vous avez pu le voir au cours de ce tutoriel, des enregistrements d'adresses temporaires sont souvent créés dans la base de données du magasin East West Food Mart : chaque fois qu'un client accède au magasin, un enregistrement d'adresse est créé et stocké dans la table SHADDR ; si ce client procède à son enregistrement ou passe une commande, ou s'il s'agit d'un client déjà référencé et qu'il met à jour ses informations d'enregistrement, un nouvel enregistrement d'adresse permanent est créé dans la table SHADDR. Dans chacun de ces cas de figure, l'ancien enregistrement d'adresse se voit affecter le type "temporaire" (une colonne de la table SHADDR est réservée à cet effet).

A mesure que le temps passe et que les clients sont de plus en plus nombreux à visiter le magasin, un nombre sans cesse croissant d'enregistrements temporaires et inutiles vient encombrer la table SHADDR. Vous allez apprendre à les supprimer au moyen de l'utilitaire de nettoyage de la base de données.

1. Affichons d'abord les enregistrements d'adresses temporaires qui se trouvent actuellement dans la table SHADDR. Veillez à utiliser la machine sur laquelle Net.Commerce est installé. Puis, à partir d'une fenêtre de commande DB2, tapez l'instruction suivante pour vous connecter à la base de données du tutoriel :

```
db2 connect to tutorial
```

2. Tapez ce qui suit :

```
db2 select * from shaddr
```

Vous obtenez la liste des enregistrements d'adresses présents dans la table SHADDR. Notez que certains comportent la valeur T dans la colonne SAADRFLG. Il s'agit des enregistrements temporaires que l'on veut supprimer. (La valeur P indique un enregistrement permanent.)

3. A partir d'une ligne de commande quelconque, accédez au répertoire :

```
WIN \ibm\netcommerce3\bin
```

```
AIX /usr/lpp/NetCommerce3/bin
```

```
SUN /opt/IBMnetc/NetCommerce3/bin
```

4. A l'invite, tapez la commande suivante :

```
ncclean -table shaddr -db tutorial
```

Tous les enregistrements d'adresses temporaires sont maintenant supprimés de la table SHADDR.

5. Voyons le résultat dans la base de données. Retournez à la fenêtre de commande DB2 et entrez l'instruction suivante à l'invite :

```
db2 select * from table shaddr
```

Seuls les enregistrements permanents subsistent dans la table SHADDR.

---

## Félicitations !

Vous voici parvenu à la fin du tutoriel East West Food Mart. Celui-ci vous a permis de vous familiariser avec les activités suivantes dans Net.Commerce.

- Utilisation de l'Administrateur Net.Commerce pour entrer et gérer les données de votre magasin

- Emploi de l'utilitaire de nettoyage de la base de données pour supprimer de la base de données les produits inutiles et les adresses temporaires du client
- Emploi de l'utilitaire d'importation globale pour importer dans la base de données des informations sur tous les produits et toutes les catégories, à l'aide d'un fichier texte
- Création de pages de magasin à l'aide de macros Net.Data
- Personnalisation des pages du magasin
- Personnalisation des pages d'erreur système
- Utilisation de JavaScript pour ajouter des caractéristiques telles que l'affichage du prix
- Configuration du magasin de façon que les clients visiteurs puissent passer des commandes sans s'enregistrer
- Création d'une fonction non prioritaire modifiant la façon dont les informations sont traitées par le système

Vous pouvez maintenant mettre à profit les connaissances acquises en suivant ce tutoriel pour créer et gérer votre propre magasin Net.Commerce. Vous trouverez ci-dessous les premières étapes des différentes méthodes disponibles pour créer un magasin. Chacune de ces méthodes vous renvoie à une procédure expliquant plus en détail le processus de création d'un magasin et présentant d'autres outils à votre disposition.

- Création d'un magasin en partant du Gestionnaire de site et du Gestionnaire de magasin
- Création d'un magasin en partant du Créateur de magasin
- Création d'un magasin à partir d'un modèle

Vous pouvez également créer un centre commercial, c'est-à-dire un site regroupant plusieurs magasins. Pour ce faire, vous disposez des méthodes suivantes :

- Création d'un centre commercial en partant du Gestionnaire de site et du Gestionnaire de magasin
- Création d'un centre commercial à partir du modèle Le Métropole

---

## Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Cela ne signifie pas qu'IBM ait l'intention de les y annoncer. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM EMEA Director of Licensing  
IBM Europe Middle-East Africa  
our Descartes La Défense 5  
2, avenue Gambetta 92066 Paris-La Défense CEDEX  
France

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations  
IBM Canada Ltd.  
3600 Steeles Avenue East  
Markham, Ontario  
L3R 9Z7  
Canada

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation  
P.O. Box 12195  
3039 Cornwallis Road  
Research Triangle Park, NC 27709-2195  
U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

Le présent document peut contenir des informations sur des produits d'autres sociétés et des références aux sites Internet desdites sociétés. IBM ne fournit aucune garantie d'aucune sorte liée au contenu, à l'exhaustivité ou à l'utilisation de telles informations.

Ce produit est basé sur le protocole SET et est en cours de test de compatibilité, mais à la date de conception de ce produit, il n'a pas encore obtenu la certification de compatibilité SET de SET Secure Electronic Transaction LLC.



Ce produit contient des composants RSA.

## Marques

Les termes qui suivent sont des marques d'International Business Machines Corporation dans certains pays :

IBM	AIX	IBM Payment Server
DB2	DB2 Universal Database	DB2 Extensions
DATABASE 2	IMS	Net.Data
MQSeries	CICS	

Microsoft, Windows, Windows NET et le logo Windows sont des marques de Microsoft Corporation.

Lotus et Domino sont des marques de Lotus Development Corporation dans certains pays.

Adobe et Acrobat sont des marques d'Adobe Systems Incorporated.

Pentium est une marque d'Intel Corporation dans certains pays.

Netscape et Netscape Communicator sont des marques de Netscape Communications Corporation dans certains pays.

Java, et toutes les marques et logos incluant Java, sont des marques de Sun Microsystems, Inc.

SET Secure Electronic Transaction, SET et le logo SET sont des marques de SET Secure Electronic Transaction LLC. Toute utilisation des marques sans licence écrite fournie par SET Secure Electronic Transaction LLC est strictement interdite.

Les images de cartes de paiement, les marques et noms commerciaux indiqués dans ce produit ne doivent être utilisés que par des commerçants autorisés par le propriétaire de la marque de carte de paiement à accepter des règlements au moyen de cette carte.

D'autres sociétés sont propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

## Informations sur la licence

Lorsque vous faites l'acquisition de la licence du logiciel Net.Commerce, vous recevez une licence pour le logiciel IBM Net.Commerce pour AIX ou pour le logiciel IBM Net.Commerce pour Windows NT. Vous recevez également une licence pour les versions des logiciels suivants :

- IBM DB2 Universal Database
- Domino\*\* Go Webserver
- IBM DB2 Extensions

Ces licences ne vous autorisent à utiliser ces produits qu'en association avec le logiciel Net.Commerce pour lequel vous avez acquis la licence.

Si vous souhaitez utiliser les logiciels Oracle et Domino Go Web Server ou Netscape Enterprise Server 3.0 plutôt que les logiciels DB2 et Domino Go Webserver, fournis sur le CD-ROM Net.Commerce, vous devez vous procurer des licences pour ces logiciels.





IBM