Client Web OMNIS

Août 2000

Table des matières

A propos de ce manuel

Ce manuel décrit le Client web OMNIS™ et comment utiliser OMNIS Studio pour développer des applications et des solutions commerciales diffusables sur le World Wide Web.

Ce manuel contient les chapitres suivants :

- **Introduction**
  présente une vue générale de la technologie Client web OMNIS et décrit les différents éléments requis dans une solution OMNIS web.
- **Tutorial Client Web OMNIS**
  vous guide lors de la création d'une application simple qui permet de naviguer dans le fichier de données d'une bibliothèque virtuelle et décrit comment la diffuser sur le web.
- **Création de *Remote forms* OMNIS**
  décrit comment créer des *Remote forms* OMNIS, les classes GUI auxquelles l'utilisateur accède à travers son navigateur ; ce chapitre inclut une description des différents modèles et assistants *Remote form* disponibles dans OMNIS Studio.
- **Classes *Remote task***
  décrit comment utiliser les *Remote tasks* en association avec les *Remote forms* et inclut la description des différents modèles et assistants de *Remote task*
- **Distribuer sa solution OMNIS Web**
  décrit comment préparer ses pages html qui contiennent le client Web OMNIS et montre comment préparer la partie serveur de votre solution Web.
- **Guide de dépannage et Foire aux questions**
  répond aux questions et aux problèmes fréquents rencontrés avec les *Remote forms* et Client web OMNIS
- **Référence**
  Liste de tous les évènements, propriétés et méthodes associés aux *Remote forms* et aux *Remote tasks*.

Chapitre 1 – Introduction

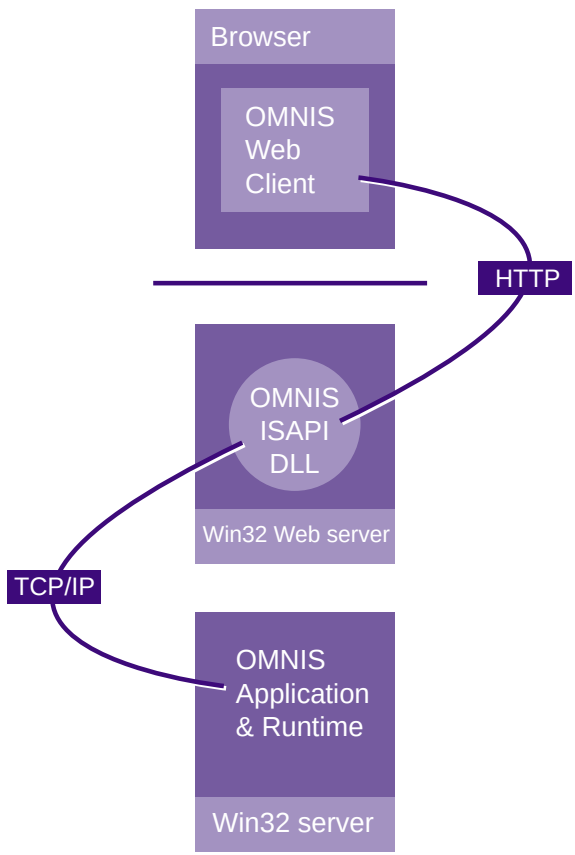
Avec OMNIS Studio et Client web OMNIS™ vous pouvez créer tout type d'application et les distribuer sur le web. L'application, écrite en OMNIS Studio, peut tout gérer, depuis un service de santé jusqu'à des présentations multimédia ou de presse, en passant par les ressources humaines, le commerce électronique, l'éducation ou l'administration. Le client Web OMNIS vous permet d'imbriquer votre application dans des pages web standards et permet à quiconque qui possède un navigateur adéquate, depuis n'importe où dans le monde, d'accéder aux données 24 heures sur 24 et 7 jours sur 7. Et ce qui rend cette technologie encore plus incroyable, c'est que les fenêtres préparées à cette effet sont entièrement réalisées avec le L4G OMNIS Studio, ses *wizards* et ses composants web, pas de Java ni d'html complexe : c'est aussi simple que ça !

La copie d'écran ci-après montre un navigateur web qui contient une application d'exemple qui permet de parcourir les ouvrages d'une bibliothèque en ligne. La fenêtre affichée dans le navigateur utilise le Client Web OMNIS, et contient différents types de composants comme un contrôle *Sidebar*, diverses listes et rubriques, des boutons, des boites à cocher, etc...

OMNIS Studio vous permet de créer vos *Remote forms* à destination des utilisateurs et de les publier à travers un navigateur web. Une *Remote form* est une classe OMNIS qui contient tous les contrôles nécessaires pour afficher vos données et permettre aux utilisateurs d'interagir avec votre application et vos bases de données via le web. Une fois connectée à OMNIS, la *Remote form* envoie des évènements vers l'application OMNIS serveur, qui à son tour, renvoie les résultats vers le client.

Vous pouvez adapter une application OMNIS Studio existante pour lui permettre un accès web en ajoutant des *Remote forms* à votre application OMNIS et en ajoutant un ou deux composants à votre serveur web et votre infrastructure réseau.

Une solution Client web OMNIS se présente en deux parties : le client et le serveur.



– **Le Client**
Le client est le navigateur de l'utilisateur final utilisé pour accéder et parcourir vos pages web (écrites en html standard) qui contiennent le client Web OMNIS. Ce dernier est imbriqué dans une page html qui affichera alors la ou les *Remote form*(*s*) stockées dans votre librairie OMNIS, localisée sur votre serveur.

– **Le Serveur**
La partie serveur de votre solution Web OMNIS comprend votre serveur web et un OMNIS Serveur, c'est-à-dire votre application OMNIS, un runtime OMNIS et votre base de données. Votre serveur web requiert une extension serveur (comme OMNISAPI.DLL), composant invisible qui gère les liens entre le client et le serveur.

Le Client

Pour accéder à votre solution web OMNIS à tout moment, depuis n'importe où dans le monde, l'utilisateur n'a besoin que d'un navigateur web standard. La première fois qu'il tentera d'accéder à votre application, il devra télécharger et installer le client web OMNIS que vous fournirez sur votre site web. Une fois le client installé, l'utilisateur pourra naviguer sur la ou les page(s) appropriée(s) de votre site web qui présente la solution web OMNIS. Cette section résume les composants requis sur la machine cliente.

Navigateur Internet

Le client web OMNIS est disponible sous forme d'un ActiveX ou d'un plug-in Netscape, le navigateur de votre client doit supporter l'une de ces deux technologies. Par exemple, certains utilisateurs accèderont aux pages qui contiennent le client Web ActiveX avec Windows Internet Explorer™, d'autres avec le plug-in Netscape pour Netscape Navigator™. Lors de la distribution de votre solution sur le web, vous devez tenir compte des différents navigateurs utilisés et adapter vos pages html qui présentent le client web OMNIS en accord.

Client Web OMNIS

Le client web OMNIS est un ActiveX ou un plug-in Netscape que vous placez dans vos pages html. Vous pouvez cependant utiliser le client web OMNIS dans tous développements qui supportent les ActiveX, y compris OMNIS lui-même ainsi que de nombreuses applications d'environnement de développement.
Sur une machine cliente Windows, le client web OMNIS utilise Wininet API pour communiquer avec l'extension serveur web OMNIS via HTTP, ou HTTPS si une transmission sécurisée est nécessaire. Les communications entre l'extension web serveur OMNIS, le runtime OMNIS et sa librairie sont établies via TCP/IP.

Pour utiliser une solution web OMNIS, un utilisateur doit télécharger et installer le Client Web OMNIS. Vous pouvez fournir un lien sur votre site web pour permettre aux utilisateurs de télécharger l'installeur du Client Web OMNIS. Les installeurs Client Web OMNIS sont fournis sur le CD OMNIS Studio et vous pouvez réaliser vos propres installeurs avec les composants nécessaires à votre application. Les derniers installeurs Client Web OMNIS sont aussi téléchargeables gratuitement sur le serveur web de votre distributeur.

Lorsque l'utilisateur accède à votre solution web à travers son navigateur, le client web OMNIS est activé et une connexion est établie avec l'extension serveur de votre serveur web (voir la section Serveur plus bas). Une fois la connexion établie, le client web OMNIS Web reçoit les données de classe de la *Remote form* plus toutes les données des variables d'instance, les pages d'icônes, les tables de police, etc, lues depuis votre librairie OMNIS. Le client web OMNIS reproduit l'instance visuelle de la *Remote form* et de tous ses contrôles, le mode saisie de données est enclenché et l'utilisateur peut alors utiliser la *Remote form*.

Vous pouvez surveiller et contrôler ce qui se passe dans la *Remote form* en surveillant les évènements (*events*) émis par l'utilisateur lorsqu'il utilise la *Remote form*. Par exemple, un *event* est généré dans la *Remote form* lorsque l'utilisateur clique sur une rubrique, lorsqu'il sélectionne une ligne dans une liste, lorsqu'il appuie sur un bouton ou tabule de rubrique en rubrique. Vous pouvez détecter et répondre à un *event* en créant un gestionnaire d'*events*, stocké avec les objets de votre *Remote form*. Lorsqu'un *event* est généré, il est envoyé depuis le client vers l'application serveur et la méthode de gestion des *events* pour l'objet est exécutée. La méthode peut faire tout ce que vous désirez, par exemple, elle peut sauver les données de la *Remote form* dans la base de données ou renvoyer le résultat d'un calcul au client. Par défaut, tous les *events* sont désactivés dans une *Remote form* pour minimiser le trafic inutile, mais vous pouvez activer des *events* spécifiques pour chaque contrôle de la *Remote form*. Lorsque vous créez votre application vous devez considérer l'impact du trafic généré par les *events* par rapport aux per*form*ances. Les détails sur la gestion des *events* et les astuces pour réduire le trafic sur le réseau sont décrits plus loin dans ce manuel.

Sous Windows, notez que le client web OMNIS Web peut potentiellement être placé sous n'importe quel environnement qui supporte ActiveX, y compris les applications écrites en Delphi ou d'autres outils de développement. Le client web OMNIS ActiveX peut être placé au sein d'une fenêtre OMNIS.

Installer le client web pour le développement

Vous devez installer le client web OMNIS pour créer et tester vos Remote *form*s avec OMNIS Studio. Vous pouvez tester les *Remote forms* pendant que vous les créez avec Ctrl/Cmnd-T, uniquement si vous avez installé le client web OMNIS. Le CD OMNIS Studio propose un installeur du client web OMNIS dans le dossier Webclient/Client. Il y a deux installeurs séparés pour ActiveX et le plug-in Netscape qui installent le client web afin que vous puissiez créer et tester vos *Remote forms*.

Le Serveur

La partie serveur de votre solution web OMNIS comprend un serveur web, une extension web serveur fournie par OMNIS, le runtime OMNIS, votre application OMNIS et vos sources de données, soit une base de données OMNIS ou un serveur de données. Tous ces éléments peuvent être ou ne pas être physiquement regroupés sur la même machine, mais typiquement ils seront placés sur le même LAN. Cette section résume les composants dont vous avez besoin sur la partie serveur de votre solution web OMNIS.

Le serveur web

Le client web OMNIS requiert un serveur web au standard Win32. Le logiciel serveur web est disponible chez différentes tierces parties, y compris les serveurs web téléchargeables sur Internet. Le serveur web gère les pages html qui contiennent le client web OMNIS, et permettent d'accéder à votre application OMNIS. Avec votre outil de recherche favori, vous pouvez rechercher "*web server software*" pour trouver des dizaines de sources de serveurs web adéquats.

Votre serveur web requiert aussi une extension serveur, fournie par OMNIS, qui gère les communications entre le client et votre application OMNIS. Consultez la section suivante.

Notez que le serveur OMNIS ou le runtime ***n'est pas un serveur web***. Le runtime OMNIS exécute votre application, contient les méthodes de gestion d'évènements, applique les règles de logique établis, etc ; il ne sert pas de pages html. Votre solution web OMNIS requiert un serveur web, comme décrit ci-dessus.

L'extension serveur web

L'extension serveur web est une pièce du logiciel fournie par OMNIS qui doit être placée sur votre serveur web et qui attend et gère les requêtes envoyées d'un client web OMNIS depuis le navigateur web d'un utilisateur. De façon similaire, l'extension passe tous les résultats envoyés par le serveur OMNIS en retour au client web OMNIS du navigateur de l'utilisateur.

L'extension serveur web est installée typiquement dans le dossier cgi-bin de votre serveur web. A l'heure actuelle, le client web OMNIS supporte MS ISAPI et CGI. Une extension serveur séparée est fournie pour chaque interface. Dans l'arborescence OMNIS Studio installée, vous pouvez retrouver ces extensions dans le dossier OMNIS/Webclient/Server/Webserver. Le dossier ISAPI contient OMNISAPI.DLL, con*form*e à l'interface MS ISAPI. Le dossier CGI contient 'nph-omniscgi.exe' qui supporte les CGI. Dans le futur, d'autres interfaces pourront être supportées, comme Netscape Server.

Si votre serveur web ne supporte pas l'interface MS ISAPI, vous devrez utiliser le programme cgi nph-omniscgi.exe.

Le Runtime OMNIS

Pour compléter votre solution web la partie serveur requiert un serveur OMNIS ou un Runtime OMNIS ainsi que votre librairie qui contient vos *Remote forms* et toute la logique de gestion nécessaire ainsi que les possibilités d'impression. Le Runtime OMNIS et votre librairie sont sur un serveur Win32 ou une machine Win32 locale à votre serveur.

Le runtime OMNIS doit être sérialisé avec un nombre d'accès concurrentiel spécifique. Par exemple, si votre runtime OMNIS est sérialisé avec un numéro 100 utilisateurs, l'application reste ouverte à tout le monde mais seulement 100 connexions simultanées sont autorisées. Le numéro de série requis par le client web OMNIS est un numéro multi-utilisateur particulier suivi d'un “W”, un numéro multi-utilisateur standard suivi d'un “M” ne fonctionnera pas dans ce cas.

L'application OMNIS

Vos *Remote forms* et tout le code qui compose votre solution web sont stockés dans un fichier librairie OMNIS. Vos *Remote forms* sont stockées sous la *form*e de *classes Remote form* OMNIS. Vous devrez aussi créer une classe *Remote task* pour gérer la connexion de chaque client. Ces classes sont décrites plus en détail plus loin dans ce manuel. La création de librairies OMNIS est décrite en détail dans le manuel *Utilisation OMNIS Studio*.

Les classes *Remote form* OMNIS sont conçues pour être utilisées avec l'environnement de développement OMNIS Studio. Vous pouvez débugger ou tester vos *Remote forms* pendant le développement avec votre navigateur web. Il existe de nombreux types de rubriques et de contrôles que vous pouvez utiliser dans vos *Remote forms*. Les contrôles de *form* sont implémentés en tant que composants externes, et comprennent des boutons de sélections, des rubriques texte mono et multi-lignes, des boutons radio, des boîtes à cocher, des *Sidebars*, des listes, des listes déroulantes et des *combo box*. De plus, vous pouvez utiliser les contrôles *page pane* intégrés à vos *Remote forms*. Vous pouvez créer vos propres contrôles sous forme de composants externes (en C++) et les utiliser dans vos *forms*. La classe *Remote form* et ses contrôles sont entièrement cross-platform, les navigateurs web Windows et Macintosh peuvent accéder aux mêmes *forms* stockées dans votre librairie OMNIS.

Notez qu'OMNIS Studio comprend un exemple de librairie client web dans l'*Examples Browser*, disponible dans le menu **Tools** de la barre de menus principale d'OMNIS. Cette librairie d'exemple propose un système d'administration scolaire et d'emprunt de livres au travers de différentes *Remote forms*.

Forms standard HTML

Comme alternative au client web OMNIS et ses *Remote forms*, OMNIS Studio vous permet d'interagir avec votre application OMNIS et sa base de données via l'internet en utilisant des *forms* html standard. Dans ce cas, vos *forms* html se connectent à une *Remote task* OMNIS directement, et aucune *Remote forms* n'est requise pour ce type d'interaction. Consultez le chapitre *Remote tasks* pour plus de détails sur la création de *forms* html standard pour un accès direct.

Chapitre 2 – Tutorial Client Web OMNIS

Développer une solution web OMNIS avec le client web OMNIS est relativement simple une fois acquie la manière de développer une application OMNIS en général. La première partie de ce chapitre décrit comment créer une simple *Remote form* pour naviguer dans les données de bibliothèque de l'exemple fourni dans le dossier *Examples* d'OMNIS Studio. Il utilise les assistants *Remote form* comme point de départ de votre *form* et implique un peu de programmation. Vous pouvez créer des *Remote forms* et les tester localement avec la version de développement OMNIS. La deuxième partie du Tutorial décrit comment distribuer la fenêtre *Book browser* et considère que vous possédiez un serveur web personnel comme celui fournit avec Microsoft FrontPage, et utilise la version de développement d'OMNIS pour exécuter l'application. Cependant, pour la distribution finale de votre solution OMNIS web solution, vous devrez vous procurer et installer un serveur web, définir le serveur Win32 pour qu'il lance le runtime OMNIS et votre librairie et acheter les licences appropriées pour le runtime serveur OMNIS.

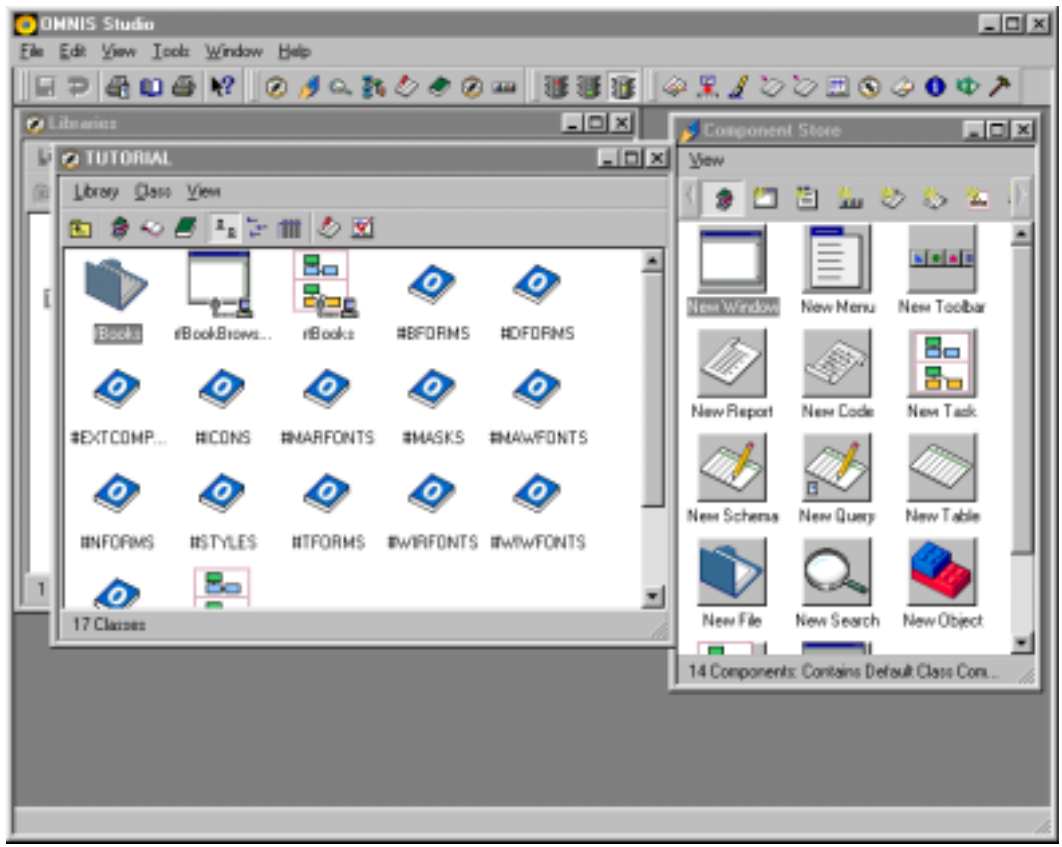
IMPORTANT : Vous devez installer le client web OMNIS pour suivre le tutorial ou créer et tester chaque *Remote form* vous-même. Le CD OMNIS Studio propose un installeur client web OMNIS dans le dossier Webclient/Client. Il y a deux installeurs séparés : ActiveX, pour Microsoft Internet Explorer et Plug-in Netscape pour Netscape Navigator. Vous pouvez utiliser l'ActiveX ou le plug-in Netscape pour le tutorial, mais vous devez lancer le bon installeur avant de commencer le tutorial.

Ce tutorial considère que vous savez utiliser le *Component Store* et le *Property Manager* OMNIS pour créer et modifier les rubriques et les contrôles, et que vous connaissiez l'emploi du *Method editor* pour saisir les méthodes et les variables. Pour plus de détails sur ces sujets, référez vous au manuel *Utilisation OMNIS Studio*.

Créer une *Remote form*

Pour développer une solution web OMNIS, vous devez créer une *Remote form* OMNIS avec l'aide d'un des assistants ('*wizards*') fournis dans OMNIS. Ouvrez d'abord la librairie Tutorial :

- Lancez OMNIS et appuyez sur Ctrl/Cmnd-O pour ouvrir la librairie
- Ouvrez le dossier **Examples** dans le dossier principal d'OMNIS Studio
- Ouvrez le dossier **Webclient** et double-cliquez sur la librairie Tutorial.lbs



La librairie ouvre un fichier de données d'exemple nommé Webclient.df1 qui contient les données de la Bibliothèque. Si le fichier de données ne peut être trouvé, la librairie vous demandera le fichier de données Webclient.df1, qui est localisé dans le dossier Examples/Webclient de votre dossier OMNIS Studio.

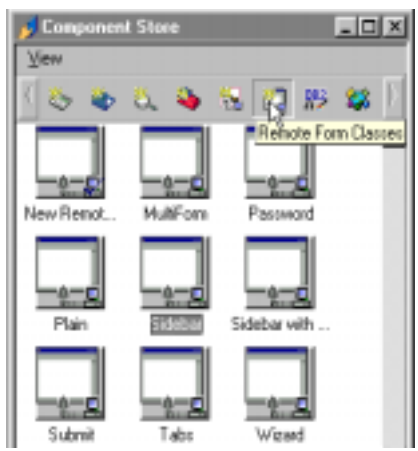
- Double-cliquez sur la librairie dans le *Class Browser* pour afficher ses classes

La librairie contient un certain nombre de classes comprenant la classe *file* **fBooks**, la classe *Remote task* **rtBooks** et la table système **#ICONS** qui contient les icônes nécessaires. La librairie contient aussi une classe *Remote form* nommée **rfBookBrowserFinal** qui est une copie de la *form* créez dans ce tutorial. La classe **Startup_Task** contient le code exécuté à l'ouverture de la librairie, qui dans ce cas ouvre le fichier de données d'exemple.

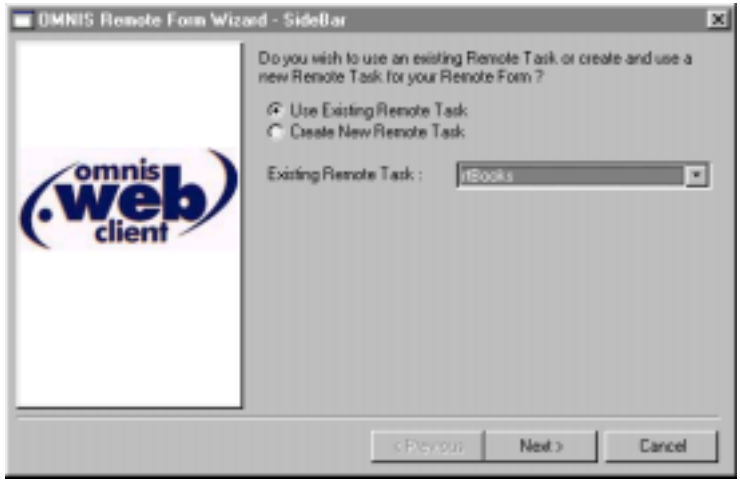
Utiliser le *Wizard Sidebar Remote form*

Vous devez d'abord ouvrir le *Component Store* et retrouver les *Remote form class wizards*.

- Appuyez sur F3/Cmnd-3 pour ouvrir le *Component Store* ou l'amener en avant-plan
- Faites défiler la barre d'outils verticale du *Component Store* et cliquez sur le bouton des *Remote form* Classes pour afficher les *templates* et *wizards Remote form*

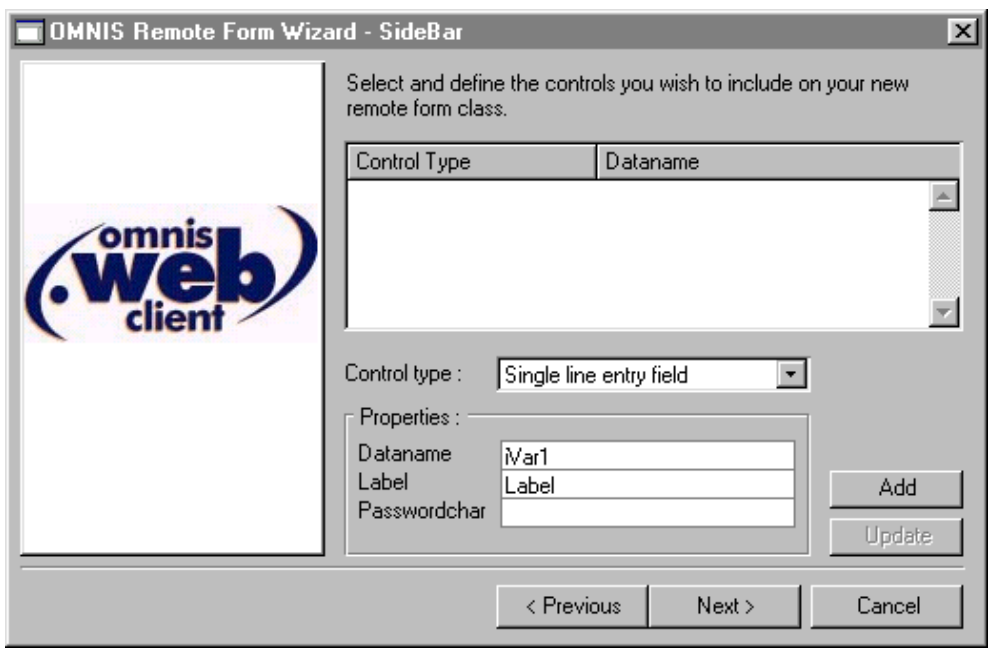


- Glissez le *wizard* ***Sidebar*** vers la librairie Tutorial dans le *Browser* de Classe (n'utilisez pas le *wizard* '*Sidebar* with pages')
- Nommez la classe **rfBookBrowser** et appuyez sur Entrée



Le *wizard* vous demande une *Remote task* ou vous laisse en créer une nouvelle. Pour ce tutorial vous pouvez utiliser la *Remote task* fournie dans la librairie tutorial.

- Dans le *wizard*, sélectionnez **rtBooks** dans la liste des *Remote tasks* existantes et cliquez sur le bouton *Next*



Le *wizard* vous permet maintenant de saisir les détails des rubriques que vous désirez placer sur votre *form*. Vous pouvez spécifier le type de contrôle, le nom de la rubrique de données rattachée et le libellé de chaque rubrique de votre *form*.

Pour la première rubrique

- Sélectionnez **Heading List** comme type de contrôle
- Saisissez **iBookList** comme nom de rubrique rattachée (*dataname*)
- Saisissez **3** dans le nombre de colonnes
- Entrez **Title,Author,Cost** (sans espaces) pour les titres de colonnes (*Column titles*)
- Entrez **200,120,50** pour les largeur de colonnes (*column widths*)
- Lorsque les détails ci-dessus sont saisis correctement, appuyez sur le bouton **Add**

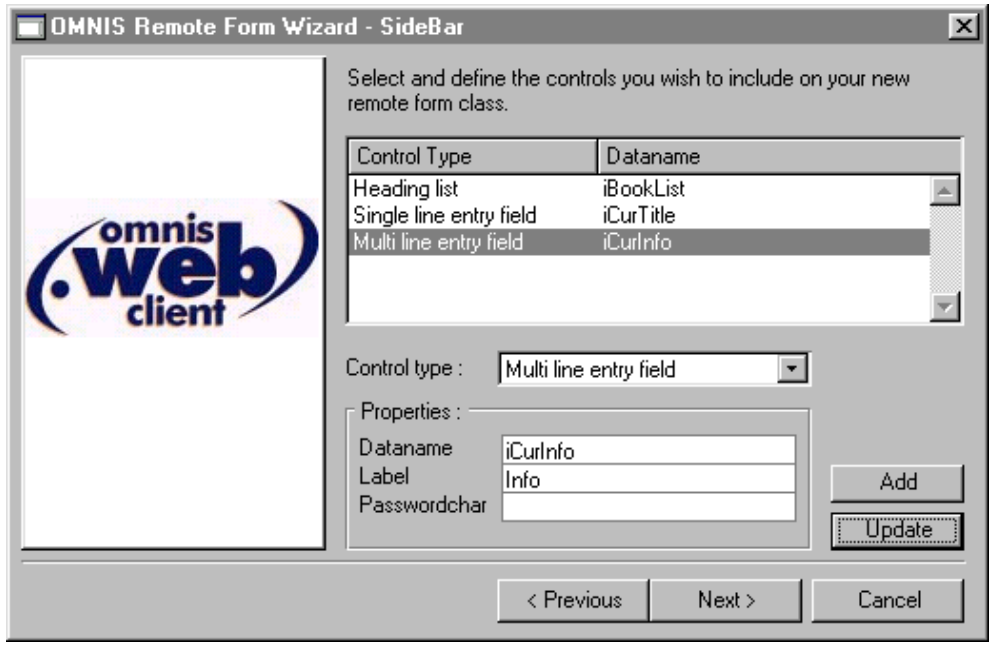
Pour la deuxième rubrique

- Sélectionnez **Single line entry field** comme type de contrôle
- Saisissez **iCurTitle** comme *dataname*
- Entrez **Title** comme label
- Lorsque les détails ci-dessus sont saisis correctement, appuyez sur le bouton **Add**

Pour la troisième rubrique

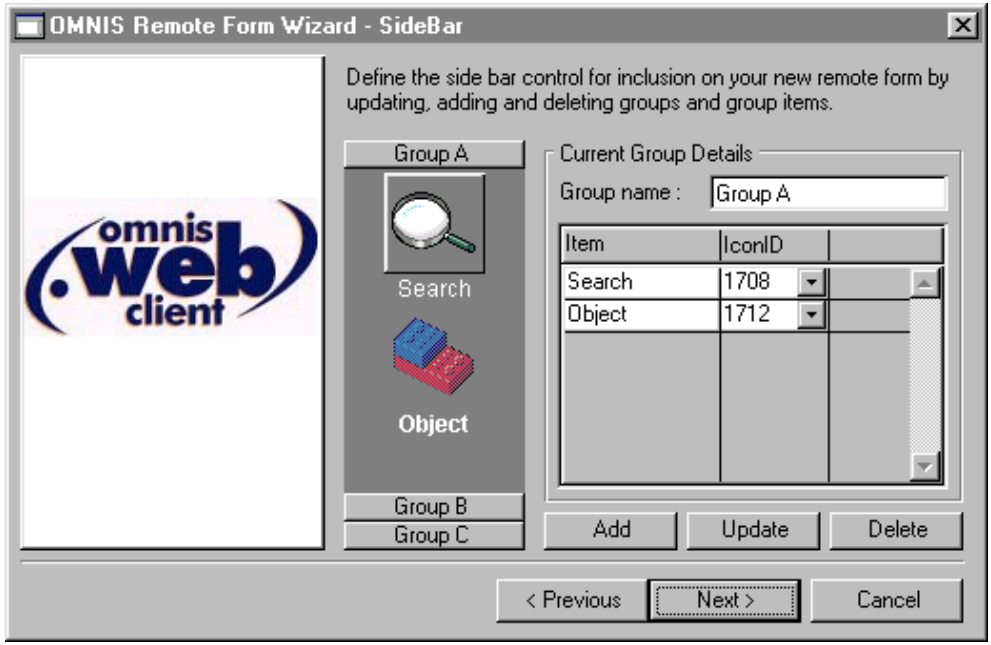
- Sélectionnez **Multi line entry field** comme type de contrôle
- Entrez **iCurInfo** comme dataname
- Saisissez **Info** comme label
- Appuyez sur le bouton **Add** lorsque les détails ci-dessus sont saisis correctement

La fenêtre du *wizard* devrait ressembler à ceci :



Lorsque les détails ci-dessus sont saisis correctement

- Cliquez sur le bouton **Next** de la fenêtre *wizard*



Le *wizard* vous permet de définir le groupe et le group items du *Sidebar* de votre *Remote form*. Pour notre exemple de *Remote form*, vous devez créer un *Sidebar* avec 5 groupes. Le *wizard* possède trois groupes que vous pouvez mettre à jour. Vous devez ajouter deux groupes supplémentaires.

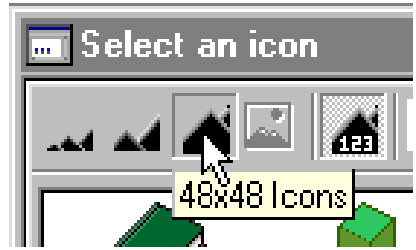
- Dans la rubrique *Group name*, remplacez le texte 'Group A' par **Bestsellers** et tabulez jusqu'au premier *item name* de la liste
- Remplacez le texte par **Bestsellers** et tabulez jusqu'à la rubrique *IconID*

Les icônes du *Sidebar* de notre Book *browser* sont stockées dans la librairie tutorial dans la table #ICONS, table système stockée dans la librairie. Vous devez saisir l'ID de l'icône pour l'icône du groupe Bestsellers. Pour cela :

- Remplacez la valeur en cours dans la rubrique *IconID* par **1**

D'une autre manière, vous pouvez sélectionnez l'icône dans le dialogue *Select an Icon.* Pour cela :

- Dans la fenêtre *wizard*, cliquez sur la flèche de la liste déroulante de la rubrique *IconID* pour ouvrir le dialogue *Select an Icon*



- Cliquez sur le bouton **48x48 Icons** de la barre d'outils du dialogue
- Ensuite, cliquez sur le bouton le plus haut et le plus à droitemarqué #ICONS

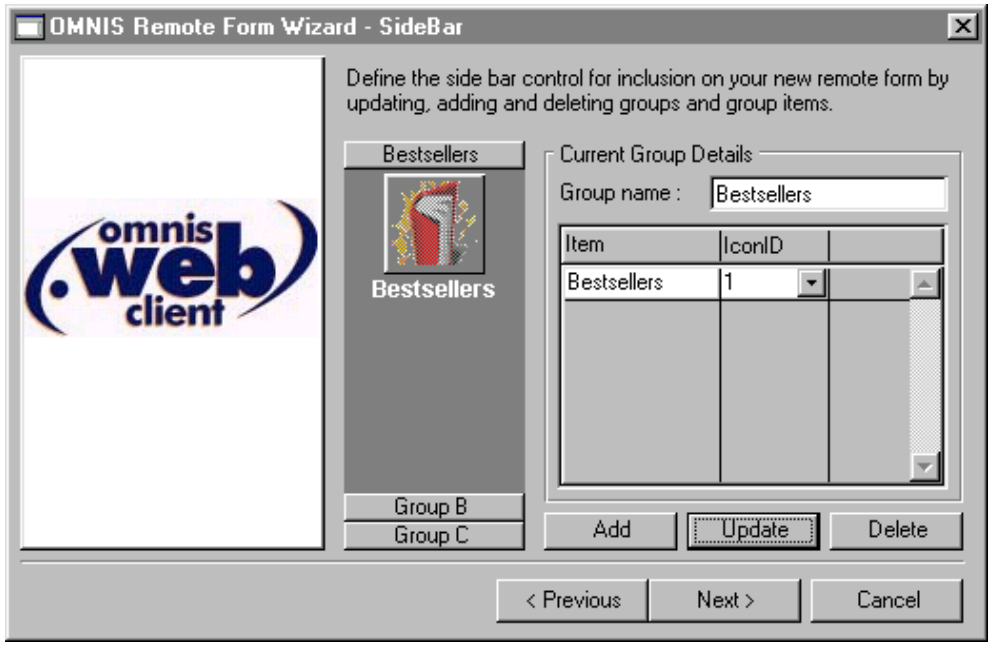
- Dans le dialogue *Select Icon*, cliquez sur l'icône numéro **1** et appuyez sur le bouton **Select**

De retour dans la fenêtre *wizard*, vous n'avez plus besoin du deuxième item du groupe, vous pouvez le détruire en utilisant le menu contextuel de l'item.

- Effectuez un clic-droit (Windows) ou un Ctrl-clic (Mac) sur la deuxième ligne de la liste et sélectionnez l'option **Delete**

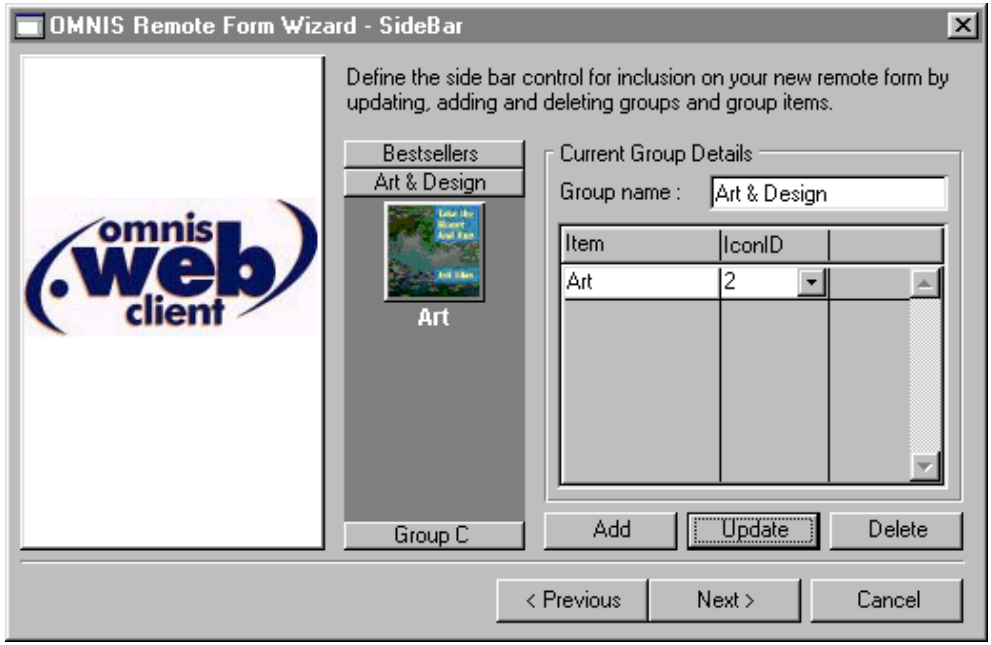
- Cliquez sur le bouton **Update** pour mettre à jour les détails du premier groupe

La fenêtre *wizard* devrait ressembler à ceci :



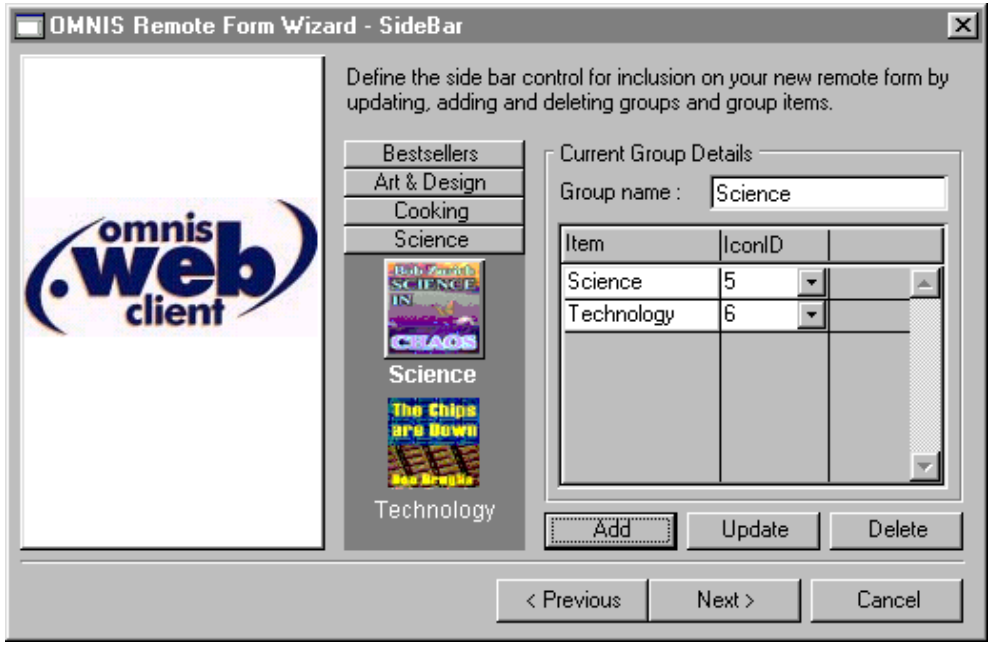
- Dans la prévisualisation du *Sidebar* cliquez sur la barre du groupe B
- Dans la rubrique *Group name*, remplacez le texte 'Group B' par **Art & Design** et tabulez jusqu'au premier item de la liste
- Remplacez le texte par **Art** et tabulez jusqu'à la rubrique *IconID*
- Remplacez la valeur en cours par **2** et tabulez jusqu'à la deuxième ligne de la liste (ou sélectionnez l'icône depuis le dialogue *Select an Icon* comme au-dessus)
- Effectuez un clic-droit (Windows) ou un Ctrl-clic (Mac) sur la deuxième ligne de la liste et sélectionnez l'option *Delete*
- Cliquez sur le bouton **Update** pour mettre à jour les détails du deuxième groupe

Maintenant, la fenêtre *wizard* devrait ressembler à ceci :



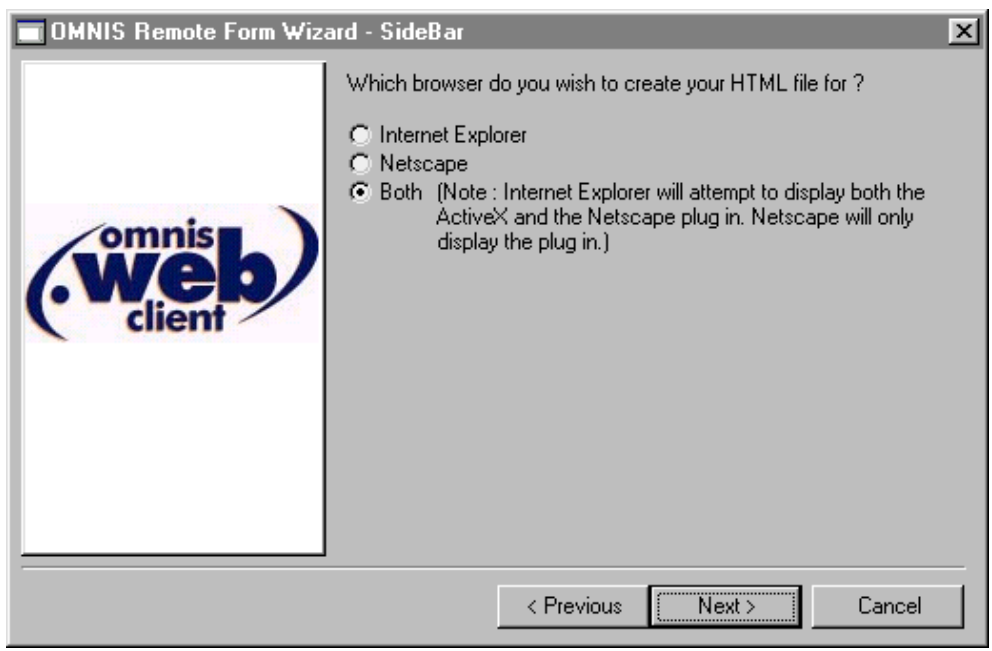
- Dans la prévisualisation du *Sidebar* cliquez sur la barre du groupe C
- Dans la rubrique *Group name* remplacez le texte 'Group C' par **Cooking** et jusqu'au premier item name de la liste
- Remplacez le texte par **Cooking** et tabulez jusqu'à la rubrique *IconID*
- Remplacez la valeur en cours par **3**
- Tabulez jusqu'à la deuxième ligne de la liste pour la modifier
- Remplacez le texte en cours par **Food** et tabulez jusqu'à la rubrique *IconID*
- Remplacez la valeur en cours par **4**
- Cliquez sur le bouton **Update** pour mettre à jour les détails du troisième groupe

- Dans la rubrique *Group name*, remplacez le texte 'Cooking' par **Science** et tabulez jusqu'au premier nom de la liste
- Remplacez le texte par **Science** et tabulez jusqu'à la rubrique *IconID*
- Remplacez la valeur en cours par **5** et tabulez jusqu'à la deuxième ligne de la liste
- Remplacez le texte par **Technology** et tabulez jusqu'à la rubrique *IconID*
- Remplacez la valeur en cours par **6**
- Cette fois, plutôt que de mettre à jour, vous devez cliquer sur le bouton **Add** pour ajouter ces détails au quatrième groupe



- Pour créer le cinquième groupe, dans la rubrique *Group name*, remplacez le texte 'Science' par **Fiction** et tabulez jusqu'au premier nom de la liste
- Remplacez le texte par **Fiction** et tabulez jusqu'à la rubrique *IconID*
- Remplacez la valeur en cours par **7** et tabulez jusqu'à la deuxième ligne de la liste
- Effectuez un clic-droit (Windows) ou un Ctrl-clic (Mac) sur la deuxième ligne de la liste et sélectionnez l'option **Delete**
- De nouveau, plutôt que de mettre à jour, vous devez cliquer sur le bouton **Add** pour ajouter les détails du cinquième groupe

- Quand tous les détails de groupe, *item names* et *IconID*s pour chaque groupe d'item sont corrects, cliquez sur le bouton **Next**

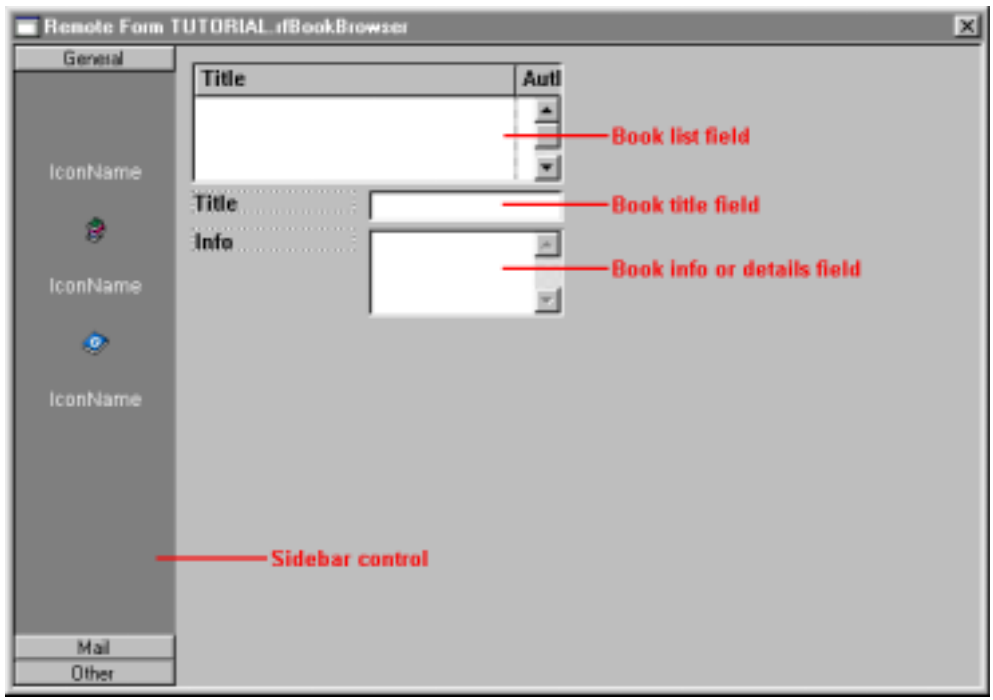


Le *wizard* vous demande alors de choisir le type de navigateur web que vous aller utiliser ; qui dépend aussi du type de client web OMNIS que vous avez installer avant de débuter ce tutorial.

- Sélectionnez **Internet Explorer** ou **Netscape** et cliquez sur le bouton **Next**

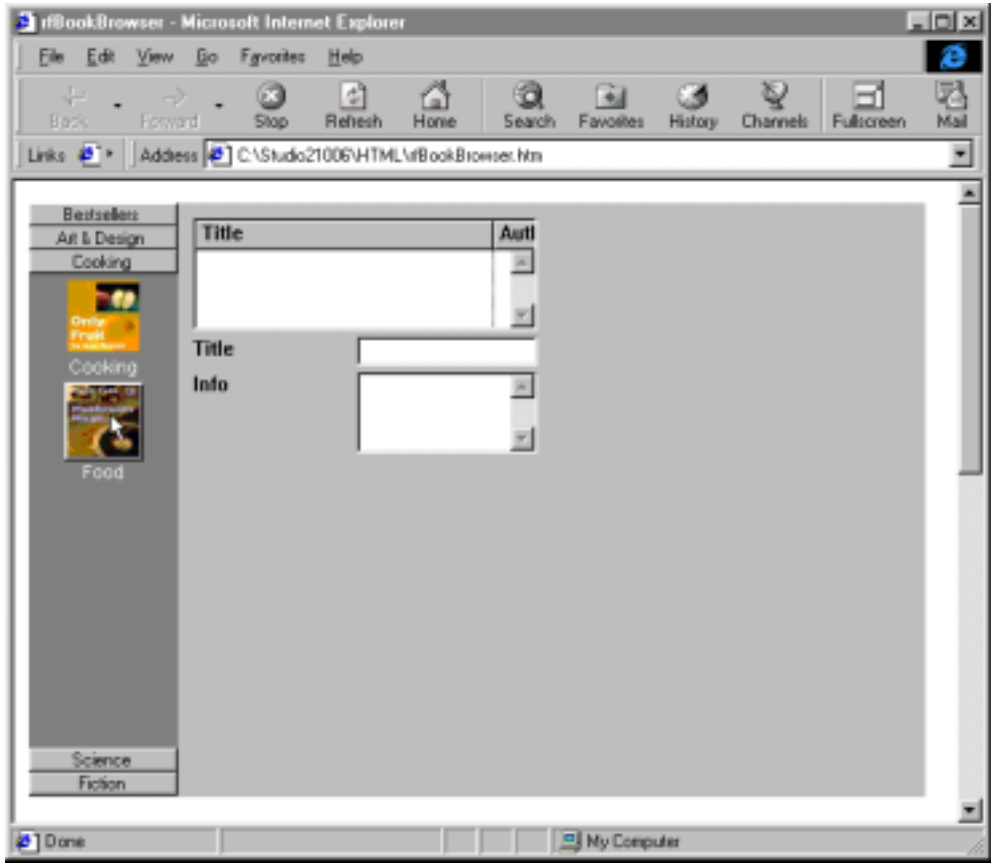
La *Remote form* **rfBookBrowser** a été ajoutée à la librairie tutorial et doit apparaître dans le *Class Browser*. Pour visualiser la *form* en mode création :

- Double-cliquez sur la *Remote form* **rfBookBrowser** dans le *Class Browser*



La *form* créée par le *wizard Sidebar* contient le contrôle *Sidebar* dans la partie gauche et les trois rubriques spécifiées, plus les libellés que vous avez saisis. La liste **BookList** (Contrôle de type *Heading List*) est placé dans la partie supérieure et contiendra la liste des livres, la rubrique **Title** (rubrique de saisie mono-ligne) est placée à côté et contiendra le titre du livre sélectionné et la rubrique **Info** (rubrique de saisie multi-lignes) est placée en bas afin de saisir les annotations du livre sélectionné. Notez que le contrôle *Sidebar* n'affiche pas les groupes que vous avez saisis, ils apparaîtront cependant lorsque vous utiliserez la *form* dans votre navigateur web.

Cette *form* ne contient aucune programmation pour rapatrier et afficher les données de la bibliothèque, mais vous pouvez ouvrir ou “tester” la *form* à tout moment en appuyant sur Ctrl/Cmnd-T. OMNIS ouvre automatiquement votre navigateur web et affiche la *form.*



A ce moment la *form* ne gère pas les données de la bibliothèque virtuelle, mais vous donne une idée de son apparence dans un navigateur web. Vous pouvez cependant cliquer sur le contrôle *Sidebar* et changer de groupe. Notez que le navigateur web utilise un modèle de page html créé par le *wizard form*. Le modèle de page est placé dans le dossier **html** du dossier principal OMNIS. Cette page html possède une imbrication du client web OMNIS par défaut avec tous les paramètres corrects définis automatiquement.

Si votre navigateur web qui abrite la *Remote form* ne s'ouvre pas à ce point, vérifiez l'installation correcte du client web OMNIS ; si vous avez installé ActiveX, vérifiez qu'il est enregistré correctement. Consultez la section Guide de dépannage et Foire Aux Questions pour les détails d'enregistrement manuel d'OCX.

- Fermez ou minimisez le navigateur web pour retourner à OMNIS ; la classe *Remote form* est toujours en avant-plan en mode création dans OMNIS

Modifier les rubriques de *Form*

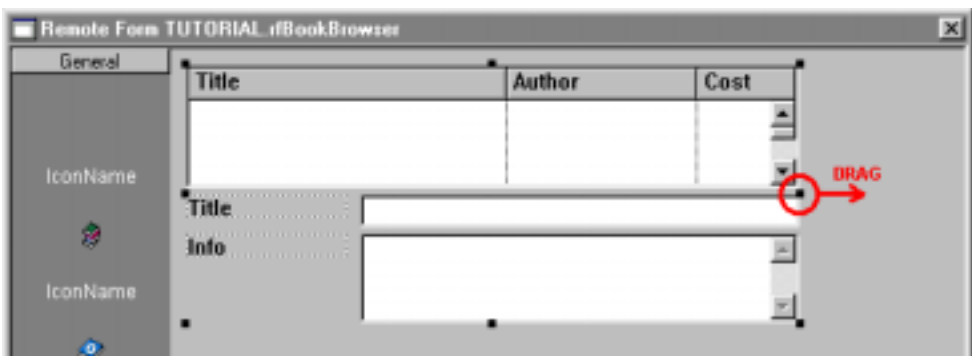
Les étapes suivantes impliquent le redimensionnement et le changement de nom des trois rubriques de la *form* ainsi que l'ajout de code derrière la *Remote form* avec en particulier les contrôles de BookList et de la *Sidebar*. Plus tard, nous ajouterons une rubrique image pour afficher la couverture du livre.

Pour redimensionner la liste Booklist et les rubriques book title et info, vous pouvez les manipuler une par une ou les trois en même temps en les sélectionnant au préalable.

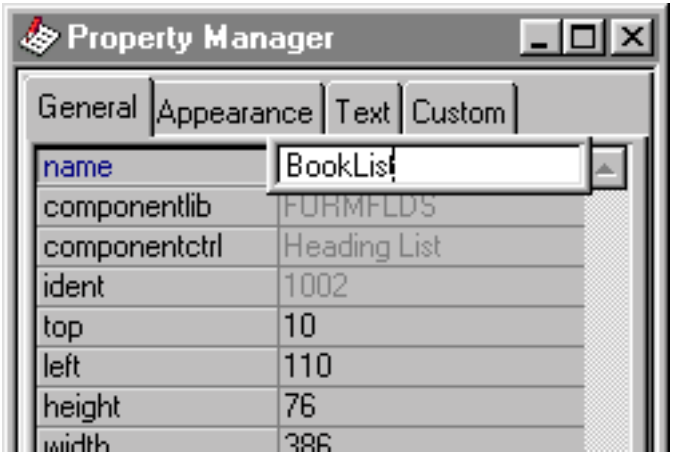
- Dans la classe *Remote form*, cliquez sur chaque rubrique et redimensionnez-la ; la colonne **Cost** doit être visible dans Booklist et vous devrez redimensionner aussi Booktitle et info pour obtenir un résultat adéquat

Sinon, pour les redimensionner en même temps :

- Effectuez un Maj-clic sur les trois rubriques pour les sélectionner ensemble
- Glissez la poignée de droite de la sélection pour agrandir les rubriques, comme montré ci-dessous



- Cliquez sur le fond de la *form* pour désélectionner les rubriques
- C'est le moment d'agrandir la rubrique Info en déplaçant sa poignée inférieure vers le bas
- Ensuite cliquez sur BookList et appuyez sur F6/Cmnd-6 pour ouvrir le *Property Manager* ou l'amener en avant-plan
- Sur l'onglet *General* du *Property Manager* modifiez la propriété **name** en **BookList**

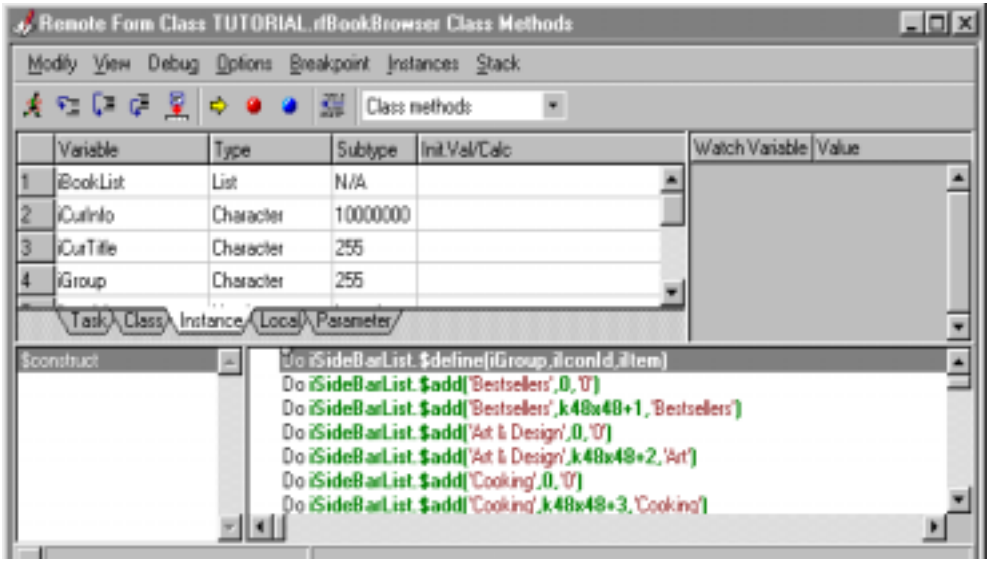


- De façon similaire, cliquez sur la rubrique Title (la rubrique texte d'une seule ligne), ouvrez le *Property Manager* et changez son nom en **Title**
- Cliquez ensuite sur la rubrique Info (la rubrique texte multi-ligne), ouvrez le *Property Manager* et modifiez son nom en **Info**

Ajouter des variables et des méthodes à la *Form*

Une fois les rubriques modifiées, vous devez ajouter quelques lignes de code derrière la *form* et ses contrôles. Vous devez ajouter quelques variables à cette *form*, modifier certaines lignes de code créées par le *wizard* et ajouter une ou deux nouvelles méthodes.

- Double-cliquez sur le fond de la *form*, en mode création, pour ouvrir le *Method editor* pour la classe *Remote form*



- Cliquez sur l'onglet **Instance** dans le volet variable
- Effectuez un clic-droit/Ctrl-click sur le volet Variable et sélectionnez **Insert New Variable** depuis le menu contextuel
- Nommez la variable **iBookGroup** et sélectionnez le Type **Number** et le sous-type **Long Integer** depuis les listes déroulantes appropriées
- Pour insérer une nouvelle variable, effectuez un clic-droit/Ctrl-clic sur le volet Variable et sélectionnez **Insert New Variable** depuis le menu contextuel
- Nommez la variable **iCurCover** et sélectionner **Picture** dans la liste déroulante de Type (ce type ne requiert pas de *subtype*)

Vous devez avoir 9 variables dans votre *Remote form*. Maintenant, vous allez modifier le code de la méthode $construct, qui doit être sélectionné dans le *Method editor.* Cette méthode construit le groupe d'items et d'icônes pour le contrôle *Sidebar* de la *form.* Modifiez la première ligne de code :

```
Do i<i>Sidebar</i>List.$define(iGroup,iIconId,iItem)
```

en ajoutant la variable **iBookGroup** dans la liste. La ligne ressemble alors à :

```
Do i<i>Sidebar</i>List.$define(iGroup,iIconId,iItem,iBookGroup)
```

Ensuite vous devez ajouter les groupes ou catégories de livre pour chaque ligne de code qui définie chaque groupe d'item du contrôle *Sidebar*. Le tableau suivant vous donne les catégories (les mêmes que celles qui sont stockées dans le fichier de données).

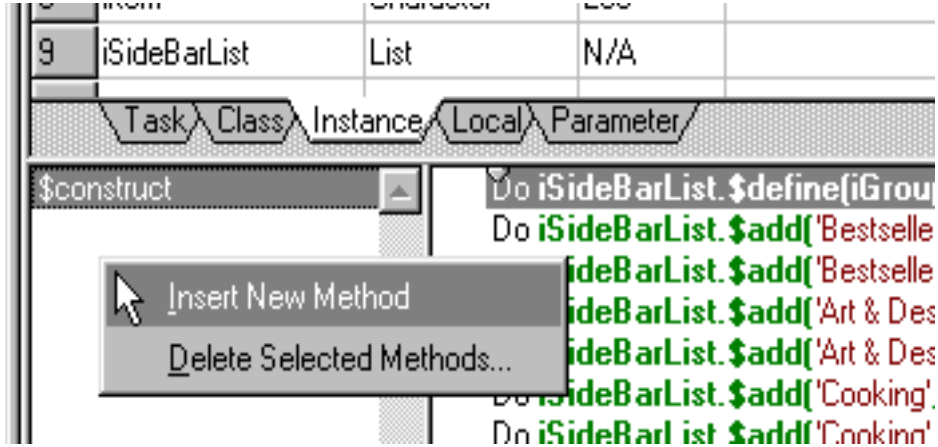
Bestsellers	0
Art	1
Cooking	2
Food	3
Science	4
Technology	5
Fiction	6

Les lignes que vous devez modifier sont emphasées (les ajouts sont soulignés et/ou en rouge), et montrent la méthode finale. Si vous préférez, vous pouvez Copier la méthode depuis le manuel électronique PDF et la Coller dans le *Method editor* pour remplacer le code existant, ou copier la méthode complète depuis la classe rfBookBrowserFinal de la librairie tutorial.

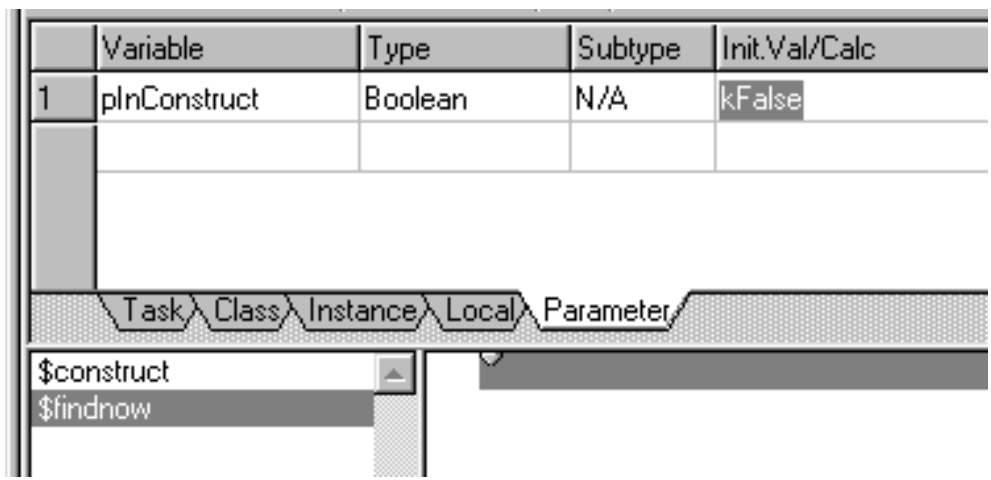
```
Do iSidebarList.$define(iGroup,iIconId,iItem,iBookGroup)
Do iSidebarList.$add('Bestsellers',0,'0')
Do iSidebarList.$add('Bestsellers',k48x48+1,'Bestsellers',0)
Do iSidebarList.$add('Art & Design',0,'0')
Do iSidebarList.$add('Art & Design',k48x48+2,'Art',1)
Do iSidebarList.$add('Cooking',0,'0')
Do iSidebarList.$add('Cooking',k48x48+3,'Cooking',2)
Do iSidebarList.$add('Cooking',k48x48+4,'Food',3)
Do iSidebarList.$add('Science',0,'0')
Do iSidebarList.$add('Science',k48x48+5,'Science',4)
Do iSidebarList.$add('Science',k48x48+6,'Technology',5)
Do iSidebarList.$add('Fiction',0,'0')
Do iSidebarList.$add('Fiction',k48x48+7,'Fiction',6)
```

Vous pouvez sauver la *form* à tout moment en sélectionnant l'option **File>>Save rfBookBrowser** dans la barre de menu principale d'OMNIS. Maintenant, vous allez ajouter une méthode à la *Remote form*.

Dans le *Method editor*, effectuez un clic-droit/Ctrl-clic sur la liste des méthode juste en dessous de la méthode $construct et sélectionnez **Insert New Method** dans le menu contextuel



- Nommez la nouvelle méthode **$findnow**
- Cliquez sur l'onglet **Parameter** du volet Variable et ajoutez une nouvelle variable par clic-droit/Ctrl-clic
- Nommez la nouvelle variable **pInConstruct**, sélectionnez le type **Boolean** et donnez lui la valeur initiale **kFalse**



- Cliquez dans le *Method editor* de nouveau et saisissez le code suivant pour la méthode **$findnow**

Pour vous économiser la saisie de cette méthode vous pouvez en Copier le code dans le manuel PDF et le Coller dans la méthode $findnow ou Copier et Coller la méthode complète depuis la classe rfBookBrowserFinal de la librairie tutorial. Certaines lignes peuvent êtres insérées en commentaires, essayez de le dé-commenter individuellement.

```
Set main file { fBooks}
Set current list iBookList
Load from list
Find on fBooks.BookTitle (Exact match)
Calculate iCurCover as fBooks.BookCover
Calculate iCurInfo as fBooks.BookInfo
Calculate iCurTitle as fBooks.BookTitle

Do $cinst.$objs.Info.$redraw()
Do $cinst.$objs.Cover.$redraw()
Do $cinst.$objs.Title.$redraw()

If not(pInConstruct)
   Do $cinst.$senddata(iCurCover,iCurInfo,iCurTitle)
End If
```

Il vous faut maintenant ajouter du code à la méthode $construct() pour appeler la méthode $findnow().

- Cliquez sur la méthode **$construct** et descendez tout en bas des lignes de code de la méthode
- Ajoutez les lignes de code suivantes ; si vous préférez, essayer de les Copier et de les Coller comme expliqué précédemment

```
; Build book list
Calculate iBookList as tBookList
Set current list iBookList
Calculate #L as 1
Do method $findnow (kTrue)
```

La méthode $construct complète, avec toutes les modifications et ajouts, est listée ci-après ; si vous avez des difficultés avec cette méthode, vous pouvez Copier et Coller cette méthode complète depuis la classe rfBookBrowserFinal de la librairie tutorial.

```
Do i Sidebar List.$define(iGroup,iIconId,iItem,iBookGroup)
Do i Sidebar List.$add('Bestsellers',0,'0')
Do i Sidebar List.$add('Bestsellers',k48x48+1,'Bestsellers',0)
Do i Sidebar List.$add('Art & Design',0,'0')
Do i Sidebar List.$add('Art & Design',k48x48+2,'Art',1)
Do i Sidebar List.$add('Cooking',0,'0')
Do i Sidebar List.$add('Cooking',k48x48+3,'Cooking',2)
Do i Sidebar List.$add('Cooking',k48x48+4,'Food',3)
Do i Sidebar List.$add('Science',0,'0')
Do i Sidebar List.$add('Science',k48x48+5,'Science',4)
Do i Sidebar List.$add('Science',k48x48+6,'Technology',5)
Do i Sidebar List.$add('Fiction',0,'0')
Do i Sidebar List.$add('Fiction',k48x48+7,'Fiction',6)

; Build book list
Calculate iBookList as tBookList
Set current list iBookList
Calculate #L as 1
Do method $findnow (kTrue)
```

- Lorsque la méthode $construct est complète, fermez le *Method editor*

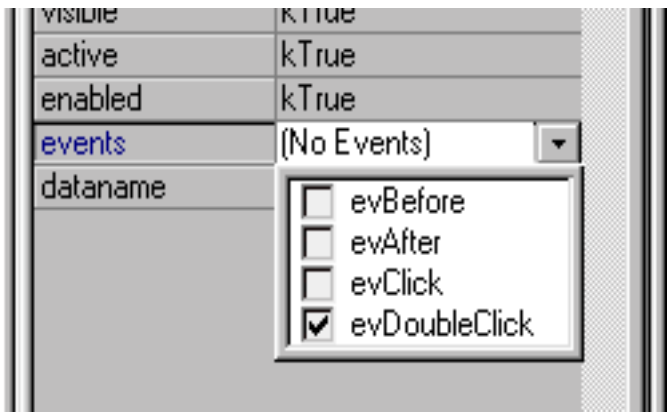
A ce moment, vous pouvez enregistrer les modifications que vous avez apporté à la *form* en sélectionnant l'option **File>>Save rfBookBrowser** de la barre de menus principale d'OMNIS. Il nous faut encore ajouter du code derrière les contrôles BookList et *Sidebar*, mais vous pouvez essayer de nouveau votre *form* dans votre navigateur web. Lorsque vous l'ouvrez pour la première fois, avec Ctrl/Cmnd-T, BookList doit contenir les livres de la catégorie Bestsellers, mais à ce point, les clics sur les contrôles *Sidebar* et BookList n'ont aucun effet.

- Pour retourner à OMNIS, fermez ou minimisez votre navigateur web

Evènements et ajouts de méthodes aux rubriques de *form*

Nous allons ajouter quelques lignes de code derrière BookList. Vous devez en premier lieu activer l'événement *evDoubleClick* pour la liste. Normalement, tous les évènements de rubriques, sont activés pour les objets de fenêtre, mais pour les objets de *Remote form* vous devez explicitement activer les évènements que vous désirez intercepter pour l'objet considéré. L'activation d'évènements se réalise dans les propriétés *events* de l'objet.

- Dans la *Remote form*, cliquez sur la rubrique BookList et appuyez sur F6/Cmnd-6 pour ouvrir le *Property Manager* ou l'amener en avant-plan
- Sous l'onglet *General*, cliquez sur la propriété *events* et sélectionnez/cochez l'événement **evDoubleClick** dans la liste déroulante



Il vous faut maintenant ajouter une méthode à **BookList**.

- Dans la *Remote form*, double-cliquez sur la rubrique **BookList** pour ouvrir le *Method editor* pour l'objet
- Saisissez le code suivant dans la méthode $event

```
On evDoubleClick
   Do method $findnow
```

- Une fois la méthode saisie, fermez la fenêtre du *Method editor*

Ensuite, vous allez saisir la méthode $event() pour le contrôle *Sidebar*. Cette fois cependant, il n'est pas utile d'activer les évènements du contrôle *Sidebar*, en particulier evIconPicked, car le *wizard Sidebar* s'en est chargé pour vous automatiquement. Pour saisir le code du gestionnaire d'évènements $event :

- Double-cliquez sur le contrôle ***Sidebar*** de la *form*

- Dans le *Method editor*, saisissez le code suivant dans la méthode $event

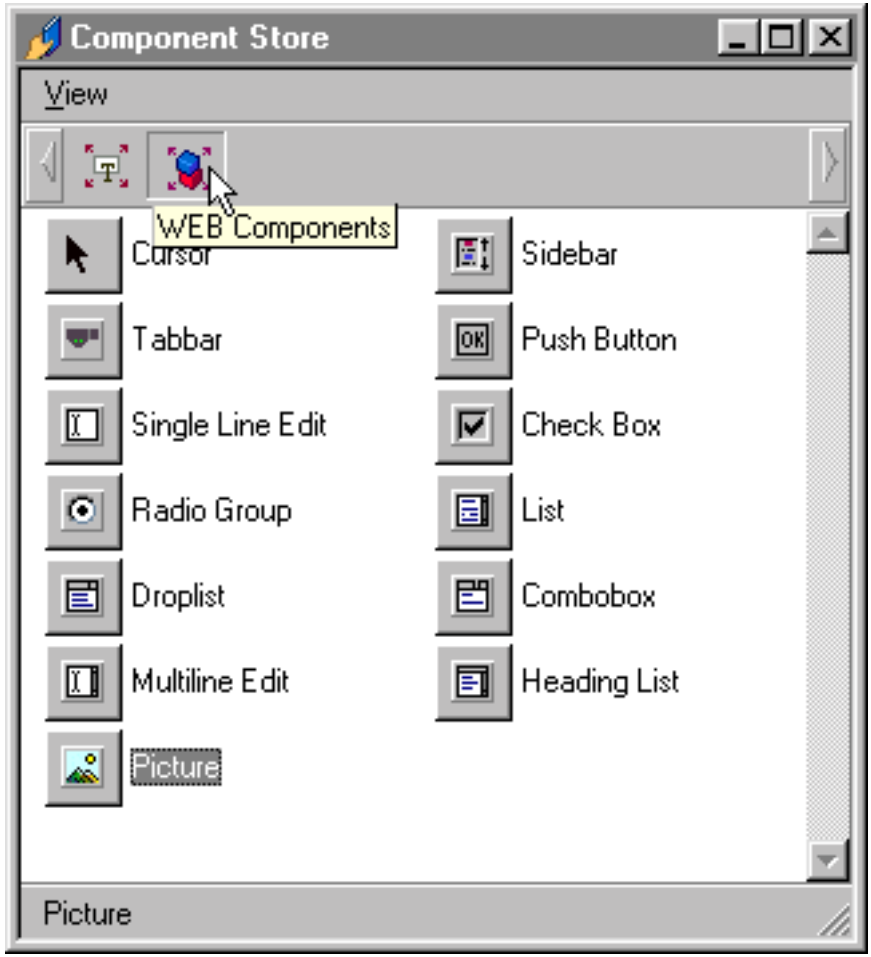
```
On evIconPicked
   Set current list i Sidebar List
   Load from list {pLinenum}
   Do method $ctask.$buildbooks (iBookGroup)
   Calculate iBookList as tBookList
   Do $cinst.$objs.BookList.$redraw()
   Do $cinst.$senddata(iBookList)
```

- Une fois la méthode saisie, fermez le *Method editor* et enregistrez votre classe avec l'option de menu **File>>Save rfBookBrowser**

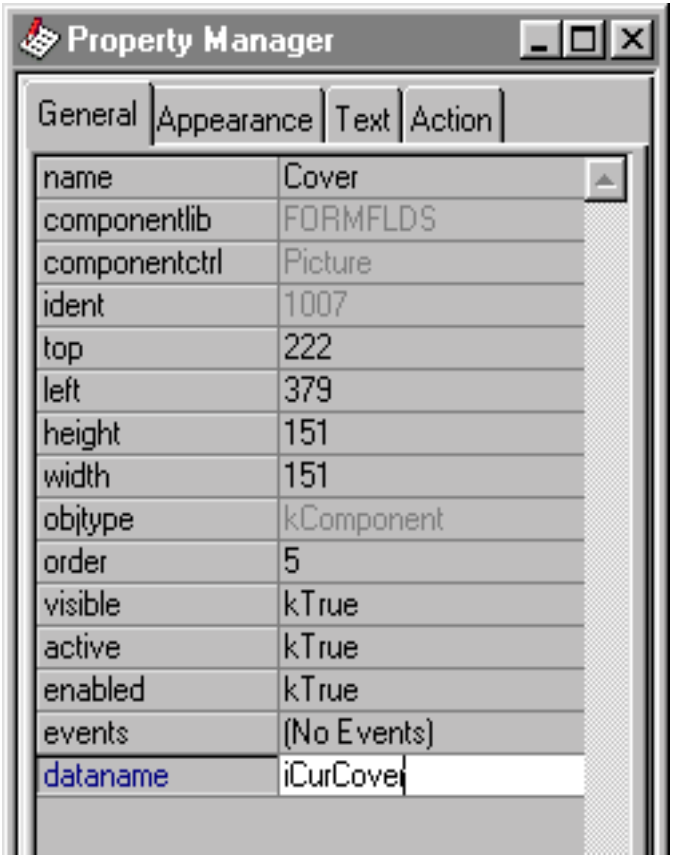
Ajouter une rubrique Picture

Pour finir, vous allez ajouter une rubrique *picture* à la *form* pour visualiser les couvertures de livres. Vous pouvez créer une rubrique *picture* avec le *wizard Sidebar*, mais cette section vous permet d'ajouter la rubrique à votre *remote* en utilisant le *Component Store*.

- En considérant que la *form* soit en avant-plan, appuyez sur F3/Cmnd-3 pour ouvrir le *Component Store* et cliquez sur le bouton **WEB Components** de la barre d'outils du *Component Store*



- Glissez l'objet **Picture** sur votre *form* et relâchez le bouton de la souris
- Avec la rubrique Picture sélectionnée, appuyez sur F6/Cmnd-6 pour ouvrir le *Property Manager* ou l'amener en avant-plan
- Sous l'onglet *General*, modifiez la propriété **name** en **Cover**
- Saisissez **iCurCover** dans la propriété **dataname**



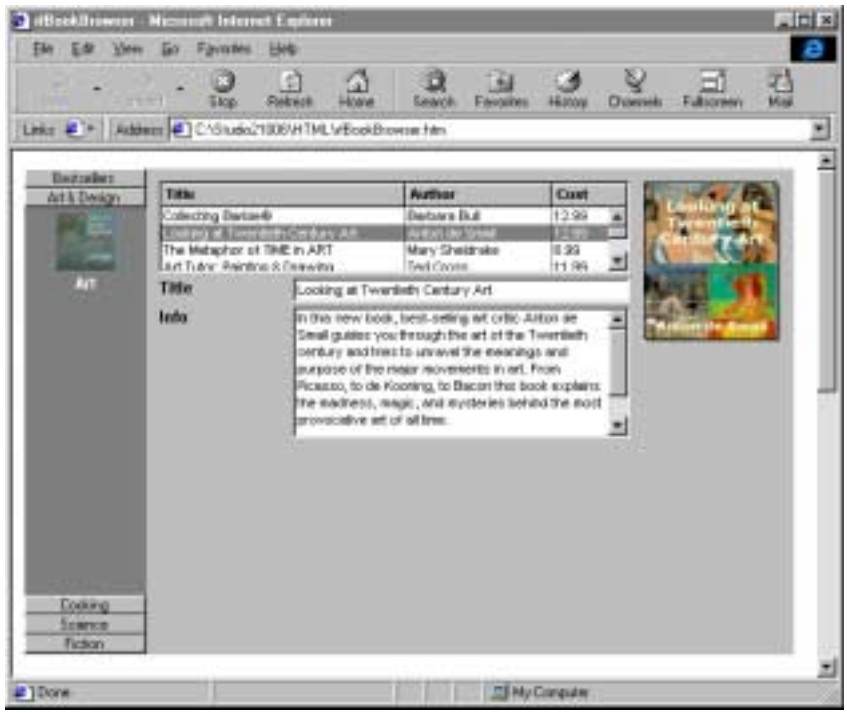
Si vous désirez retirer les ascenseurs de la rubrique *picture*, redimensionner et positionner celle-ci dans la *form* :

- Cliquez sur la rubrique *Picture* et appuyez sur F6/Cmnd-6 pour ouvrir le *Property Manager* ou l'amener en avant-plan
- Sous l'onglet *Appearance*, définissez les propriétés **vertscroll** et **horzscroll** à **kFalse**
- Sous l'onglet *Appearance*, définissez la propriété **effect** à **kShadowBorder**
- Sous l'onglet *General*, définissez la propriété **height** à **132** et la propriété **width** à **112**
- Elargissez un peu la *form* et déplacez la rubrique *picture* dans le coin supérieur droit de la *form*, à droite de la rubrique BookList

Tester la *Form* et parcourir la Bibliothèque

Pour tester la *Remote form* :

- Appuyez sur Ctrl/Cmnd-T pour essayer votre *form* et parcourir les données de la Bibliothèque



Cette *form* vous permet de chercher les informations sur un ouvrage. Vous pouvez cliquer sur le contrôle *Sidebar* pour changer de groupe ou de sous-groupe et visualiser les différentes catégories d'ouvrages proposés. Vous pouvez aussi ***double-cliquer*** sur *BookList* pour visualiser les détails et la couverture de chacun des ouvrages.

Debugger une *Form*

Lorsque vous développez une *Remote form*, vous pouvez définir des points d'arrêt (*Breakpoints*) dans votre code dans le *Method editor* et utiliser le mode trace. Si vous définissez un *breakpoint* dans l'une de vos méthodes, ouvrez la *Remote form* dans votre navigateur web (avec Ctrl/Cmnd-T) et utilisez la *form*, OMNIS basculera en mode debugger du mode Design. Si le mode Trace est activé, OMNIS suivra toutes les méthodes appelées par la *form*, vous permettant ainsi de 'voir ce qui se passe'.

Distribuer des *Remote form*

Pour distribuer une solution web OMNIS, vous avez besoin d'un serveur web Win32 sur un serveur Win32 pour installer la version runtime de l'exécutable OMNIS. Cependant, pour les besoins de ce tutorial, vous pouvez utiliser un serveur web personnel (comme celui proposé par Microsoft FrontPage) et la version de développement d'OMNIS sur votre machine locale. Le processus de distribution est exactement le même dans les deux cas et impliquent :

– l'installation et la configuration du serveur OMNIS, c'est-à-dire, le runtime OMNIS et votre librairie OMNIS

– l'installation et la configuration de votre serveur web, la création de pages html

Si vous désirez mettre en place la librairie du tutorial sur votre serveur web, plutôt que de l'utiliser localement, vous devez modifier la méthode $construct() de la *Startup_Task*. Vous devez en particulier modifier la commande **Open data file** et son code associé qui ouvre le fichier de données Webclient et vous devrez retirer la commande **Prompt for data file**. Vous ne devez utiliser aucune commande dans l'application serveur OMNIS qui fasse intervenir un dialogue.

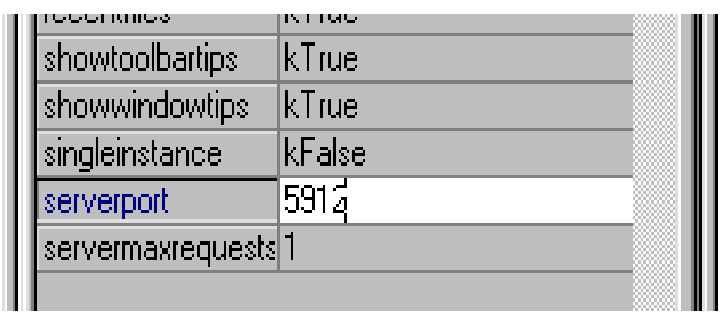
Serveur OMNIS

Le tutorial utilise la version de développement d'OMNIS mais si vous distribuez votre librairie sur le web vous devez installer et sérialiser spécifiquement un runtime OMNIS.

Pour la version de développement OMNIS :

- Lancez OMNIS, ouvrez la librairie Tutorial et connectez vous au fichier de données Books

- Sélectionnez l'option de menu **Tools>>Options** depuis la barre de menus principale d'OMNIS



- Dans les préférences OMNIS dans le *Property Manager*, définissez la propriété **serverport** à 5912 ou, si cette valeur est déjà en usage, choisissez n'importe quelle autre valeur

- Relancez OMNIS, ré-ouvrez la librairie et conservez-la en fonctionnement en arrière-plan

Pour le runtime OMNIS :

- Installez le runtime OMNIS sur votre serveur Win32 et sérialiser le avec le numéro de série
- Copiez la librairie tutorial et le fichier de données (Webclient.df1) sur votre serveur Win32
- Lancez le runtime OMNIS
- Sélectionnez l'option de menu **File>>Server Configuration** depuis la barre de menus principale d'OMNIS
- Saisissez la valeur 5912 Dans la rubrique *port number* et fermez le dialogue
- Ouvrez la librairie Tutorial et connectez vous au fichier de données ; conservez la librairie en fonctionnement en arrière-plan

Serveur Web et fichiers HTML

Le tutorial utilise un serveur web personnel défini pour MS FrontPage, mais si vous distribuez votre application sur le web vous devez vous procurer et installer un serveur Web Win32. Il est possible de trouver des serveurs gratuits (freeware) ou de démonstration sur Internet. Dans tous les cas la configuration du serveur web est la même.

- Installez votre serveur web ou configurez votre serveur web personnel de MS FrontPage ; Consultez l'Aide de MS FrontPage pour les détails de configuration
- Localisez l'arborescence OMNIS et le dossier OMNIS/WebClient/Server
- Copiez le fichier OMNISAPI.DLL dans le dossier **cgi-bin** de votre serveur web
- Localisez ensuite le dossier **Html** dans votre dossier principal OMNIS
- Copiez le fichier **rfBookBrowser.htm** sur votre serveur personnel ou votre serveur web
- Ouvrez le fichier **rfBookBrowser.htm** dans un éditeur de texte ou Html
- Dans le source Html du fichier rfBookBrowser.htm, modifiez les paramètres suivants, en particulier les paramètres WebServerURL et WebServerScript

Pour Internet Explorer vous devez modifier les paramètres suivants pour ActiveX :

```
<param name="OmnisServer" value="5912"> ;; le numéro de port
<param name="OmnisLibrary" value="TUTORIAL">
   ; le nom de librairie sans extension
<param name="OmnisClass" value="rfBookBrowser">
   ; Le nom de la Remote form
<param name="WebServerURL" value="http://www.yourdomain.com">
   ; le nom de domaine ou l'adresse IP de votre serveur web
   ; ou serveur web personnel
<param name="WebServerScript" value="/cgi-bin/omnisapi.dll">
   ; l'emplacement de l'extension web serveur OMNIS
```

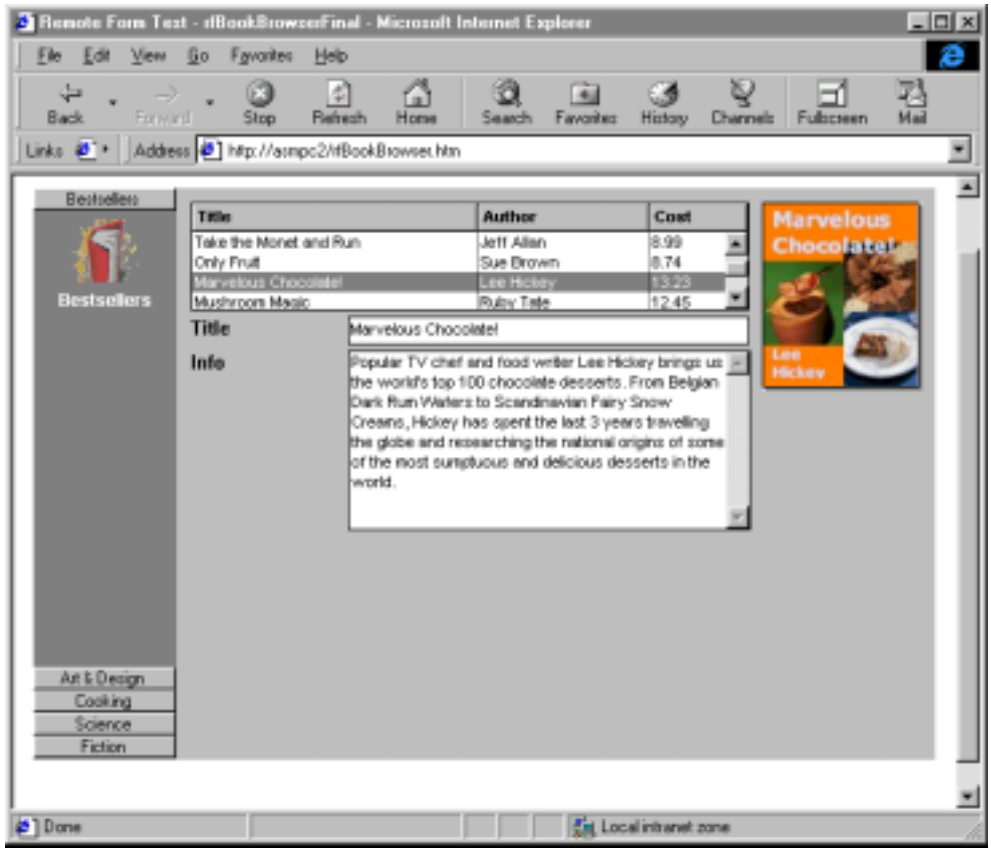
Les paramètres pour le plug-in Netscape doivent être modifiés comme suit :

```
<EMBED type=application/OMNIS-RCC-plugin name="rcc1"
   width=500 height=500
   OmnisServer="5912" ;; le numéro de port
   OmnisLibrary="TUTORIAL"
   ; le nom de librairie sans extension
   OmnisClass="rfBookBrowser"
   ; le nom de la Remote form
   WebServerURL="http://www.yourdomain.com" ; le nom de domaine ou
   l'adresse IP de votre serveur web
   WebServerScript="/cgi-bin/omnisapi.dll" ; l'emplacement de
   l'extension web serveur OMNIS
   Param1="pOption1=Data" .. .. Param9="pOption9=Data">
```

Les paramètres *port number*, *library name* et *Remote form* sont déjà saisis automatiquement. Si ce n'est pas le cas, saisissez les valeurs inscrites ci-avant. Si votre serveur web ne supporte pas l'interface MS ISAPI, vous devrez utiliser le programme cgi nph-omniscgi.exe. Dans ce cas, vous devrez ajouter le programme nph-omniscgi.exe dans votre dossier cgi-bin et utiliser ce nom dans le paramètre **WebServerScript** de votre (vos) fichier(s) html.

- Enregistrez et fermez le fichier rfBookBrowser.htm
- Si vous distribuez votre application sur un serveur distant, copiez le fichier rfBookBrowser.htm sur votre serveur web via FTP
- Ouvrez votre navigateur web et localisez le fichier **rfBookBrowser.htm** avec “[VotreNomDeDomaine ou VotreAdresseIP]/rfBookBrowser.htm”, où VotreNomDeDomaine peut être http://www.votredomain.com et où AdresseIP peut être l'adresse de votre serveur web personnel

L'interface de consultation de la bibliothèque, c'est à dire la *form* doit apparaitre et vous permettre de consulter les données de la bibliothèque.



Voir la librairie d'exemple du Client Web OMNIS

Vous trouverez une version améliorée de l'exemple "*Books form*" dans la librairie d'exemple Client web OMNIS disponible aussi à traver l'*Examples Browser*. Les améliorations concernent l'implémentation d'un "panier de course" pour stocker les livres que l'utilisateur désire acheter, et une fonctionnalité d'encaissement on-line par saisie de numéro de carte de crédit. Vous pouvez ouvrir cet exemple via l'*Examples Browser* en utilisant l'option de menu **Tools>>Examples Browser** de la barre de menus principale d'OMNIS. Vous pouvez aussi utiliser cette application sur Internet via le site web d'OMNIS Software : www.omnis-software.com.

Chapitre 3 – Créer des *Remote forms* OMNIS

Pour créer une solution web OMNIS vous devez créer une librairie qui contiendra des classes *Remote form* OMNIS et des classes *Remote task.* Vous pouvez créer et tester les *Remote forms* sur votre propre machine avec votre propre navigateur web. Vous pouvez ainsi développer votre librairie OMNIS avant de l'installer sur votre serveur web, d'installer l'extension serveur web et le serveur web OMNIS, ou avant même de créer vos pages html.

Les informations générales sur le développement d'applications OMNIS, la créations de *forms* et la programmation OMNIS sont trouvées dans les manuels d'*Utilisation OMNIS Studio* et *Programmation OMNIS* disponibles avec OMNIS Studio.

Vous pouvez créer vos *Remote forms* OMNIS à partir de rien ou employer les *templates* et *wizards* disponibles dans le *Component Store*. Les *wizards* vous offrent une maîtrise complète sur les rubriques et les contrôles de votre *form* et vous économise une grande part de temps. Tous les *wizards* créent une nouvelle *Remote task* ou utilisent une *task* existante.

Créer des *Remote forms* avec les *Wizards*

Vous pouvez créer une *Remote form* à l'aide de l'un des *wizards* disponibles dans le *Component Store*. Les *wizards* vous permettent de spécifier les rubriques et les contrôles que vous désirez placer sur votre *form*. Les *wizards* sont résumés ici et décrits plus en détails plus loin.

- **Plain**
  Crée une *form* vierge et la lie à une *Remote task* ; vous devez créer vos propres variables d'instance et vos propres contrôles de *form* pour celle-ci
- **Tabs**
  Crée une *Remote form* qui contient une *tab strip,* un contrôle *page pane* et les rubriques sur chaque volet (*pane*) ; crée une variable d'instance pour chaque rubrique de la *form*
- **Wizard**
  Crée une *Remote form* qui contient une contrôle *page pane,* les rubriques sur chaque volet et les boutons *Next*, *Previous* et *Finish* ; crée une variable d'instance pour chaque rubrique de la *form*

- **Submit**
  Crée une *form* standard qui contient vos rubriques et les boutons *Submit* et *Clear* ; crée une variable d'instance pour chaque rubrique de la *form*

- **Password**
  Crée une *form* standard de demande de mot de passe avec les rubriques *Password* et *Username* ; crée une variable d'instance pour chaque rubrique de la *form*

- **Sidebar**
  crée une *form* qui contient un nombre de rubriques et un contrôle *Sidebar* ; crée une variable d'instance pour chaque rubrique de la *form*

- **Sidebar with pages**
  Crée une *form* qui contient un contrôle *page pane* et un contrôle *Sidebar* ; vous spécifiez les rubriques de chaque *pane* et le lien de chaque *pane* à un groupe particulier du contrôle *Sidebar*; crée une variable d'instance pour chaque rubrique de la *form*

- **Multiform**
  vous permet de créer un site web complet avec page d'index, barre de navigation et de nombreuses autres *Remote forms* créées avec les précédants *wizards*

Tous les *wizards* demandent une classe *Remote task* : une instance de *Remote task* gère les connexions clientes, et regroupe toutes les instances associées à une *Remote form* particulière. Vous pouvez sélectionner une *Remote task* existante de votre librairie ou en créer une nouvelle basée sur les *templates* fournis. Pour la majeure partie des types de *Remote form* vous pouvez sélectionner le *template "Plain Remote task"*.

Tous les *wizards*, excepté *Plain*, génère un fichier html qui devra être placé dans le dossier **html** de la racine OMNIS. Cette page html vous permet de tester la *form* en mode création avec Ctrl/Cmnd-T et fournit un *template* pour vos pages html finales de votre site web. Durant chaque utilisation de *wizard* il vous sera demandé de sélectionner le type de navigateur web pour lequel vous voulez créer les pages html. Vous pouvez sélectionner Internet Explorer ou Netscape ou les deux, dans tous les cas, le plug-in approprié sera imbriqué dans vos pages html *template*.

Tous les *wizards* qui créent des *forms* qui contiennent des rubriques ou des contrôles, vous demanderont le **dataname** et le type de contrôle de chaque rubrique que vous désirez insérer sur votre *form.* Une variable d'instance est créée pour chaque pour l'ajouter à la *form*.

Toutes les *Remote forms* créées à l'aide d'un *wizard* possèdent la méthode $construct() avec une variable paramètre nommée **pParams** de type *Row Variable*. Cette *row variable* contient les paramètres de la *Remote form*.

Une fois la/les *Remote form*(s) créée(s) par un des *Remote form Wizards*, vous pouvez les modifier ou ajouter vos variables d'instance et vos contrôles de *form*, selon vos besoins, pour compléter vos *Remote forms*.

Form Wizard Plain

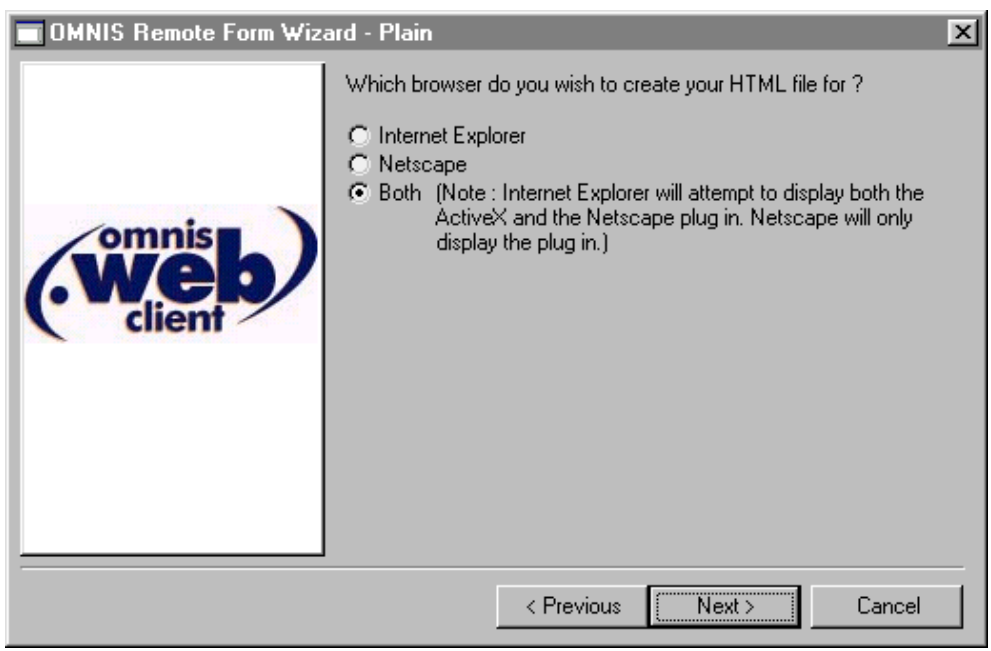
Le *form wizard Plain* crée une *form* vierge et la lie à une *Remote task*. Le *wizard* vous demande une *Remote task* existante ou vous permet d'en créer une nouvelle basée sur les *templates* fournis. Vous devez ajouter vos propres variables d'instance et vos propres contrôles *form* à ce type de *Remote form*.

Pour créer une *Plain Form*

- Ouvrez votre librairie et affichez ses classes dans le *Class Browser*
- Ouvrez le *Component Store* et cliquez sur le bouton *Remote form*
- Sélectionnez le *Remote form wizard Plain* et glissez son icône vers votre librairie dans le *Class Browser* OMNIS
- Nommez la nouvelle *Remote form* et appuyez sur Entrée
- Sélectionnez une *Remote task* existante dans la liste des *tasks* fournie et cliquez sur le bouton *Next*

ou

- Cliquez sur le bouton *Create New Task*, sélectionnez un *template* depuis la liste fournie, nommez la nouvelle *Remote task* et cliquez sur le bouton *Next*



Erreur! Nom du fichier non spécifié.

- Lorsque l'option de navigateur est sélectionnée, cliquez sur le bouton *Next*

OMNIS crée une nouvelle *Remote form* dans votre librairie. Double-cliquez sur la *form* dans le *Class Browser* pour la modifier.

Form Wizard Tabs

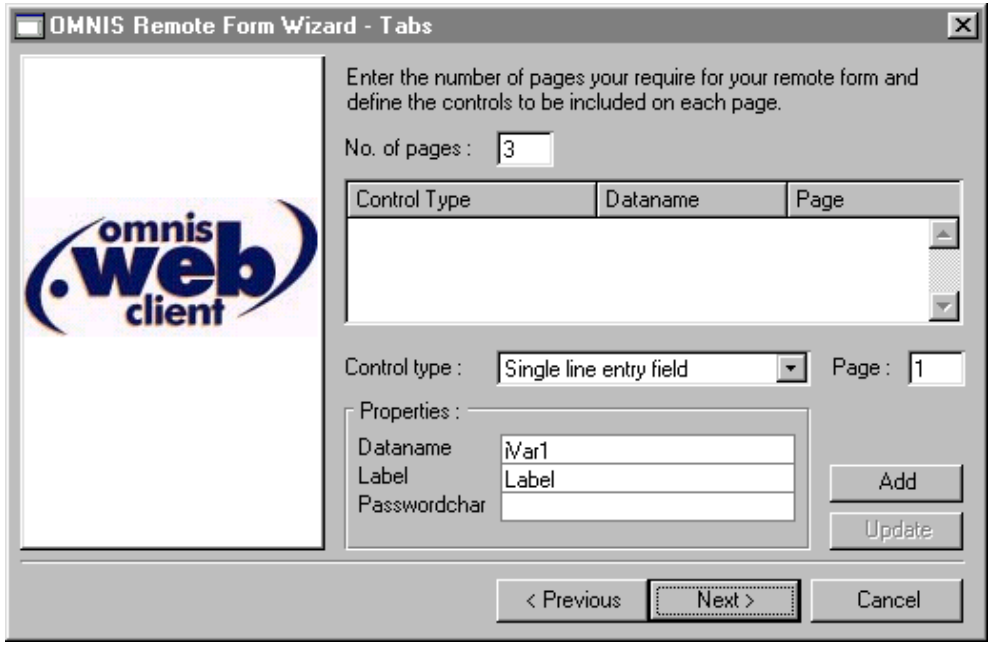
Le *form wizard Tabs* crée une *Remote form* qui contient un bandeau d'onglets et un contrôle *page pane*. Dans le *wizard*, vous pouvez ajouter des rubriques et des contrôles pour chaque *pane* de la *form*. La *form* résultante permet à l'utilisateur de saisir des données sur plusieurs volets (*panes*) en cliquant sur les différents onglets (*Tabs*).

Pour créer une *Form Tabs*

- Ouvrez votre librairie et affichez ses classes dans le *Class Browser*
- Ouvrez le *Component Store* et cliquez sur le bouton des classes *Remote form*
- Sélectionnez le *form wizard Tabs Remote* et glissez son icône sur votre librairie ouverte dans le *Class Browser* OMNIS
- Nommez la nouvelle *Remote form* et validez
- Sélectionnez une *Remote task* existante depuis la liste des *tasks* fournie et cliquez sur le bouton *Next*

ou

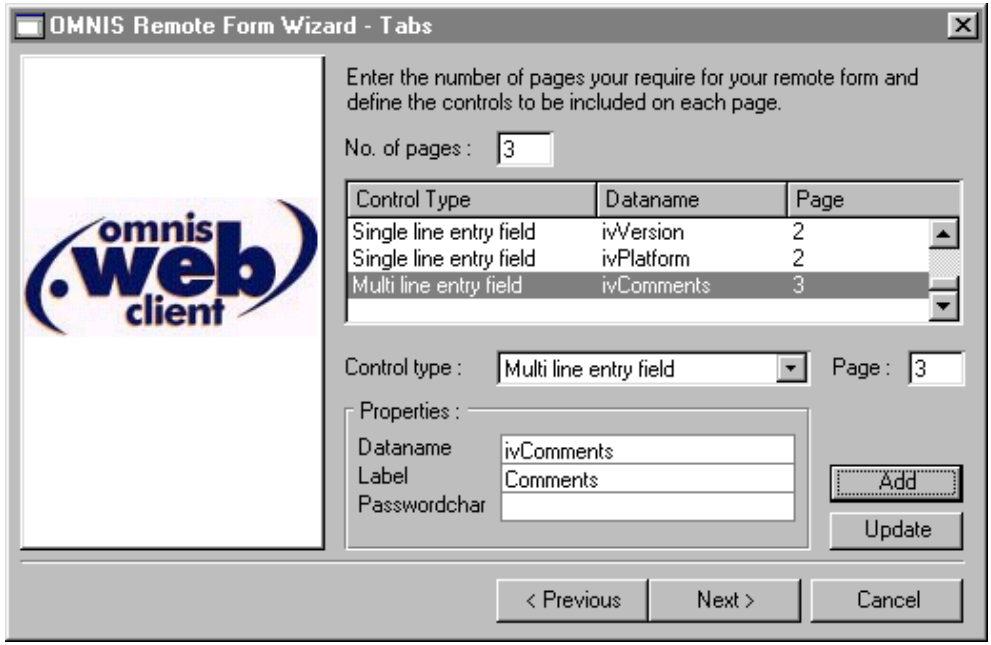
- Cliquez sur l'option *Create New Task*, sélectionnez un *template* dans la liste fournie comme *Plain Remote task*, nommez la nouvelle *Remote task* et cliquez sur le bouton *Next*



- Saisissez le nombre de volets de votre *form*
- Sélectionnez le type de contrôle de la première rubrique de la première page
- Saisissez la propriété *dataname* et le libellé texte de la première rubrique et cliquez sur le bouton *Add*
- Sélectionnez le type de contrôle, saisissez la propriété *dataname* et le libellé texte de la deuxième rubrique de la première page et cliquez sur le bouton *Add*
- Ajoutez les autres rubriques de la première page puis sélectionnez la seconde page et ajoutez les rubriques comme ci-dessus

La liste de la fenêtre *wizard* affiche les rubriques que vous avez ajouté à chaque page de la *form*. Vous pouvez cliquer sur une des rubrique de la liste, modifier ses détails et cliquer sur le bouton *Update*.

Les copies d'écran suivantes montre un *wizard* qui crée 3 pages avec quelques rubriques sur les deux premières pages et une seule rubrique texte multi-ligne sur la troisième page.



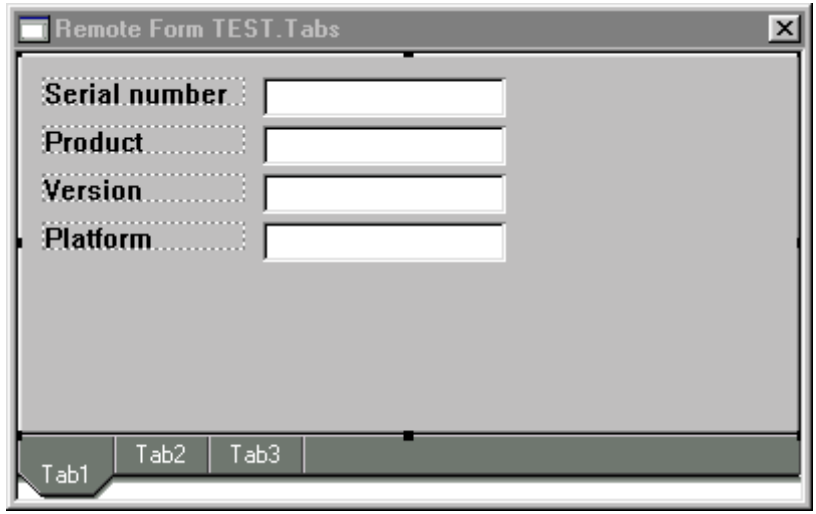
- Lorsque les rubriques de chaque page de votre *form* sont saisies, cliquez sur le bouton *Next*

- Sélectionnez le type de navigateur web et cliquez sur le bouton *Next*

Le *wizard* crée alors la *Remote form* selon les détails que vous lui avez fourni. Notez qu'OMNIS crée aussi une *Remote task*.

- Pour modifier la *form*, double-cliquez dessus dans le *Class Browser*

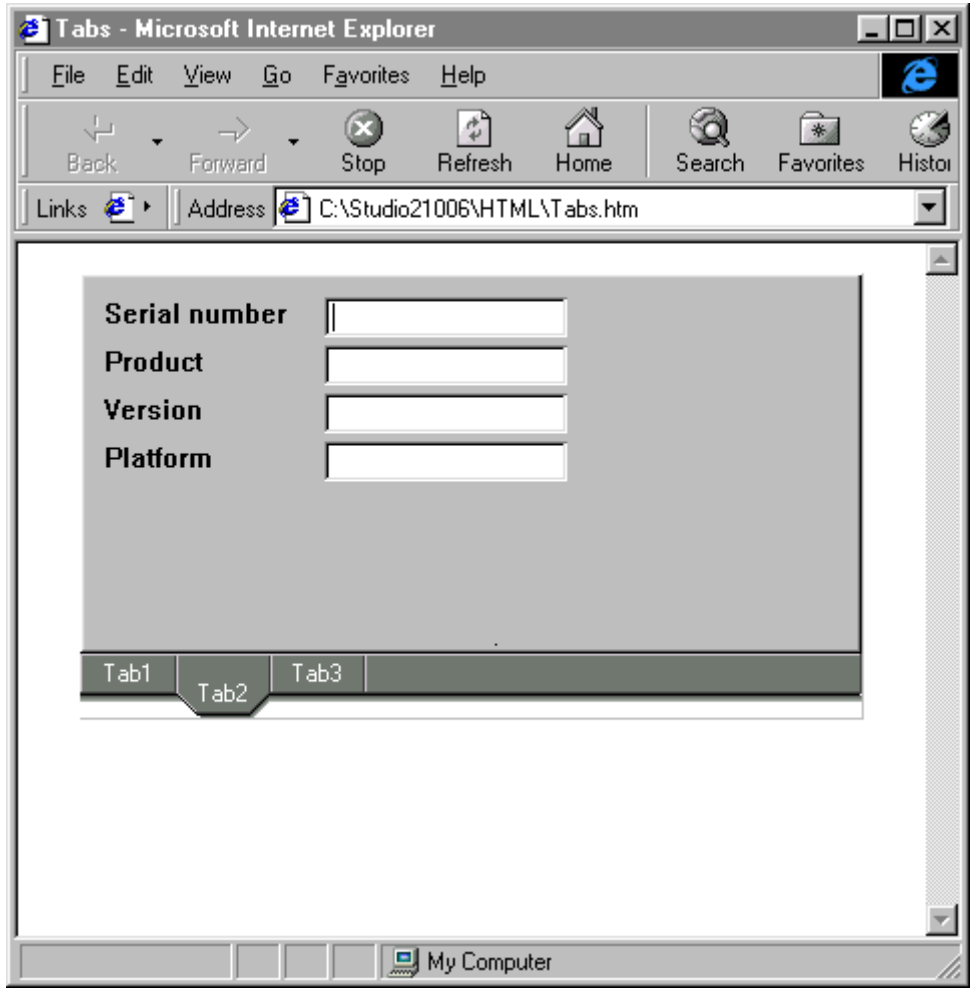
Pour examiner le contenu de chaque page en mode création, vous devez cliquer sur la rubrique *page pane* (cliquez sur le fond grisé, en dehors des rubriques et libellés), ouvrez le *Property Manager* ou amenez-le en avant-plan F6/Cmnd-6 et modifiez la propriété **currentpage**. Par exemple, pour afficher la page deux, cliquez sur la rubrique *page pane*, saisissez 2 dans la propriété **currentpage** du *Property Manager* et appuyez sur Entrée (notez que le *tab strip* en haut de la fenêtre ne change pas en mode création). Vous devriez obtenir quelque chose qui ressemble à la copie d'écran ci-dessous.



Pour tester la *form*

- Appuyez sur Ctrl/Cmnd-T ou effectuez un clic-droit/Ctrl-click sur la *form* et sélectionnez l'option de menu ***Test Form*** du menu contextuel

Lorsque vous appuyez sur Ctrl/Cmnd-T, OMNIS ouvre le navigateur automatiquement et affiche la *form*.



La *form* est ouverte dans votre navigateur web et vous pouvez cliquer sur les onglets pour changer de page en cours dans la *form*. A cette étape, votre *form* ne gère pas les données que vous saisissez dans votre *form*, mais vous présente une idée de son apparence pour l'utilisateur final.

Notez que le navigateur utilise le fichier html *template* créé pour vous par le *form wizard*. La page *template* est placée dans le dossier **html** du dossier principal d'OMNIS. Cette page html possède une imbrication du client web OMNIS par défaut, les paramètres nécessaires à son bon fonctionnement sont générés automatiquement.

Si votre navigateur web ne s'ouvre pas, vérifiez l'installation de votre client web OMNIS.

- Fermez ou minimisez le navigateur lorsque vous désirez revenir à OMNIS

Wizard Form Wizard

Le *Wizard form wizard* crée une *Remote form* qui contient un contrôle *page pane* et un ensemble de boutons *Next*, *Previous* et *Finish*. La *form* résultante ressemble à un *wizard* qui autorise l'utilisateur à saisir ses données sur plusieurs pages pane avec le bouton *Next* et vous offre un plus grand contrôle sur l'ordre d'apparition des demandes d'informations à destination de l'utilisateur. Dans le *wizard* vous pouvez ajouter des rubriques et des contrôles à chaque *form*. Une fois la *form* créée vous devez programmer le bouton *Finish* pour traiter le contenu de la *form*.

Pour créer une *Form Wizard*

- Ouvrez le *Component Store* et cliquez sur le bouton des classes *Remote form*
- Sélectionnez *Wizard Remote form wizard* et déplacez son icône sur votre librairie ouverte dans le *Class Browser* OMNIS
- Nommez la nouvelle *Remote form* et appuyez sur Entrée
- Sélectionnez une *Remote task* existante depuis la liste des *tasks* fournies et cliquez sur le bouton *Next*

ou

- Cliquez sur l'option *Create New Task*, sélectionnez un *template* dans la liste fournie comme *Plain Remote task*, nommez la nouvelle *Remote task* et cliquez sur le bouton *Next*
- Saisissez le nombre de *panes* de votre *form*
- Saisissez les détails pour chaque rubrique de la première page, c'est-à-dire les types de contrôle, leurs dataname et libellés ; cliquez sur le bouton *Add* pour chacune d'elles
- Ajoutez les rubriques restantes aux autres pages de votre *form*

La liste de la fenêtre *wizard* montre les rubriques que vous avez ajouter à chaque page de votre *form*. A tout moment, vous pouvez cliquer sur une rubrique de la liste, modifier ses détails et cliquer sur le bouton Update pour mettre les informations modifiées à jour. Vous pouvez détruire une rubrique dans la liste par un clic-droit/Ctrl-clic sur la rubrique concernée et en sélectionnant l'option *Delete Field* dans le menu contextuel.

- Lorsque les rubriques de chaque page de votre *form* de style *wizard* sont saisies, cliquez sur le bouton *Next*
- Sélectionnez le type de navigateur web et cliquez sur le bouton *Next*

Le *wizard* crée la *Remote form* selon les détails que vous avez saisi. Notez qu'OMNIS crée aussi une *Remote task*.

- Pour modifier la *form*, double-cliquez sur elle dans le *Class Browser*

Pour tester la *form*

- Apuyez sur Ctrl/Cmnd-T ou effectuez un clic-droit/Ctrl-click sur la *form* et sélectionnez l'option *Test Form* dans le menu contextuel

Form Wizard Submit

Le *form wizard Submit* crée une *submit form* qui contiendra vos rubriques ainsi que les boutons *Submit* et *Clear*. Le *wizard* vous permet d'ajouter des rubriques et des contrôles à votre *form*. Dans la *form résultante*, vous devrez programmer le bouton *Submit* pour qu'il traite les données de la *form*.

Pour créer une *Submit Form*

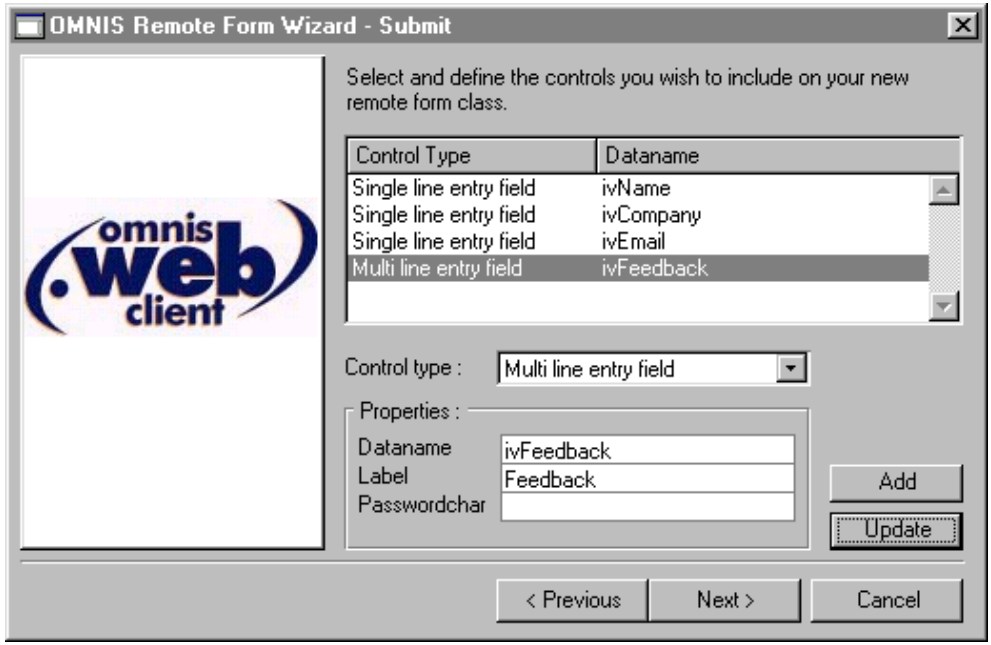
- Ouvrez le *Component Store* et cliquez sur le bouton Classes *Remote form*
- Sélectionnez le *Remote form wizard Submit* et glissez son icône dans votre librairie ouverte dans le *Class Browser* OMNIS
- Nommez la nouvelle *Remote form* et appuyez sur Entrée
- Sélectionnez une *Remote task* existante depuis la liste des *tasks* fournie et cliquez sur le bouton *Next*

ou

- Cliquez sur l'option *Create New Task*, sélectionnez un *template* dans la liste fournie comme *Plain Remote task*, nommez la nouvelle *Remote task* et cliquez sur le bouton *Next*
- Saisissez les détails pour chaque rubrique de la *form*
- Utilisez le bouton *Add* pour ajouter chaque rubrique à votre *form*

La liste de la fenêtre *wizard* montre les rubriques que vous avez ajouter à chaque page de votre *form*. A tout moment, vous pouvez cliquer sur une rubrique de la liste, modifier ses détails et cliquer sur le bouton Update pour mettre les informations modifiées à jour. Vous pouvez détruire une rubrique dans la liste par un clic-droit/Ctrl-clic sur la rubrique concernée et en sélectionnant l'option *Delete Field* dans le menu contextuel.

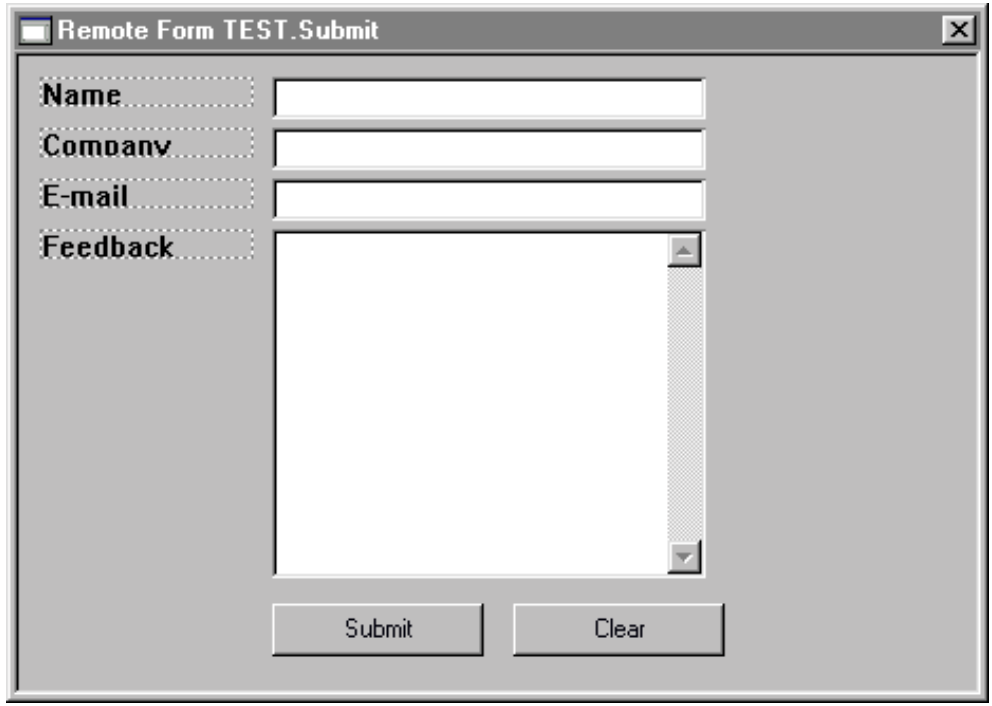
La copie d'écran suivante montre un *wizard Submit* qui défini un certain nombre de rubriques pour une *form* typique de "retour d'information", qui comprend une rubrique de saisie pour le nom (Name), l'entreprise (Company name), l'adresse E-mail(E-mail address), et une rubrique multi-ligne pour les commentaires.



- Lorsque les détails de votre *form* sont saisis, cliquez sur le bouton *Next*
- Sélectionnez le type de navigateur web et cliquez sur le bouton *Next*

Le *wizard* crée alors la *Remote form* avec les détails que vous avez saisi. Notez qu'OMNIS crée aussi une *Remote task*.

- Pour modifier la *form*, double-cliquez dessus dans le *Class Browser*



Pour tester la *form*

- Appuyez sur Ctrl/Cmnd-T, ou effectuez un clic-droit/Ctrl-clic sur la *form* et sélectionnez l'option ***Test Form*** dans le menu contextuel

Wizard Password Form

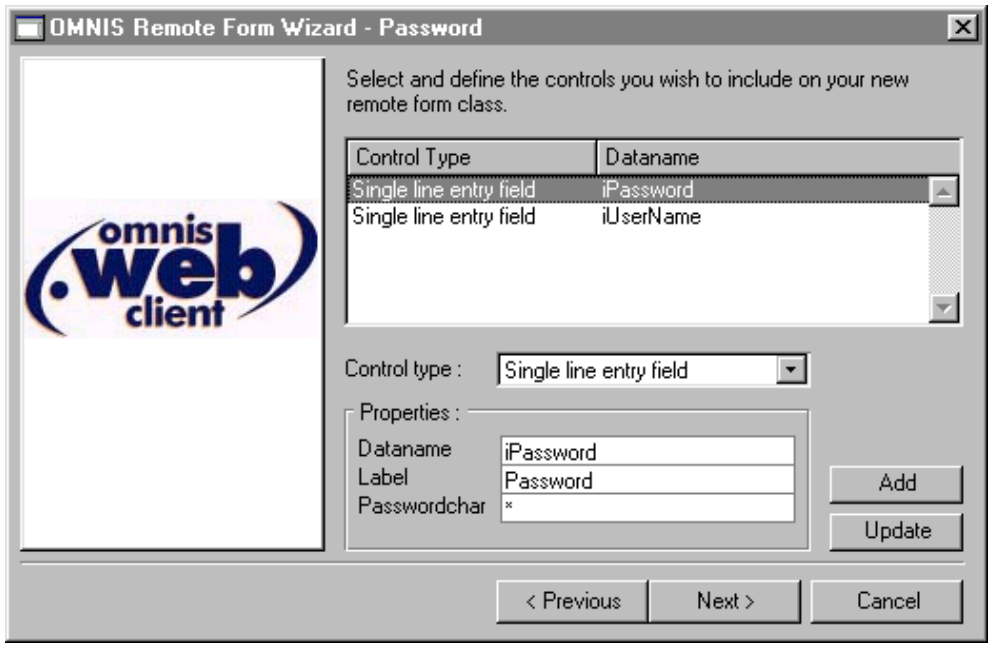
Le *wizard Password form* crée une version standard d'une *form* password qui propose les rubriques standards *Password* et *Username*. Le *wizard* spécifie ces deux rubriques par défaut mais vous pouvez les modifier ou ajouter vos propres rubriques. Dans la *form* résultante, vous devrez programmer le *Submit* pour envoyer le contenu des rubriques *username* et *password* à votre application serveur.

Pour créer une Password *Form*

- Ouvrez le *Component Store* et cliquez sur le bouton des classes *Remote form*
- Sélectionnez le *wizard Password Remote form* et glissez son icône sur votre librairie ouverte dans le *Class Browser* d'OMNIS
- Nommez la nouvelle *Remote form* et appuyez sur Entrée
- Sélectionnez une *Remote task* existante depuis la liste de *tasks* fournie et cliquez sur le bouton *Next*

ou

- Cliquez sur l'option *Create New Task*, sélectionnez un *template* depuis la liste fournie comme *Plain Remote task*, nommez la nouvelle *Remote task* et cliquez sur le bouton *Next*



- Si vous le désirez, vous pouvez modifier les détails des rubriques Password et Username en utilisant le bouton *Update* ou ajouter quelques rubriques si nécessaire
- Lorsque les détails de votre *form* sont saisis, cliquez sur le bouton *Next*
- Sélectionnez le type de navigateur web et cliquez sur le bouton *Next*

Le *wizard* crée alors la *Remote form* selon les détails que vous avez saisi. Notez qu'OMNIS crée aussi une *Remote task*.

- Pour modifier la *form*, double-cliquez son icône dans le *Class Browser*

Pour tester la *form*

- Appuyez sur Ctrl/Cmnd-T ou effectuez un clic-droit/Ctrl-clic sur la *form* et sélectionnez l'option ***Test Form*** dans le menu contextuel

Form Wizard Sidebar

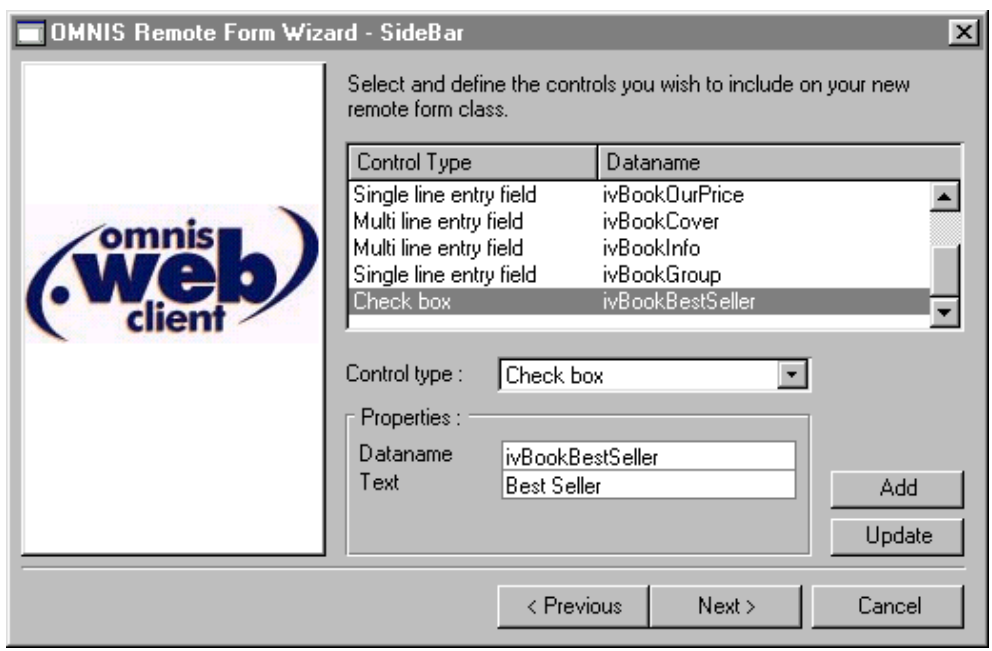
Le *form wizard Sidebar* crée une *form* qui contient un certain nombre de rubriques et un contrôle *Sidebar*. Dans le *wizard* vous pouvez spécifier les groupes et les objets de groupe du contrôle *Sidebar*. Le *wizard* crée un liste qui contient les objets de groupe de votre *Sidebar*. Lorsque vous créerez votre *form* vous devrez ajouter quelques lignes de code deriière le *Sidebar* pour lui permettre de répondre aux clics et double-clics.

Pour créer une *Sidebar Form*

- Ouvrez le *Component Store* et cliquez sur le bouton des classes *Remote form*
- Sélectionnez le *Remote form wizard Sidebar* et glissez son icône sur votre librairie ouverte dans le *Class Browser* d'OMNIS
- Nommez la nouvelle *Remote form* et appuyez sur Entrée
- Sélectionnez une *Remote task* existante dans la liste des *tasks* fournie et cliquez sur le bouton *Next*

ou

- Cliquez sur l'option *Create New Task*, sélectionnez un *template* depuis la liste fournie comme *Plain Remote task*, nommez la nouvelle *Remote task* et cliquez sur le bouton *Next*



- Saisissez les détails de vos rubriques pour créer votre *form* et cliquez sur le bouton *Next*

Pour le *form wizard Sidebar* vous devez saisir différents groupes et séparer les détails de groupe. Le *wizard* vous permet de saisir les noms des groupes (*group name*), ainsi que les noms (*name*) et les numéros d'icône (*IconID*) pour chaque item d'un groupe. Lorsque vous cliquez sur la rubrique IconID, le dialogue de sélection d'icône s'ouvre et vous permet de choisir une icône parmi celles stockées dans la table système #ICONS de la librairie en cours ou des fichiers de données OMNISPIC ou USERPIC.

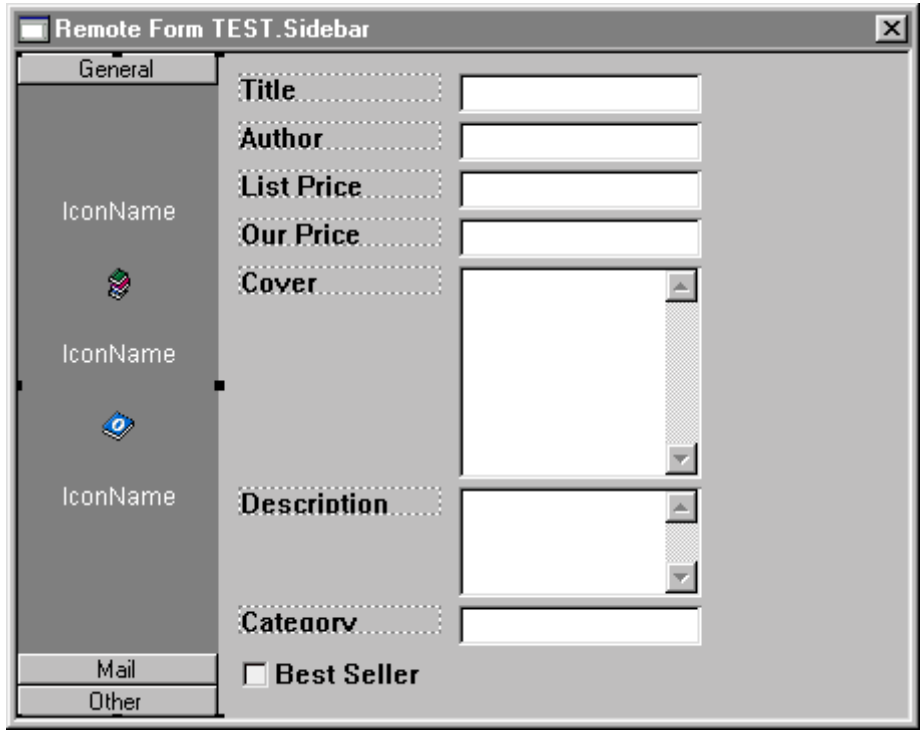
La copie d'écran suivante montre trois groupes et deux items dans chacun d'eux.

- Lorsque vos détails de groupe sont saisis pour la *form*, cliquez sur le bouton *Next*
- Sélectionnez le type de navigateur et cliquez sur le bouton *Next*

Le *wizard* crée alors la *Remote form* en utilisant les détails que vous lui avez fourni. Notez qu'OMNIS crée aussi la *Remote task*.

- Pour modifier la *form*, double-cliquez sur son icône dans le *Class Browser*

La copie d'écran suivante montre la *form* créée avec les détails ci-dessus. Notez qu'en mode création, votre *Sidebar* n'affiche pas les groupes ou les catégories que vous avez saisis dans le *wizard*. Les détails du *Sidebar* sont rassemblés dans une liste qui appartient à la *form* et qui n'est créée qu'en mode runtime.



Pour tester la *form*

- Appuyez sur Ctrl/Cmnd-T, ou effectuez un clic-droit/Ctrl-click sur la *form* et sélectionnez l'option ***Test Form*** dans le menu contextuel

De retour en mode création, vous devrez ajouter une méthode derrière le contrôle *Sidebar* pour répondre aux clics. Par exemple, vous pouvez ajouter la méthode suivante pour charger le numéro de ligne dans la liste de l'item cliqué et exécuter une méthode qui utilisera cette information.

```
On evIconPicked
   Set current list i SidebarList ;; la liste du Sidebar
   Load from list {pLinenum} ;; charge le num. de ligne de l'item
   Do ...quelquechose
```

Form Wizard Sidebar with pages

Le *Form wizard Sidebar with pages* crée une forme qui contient un *page pane* et un contrôle *Sidebar*. Dans le *wizard* vous pouvez spécifier les groupes et les groupes d'items du contrôle *Sidebar*. Le *wizard* crée une liste qui contient les groupes d'items de votre *Sidebar*. Le *wizard* vous permet de spécifier les rubriques pour chaque *pane* et vous permet de lier chaque *pane* à un groupe particulier du contrôle *Sidebar*.

Pour créer une *Form Sidebar with pages*

- Ouvrez le *Component Store* et cliquez sur le bouton des classes *Remote form*
- Sélectionnez le *Remote form wizard Sidebar with pages* et glissez son icône sur votre librairie dans le *Class Browser* d'OMNIS
- Nommez la nouvelle *Remote form* et appuyez sur Entrée
- Sélectionnez une *Remote task* existante depuis la liste des *tasks* fournie et cliquez sur le bouton *Next*

ou

- Cliquez sur l'option *Create New Task*, sélectionnez un *template* depuis la liste fournie comme *Plain Remote task*, nommez la nouvelle *Remote task* et cliquez sur le bouton *Next*
- Spécifiez le nombre de pages de votre *form* et saisissez les détails des rubriques que vous désirez voir apparaître sur chaque page, puis cliquez sur le bouton *Next*
- Il vous faut maintenant spécifier les détails de groupe de votre *Sidebar*
- Lorsque tous les détails de votre *form* sont saisis, cliquez sur le bouton *Next*
- Sélectionnez le type de navigateur et cliquez sur le bouton *Next*

Le *wizard* crée alors la *Remote form* en utilisant les informations que vous lui avez fourni. Notez qu'OMNIS crée aussi une *Remote task*.

- Pour modifier la *form*, double-cliquez sur son icône dans le *Class Browser*

Pour tester la *form*

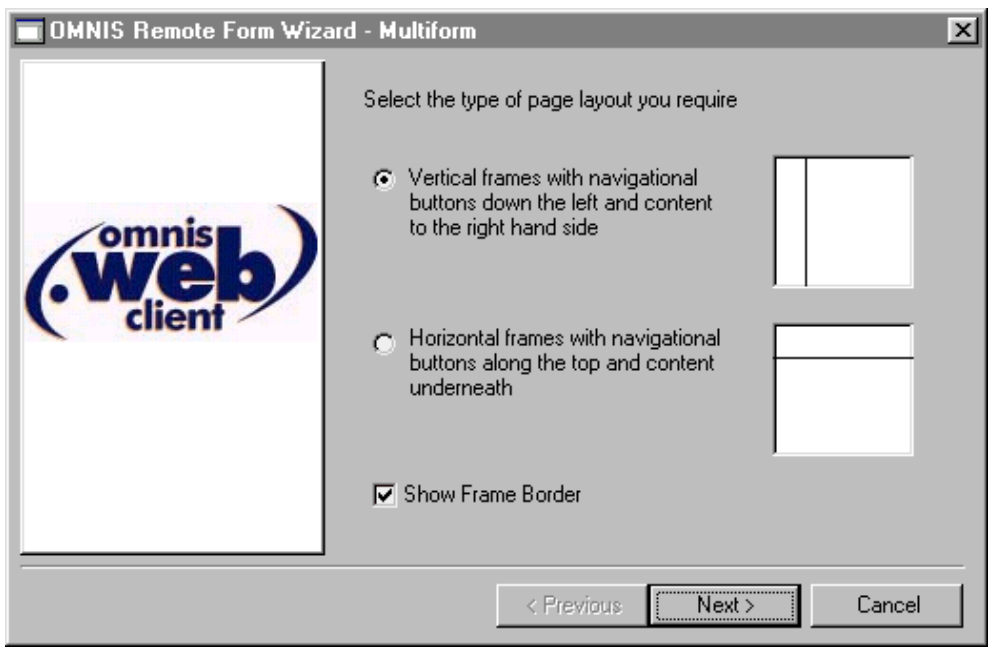
- Appuyez sur Ctrl/Cmnd-T ou effectuez un clic-droit/Ctrl-click sur la *form* et sélectionnez l'option *Test Form* dans le menu contextuel

Form Wizard Multiform

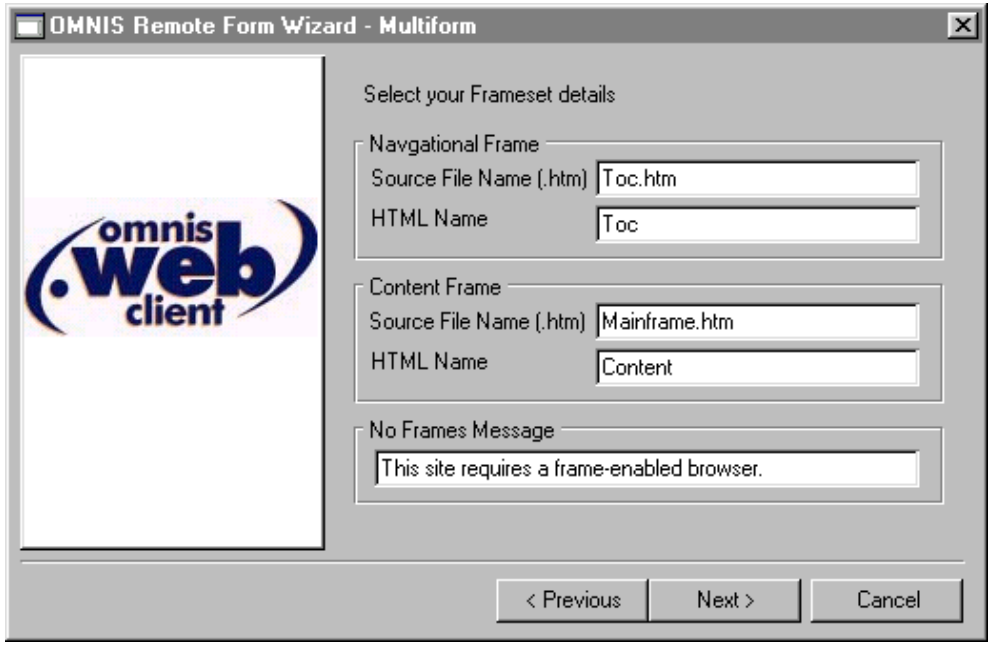
Le *Form wizard Multiform* vous permet de créer un site web complet qui contiendra une home page, une barre de navigation et un certain nombre de *Remote forms* OMNIS. Le *wizard* crée une page html avec un frame horizontal ou vertical et des boutons de navigation liés à différentes *forms*. Dans le *wizard*, vous pouvez sélectionner des *Remote forms* existantes à inclure à votre site web ou créer de nouvelles *Remote forms* à partir des *form wizards* et *templates* proposés.

Pour créer un site web *Multi Form*

- Ouvrez le *Component Store* et cliquez sur le bouton des classes *Remote form*
- Sélectionnez le *Remote form wizard Multi form* et glissez son icône sur votre librairie dans le *Class Browser* d'OMNIS
- Nommez la nouvelle *Remote form* et appuyez sur Entrée



- Sélectionnez l'orientation de vos pages html, avec une barre de navigation soit **Verticale** soit **Horizontale** et décochez l'option **Show frame border** si vous désirez cacher les bords de frames
- Une fois l'orientation choisie, vous devez cliquer sur le bouton *Next*

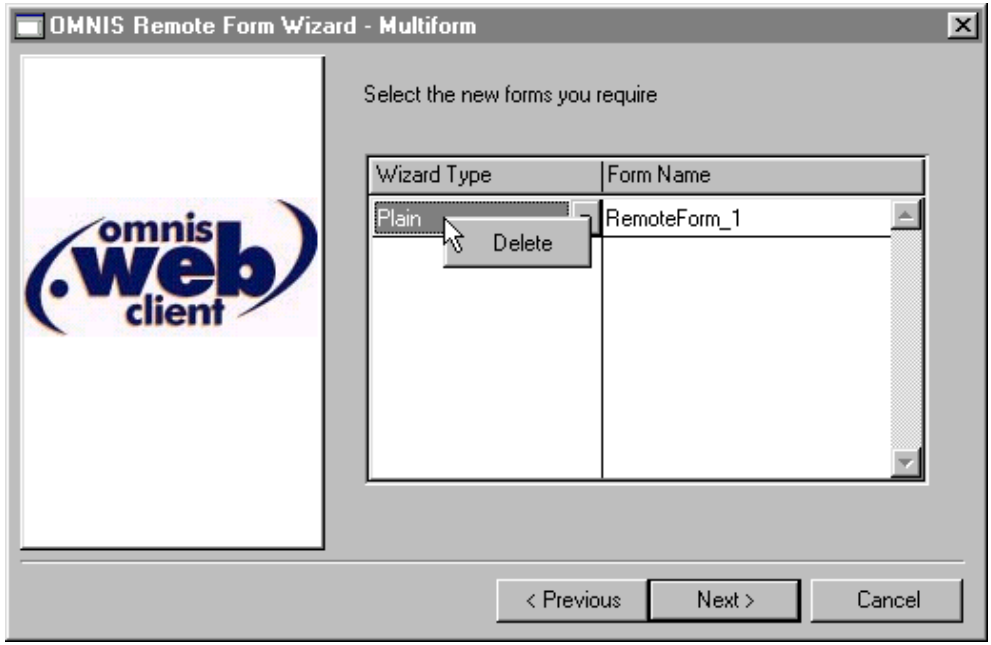


Ensuite, le *wizard Multiform* vous demande de préciser les détails de *Frameset*. Vous pouvez saisir le nom du fichier html du frame de navigation ou de la table des matières et le nom du frame principal ou *Home page*. Ces fichiers sont ajoutés au dossier **html** du dossier principal d'OMNIS et vous pourrez les modifier par la suite.

- Saisissez les noms de fichier pour les frames de navigation et de contenu ou acceptez les noms proposés par défaut et cliquez sur le bouton *Next*

Maintenant, le *wizard Multiform* vous permet de sélectionner les *Remote forms* existantes dans votre librairie que vous voulez inclure dans votre site web. Si votre librairie ne contient aucune *Remote forms*, cliquez sur le bouton *Next* pour passer à l'étape suivante immédiatement. Sinon,

- Sélectionnez le nom d'une ou plusieurs *Remote forms* de votre librairie que vous désirez inclure dans votre site web et cliquez sur le bouton *Next*



Le *wizard Multiform* vous permet de saisir une liste de nouvelles *Remote forms* à ajouter à votre site web. Si votre librairie ne contient pas de *Remote forms* vous devrez spécifier une liste de nouvelles *remote forms* dans la table, ce qui vous permet de préciser le type de *Wizard* et le nom de la *Form* pour chaque nouvelle *Remote form* requise.

Si vous ne voulez pas ajouter de nouvelles *Remote forms*, vous pouvez détruire le contenu de la liste en effectuant un clic-droit/Ctrl-clic sur la liste d'items et en sélectionnant l'option *Delete* comme montré ci-avant.

- Saisissez la liste des nouvelles *Remote forms* requises ou effacez la liste comme au-dessus puis cliquez sur le bouton *Next*
- Sélectionnez le type de navigateur web et cliquez sur le bouton *Next*

Si votre liste comprend de nouvelles *Remote forms* à inclure, les *Remote form wizards* appropriés s'ouvrent. Lorsque les *wizards* sont renseignés pour chaque nouvelle *Remote forms*, vous êtes ramené au *Class Browser*. Le *wizard Multiform* crée les pages html spécifiées et ajoute les nouvelles *Remote forms* à votre librairie. Pour tester votre site web *multi-form*, vous devez ouvrir le fichier html approprié. Pour cela :

- Ouvrez le dossier html du dossier principal d'OMNIS
- Double-cliquez sur le fichier **Frameset.htm** pour l'ouvrir dans votre navigateur web

Debugger et tester les *Remote forms*

Afin de lancer et debugger une *Remote form*, la version de développement d'OMNIS vous permet d'ouvrir une *Remote form* localement dans votre navigateur web. Vous pouvez appuyer sur Ctrl/Cmnd-T pendant la création de votre *Remote form* ou sélectionner l'option *Test Form* depuis le menu contextuel de la classe *form*, pour afficher votre *form* dans votre navigateur. Le Client Web OMNIS établit une connection distante vers la version de développement d'OMNIS. Vous pouvez définir des points d'arrêt (*breakpoints*) dans le code de la *form* et le parcourir pas à pas comme d'accoutumé.

Créer une nouvelle *Remote form*

Vous pouvez créer une nouvelle *Remote form* à partir de rien avec le *template Remote form* disponible dans le *Component Store* ou depuis l'option de menu **Class>>New** du *Class Browser*. Cependant, ce type de *form* ne contient aucune rubrique ou variable d'instance, et vous devrez la lier manuellement à une *Remote task* en spécifiant sa propriété **$designtaskname**.

Pour créer une nouvelle *Remote form*

- Ouvrez le *Component Store* et glissez l'icône du *template Remote form* sur votre librairie dans le *Class Browser* d'OMNIS
- Nommez la nouvelle *Remote form* et appuyez sur Entrée
- Double-cliquer sur la *Remote form* dans le *Class Browser* et ajoutez toutes les variables d'instance, les rubriques et les contrôles requis

Propriétés de *Remote form*

Les classes *Remote form* possèdent tous les attributs standards des classes OMNIS ainsi que la propriété **iconpages**, affichées sous le *tab General* du *Property Manager*.

Icon Pages

Certains composants de *Remote form*, comme les boutons peuvent afficher une icône. Les icônes de tous les composants de *Remote form* sont envoyées au client, avec les données de classe (*class data*), lorsque la *form* est instanciée. Cependant, OMNIS n'envoie pas les icônes individuelles au client, seulement des pages complètes, vous devez donc faire attention et utiliser un minimum de pages d'icônes possible. La propriété **iconpages** d'une *form* spécifie une liste de pages d'icônes qui sont envoyées au client. Les icônes des objets de la *Remote form* peuvent être stockées dans la table système #ICONS de la librairie en cours ou dans les fichiers de données OMNISPIC ou USERPIC.

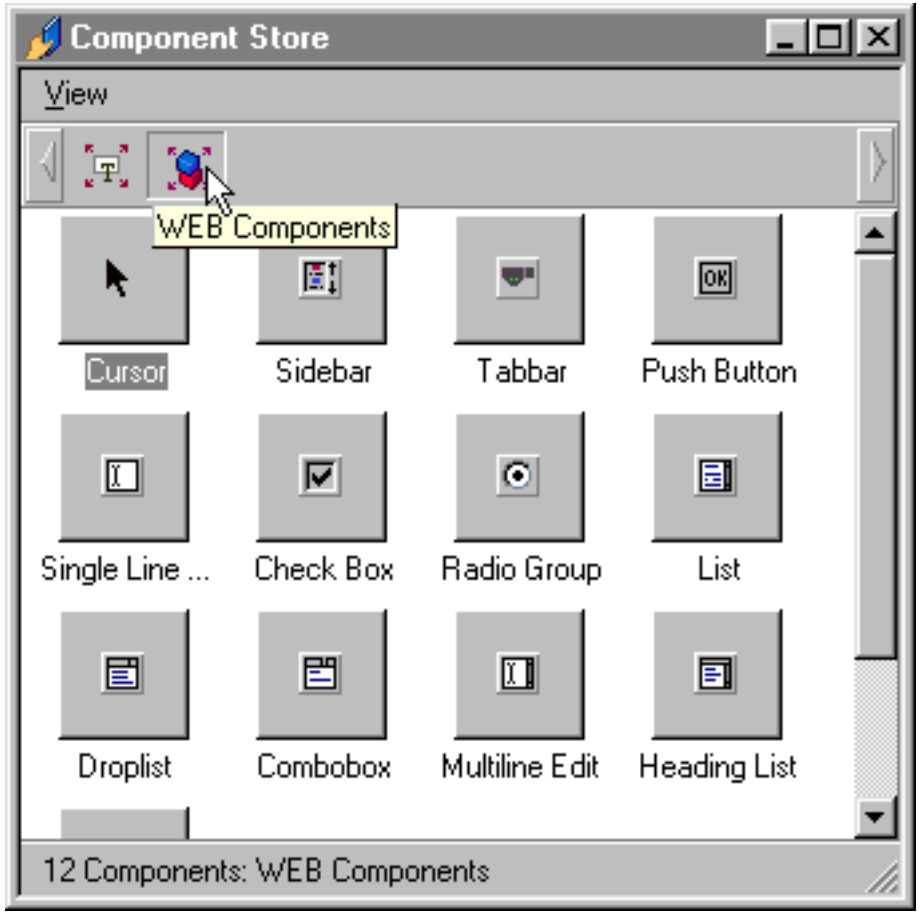
Dans le *Property Manager* vous pouvez cliquer sur la liste déroulante de la propriété **iconpages** et sélectionner une ou plusieurs pages d'icônes. Les noms des pages d'icônes sont stockés dans la propriété **iconpages**, séparés par des virgules. Les pages d'icônes de source différentes (#ICONS, OMNISPIC et USERPIC, dans cet ordre) sont séparées par des points virgules. Par exemple, deux pages nommées Books et *General* de la table système #ICONS de la librairie en cours et une page nommée Embedded Colors du fichier de données USERPIC sont stockées, l'on obtient : Books,*General*;;Embedded Colors.

Contrôles de *Remote form*

Les *Remote form wizards* ajoutent des contrôles à vos *forms* automatiquement, mais si vous créez vos propres *forms* ou si vous ajoutez vos propres contrôles à une *form* existante, le *Component Store* d'OMNIS vous permet de créer différents types de contrôle. Lorsque vous créez un contrôle dans votre *Remote form* vous pouvez modifier ses propriétés dans le *Property Manager*.

Pour créer un contrôle de *Remote form* depuis le *Component Store*

- Ouvrez votre *Remote form* en mode création
- Appuyez sur F3/Cmnd-3 pour ouvrir le *Component Store* ou l'amener en avant-plan
- Cliquez sur le bouton *WEB Components* du *Component Store* ; notez que les objets Label, Border et Page sont dans le groupe *WEB Background Objects* du *Component Store*, séléctionné par défaut



- Glissez le type de contrôle requis depuis le *Component Store* jusqu'à votre *Remote form*

ou, pour dessiner un contrôle de *form* d'une taille particulière

- Sélectionnez le type de contrôle requis dans le *Component Store* en cliquant sur son icône
- Cliquez et positionnez/dimensionnez le masque sur *Remote form* pour définir la taille du composant à intégrer

ou, placez un contrôle dans votre *form* automatiquement

- Double-cliquez sur le type de contrôle approprié dans le *Component Store*

Lorsque vous double-cliquez sur une icône du *Component Store*, un contrôle du même type apparaît au centre de votre *Remote form*. Vous pouvez répéter cette opération plusieurs fois afin de placer plusieurs copies du même type de contrôle sur votre *form*.

Pour plus de détails sur la création de rubriques et de contrôles depuis le *Component Store* et la modification de leurs propriétés via le *Property Manager*, consultez le manuel *Utilisation OMNIS Studio*.

Les *web components* suivants sont disponibles dans le *Component Store*.

Labels

Objet texte standard pour libeller les objets de *form*. Vous pouvez saisir ou modifier le texte de cette objet par sa propriété **text**.

Border

Rectangles avec styles de bordures variées. Sous le *tab* Custom du *Property Manager*, vous pouvez définir le style de bord pour cet objet en spécifiant les propriétés **outer** et **inner** ainsi que **bordergap**.

Pushbuttons

Les contrôle *Pushbutton* permettent d'exécuter une méthode lorsque l'utilisateur clique sur l'objet. Vous pouvez saisir ou modifier le texte du bouton via sa propriété **texte**. Notez que le bouton ne possède pas de propriété **dataname**. Vous pouvez activer spécifiquement les *events* du bouton en sélectionnant les *events* appropriés dans la propriété *events* de l'objet. Par exemple, vous pouvez spécifier **evClick** afin que le bouton réponde aux simples clics. Lorsque l'*event* est émis par un clic la méthode derrière l'objet est exécutée. Consultez la section *events* plus loin dans ce chapitre.

Rubriques Single Line Entry

Les contrôles Single-line permettent de saisir et afficher des données. La seule propriété importante de ce contrôle est sa propriété **dataname**, c'est à dire, le nom de la variable d'instance que la rubrique utilise pour afficher ou insérer des données. Si vous avez créer votre *Remote form* avec l'un des *form wizards*, la propriété **dataname** de chaque rubrique est définie pour vous automatiquement, directement à une colonne spécifique ou une zone de données de votre base. Une variable d'instance pour un contrôle *entry field* doit être de type **Character** ou **National** et son nom doit être saisi dans la propriété **dataname**. Vous pouvez activer ses *events evBefore* ou *evAfter* via sa propriété **events** ; ces *events* détèctent lorsque l'utilisateur entre ou quitte la rubrique et vous permettent de créer une méthode pour intercepter ces *events*. Consultez la section *events* plus loin dans ce chapitre.

Display Format Date and Time

Les rubriques *Single-* et *multi-line* possèdent **displayformat** qui contrôlent les variables date & time affichées sur la machine cliente. Ils peuvent être de différents formats : *None*, *Time*, *ShortDate*, *ShortDateTime*, *LongDate*, et *LongDateTime*. La façon d'afficher les données sur l'interface client dépendra entièrement de la configuration définie dans le **Panneau de Configuration>>Paramètres régionaux** (Sous Windows).

Rubriques Multi Line

Les rubriques *Multi-line* possèdent un ascenceur vertical qui permet de saisir ou d'afficher des données longues. Les propriétés et les *events* d'une rubrique *multi-line* sont les mêmes que pour les rubriques *single-line* ; voir au-dessus.

Check Box

Contrôle que l'utilisateur peut cocher ou décocher. Si l'utilisateur coche le contrôle, la variable derrière la rubrique prend la valeur 1, sinon, sa valeur est zéro. La variable d'instance d'une *check box* doit être de type **Boolean** et son nom doit être saisi dans la propriété **dataname**. Vous pouvez saisir ou modifier le texte associé à la *check box* en modifiant sa propriété **text**.

Objets *Radio Group*

Rubrique à choix unique qui permet à l'utilisateur de sélectionner un choix exclusif parmi plusieurs propositions en cliquant sur un bouton radio. Vous pouvez saisir ou modifier le texte associé à chaque *radio button* du groupe dans une liste séparé par virgule de la propriété **text**. La variable d'instance d'un *Radio group* doit être de type **Number** pour accepter les valeurs numériques des bouton sélectionnés ; le nom de la variable doit être saisi dans la propriété **dataname**. Sous le *Tab* Custom du *Property Manager*, vous pouvez définir les propriétés **columncount**, **minvalue** et **maxvalue** pour le groupe, et vous pouvez spécifier si le groupe est aligné horizontalement ou non avec la propriété **horizontal**.

List Box

Les contrôles *Multi-line list* affichent les données d'une variable liste OMNIS et permet à l'utilisateur de cliquer ou double-cliquer une ligne pour faire une sélection. La variable d'instance d'un contrôle List doit être du type **List** et son nom doit être saisi dans la propriété **dataname** de l'objet. Vous pouvez activer certains *events* spécifiques pour le contrôle en sélectionnant les *events* appropriés dans la propriété **events** de l'objet. Par exemple, vous pouvez spécifier **evDoubleClick** afin que la liste réponde aux double-clics. Lorsque l'*event* est émis par un double-clic, la méthode derrière la liste est exécutée. Consultez la section *events* plus loin dans ce chapitre.

Drop Lists

Les contrôles *Drop List* permettent d'afficher les données d'une variable liste OMNIS dans une liste déroulante. La variable d'instance d'un contrôle *Drop list* doit être de type **List** et son nom sera saisi dans la propriété **dataname**.

Combo Box

Les contrôles *Combo Box* combinent une *dropdown list* et une rubrique *entry field* qui permettent à l'utilisateur de sélectionner une valeur depuis la liste ou de saisir sa propre valeur ; le contrôle requiert deux variables, une variable de type **character** et une variable de type **list** OMNIS. La propriété **dataname** d'une *Combo box* spécifie la variable de type

Character. Sous l'onglet *Custom* du *Property Manager*, la propriété **::listname** spécifie la variable de type *List* pour la partie liste du contrôle. Vous pouvez activer des *events* spécifiques pour le contrôle *Combo box* en sélectionnant les *events* appropriés dans la propriété **events** de l'objet. Consultez la section *events* plus loin dans ce chapitre.

Headed List

Les contrôles *Headed List* sont des contrôles *Multi-line list* associés à des en-tête de style *button* et qui permettent d'ajuster les largeur de colonnes de la liste. La variable d'instance d'un contrôle *Headed list* doit être de type **List** et son nom devra être précisé dans la propriété **dataname**. Sous l'onglet *Custom* du *Property Manager*, vous pouvez spécifier le nombre de colonnes dans la propriété **colcount**, les en-têtes de colonne dans la propriété **columnnames** et la largeur de celles-ci dans la propriété **columnwidths** de l'objet. Comme pour une simple *list box*, vous pouvez activer des *events* spécifiques pour le contrôle *headed list* en sélectionnant les *events* appropriés dans la propriété *events* de l'objet.

Objet *Picture*

Les objets *Picture* sont des contrôles graphiques qui permettent d'afficher des données images. La variable d'instance d'un objet *Picture* doit être de type **Picture** et son nom sera saisi dans la propriété **dataname** de l'objet.

Sidebar

Les *Sidebars* sont des barres de sélection qui permettent à l'utilisateur de choisir une icône parmi un certain nombre de groupes. La variable d'instance d'un contrôle *Sidebar* doit être de type **List** et son nom devra être saisi dans la propriété **dataname** de l'objet. La variable *list* du contrôle *Sidebar* stocke les données statiques qui représentent les noms et les ID d'icônes pour les groupes et les items de groupe du contrôle *Sidebar*. Vous pouvez construire la liste d'un contrôle *Sidebar* avec la méthode $construct() de votre *form*, en utilisant la méthode $add() et les paramètres suivants :

```
Do ListName.$add('GroupName',IconID,'GroupItemName')
```

La méthode suivante construit une liste pour un contrôle *Sidebar* qui contiendra 3 groupes (Cooking, Science et Fiction) avec deux items dans chaque groupe.

```
; declare i<i>Sidebar</i>List of type List
Do i<i>Sidebar</i>List.$define(iGroup,iIconId,iItem)
Do i<i>Sidebar</i>List.$add('Cooking',0,'0') ;; définition d'un groupe
Do i<i>Sidebar</i>List.$add('Cooking',k48x48+3,'Cooking') ;; item de groupe
Do i<i>Sidebar</i>List.$add('Cooking',k48x48+4,'Food') ;; item de groupe
Do i<i>Sidebar</i>List.$add('Science',0,'0')
Do i<i>Sidebar</i>List.$add('Science',k48x48+5,'Science')
Do i<i>Sidebar</i>List.$add('Science',k48x48+6,'Technology')
Do i<i>Sidebar</i>List.$add('Fiction',0,'0')
Do i<i>Sidebar</i>List.$add('Fiction',k48x48+7,'Fiction')
Do i<i>Sidebar</i>List.$add('Fiction',k48x48+7,'Science Fiction')
```

Pour positionner un contrôle *Sidebar* dans une *Remote form* vous pouvez modifier sa propriété **edgefloat** ; vous pouvez la définir, par exemple, à **kEFposnLeftToolBar** afin que le contrôle *Sidebar* “colle” au côté gauche de la *form*. Vous pouvez modifier les autres propriétés du contrôle *Sidebar*, sous les *Tabs Sidebar* et *Text* du *Property Manager* ; pour spécifier, par exemple, le remplissage du fond de l'objet.

Tab Bar

L'objet *Tab Bar* est un bandeau de sélection qui permet à l'utilisateur de cliquer sur un *tab*, pour changer de *page pane* par exemple. Notez qu'un contrôle *tab bar* ne possède pas de propriété **dataname**. Vous pouvez en définir les propriétés sous le *tab TabBar* du *Property Manager* ; vous pouvez, par exemple, spécifier la position de l'objet gràce à sa propriété **position**. Vous pouvez activer des *events* spécifiques au *tab bar* en sélectionnant les *events* appropriés dans la propriété **events** de l'objet. Spécifiquement, **evClick** vous permet de détecter le *tab* sélectionné, en renvoyant le numéro de ce dernier au paramètre *d'event pLineNumber*. Par exemple, La méthode de gestion d'*event* suivante détecte le *tab* cliqué et défini la page en cours d'un contrôle *page pane* associé (nommé Mainpage) dans une *form* :

```
On evClick
   Do $cinst.$objs.Mainpage.$currentpage.$assign(pLineNumber)
```

Page Pane

Contrôle *Multi-page* qui vous permet d'ajouter des rubriques du texte et des contrôles sur *pane* ; ce contrôle est utilisé en corrélation avec un contrôle *tab bar* pour naviguer parmi les *panes*. Sous le *tab General* du *Property Manager*, vous pouvez spécifier le nombre de panes dans la propriété **pagecount** et le pane en cours dans la propriété **currentpage** du contrôle. Notez qu'un contrôle *page pane* ne possèdent pas de propriété **dataname** et ne renvoie pas d'*events*. L'exemple suivant montre la méthode de gestion d'*event* d'un *tab bar* et montre comment vous pouvez changer de *pane* sur un contrôle *page pane*. Notez que la *Remote form* contient deux *page pane*.

```
On evClick
   Do $cinst.$objs.Mainpage.$currentpage.$assign(pLineNumber)
   Switch pLineNumber ;; le tab 1 est cliqué
   Case 1
      Calculate $cinst.$objs.Subpage.$currentpage as 4
   Case 3
      Calculate $cinst.$objs.Subpage.$currentpage as 5
   Case 4
      Calculate $cinst.$objs.Subpage.$currentpage as 6
   Default
      Calculate $cinst.$objs.Subpage.$currentpage as 1
   End Switch
   Do $cinst.$senddata(#NULL)
```

Programmer les *Remote forms*

Lors de l'écriture de méthodes pour une classe *Remote form* ou ses contrôles, vous devez considérer que la classe peut être instanciées plusieurs fois, selon le nombre de clients connectés simultanément. Bien que vous puissiez utiliser des commandes ou la notation OMNIS et créer des méthodes de toute taille, choisissez vos commandes avec attention et mettez autant de soin dans votre programmation pour certaines opérations.

Vous ne pouvez pas modifier une instance de *Remote form* en mode runtime, c'est-à-dire, que vous ne pouvez pas rajouter d'objet à l'instance de la *form* ou en retirer. La même règle s'applique aux variables d'instance. Ce type de modification doit être effectué dans la classe *Remote form* avant l'ouverture de l'instance, si nécessaire. De plus, une notation invalide génèrera une erreur lors de son exécution.

Les commandes relatives aux fenêtres comme ‘Redraw’ ne fonctionnent pas et ne devront pas être utilisées. Utilisez dans ce cas la notation $cobj.$redraw() ou NomObjet.$redraw().

Aucun appel n'est fait à la méthode $control() tâche de classe *Remote form*. Les *events* ne sont envoyés qu'aux méthodes $event() des objets. Aucun *events* n'est généré à destination de la *Remote form* elle-même.

Optimiser la gestion de données

Lors de l'exécution d'*events* sur le serveur, OMNIS retourne par défaut toutes les variables d'instance au client. Pour optimiser les variables retournées, vous pouvez spécifier l'état de chaque donnée à retourner. Pour cela, utilisez la méthode $cinst.$senddata(ivVar1,…ivVarN).Une fois que $senddata() est exécutée, OMNIS ne retournera que les données des variables spécifiées. Ainsi, lorsque vous modifiez les données d'un contrôle, il est normalement suffisant de procéder à un $redraw pour le contrôle avec $cobj.$redraw(). Dans les *Remote forms* vous devrez aussi exécuter un $senddata(), en spécifiant le **dataname** du contrôle concerné. Vous pouvez exécuter une

méthode $senddata() plusieurs fois, et vous pouvez spécifier plusieurs fois les mêmes variables d'instance.

Si vous émettez un $cinst.$senddata() sans spécifier de variables d'instance, la liste des variables à renvoyer préalablement définie est effacée et toutes les variables seront envoyées, à moins que vous n'émettiez un nouveau $senddata(). Si vous n'émettez pas de $senddata(), seules les nouvelles variables spécifiées seront envoyées. La notation $cinst.$senddata(#NULL) indique à OMNIS de ne plus envoyer de données. Vous pouvez aussi utiliser $cinst.$senddata(#NULL,ivVar1) pour vous assurer que seule **ivVar1** est envoyée.

Icon page

Vous pouvez ajouter des *icon pages* avec RemoteFormRef.$iconpages.$add(iconID). OMNIS ajoute le nom de la page qui contient les *icon page* qui contiennent elle-même l'icône spécifiée dans la propriété $iconpages de votre *Remote form*.

Fenêtres ouvertes et demandes utilisateur

Vous ne devez pas utiliser de commandes ou de programmation dans votre librairie qui ouvre une *message box* ou un dialogue modal. Si, par exemple, vous utilisez la commande standard *OK message* et qu'elle est exécutée, le message apparaît sur le serveur OMNIS, pas dans le client OMNIS et l'application serveur est stoppée jusqu'à réponse du message par le bouton OK, sur le serveur ; voir plus bas une méthode d'envoi de message modal au client. Si un message ou un dialogue reste ouvert dans l'application serveur OMNIS, toutes les interactions avec les clients connectés au serveur sont suspendues et le navigateur client se fige. Cette situation ne cause pas de problème sérieux pendant le développement, puisque vous pouvez rebasculer vers OMNIS et vous affranchir du message bloquant, mais pour une application distribuée sur le web c'est un désastre qui réduira grandement l'usage de votre application !

Il existe de nombreuses commandes que vous ne devez pas utiliser pour les raisons citées ci avant, elles comprennent les commandes : *OK message, Yes/No message, Working message, Prompt for input, Prompt for data file,* et ainsi de suite.

Vous devez aussi considérer le résultat de l'utilisation de vos commandes ou notation qui ouvre une fenêtre, installe un menu ou une *toolbar*, imprime un état et ainsi de suite. Ces commandes s'appliquent toutes à l'application serveur et leur résultat n'est pas visible à partir du navigateur web client. Si, par exemple, vous lancez la commande *Open window*, une classe *window* OMNIS s'ouvre au sein de l'application serveur et non pas au travers du navigateur web client ; Ceci est valable pour une fenêtre d'administration du serveur, par exemple, mais ne peut être employé pour un client. Donc, conservez à l'esprit que toutes les interactions avec l'utilisateur via le web doivent être activées via les *Remote forms* ouvertes chez le client.

OK Message

Vous pouvez afficher un message chez le client avec la méthode $showmessage(). Par exemple, le code suivant pour un *pushbutton* ouvre un dialogue message.

```
On evClick ;; le bouton doit avoir l'event evClick activé
   Do $cinst.$showmessage('Message','Titre')
```

Smart list

Le client web OMNIS et les *Remote forms* ne supportent pas les *smart lists* OMNIS.

Instances de *Remote form*

Une fois le client web OMNIS connecté au serveur OMNIS, l'utilisateur peut tabuler à travers les différentes rubriques de la *form* et basculer parmi les pages qui lui sont proposées, saisir des données, cliquer sur des boutons, sélectionner des options et ainsi de suite. Lorsque l'utilisateur génère un *event*, qui a été activé, l'*event* est exécuté sur le serveur. Seuls les *events* activés entraînent une communication entre le client et le serveur, tous les autres *events* sont ignorés.

Il existe un groupe nommé $iremoteforms dans $root. Ce groupe contient la chaîne des instances de *Remote form*. Sur le serveur, seule la partie non-visuelle (principalement ses méthodes) de la *form* sont instanciées. La partie visuelle de l'instance de *form* est instanciée chez le client.

La méthode $construct d'une classe *Remote form* peut spécifier des variables paramètres de type **Row** comme premiers paramètres afin d'accéder aux paramètres ActiveX. Lorsque la méthode $construct de la *form* termine son exécution, les données de la *form*, les variables d'instance et toutes les modifications de propriété effectuées à la *form* et ses composants pendant la "construction" sont retournés au client.

Form Cache

Mémoire cache des *Remote forms* coté serveur

Afin d'améliorer les performances de connexion, le runtime serveur OMNIS garde en mémoire cache de nombreux items lors de la connexion du premier client à une *Remote form*. Les connexions suivantes se servent de ces données mises en cache. Les items suivants sont cachés : les données de classe, la définition des variables d'instance, les *icon pages* et les tables de police.

Vous pouvez implémenter quelques processus de mise à jour comme lorsque la *Remote form* a été modifiée, mais la mémoire cache doit être effacée. Si elle n'est pas effacée, les nouvelles connexions continueront à s'effectuer sur les anciennes données de classe.

Vous pouvez utiliser la méthode de $root nommée $clearcachedforms() dans votre application pour effacer tous les items placés en mémoire cache du serveur OMNIS.

Mémoire cache des *Remote forms* coté client

En plus de la mémoire cache des *Remote form* du serveur, le client web OMNIS peut lui aussi stocker en mémoire cache n'importe quelle *Remote forms* téléchargée depuis le serveur sur le disque des postes client. Le client stocke en mémoire cache les données de classe *Remote form*, la définition des variables d'instance, les *icon pages* et les tables de police. La mémoire cache client stocke aussi les dates de modification de chaque item. Lorsque le client se connecte à la *form* ultérieurement, ces dates de modification stockées sont envoyées au serveur et le serveur ne retourne que les items modifiés.

NOTE IMPORTANTE : Les *icon pages* reçoivent leurs données modifiées depuis la table système #ICONS de la librairie OMNIS en cours. Si vous incluez des icônes depuis les fichiers d'icônes OMNISPIC ou USERPIC et que vous les mettez à jour, vous devez modifier la classe #ICONS pour forcer la mise à jour de toutes les icônes du client ; c'est la seule façon qu'a OMNIS d'indiquer la modification d'OMNISPIC.DF1 ou d'USERPIC.DF1, plutôt que de vérifier la date de modification de #ICONS.

Events

Les *Remote forms* ne génèrent pas d'*events* eux-mêmes, seule la méthode $construct de la *form* est appelée à l'instanciation de celle-ci. La plupart des objets de *Remote form* le font cependant, bien que les *events* soient désactivés par défaut. Vous pouvez activer spécifiquement les *events* d'un objet en définissant sa propriété *events* à travers le *Property Manager*. Lorsqu'ils sont activés, les *events* sont envoyés à la méthode $event() de l'objet. Vous pouvez ajouter votre programmation à la méthode $event() d'un objet, qui sera appelée lorsque l'*event* sera généré. Pour plus d'informations sur la gestion des *events*, consultez le manuel Programmation *OMNIS*.

La plupart des objets de *Remote form* renvoie les *events* **evBefore** et **evAfter**, générés lorsque l'utilisateur s'apprête à entrer dans une rubrique ou à la quitter, respectivement. Vous pouvez utiliser la commande *On* pour détecter les *events* dans vos méthodes de gestion d'*events*. Par exemple, dans la méthode $event() d'une rubrique *single-line entry*, vous pouvez utiliser la méthode suivante pour détecter les events **evBefore** ou **evAfter**.

```
On evBefore
   ; do quelquechose..
On evAfter
   ; do quelquechose d'autre..
```

Les *Pushbuttons* et tous les contrôles de type *list* renvoient les *events* **evClick** et **evDoubleClick**, ainsi que **evBefore** et **evAfter**. La liste *Booklist* (type *heading list*) de notre librairie tutorial possède la méthode $event() suivante, qui est exécutée lorsque l'utilisateur double-clic sur une ligne de la liste :

```
On evDoubleClick
   Do method $findnow
   ; exécute la méthode $findnow de la form.
      ; la méthode charge lés détails du livre sélectionné
      ; et redessine les rubriques de la form
```

Les autres contrôles de *form* possèdent leurs *events* particuliers ; le contrôle *Sidebar*, par exemple, possède les *events* **evIconPicked, evSetPicked** et **evSetBeingPicked** ; le premier est émis lorsqu'une icône est sélectionnée, les deux autres sont émis lorsque l'utilisateur sélectionne une icône ou un groupe. Le contrôle *Sidebar* utilisé dans la librairie tutorial possède la méthode $event() suivante, qui est exécutée lorsque l'utilisateur sélectionne une icône :

```
On evIconPicked
   Set current list i Sidebar List
   ; utilise la variable de liste du Sidebar comme liste en cours
   Load from list {pLinenum}
   ; charge la ligne de la liste en se référant à son numéro
      ; le paramètre pLinenum est envoyé à la méthode
   Do method $ctask.$buildbooks (iBookGroup)
   ; exécute la méthode $buildbooks de la task en cours avec la
      ; valeur en cour de iBookGroup tirée de la ligne sélectionnée
   Calculate iBookList as tBookList
   ; transfert les données de tBookList vers la var. d'instance
      ; iBookList de la Remote form
   Do $cinst.$objs.BookList.$redraw()
   ; redessine la rubrique Booklist de la form
   Do $cinst.$senddata(iBookList)
   ; envoie les données de iBookList au client
```

Chapitre 4 – Classes *Remote task*

Une *Remote task* est un type de classe OMNIS qui gère la connexion entre une *Remote form* et l'application serveur OMNIS, et qui dans certains cas effectue quelques opérations de gestion et de traitement d'*event* en général réservé à la partie serveur. Ainsi, toutes les *Remote forms* requièrent une *Remote task.* Lorsque vous créez une *Remote form* avec l'un des *form wizards*, OMNIS crée une classe *Remote task* pour vous automatiquement. Si vous désirez créer vos propres classes *Remote task*, le *Component Store* contient un certain nombre de *templates* et de *wizards*.

Comme alternative à l'emploi du client web OMNIS et des *Remote forms*, OMNIS Studio vous permet d'interagir avec votre application OMNIS et votre base de données à travers l'internet en utilisant des *forms* html standard et des *Remote tasks*. Dans ce cas, votre *forms* html se connecte directement à une classe *Remote task* OMNIS et aucune *Remote form* n'est nécessaire dans ce type d'interaction. Plus de détails sur les interactions directes avec les html *forms* sont données plus loin dans ce chapitre.

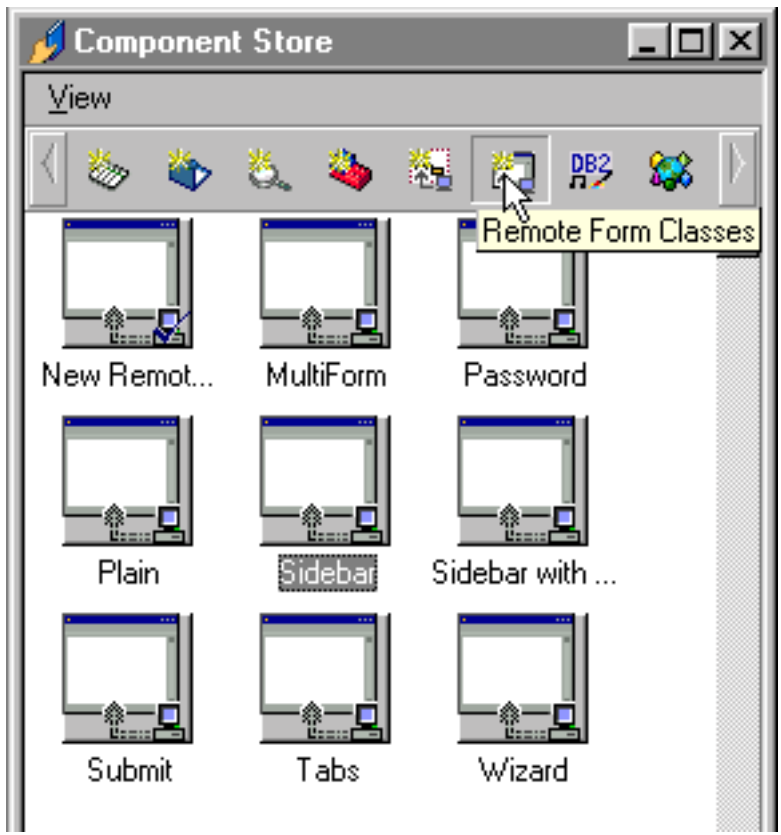
Création de classes *Remote task* Classes avec les *Wizards*

Vous pouvez créer une *Remote task* OMNIS de vous-même ou en utilisant les *templates* ou les *wizards* fournis dans le *Component Store*. Les *templates* suivants sont disponibles :

- **Plain Remote task**
  crée une *Remote task* vide qui contient une méthode $event avec le code de gestion des *events* evBusy et evIdle
- **Monitor**
  crée une *task* et une fenêtre pour suivre l'activité des connexions distantes.
- **HTML Report**
  crée une *task* qui génère des *reports* html à la demande
- **Submit**
  crée une *task* et un fichier html de type submit *form* qui interagit directement avec OMNIS.

Pour créer une classe *Remote task* class avec un *wizard*

- Ouvrez votre librairie dans le *Class Browser* d'OMNIS
- Affichez les classes de votre librairie avec l'option de menu **View>>Down One Level** de la barre de menu du *Browser*
- Appuyez sur F3/Cmnd-3 pour ouvrir le *Component Store* ou l'amener en avant-plan
- Parcourez la barre d'outils du *Component Store* et cliquez sur le bouton ***Remote task Classes*** pour afficher les *wizards* et *templates* de *Remote task*



- Glissez l'icône d'un *wizard* sur votre librairie dans le *Class Browser*
- Nommez la nouvelle classe et appuyez sur Entrée
- Suivez les instructions du *wizard*

Wizard Plain Remote task

Le *wizard Plain Task* crée un *template Remote task* basique qui convient à une connexion aux plus simples des *Remote forms*. La *Plain Remote task* possède aussi une méthode $event() qui contient une méthode de gestion des *template event* et qui détecte les *events* **evBusy** et **evIdle** dans la *task*. Vous pouvez ajouter votre propre code pour gérer ces *events.*

La *Plain Remote task* possède une méthode $construct() qui contient une variable paramètre nommée *pParams* de type *Row Variable*. Cette *row variable* peut recevoir tous les paramètres des ActiveX/plug-in, comprenant le nom de la *Remote form*, le nom de la *task*, etc, et jusqu'à neuf paramètres supplémentaires imbriqués dans la page html qui contient la *form*.

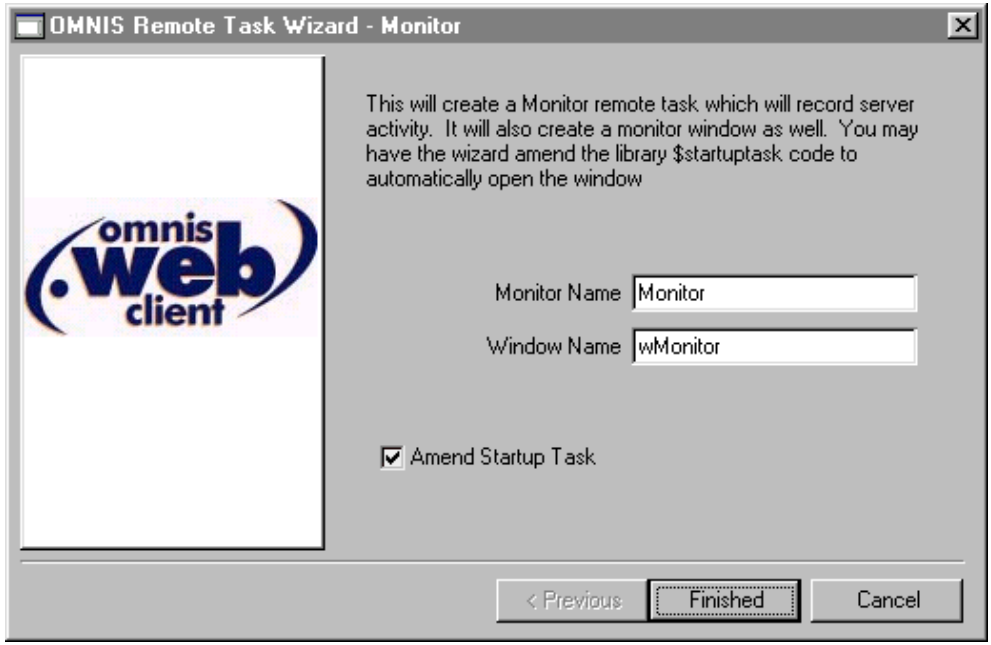
Lorsque vous créez une *task* avec le *wizard Plain Task* vous pouvez spécifier l'option **Inherit from Monitor task**. Cette option ajoute un ensemble de classes “monitor” à votre librairie qui permettent d'enregistrer les connexions associées aux clients qui utiliseront la nouvelle *plain task* que vous créez dans votre librairie. Si vous cochez l'option **Monitor**, le *wizard* vous demande les détails de création de la nouvelle *task monitor*. Si votre librairie ne contient pas de *task monitor,* vous devez spécifier l'option **Create New Monitor Task**. Si, cependant, votre librairie contient une *task monitor, vous* pouvez spécifier l'option **Use Existing Monitor Task** pour ajouter la nouvelle *plain task* à votre librairie liée à la *task monitor* existante. Consultez la section *Monitor Wizard* pour plus de détails.

Wizard Monitor Remote task

Le *wizard Monitor* crée un certain nombre de classes "*monitor*", qui comprennent une nouvelle *task* et une fenêtre "*monitor"*. Celles-ci vous permettent d'enregistrer les connexions entre les clients qui utilisent le client web OMNIS et le serveur OMNIS.

Pour créer une *Monitor window* et ses classes associées

- Affichez les classes de votre librairie avec l'option de menu **View>>Down One Level** de la barre de menus du *Class Browser*
- Appuyez sur F3/Cmnd-3 pour ouvrir le *Component Store* ou l'amener en avant-plan
- Parcourez la barre d'outils du *Component Store* et cliquez sur le bouton ***Remote task Classes*** pour afficher les *wizards* et *templates* de *Remote task*
- Glissez le *wizard* **Monitor** sur votre librairie dans le *Class Browser*
- Nommez la nouvelle classe ou conservez son nom par défaut et appuyez sur Entrée
- Suivez les instructions du *wizard*



Le *wizard* vous demande de saisir le nom de la nouvelle classe *Monitor Remote task* et de la fenêtre ; dans la plupart des cas cependant, vous pouvez accepter les noms par défaut. L'option **Amend Startup Task** vous permet d'ajouter du code à la Startup_Task de la librairie en cours pour ouvrir la fenêtre *Monitor* au lancement de la librairie ; cette option est cochée par défaut, et dans la plupart des cas, il est plus pratique de laisser OMNIS modifier le code de démarrage seul.

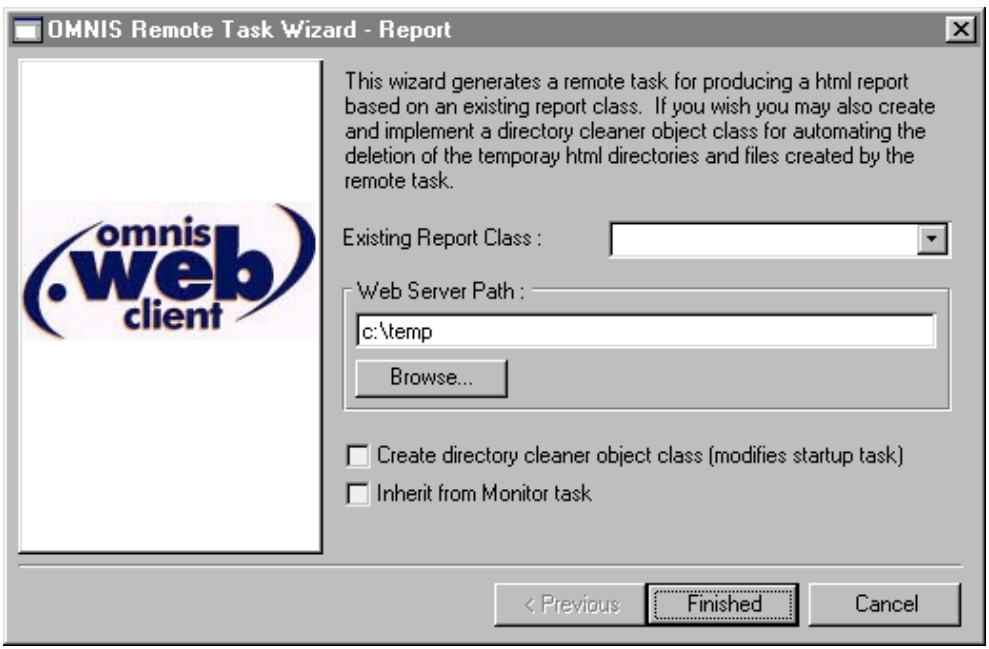
La fenêtre *Monitor*

La fenêtre *Monitor*, nommée wMonitor par défaut, présente trois *panes*. Le *pane* **Connections** qui affiche les connexions groupées par nom de *Remote form*. Le *pane* **History** et le pane **Server Usage** qui vous permet de surveiller le trafic de votre serveur OMNIS et qui fournit quelques informations générales sur l'emploi du serveur. Vous pouvez imprimer ces dernières informations en appuyant sur le bouton **Print Report**.

Wizard HTML Report Task

Le *wizard HTML report task* crée une *Remote task* qui génère des états html à la demande et les renvoie au client. La méthode $construct() contient le code d'impression d'état vers un document html en utilisant un objet-répertoire temporaire et retourne une URL ou un message d'erreur au client.

Le *wizard* vous permet de baser votre *Html report* sur une classe *report* OMNIS existante. Vous pouvez aussi spécifier le dossier où les pages temporaires seront placées et vous pouvez ajouter une classe objet à votre librairie pour effacer le contenu de ce dossier. De plus, vous pouvez ajouter une nouvelle *HTML report task* à une *Monitor task*.



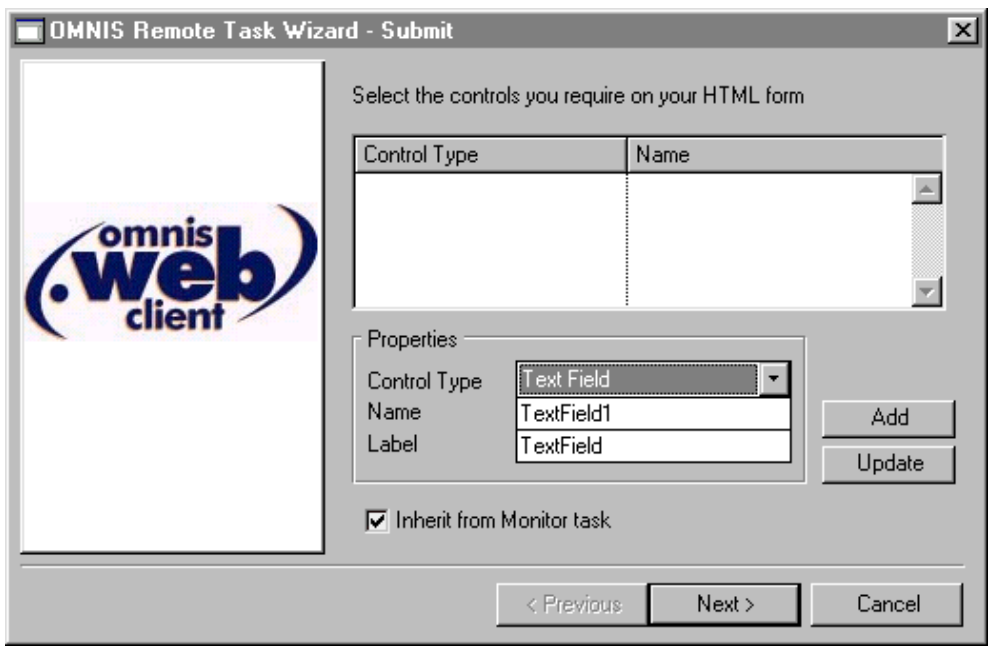
Le *wizard HTML Report* crée un fichier html qui contient un bouton *Print Report*, sous le nom TaskName.htm, et le place dans le dossier OMNIS/HTML. Lorsque le bouton est cliqué sur la *form* html, la méthode $construct() de la *task* crée une nouvelle page html dans le dossier webserver\omnishtml\[reportname] qui contient l'état. Cette page est ensuite envoyée à l'utilisateur. L'objet '*Cleaner*' nettoie le dossier omnishtml de votre webserver environ toutes les minutes.

Wizard Submit *Remote task*

Le *wizard Submit* crée une *Remote task* et un fichier html qui contient une *submit form*. La *form* html créée par le *wizard* vous permet un accès direct à votre base de données et n'utilise pas de *Remote form* ou le client web OMNIS. De plus amples détails sont donnés plus dans ce chapitre à propos de l'utilisation des *forms* html standard pour l'accès direct.

La méthode $construct() de la *Submit task* contient des variables pour gérer les données de la *form* du fichier html associé. Le *wizard* place le fichier html *Submit* dans votre dossier OMNIS/HTML. La *Submit task* peut être héritée d'une *monitor task*.

Le premier *pane* du *wizard* vous permet de définir les rubriques qui apparaîtront sur votre *Submit form* html. Un bouton **Submit** et **Reset** sont inclus dans la *form* automatiquement.



Vous devrez compléter le code de la méthode $construct() de votre *Remote task* afin qu'elle retourne l'URL que vous désirez lorsque l'utilisateur appuie sur le bouton **Submit** de la *form*. Cette manipulation est documentée dans la méthode de *task* $construct().

Créer une nouvelle classe *Remote task*

Normalement, vous utiliserez les *wizards* et *templates* du *Component Store* pour créer une *Remote task*, mais vous pouvez créer une *task* entièrement vide en utilisant le *template New Remote task*, soit depuis le *Component Store*, soit via l'option de menu **Class>>New** de la barre de menus du *Class Browser*.

Pour créer une nouvelle classe *Remote task*

- Ouvrez votre librairie dans le *Class Browser* d'OMNIS
- Affichez les classes de votre librairie via l'option de menu **View>>Down One Level** de la barre de menus du *Browser*
- Glissez le *template* nommé "*New Remote task*" depuis le *Component Store* vers votre librairie dans le *Browser*

ou

- Sélectionnez l'option de menu **Class>>New>>Remote task** depuis la barre de menus du *Browser*
- Nommez la nouvelle *task*
- Double-cliquez sur la classe *task* pour la modifier

Une classe *Remote task* se modifie à travers le *Method editor*. Vous pouvez placer dans $construct() le code que vous désirez exécuter lorsque la *Remote task* est instanciée. Vous pouvez ajouter vos propriétés et méthodes à la *task*, ainsi que les vos variables et variables d'instance.

Instances de *Remote task*

Lorsque le client web OMNIS se connecte pour la première fois, OMNIS crée une instance de la classe *Remote task* spécifiée par la classe *Remote form* à laquelle vous vous connectez (Propriété $designtask de la *form*). La méthode $construct de la classe *Remote task* peut spécifier une variable paramètre de type *Row* comme premier paramètre, et qui contiendra jusqu'à neuf paramètres optionnels que vous pouvez spécifier dans votre code html. Si vous voulez vérifier le contenu du paramètre **Param1,** vous pouvez utiliser ***params***.*pOption1* en considérant que vous avez nommé votre variable *row* ***params***.

Lorsque la méthode $construct de la *Remote task* s'achève, elle ne doit pas retourner de valeur si la connexion se poursuit, comme avec un *Quit method* sans valeur de retour. Si vous devez renvoyer une erreur, vous pouvez utiliser un *Quit method 'Le message d'erreur'*. Dans ce cas, l'instance de *task* est détruite et la connexion est rompue. Si vous ne

retournez pas d'erreur, OMNIS crée alors l'instance de la classe *Remote form* et appelle la méthode $construct de la *form*.

Events

Pour les *Remote tasks*, les *events* **evBusy** et **evIdle** *sont envoyés* à la méthode $event() pendant la durée de la connexion : **evBusy** est envoyé lorsqu'OMNIS reçoit une requête d'un client, **evIdle** est envoyé lorsqu'OMNIS s'apprête à retourner le résultat de la requête. L'exemple suivant montre le code d'une méthode $event() d'une *Monitor task* créée par le *wizard Monitor task* :

```
On evBusy
   If iMonitorOpen
      Do iMonitorRef.$setstatus($cinst,kTrue) Returns lServerBusyFlag
      ; iMonitorRef est une référence à une fenêtre Monitor
      ; créée par le wizard Monitor task
      If lServerBusyFlag
         Quit event handler (Discard event)
      End If
   End If
On evIdle
   If iMonitorOpen
      Do iMonitorRef.$setstatus($cinst,kFalse)
   End If
```

De plus, les *tasks* renvoient l'*event* ***evRejected*** généré lorsqu'OMNIS rejette une connexion. Cette situation se produit habituellement si un trop grand nombre d'utilisateurs tente de se connecter à OMNIS, ou si la valeur de la propriété **$maxusers** de la *Remote task* est dépassée. Le paramètre **pErrorText** renvoie "Too many users connecting to server" pour le premier cas et "Too many users connecting to task [nom de *task*]" pour le second.

Statistiques d'accès client

Les instances de *Remote tasks* présentent un certain nombre de propriétés qui vous permettent de suivre les connexions de l'application serveur OMNIS.

– **$connectbytessent**
spécifie le nombre d'octets envoyés au client pendant la connexion. Cette propriété est définie après l'exécution de $construct().

– **$requests**
spécifie le nombre d'*events* exécutés sur le serveur. Les messages de connexion et déconnexion en sont exclus. Mis à jour avant le message evBusy.

– **$reqtotbytesreceived**
le nombre total d'octets reçus du client pour toutes les requêtes. Pour calculer une

moyenne par requête, vous pouvez diviser cette valeur par $requests. Mis à jour avant le message evBusy.

- **$reqtotbytessent**
  le nombre total d'octets envoyés au client pour toutes les requêtes. Pour calculer une moyenne par requête, vous pouvez diviser cette valeur par $requests. Mis à jour avant le message evIdle.
- **$reqmaxbytesreceived**
  le plus grand bloc, exprimé en octets reçus du client lors de toutes ses requêtes. Mis à jour avant le message evBusy.
- **$reqmaxbytessent**
  le plus grand bloc, exprimé en octets envoyé par le client lors de toutes ses requêtes. Mis à jour avant le message evIdle.
- **$reqcurbytesreceived**
  le nombre d'octets reçus du client pour la requête en cours. Mis à jour avant le message evBusy pour la requête en cours.
- **$reqcurbytessent**
  le nombre d'octets envoyés au client pour la requête en cours. Mis à jour avant le message evIdle.

Connexion via une form HTML standard

En plus du client web OMNIS, OMNIS Studio vous permet d'interagir avec votre application OMNIS en utilisant les *forms* html standards. Dans ce cas, votre *form* html doit se connecter à une classe *Remote task* OMNIS sans se connecter à une *Remote form*. Par exemple :

```
<form method="GET" action="/cgi-bin/omnisapi.dll">
<input type="hidden" name="OmnisServer" value="PortNumber">
<input type="hidden" name="OmnisClass" value="RemoteTaskName">
 <input type="hidden" name="OmnisLibrary" value="LibraryName">
<p><input type="password" name="pPassword" size="20"></p>
 <p><input type="text" name="pQuery" size="80"></p>
 <p><input type="submit" value="Submit" name="B1">
 <input type="reset" value="Reset" name="B2"></p>
</form>
```

La *form* possède les champs spéciaux :

– **OmnisServer**
spécifie le numéro du port ou le nom du service du serveur OMNIS, c'est à dire la valeur spécifiée dans la préférence $serverport du runtime serveur OMNIS.

– **OmnisClass**
le nom de la classe *Remote task* à laquelle se connecter.

– **OmnisLibrary**
le nom interne de la librairie à laquelle se connecter, c'est à dire, le nom de votre librairie moins l'extension .lbs ; la librairie et le serveur OMNIS doivent fonctionner sur le serveur pour que la connexion à la *form* puisse s'établir.

Le code source html suivant implémente une *form* feedback. Les paramètres spécifiques OMNIS sont marqués en gras ; la suite du code source spécifie les champs de la *form* et le libellés texte ainsi que le bouton standard *Submit*.

```
<form method="GET" action="/cgi-bin/omnisapi.dll">
   <input type="hidden" name="OmnisClass" value="tThinClientFeedback"> ;; Le
   nom de la classe Remote task
   <input type="hidden" name="OmnisLibrary" value="Webclient"> ;; le nom de
   la librairie
   <input type="hidden" name="OmnisServer" value="5912"> ;; le numéro de port
   <table border="0" cellspacing="0" cellpadding="0" width="760">
   <tr>
      <td width="788" valign="top"><div align="right">
      <p><strong><font face="Arial">Developer
               Name:</font></strong></td>
      <td width="564" height="25"><font face="Arial">
      <input type="text" name="Name" size="27"></font></td>
   </tr>
   <tr>
      <td width="788"><div align="right">
      <p><strong><font face="Arial">Serial
               No:</font></strong></td>
      <td width="564" height="25"><font face="Arial">
      <input type="text" name="Serial" size="27"></font></td>
   </tr>
   <tr>
      <td width="788" valign="top"><strong><font face="Arial">
            <div align="right">
      <p>Platform:</font></strong></td>
      <td width="564" height="23"><table border="0"
            cellspacing="0" cellpadding="0" width="520">
         <tr>
            <td width="135"><font face="Arial">
            <input type="checkbox" name="Macintosh" value="YES">
            <strong>Macintosh</strong></font></td>
```

```
            <td width="385"><font face="Arial">
            <strong><input type="checkbox" name="Windows"
                    value="YES">Windows</strong></font></td>
         </tr>
      </table>
      </td>
   </tr>
   <tr>
      <td width="788" valign="top"><strong><font face="Arial">
      <div align="right"><p>Client:</font></strong></td>
      <td width="564" height="23">
      <table border="0" cellspacing="0" cellpadding="0"
               width="601">
         <tr>
            <td width="136" height="13"><font face="Arial">
            <input type="checkbox" name="ActiveX" value="YES">
            <strong>ActiveX   </strong></font></td>
            <td width="135" height="13"><font face="Arial">
            <strong><input type="checkbox" name="Netscape"
                  value="YES">Netscape </strong></font></td>
            <td width="330" height="13"><font face="Arial"><strong>
            <input type="checkbox" name="RAWHTML"
                  value="YES">RawHTML</strong></font></td>
         </tr>
      </table>
      </td>
   </tr>
   <tr>
      <td width="788" valign="top"><strong><font face="Arial">
               <div align="right">
      <p>Comments:</font></strong></td>
      <td width="564" height="249" valign="top">
      <font face="Arial"><textarea rows="11" name="Comments"
               cols="57"></textarea></font></td>
   </tr>
   </table>
   <div align="center"><center><p><font face="Arial">
   <input type="submit" value="Send Comments" name="B1"></font></p>
   </center></div>
</form>
```

Lorsque le bouton ‘Submit’ est pressé, l'extension web serveur (par exemple omnisapi.dll) est exécutée et tous les paramètres de la *form* sont transmis. L'extension web serveur envoie la requête à la file d'attente OMNIS. OMNIS crée une instance de la classe *Remote task* spécifiée et appelle sa méthode $construct. Lorsque $construct s'achève, l'instance de la *task* est détruite.

Wizards Task et $construct

Toutes les *Remote tasks* créée à l'aide de *wizards* possèdent une méthode $construct() qui contient une variable paramètre de type *Row* et nommée pParams. Tous les champs html de la *form* peuvent être atteints par cette variable *row*. Par exemple, en utilisant la *form* ci-dessous, vous pouvez vérifier la valeur du champ **password** avec *pParams.pPassword*. Notez que le nom de colonne dans la variable *row* est le nom donné au contrôle de la *form* html.

Lorsque la méthode $construct de la *task* s'achève, elle peut passer l'URL de la page de résultat si nécessaire, c'est à dire, *Quit method 'www.monsiteweb.fr/result.htm'*. Le résultat peut être une simple page de remerciements annonçant au client que ses informations ont bien été reçues ou un *html report* généré par OMNIS et sa destination d'impression **html**.

Si vous devez retourner un message d'erreur, utilisez *Quit method 'nnn Le message'd'erreur*, où *nnn* est un code d'erreur à trois chiffres suivi d'un seul espace avant le texte du message d'erreur.

Chapitre 5 – Distribuer sa solution web OMNIS

Une fois vos *Remote forms* et *tasks* créées dans une librairie OMNIS, vous êtes prêt à distribuer votre application via le web. Vous devez placer votre librairie OMNIS sur un serveur Win32 avec un runtime OMNIS et implémenter vos pages html sur votre serveur web intranet ou internet. Lorsque les utilisateurs accèderont à votre site web pour la première fois, ils devront télécharger le client web OMNIS. Vous pouvez utiliser les installeurs fournis sur le CD OMNIS Studio ou créer vos propres installeurs mais dans tous les cas, vos clients devront télécharger le client web avant d'utiliser votre solution.

Ce chapitre aborde les sujets suivants :

– la définition de vos pages html qui contiennent un client web OMNIS ;
ce qui implique l'édition de pages html *template* créée pour vous par les *form wizards*

– l'installation de votre serveur web et de l'extension serveur web fournie avec OMNIS

– l'installation de votre runtime serveur OMNIS

– la copie de vos fichiers html et vos installeurs client sur votre serveur web ;
vous pouvez créer cependant vos propres installeurs si nécessaire

Modifier vos pages html

Le client web OMNIS est typiquement placé dans une page html à l'aide de n'importe quel éditeur html. Si aucun outil html n'est disponible, le client web peut être inséré à l'aide d'un éditeur de standard. Lorsque vous utilisez un *Wizard Remote form* pour créer vos *Remote forms*, OMNIS crée un fichier html *template* automatiquement et le place dans le dossier **html** du dossier principal OMNIS. Vous pouvez utiliser ces *templates* comme base de vos pages html ou copier l'ActiveX et ses paramètres et les placer sur vos propres pages html. Notez que les *templates* contiennent le code html pour l'ActiveX pour Internet Explorer et le plug-in Netscape. Ainsi, si vous utilisez IE seulement vous pouvez mettre en commentaires le code qui concerne le plug-in Netscape dans les pages html *template*. Netscape, de son côté, ignore le code ActiveX. Vous pouvez aussi utiliser le code ASP pour examiner le navigateur utilisé et afficher le client web approprié à vos clients.

Imbriquer le client web ActiveX

Le code html et paramètres du composant ActiveX sont les suivants.

```
<object classid=" clsid:13510606-30FA-11D2-B383-444553540000"
   width="500" height="250">
<param name="OmnisServer" value="ServerPortNumber">
<param name="OmnisLibrary" value="LibraryName">
<param name="OmnisClass" value="RemoteFormName">
<param name="WebServerUrl" value="www.yourwebserver.com">
<param name="WebServerScript" value="/cgi-bin/omnisapi.dll">
<param name="Param1" value="pOption1=Data">
 ...
<param name="Param9" value="pOption9=Data">
</object>
```

Le client web OMNIS ActiveX possède les paramètres suivants.

– **classid**
  l'ID de l'objet ActiveX ; comprend sa hauteur et sa largeur telle que définies selon la *Remote form*.

– **OmnisServer**
  le numéro de port du serveur ou le nom du service enregistré sur le serveur et spécifié dans la préférence $serverport d'OMNIS. Le paramètre *OmnisServer* peut être exprimé sous la forme AdresseIP:port, si le serveur OMNIS est lancé sur une machine différente de votre serveur web. Par exemple,
  “111.222.333.444:5912 “
  “111.222.333.444:omnis” ;; où "omnis" est le nom de service
  “www.myhost.com:5912”
  “www.myhost.com:omnis”

- **OmnisLibrary**
  le nom interne de la librairie à laquelle se connecter ; par défaut le nom de la librairie moins l'extension .lbs.
- **OmnisClass**
  le nom de la *Remote form* à laquelle se connecter.
- **WebServerUrl**
  l'URL de votre serveur web.
- **WebServerScript**
  l'emplacement de l'extension serveur web, comme omnisapi.dll, sur votre serveur web ; typiquement dans le dossier /cgi-bin.
- **Param1 à Param9**
  spécifie les paramètres supplémentaires, qui peuvent être envoyés au serveur OMNIS. Dans la clause de valeur, vous devez spécifier les noms de paramètres et les données séparées par le signe égal ("=").

Le code html suivant permet une connexion à une *Remote form* nommée **rfBooks** de l'exemple Books sur le serveur web de Mitford House, Principal centre de développement d'OMNIS Software.

```
<object classid="clsid:13510606-30FA-11D2-B383-444553540000"
   width="298" height="287">
 <param name="_Version" value="65536">
 <param name="_ExtentX" value="7161">
 <param name="_ExtentY" value="7373">
 <param name="_StockProps" value="0">
 <param name="OmnisServer" value="5912"> ;; le numéro de port
 <param name="OmnisLibrary" value="WEBCLIENT"> ;; le nom de la
   librairie
 <param name="OmnisClass" value="rfBooks"> ;; la remote form
 <param name="WebServerUrl" value="mhthinserver.mh.omnis-
   software.com"> ;; le nom du serveur web
 <param name="WebServerScript" value="/cgi-bin/omnisapi.dll">
    ;; l'emplacement de l'extension omnisapi.dll
</object>
```

Vous pouvez examiner les pages html *template* dans le dossier OMNIS\HTML pour voir comment le client web OMNIS est imbriqué. Les pages *template* comprennent déjà les paramètres numéro/nom, noms de librairie et de classe *Remote form* renseignés, mais vous devrez les modifier ou implémenter vos propres paramètres, comme **WebServerUrl** et **WebServerScript**, pour permettre l'accès distant à votre *Remote form*.

Imbriquer le Plug-in Client Web Netscape

Le code html et les paramètres pour le plug-in Netscape sont les suivants.

```
<EMBED  type=application/OMNIS-RCC-plugin
        name="rcc1"
        width=500
        height=500
        OmnisServer="ServerPortNumber"
        OmnisLibrary="LibraryName"
        OmnisClass="Remote FormName"
        WebServerUrl="www.yourdomainname.com"
        WebServerScript="/cgi-bin/omnisapi.dll">
```

– **OmnisServer**
le numéro de port du serveur ou le nom de service tel qu'il est enregistré sur le serveur et spécifié dans la préférence $serverport d'OMNIS.

– **OmnisLibrary**
le nom interne de la librairie à laquelle se connecter ; par défaut le nom de la librairie moins l'extension .lbs.

– **OmnisClass**
le nom de la *Remote form* à laquelle se connecter.

– **WebServerUrl**
l'URL de votre serveur web.

– **WebServerScript**
l'emplacement de l'extension serveur web, comme omnisapi.dll, sur votre serveur web ; typiquement dans le dossier /cgi-bin.

CGI

Si votre serveur web ne supporte pas l'interface MS ISAPI, vous devrez utiliser le programme cgi nph-omniscgi.exe. Dans ce cas, vous devez ajouter le programme cgi nph-omniscgi.exe à votre dossier cgi-bin et utiliser ce nom dans le paramètre **WebServerScript** de vos fichiers html.

Configuration de l'OMNIS Serveur

Le serveur OMNIS *ou* runtime est le programme qui utilise votre application OMNIS. Vous devrez installer votre runtime serveur OMNIS avec un numéro de série spécial sur une machine-serveur Win32. Notez que vous ne pouvez pas utiliser un numéro de série multi-utilisateur standard pour votre serveur OMNIS. Le serveur OMNIS n'a pas besoin d'être physiquement localisé sur votre machine-serveur, mais le serveur qui l'héberge doit impérativement être une machine Win32 (NT/95/98). Pour les tests et le debuggage, vous pouvez utiliser la version de développement d'OMNIS mais pour la distribution, vous devez utiliser un runtime OMNIS.

Définir le numéro de port OMNIS

Vous devez définir le numéro de port ou le nom de service de votre runtime serveur OMNIS. Le port doit être disponible sur le runtime serveur OMNIS et le serveur web. Cette valeur peut aller de 1 à 32767, mais vous ne devez pas utiliser la valeur 80 ou une autre valeur qui fait référence aux services e-mail, FTP ou autres. Utilisez une valeur élevée, de quatre ou cinq chiffre comme 5912. Le numéro de port OMNIS est stocké dans le fichier OMNIS.CFG du dossier STUDIO.

Dans la version de développement d'OMNIS vous pouvez définir le numéro de port dans les préférences OMNIS via l'option de menu **Tools>>Options/Preferences**. Cette option ouvre le *Property Manager* et affiche les préférences OMNIS. Vous pouvez définir le port serveur OMNIS via la propriété **serverport**. Cependant, comme vous allez utiliser votre solution web avec un runtime OMNIS , vous devez définir son numéro de port serveur aussi. Vous pouvez définir le numéro de port du serveur OMNIS via le dialogue de Configuration du serveur, disponible via l'option de menu **File>>Server Configuration** (visible uniquement avec les versions Win32 et seulement si le runtime est sérialisé avec un numéro de série serveur web). Le dialogue vous permet de saisir la valeur de port et le nombre de maximum de requêtes acceptées. Si le serveur OMNIS est déjà en action, il vous sera demandé de redémarrer OMNIS, sinon, le runtime OMNIS commence à scruter le port que vous lui avez indiqué.

Vous pouvez définir le numéro de port par notation, *Calculate $prefs.$serverport as 5912*, bien que vous deviez vous assurer que cette valeur correspond à celle spécifiée dans vos pages html qui contiennent le client web.

L'extension web serveur (omnisapi.dll) communique via TCP/IP avec OMNIS. Pour cette raison, votre serveur web et le serveur qui héberge votre runtime OMNIS requièrent WinSock.

Serveurs NT

Pour les serveurs NT seulement, un programme nommé NTSERV.EXE est fourni sur le CD Studio, qui installe un service NT à lancer sur le serveur OMNIS. Par défaut, le nom du service NT est *OMNIS Server*. Ne confondez pas le nom du service NT avec le nom de service TCP/IP que vous spécifiez dans la propriété $serverport pour la communication. Ils ne sont pas liés.

Si vous n'installez pas OMNIS comme service NT pendant l'installation, vous pouvez l'enregistrer en lançant le programme NTSERV.EXE avec la ligne de commande suivante :

```
NTSERV -install
```

Pour retirer le service OMNIS, utilisez la ligne de commande suivante :

```
NTSERV -remove
```

Pour lancer OMNIS comme application console pour le debuggage, utilisez la ligne de commande suivante :

```
NTSERV -debug <params>
```

Il est important lorsque vous utilisez NTSERV.exe que le fichier soit dans le même dossier que l'OMNIS.exe utilisé.

Configuration du Serveur web

Installation de l'extension serveur web

Vous devez installer l'extension serveur web OMNISAPI.DLL, fournie avec OMNIS, sur votre serveur web dans son dossier /cgi-bin. Pour vous connecter à votre extension serveur web, le runtime OMNIS et votre librairie OMNIS doivent être lancés sur le serveur. Le nombre possible de connexions concurrentes est spécifiée par le numéro de série web du runtime OMNIS.

De façon typique, un serveur web possède un emplacement pour stocker vos codes exécutables accessibles via HTTP à travers internet. Ce dossier est très souvent nommé /cgi-bin, mais ce n'est pas une contrainte. Vous devez placer votre extension serveur, telle que omnisapi.dll, dans ce dossier et vous assurez que le serveur web est configuré correctement afin qu'il tienne compte de ces modifications. Enfin, vous devrez ajouter le chemin d'accès à omnisapi.dll dans le paramètre de script *WebServerScript* de vos fichiers html, par exemple WebServerScript=/cgi-bin/omnisapi.dll.

Si votre serveur web ne supporte pas l'interface MS ISAPI, vous devrez utiliser le programme cgi nph-omniscgi.exe et modifier vos pages selon ces modifications.

Hébergement chez un FAI

Si votre site web est hébergé chez un tiers, typiquement votre Fournisseur d'Accès Internet (FAI), vous devrez placer votre extension serveur web OMNIS dans le dossier cgi-bin du

serveur du FAI. Votre FAI peut désirer tester votre extension serveur web, ce qui est la règle pour tous fichiers placés dans leur dossier cgi-bin et qui peut s'avérer coûteux. D'une autre manière, vous pouvez considérer la solution d'avoir votre serveur web “local” spécifiquement pour héberger votre solution web OMNIS et, si nécessaire, le lier à votre site web principal hébergé chez votre FAI.

Si votre solution web OMNIS utilise un site web hébergé chez un FAI, vous devrez ajuster vos réglages de port aussi bien au niveau du serveur OMNIS que de vos fichiers html. Dans ce cas, vous pouvez utiliser *DomainName:Port* ou *IPAddress:Port* dans vos réglages de port.

Sockets sécurisés

Vous pouvez utiliser des sockets sécurisés, si disponibles et si la page qui contient le client web OMNIS est sécurisée, ou si votre adresse *WebServerUrl* spécifie une URL complète qui comprend le préfixe https: comme suit :

https://UrlComplète.

Installeurs client web OMNIS

Pour utiliser votre solution web OMNIS, un utilisateur doit d'abord télécharger et installer le client web OMNIS. Vous pouvez fournir un lien sur votre site web pour permettre à l'utilisateur de télécharger l'installeur. Les installeurs pour ActiveX ou le plug-in Netscape sont fournis sur le CD OMNIS Studio et sont en distribution libre sur le site web OMNIS via la home page à http://www.omnis-software.com.

Créer vos propres installeurs

Dans la plupart des cas, vous pouvez utiliser les installeurs des clients web OMNIS tels quels. Par exemple, vous pouvez établir un lien avec la Home page du site web OMNIS afin que vos utilisateurs puissent télécharger la dernière version du client web, ou placer sur votre site web les liens appropriés. Si vous désirez créer votre propre installeur du client web OMNIS, soit pour minimiser les composants livrés en standard, soit pour ajouter vos propres composants web. Vous pouvez utiliser n'importe quel installeur du marché, comme InstallShield, pour créer vos propres installeurs de client web. Cette section décrit les composants web livrés par les installeurs OMNIS.

Composants du client web

Comme expliqué précédemment, le client web OMNIS est un contrôle ActiveX ou un plug-in Netscape. Ce sont de petits objets, qui se lient dynamiquement à des DLLs (Shared Libraries sur Macintosh). Si vous désirez créer un installeur pour le contrôle ActiveX et le plug-in Netscape, selon le navigateur supporté par vos pages/application, les éléments suivants composent le client web OMNIS :

– **Orfc.ocx** & **Orfc.dat**
  l'objet ActiveX

ou
np_orfc.dll & **np_orfc.dat**
le plug-in Netscape

– **Orfcgui.dll**
librairie qui fournie le GUI du client web (requis)

– **Orfcmain.dll**
le moteur du client web (requis)

Composants de *Remote form*

Les objets de classe *Remote form* sont des composants externes écrits spécialement pour les classes *Remote form*. Aucun autre objet OMNIS ne peut être placé dans une classe *Remote form*, à l'exception des *Page pane* standards. Les composants web sont contenus dans les installeurs du client web OMNIS fourni sur le CD OMNIS Studio, disponibles aussi sur le site web OMNIS. Si vous créez vos propres installeurs, vous devrez inclure les composants web, et omettre certains composants qui ne sont pas utilisés dans votre application.

Les composants de *form* sont livrés dans les librairies suivantes, chacune contenant un certain nombre de composants de classe *Remote form* :

– **Formback.dll**
contient l'objet *background Label* et les composants web *Border*.

– **Formflds.dll**
contient les composants *Push button*, *Single line entry field*, *Multi line entry field*, *Checkbox*, *Radio group*, *List box*, *Drop list*, *Combo box*, *Heading list* et *Picture web*.

– **Formsbar.dll**
contient le contrôle *Sidebar*.

– **Formtbar.dll**
contient le contrôle *Tab bar*.

Chapitre 6 – Problèmes et réponses

Ce chapitre contient des notes et réponses aux questions les plus fréquentes qui résolvent la plus grande partie des soucis que vous pouvez rencontrer lors de la mise en place de votre client web OMNIS.

Général

Q : J'ai une application OMNIS Studio existante. Dois-je recommencer depuis le début pour bénéficier des avantages du client web OMNIS, ou puis-je adapter mon application existante ?
A : Oui, vous pouvez adapter votre application existante pour bénéficier du client web OMNIS et lui ajouter une interface web. Développer une solution web OMNIS avec le client web OMNIS est exactement la même chose que développer une application OMNIS. De plus, si vous avez une application OMNIS 7, vous pouvez la convertir en OMNIS Studio et l'adapter pour un accès internet avec le client web.

Remote forms

Q : J'ai ajouté une méthode de gestion d'*event* derrière un bouton, mais lorsque je teste ma *form* et clique sur le bouton, rien ne se passe, la méthode semble ne pas fonctionner. Pourquoi ?
A : C'est probablement parce que vous n'avez pas activé l'*event* evClick pour le bouton. Par défaut, tous les *events* de tous les contrôles de *form* sont désactivés ; vous devez activer individuellement les *events* de chaque objet via leur propriété *events*.

Q : Dois-je me préoccuper des utilisateurs Windows qui emploient des polices de petite taille ou de grande taille ?
A : Non, les objets de *Remote form* n'utilisent que des polices TrueType, et le navigateur client mettra les polices à l'échelle en mode runtime selon la définition de la machine client.

Q : Je veux placer le focus sur une rubrique particulière lors de l'ouverture de ma *form*. Comment faire ?
A : Spécifiez le numéro de cette rubrique (dans la propriété **order** de la rubrique), dans la propriété **startfield** de la *Remote form*.

Navigateur

Q : J'ai créé ma *Remote form*, mais lorsque j'appuie sur Ctrl/Cmnd-T mon navigateur web n'apparaît pas et je ne peux pas voir ma *form*. Que se passe-t'il ?
A : Avez-vous installer le client web ? Vous devez installer le client web OMNIS pour créer et tester vos *Remote forms* avec la version de développement d'OMNIS. Si le client web n'est pas installé et que vous appuyez sur Ctrl/Cmnd-T pour tester votre *form*, votre navigateur web n'apparaîtra pas. Pour remédier à cette situation, installez le client web à l'aide de l'un des installeurs fournis sur le CD OMNIS Studio. Si vous avez installé le client web et que votre *form* n'apparaît toujours pas, c'est peut-être parce que votre ActiveX n'est pas enregistré correctement. L'installeur enregistre automatiquement l'ActiveX, mais s'il n'a pas réussi à le faire, pour différentes raisons, vous pouvez l'enregistrer manuellement avec le programme REGSVR32.EXE. Sous Windows, ouvrez le dialogue Exécuter du bouton Démarrer, et saisissez la commande suivante, en plaçant le chemin correct de l'.OCX :

```
regsvr32.exe c:\windows\Webclient\Orfc.ocx
```

Q : Ctrl/Cmnd-T affiche deux *forms* avec Internet Explorer. Que se passe-t'il ?
A : Si vous possédez Netscape Navigator sur votre machine et Internet Explorer (IE) et que vous avez installé l'ActiveX et le plug-in Netscape, IE essaiera d'utiliser le plug-in Netscape depuis Navigator et affichera les deux objets. Ctrl/Cmnd-T utilise un fichier *template* dans le dossier HTML pour créer une page de test html. Ce fichier possède le code source pour les objets Netscape et Explorer. Pour empêcher ce comportement, retirez le source pour ActiveX ou Netscape du fichier ***template.htm***, en détruisant les fichiers html créés par Ctrl-T dans le dossier HTML (ne détruisez pas *template.htm*), puis re-testez votre *form* avec Ctrl/Cmnd-T.

Q : Pourquoi j'obtiens le message “unable to locate class” ?
A : Soit le nom de votre librairie OMNIS et le paramètre OmnisLibrary de votre fichier html ne correspondent pas, soit votre librairie n'est pas ouverte ; le runtime serveur OMNIS et votre librairie doivent être lancés en permanence pour permettre à vos clients d'accéder à votre application.

Q : Pourquoi un message d'alerte d'IE apparaît lorsque j'appuie sur Ctrl/Cmnd-T?
A : Sans doute parce que vous utilisez une ancienne version béta du client web OMNIS. Dans ce cas, vous devez mettre à jour votre client web OMNIS.

OMNIS Server

Q : Puis-je définir la propriété serverport d'OMNIS à 80 ?
A : Non ! OMNIS ne doit pas utiliser le port 80, ou n'importe quelle valeur ou nom de service déjà utilisé. OMNIS nécessite les services d'un serveur web standard. De ce fait, définissez le port OMNIS avec une valeur relativement élevée, qui contiendra au moins quatre chiffres, à moins que votre département informatique vous alloue une valeur. N'oubliez pas, le numéro de port serveur d'OMNIS doit être le même que celui spécifié dans le paramètre OmnisServer de vos fichiers html.

Q : Puis-je sérialiser le runtime OMNIS pour ma solution web avec mon numéro de série multi-utilisateur actuel ?
A : Non. Vous devez sérialiser le runtime serveur OMNIS avec un numéro web spécial. Ces numéros contiennent un “W” et permettent un nombre spécifié d'utilisateurs simultanés. Contactez Aware pour obtenir ces numéros.

Q : Pourquoi obtiens-je le message “server busy” ?
A : Il y a trop d'utilisateurs en cours, le serveur ne peut accepter d'autres connexions. Vous pouvez aussi obtenir ce message avec la version de développement si la connexion vers votre navigateur n'est pas close correctement. Dans ce cas, OMNIS pense que d'autres utilisateurs sont toujours connectés. De telles connexions peuvent éventuellement se terminer sur un timeout, mais pour éviter cela, fermez et réouvrez votre librairie.

Chapitre 7 – Référence

Ce chapitre liste les *events* et notations du client web OMNIS, ainsi que les notations pour les *Remote forms* et les *Remote tasks*.

Events

Classes *Remote task*

Event	**Description**
evBusy	envoyé lorsqu'OMNIS a reçu une requête du client
evIdle	envoyé lorsqu'OMNIS s'apprête à retourner le résultat de la requête
evRejected	généré lorsqu'OMNIS rejette une connexion cliente ; habituellement si trop d'utilisateurs essaient de se connecter à OMNIS, ou si la valeur de $maxusers de la tâche est dépassée.

Objets de *Remote form*

Tous les objets de *Remote form* peuvent envoyer les *events* evBefore et evAfter s'ils sont activés dans la propriété $events. De plus, les objets comme les *pushbutton* et la plupart des types liste envoient les *events* evClick et evDoubleClick. Certains objets possèdent leurs propriétés spécifiques comme les *Sidebar*.

Général

Events	**Description**
evBefore	envoyé lorsque l'utilisateur s'apprête à entrer dans une rubrique de *form*
evAfter	envoyé lorsque l'utilisateur s'apprête à quitter une rubrique de *form*
evClick	envoyé lorsque l'utilisateur a cliqué une rubrique de *form*
evDoubleClick	envoyé lorsque l'utilisateur a double-cliqué sur une rubrique de *form*

Sidebar

Events	Description
evIconPicked	envoyé lorsqu'une icône de *Sidebar* est choisie
evSetPicked	envoyé lorsqu'une *icon strip* ou un *group bar* est sélectionné
evSetBeingPicked	envoyé lorsqu'une icône de *Sidebar* va être choisie

Notation

Préférences OMNIS

Vous pouvez définir les préférences OMNIS en utilisant l'option de menu **Tools>>Options/Preferences** de la barre de menus principale d'OMNIS ou via une expression de notation dans le groupe $root.$prefs.

Propriétés

Propriété	Description
$serverport	spécifie le port serveur sur lequel OMNIS surveille les connexions ; nombre entre 1 et 32767, ou nom de service ; il ne doit pas être défini à 80 ou un autre nombre ou nom de service déjà utilisé. La définition de port serveur dans OMNIS doit correspondre au nombre ou au nom spécifié dans le paramètre OmnisServer de vos fichiers html.
$servermaxrquests	le nombre maximum de connexions concurrentes sur le serveur OMNIS ; nombre compris entre 1 et 511.
$webbrowser	le nom et le chemin du serveur web qui peut être utilisé pour tester les *Remote forms* en local ; laissé à vide pour utiliser le navigateur par défaut. Spécifie aussi le navigateur utilisé pour le système d'aide OMNIS si aucun navigateur compatible ActiveX n'est disponible.

Méthodes

Méthodes	Description
$clearcachedforms()	efface toutes les *Remote forms* stockées dans le cache du serveur OMNIS
$promptforicon()	ouvre le dialogue de sélection d'icône ; implémenté pour les *templates* et les *wizards* du client web : utilisé seulement en développement, vous ne devez pas utiliser cette méthode en mode runtime dans une solution web distribuée.

Classes *Remote form*

Propriétés

Les *Remote forms* possèdent en standard les propriétés suivantes :

Propriétés	Description
$designtaskname	nom de *Remote task* ; vous devez spécifier un nom de classe *Remote task*. Les *Remote forms* ne peuvent pas fonctionner sans classe *Remote task*.
$iconpages	spécifie une liste d'*icon pages* requise pour la classe, elles sont envoyées au client avec les données de classe. Vous pouvez ajouter des icon pages avec $iconpages.$add(iconID). OMNIS ajoutera la *page* de l'*icon* à $iconpages si elle n'est pas déjà spécifiée. OMNIS n'envoie pas d'icônes spécifiques au client, il envoie des pages complète. De ce fait, vous devez choisir des icônes sur un minimum de pages possible, ou placer toutes les icônes d'une *form* sur une seule page. Les *icons pages* sont séparées par des virgules et les pages qui proviennent de la table système #ICONS, du fichier OMNISPIC ou du fichier USERPIC (dans cet ordre) sont séparées par des points-virgules. Par exemple, une définition d'*iconpages* pourrait être : “Books,General;;Embedded Colors” spécifiant deux pages de la table #ICONS de la librairie en cours et une page du fichier de données USERPIC.

Méthodes

Les classes *Remote form* possèdent en standard les méthodes de classe $construct, $destruct et $redraw. Notez que la méthode $open() n'existe que pour le développement et ne doit pas être utilisée en mode runtime.

Méthodes	Description
$open()	pour l'environnement de développement seulement : non supporté en mode runtime. $open() vous permet de tester vos *forms* pendant que vous debuggez votre librairie ; action identique à Ctrl/Cmnd-T. Lorsque la méthode est exécutée, OMNIS ouvre la page html associée à la *Remote form* dans votre navigateur local. La page html doit être placée dans le dossier **html** du dossier OMNIS, et possède le même nom que la *Remote form* suivi de ".html". Lors de la création des *Remote forms* avec les *wizards*, les pages html sont créés automatiquement pour vous. Vous pouvez spécifier une page html alternative dans le paramètre 1, auquel cas, le chemin et le nom complet de la page doivent être fournis.
$canclose()	retourne toujours kFalse. Vous ne pouvez pas fermer directement une *Remote form*. Vous devez fermer sa *Remote task.*
$senddata()	$senddata(iVar1,iVar2,..) spécifie les variables d'instance à retourner au client à la fin de l'exécution de l'*event* sur le serveur.
$setcurfield()	$setcurfield(NuméroRub|Nom|Ident) défini la rubrique en cours sur la machine cliente. Vous pouvez spécifier le nom de l'objet, son numéro ou son ident.
$showmessage()	$showmessage('Message','Titre') affiche un message Ok sur la machine client avec le message et le titre spécifié. Un seul message peut être affiché à chaque communication. L'exécution de $showmessage() une seconde fois pendant la même communication remplacera le texte du message.
$showurl()	$showurl('URL','NomDeFrame') affiche la page html spécifiée dans une nouvelle fenêtre ou un frame de la machine cliente. Le premier paramètre spécifie l'URL de la page html. Le second paramètre spécifie le nom du frame. S'il est laissé vide, la page sera affichée dans une nouvelle fenêtre, sinon elle est affichée dans le frame spécifié de la fenêtre.

Objets de *Remote form*

Propriétés générales

Les objets de *Remote form* possèdent les propriétés standards des objets de classes *window*. Notez que tous les objets de *Remote form* sont des composants externes.

Propriétés	Description
$componentlib	le nom de la librairie composant ; nom de librairie ; l'un des suivants : Formback, Formflds, Formsbar ou Formtbar
$componentctrl	le type de contrôle
$objtype	composant externe de type kComponent
$events	spécifie les *events* activés pour l'objet ; seuls ces *events* sont envoyés au serveur
$dataname	Ne doit spécifier que des variables d'instance

Objets *Combo Box*

Propriété	Description
$dataname	le *dataname* de la rubrique de saisie de la combo box
$listname	variable liste de la liste de la combo box

Objets *Radio Group*

Propriété	Description
$horizontal	si vrai, les *radio buttons* sont dessinés horizontalement, de gauche à droite.
$columncount	spécifie le nombre de colonne de *Radio buttons* (par défaut 1). 9 *Radio buttons* et 3 colonnes afficheront une grille de boutons présentés 3x3. la largeur de chaque colonne est déterminée par la longueur du texte.
$minvalue	valeur entière du premier *button* (par défaut zéro)
$maxvalue	valeur entière du dernier *radio button* (par défaut 2). Ensemble, la valeur des propriétés $minvalue et $maxvalue détermine le nombre de *radio buttons*.

Objet Heading List

Propriété	Description
$colcount	nombre de colonnes à afficher.
$columnnames	liste au format "séparé par virgule" de noms de colonne.
$columnwidths	liste au format "séparé par virgule" des largeurs de colonne exprimées en pixels.

Objet *Sidebar*

Propriété	Description
$fillcolor	couleur de remplissage du fond du contrôle *Sidebar*
$flipswitch	contrôle si les *icon bars* "glissent" lorsqu'elles sont cliquées ; kFalse par défaut ce qui indique qu'elles "glissent"
$labelcolor	couleur du label de l'icône
$selectedlabelcolor	couleur du label de l'icône sélectionnée
$selectcurrent	si vrai, l'icône en cours est surlignée
$currenticon	spécifie l'icône sélectionnée dans le groupe en cours
$currentset	spécifie le groupe sélectionné en cours
$show3dundermouse	si vrai, les icônes sont dessinées en 3d si elles sont sous la souris
$showiconnames	si vrai, les noms d'icônes sont affichés
$buttonfillcolor	la couleur de remplissage du bouton
$washstartcolor	la couleur de départ pour l'effet *wash* (dégradé de couleurs) du contrôle *Sidebar* ; visible si *washstrip* est activé
$washendcolor	la couleur de fin pour l'effet *wash* du contrôle *Sidebar* ; visible si *washstrip* est activé
$washdirection	la direction pour l'effet *wash* du contrôle *Sidebar* ; visible si *washstrip* est activé
$tilebmp	l'icône répliquée en fond de *SideBar* ; visible si *tilestrip* est activé
$tilestrip	si vrai, le fond du *Sidebar* est répliqué
$washstrip	si vrai, le fond du *Sidebar* affiche son effet wash (dégradé de couleurs)

Objet *Tab bar*

Propriété	Description
$nosofttab	nombre de *tabs*(onglets)
$currenttab	le *tab* en cours
$tabtext	le texte du *tab* en cours
$tabtip	le texte d'aide (*tip text*) du *tab* en cours
$position	la position ou l'orientation des *tabs*, soit *Top*, soit *Bottom*, soit *Left*, soit *Right*
$style	le style des *tabs* ; une constant : kDefaultWebTab, kSquareWebTab, kTriangleWebTab

Méthodes

Les objets de *Remote form* contiennent la méthode $redraw(), que vous pouvez exécuter avec la notation NomObjet.$redraw() ou $cobj.$redraw().

Classes *Remote task*

Propriétés

Propriété	Description
$maxtime	le nombre de minutes maximum autorisées à l'utilisateur ; si définie à zéro, la propriété n'énonce aucune limite.
$timeout	le nombre de minutes, qui peuvent s'écouler sans actions de la part de l'utilisateur ; si définie à zéro, la propriété n'énonce aucune limite.
$maxusers	nombre maximum de clients qui peuvent se connecter simultanément à cette tâche ; si définie à zéro, la propriété n'énonce aucune limite.

Méthodes

Méthodes	Description
$construct()	appelée lorsqu'un client se connecte à la tâche. Le premier paramètre contient une variable *row*. Cette variable *row* contient les paramètres de connexion.
$destruct()	appelée lorsque le client se déconnecte.
$open()	En mode création seulement : non supporté en mode runtime. $open vous permet de tester vos *Remote tasks* pour le débuggage de votre librairie. Lorsqu'elle est exécutée, OMNIS ouvre la page html associée à la *Remote task* dans le navigateur local. La page html doit être située dans le dossier **html** du dossier principal d'OMNIS et doit avoir le même nom que la *Remote task* suivie de ".html". Lors de la création d'une *Remote tasks* avec les *wizards* fournis, les pages html sont créées automatiquement. Vous pouvez spécifier une page html alternative dans le paramètre 1, auquel cas, vous devrez fournir le chemin et le nom complet du fichier alternatif.
$close()	l'exécution de $close force le serveur OMNIS à déconnecter le client et fermer toutes les instances de *form*, avant de fermer l'instance de *task*.
$event()	gère les *events* de l'instance de task : les *events* suivants sont envoyés pendant la connexion via le client web OMNIS. evBusy - envoyés lorsqu'OMNIS a reçu une requête du client. evIdle - envoyé lorsqu'OMNIS s'apprête à retourner le résultat de la requête. evRejected - envoyé lorsqu'OMNIS rejette une connexion client. Note : A la réception de l'*event* evBusy, vous pouvez éviter son traitement en le rejetant (*discard even*t). Dans ce cas, le client recevra un message “Server busy”.

Instance de *Remote task*

Propriétés

Propriété	Description
$ident	ID unique entre 1 et 2^31. Réinitialisé lorsqu'OMNIS est quitté ou à la déconnexion du dernier client
$order	valeur d'index de l'instance de *task*. (Unique seulement pour les instances de la même classe *task*, chaque classe *task* maintient ses propres instances)
$maxtime	temps maximum (en minutes), mesuré depuis $connectiontime, autorisé pour la connexion de l'utilisateur.
$timeout	temps maximum (en minutes), mesuré depuis $lastresponse, autorisé pour la connexion de l'utilisateur.
$clientaddress	adresse TCP/IP du client
$connectiontime	la date et l'heure de la connexion de l'utilisateur
$lastresponse	la date et l'heure de la dernière communication du client avec le serveur
$connectbytessent	spécifie le nombre d'octets envoyés au client pendant la connexion. Cette propriété est définie après l'exécution de $construct.
$requests	spécifie le nombre d'*events* exécuté sur le serveur. Sont exclus les messages de connexion et de déconnexion. Mis à jour avant le message evBusy.
$reqtotbytesreceived	nombre d'octets reçus du client pour toutes les requêtes. Pour calculer une moyenne par requête, vous pouvez diviser cette valeur par $requests. Mis à jour avant le message evBusy.
$reqtotbytessent	nombre total d'octets envoyé au client pour toutes les requêtes. Pour calculer une moyenne par requête, vous pouvez diviser cette valeur par by $requests. Mis à jour avant le message evIdle.
$reqmaxbytesreceived	le bloc le plus large, en octets, reçu du client pour toutes les requêtes. Mis à jour avant le message evBusy.
$reqmaxbytessent	le bloc le plus large, en octets, envoyé au client pour toutes les requêtes. Mis à jour avant le message evIdle.
$reqcurbytesreceived	nombre d'octets reçus depuis le client pour la requête en cours. Mis à jour avant le message evBusy pour la requête en cours.
$reqcurbytessent	nombre d'octets reçus depuis le client pour la requête en cours. Mis à jour avant le message evIdle.

Index

Glossaire Anglais-Français

A

Add-on tool	Outil extension - Placé dans le menu *Tools*

B

Browser	Navigateur, interface de navigation – Ancien "Tableau de Bord"

C

Catalog	Catalogue
Class Browser	Navigateur de Classes
CGI	*Common Graphic Interface* – Interface graphique commune
CGI-encoded	Encodé par un moteur CGI
Checkbox	Boite à cocher
Column	Colonne
Component	Composant
Component Store	Galerie de Composants "Container"
Container	Rubrique qui contient d'autres rubriques

D

DAM	Data Access Manager – Gestionnaire d'accès aux données
Dataname	Nom de la rubrique de données
Design mode	Mode création/développement
Driver	Pilote (de gestion de données)

E

Event	Evénement

F

Field	OMNIS : Rubrique – HTML : Champ
Form	Fenêtre "formulaire"
Form wizard	Assistant de création de *form*

G

Grid	Grille, table, tableau

H

Header	En-tête
Heading List	Liste à en-tête
Height	Hauteur
Host	Hôte

I

IconID	Numéro d'identification de l'icône
ID	Numéro d'identification
IDE	*Interface Development Environment* - Interface de l'Environnement de Développement
Item	Elément

J

K

L

Label	Libellé
Length	Longueur

M

Method editor	Editeur de méthodes

N

Next	Suivant
Node	Noeud
Notation Inspector	Inspecteur de notation

O

P

Page pane	Volet
Picture	Image
Previous	Précédent
Property Manager	Gestionnaire de propriétés

Q

R

Remote form (classe)	Formulaires à usage distants (via le web)
Remote task (classe)	Tâche de gestion d'évènements distants (via le web)
Report (classe)	Etat

S

Select	Sélectionner
Sidebar	Menu graphique
Size	Taille
Stream	Flux (de données)

Subclass ..... Sous-classe

T

Tab ..... Onglet
Task ..... Tâche
Tip Text ..... Vignette d'aide/Info-bulle
Title ..... Titre
Tool ..... Outil
Toolbar (classe) ..... Barre d'outils
Tree ..... Arborescence
True color ..... Format d'image 24 bit (

U

V

VCS ..... Version Control System — Système de Contrôle des Versions

W

Width ..... Largeur
Window (classe) ..... Fenêtre
Wizard ..... Assistant

X

Y

Z