



Projet Gestion d'un Carnet de Rendez-vous

Structures de Données



Gestion d'un carnet de rendez-vous

Travail à faire :

- Créer un ensemble de classes pour gérer un carnet de rendez-vous. Le code ne doit pas utiliser de bibliothèque graphique
- Créer une interface utilisateur (graphique ou console) pour pouvoir afficher et manipuler ce carnet de rendez-vous



Gestion d'un carnet de rendez-vous

Travail à faire :

- Groupes de 4 étudiants
- Durée : 6 semaines
- Rapport à rédiger :
 - Expliquer le fonctionnement de l'interface.
 - Expliquer l'organisation du programme (organigramme, schéma)
 - Expliquer les choix qui ont été faits au niveau de l'organisation du programme
 - Définir la contribution de chacun des étudiants (qui a fait quoi ?)



Gestion d'un carnet de rendez-vous

Stocker des informations de plusieurs types :

- Une structure de données pour les personnes qui permet d'enregistrer les informations: nom prénom téléphone personnel et adresse email
- Une structure de données rendez-vous qui permet d'enregistrer un rendez-vous entre plusieurs personnes.



Gestion d'un carnet de rendez-vous

Les informations :

- Pour chaque personne
 - Nom
 - Prénom
 - Téléphone portable
 - Adresse de courrier électronique
- Pour chaque rendez-vous
 - Nom du rendez-vous
 - Date
 - Heures de début et de fin
 - Les personnes qui assistent à ce rendez-vous



Gestion d'un carnet de rendez-vous

Les règles :

- Les informations pour les personnes et les rendez-vous sont dans deux structures de données différentes.
- Une personne ne peut pas être présente à plusieurs rendez-vous au même moment.



Gestion d'un carnet de rendez-vous

Les manipulations du carnet de rendez-vous :

- Ajout d'une personne
 - Spécifier le nom de famille
 - Prénom
 - Numéro de téléphone portable
 - Adresse de courrier électronique



Gestion d'un carnet de rendez-vous

Les manipulations du carnet de rendez-vous :

- Ajout d'un rendez-vous
 - Spécifier le nom du rendez-vous
 - Date
 - Heure de début et heure de fin
 - Liste des personnes
 - Attention : Une personne ne peut pas être présente à plusieurs rendez-vous au même moment. Afficher un message si c'est le cas.



Gestion d'un carnet de rendez-vous

Les manipulations du carnet de rendez-vous :

- Suppression d'une personne
 - Spécifier le nom et prénom de la personne
 - Si la personne n'a aucun rendez-vous, la personne est supprimée. Sinon, un message d'erreur est envoyé.
- Suppression d'un rendez-vous
 - Spécifier le nom du rendez-vous
 - Le rendez-vous est supprimé



Gestion d'un carnet de rendez-vous

Les manipulations du carnet de rendez-vous :

- Modification d'une personne
 - Spécifier le nom de la famille et le prénom de la personne
 - Numéro de téléphone mobile et adresse de courrier électronique peuvent être modifiés
- Modification d'un rendez-vous
 - Spécifier le nom du rendez-vous
 - Modifier la date, les horaires et la liste des personnes.
 - **Attention : Une personne ne peut pas être présente à plusieurs rendez-vous au même moment. Si c'est le cas, afficher un message d'erreur.**



Gestion d'un carnet de rendez-vous

Les manipulations du carnet de rendez-vous :

- Afficher tous les rendez-vous pour une date donnée par l'utilisateur.
- Afficher tous les rendez-vous pour une personne dont le nom est donné par l'utilisateur.
- Déterminer si une personne n'a pas de rendez-vous pour une date et un horaire donnés par l'utilisateur.
- Afficher pour un rendez-vous les noms, prénoms, numéros de téléphone et adresses de courrier électronique de toutes les personnes assistant à un rendez-vous.



Gestion d'un carnet de rendez-vous

Les structures données :

- Une structure de données pour les personnes avec les champs pour :
 - Nom
 - Prénom
 - Téléphone mobile
 - Adresse de courrier électronique
- Une structure de données pour les rendez-vous :
 - Nom du rendez-vous
 - Date
 - Heures de début et de fin
 - Les personnes qui assistent à ce rendez-vous



Gestion d'un carnet de rendez-vous

Exemple de structures de données :

- Listes chaînées avec les fonctions suivantes :
 - Chaque élément a une chaîne de caractères comme identificateur
 - Insérer un élément avec comme paramètre le numéro de l'élément
 - Insérer un élément avec comme paramètre la chaîne de caractères, l'élément doit être inséré de façon à respecter l'ordre alphabétique des éléments
 - Supprimer un élément avec comme paramètre le numéro de l'élément
 - Supprimer un élément avec comme paramètre la chaîne de caractères
 - Chercher un élément avec comme paramètre la chaîne de caractères
 - Retourner le nombre d'éléments



Gestion d'un carnet de rendez-vous

Les méthodes privées :

- Insérer un élément avec comme paramètre le numéro de l'élément
- Supprimer un élément avec comme paramètre le numéro de l'élément
- Chercher un élément avec comme paramètre la chaîne de caractères

Ces structures de données seront utilisées pour les personnes et les rendez-vous. Tous les éléments sont classés par ordre alphabétique. Classement suivant les noms seulement pour les personnes.



Gestion d'un carnet de rendez-vous

Remarques importantes :

- Bien séparer le code de la gestion du carnet de rendez-vous et de l'interface utilisateur (graphique ou console)
- La mémoire allouée doit être gérée correctement, en particulier, dans le cas de la suppression d'éléments
- Le code doit être optimisé (boucles, parcours des tableaux, allocation et libération de la mémoire)



Gestion d'un carnet de rendez-vous

Interface graphique basée sur les MFC:

- Bibliothèque qui permet de construire une interface graphique avec Visual Studio
- Création de l'interface avec la souris (Création des ressources)
- Écrire le code pour définir les fonctionnalités des éléments de l'interface.



Gestion d'un carnet de rendez-vous

Une application boîte de dialogue :

- Ne comporte qu'une seule fenêtre
- N'utilise pas l'architecture document vue
- Ne permet pas en standard la gestion d'une barre d'outils
- Simple à mettre en œuvre : une boîte de dialogue permet de développer très vite une petite application avec une interface minimaliste pouvant convenir pour de petits programmes utilitaires.
- D'autres modèles d'application sont possibles.

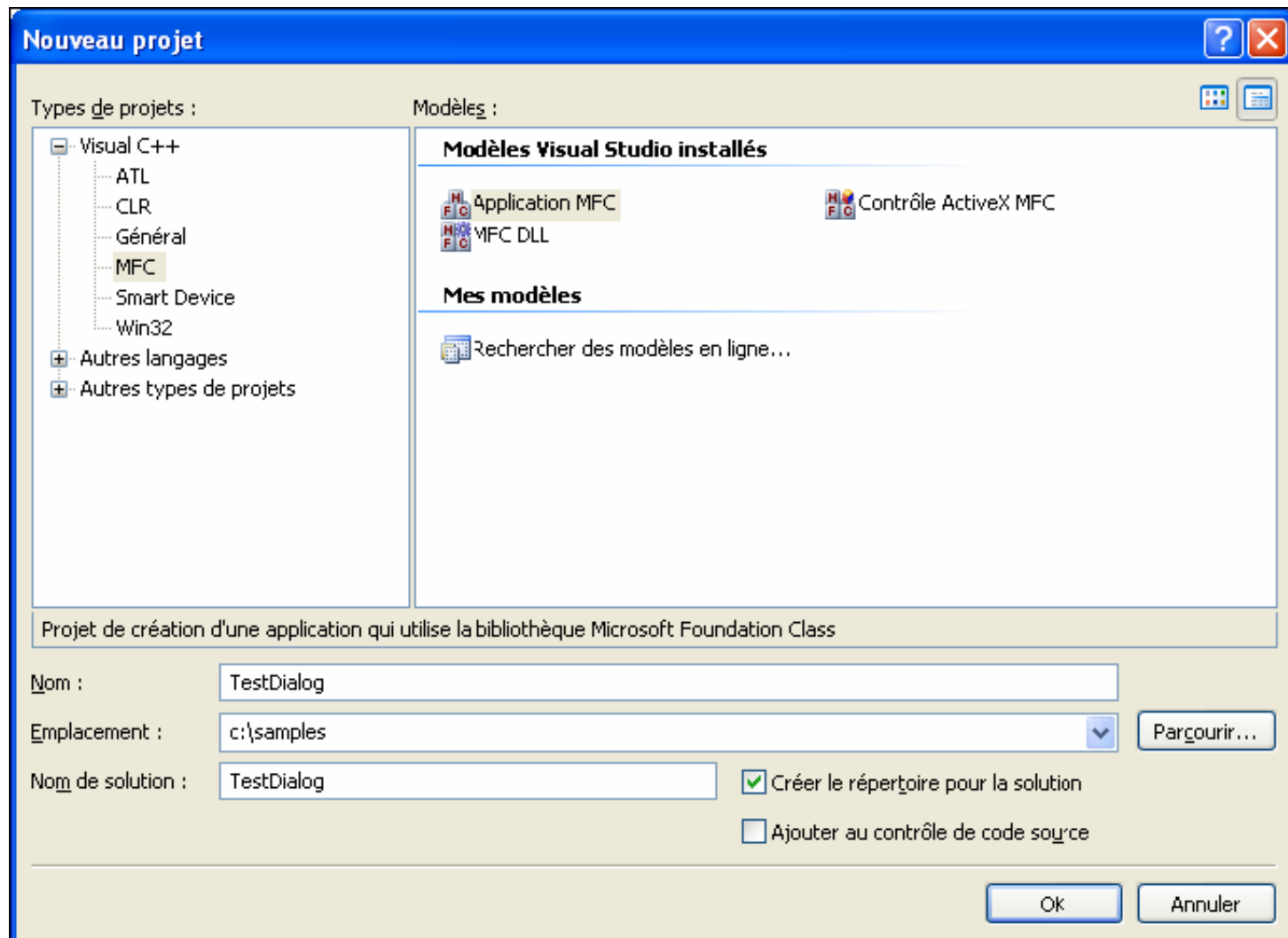


Gestion d'un carnet de rendez-vous

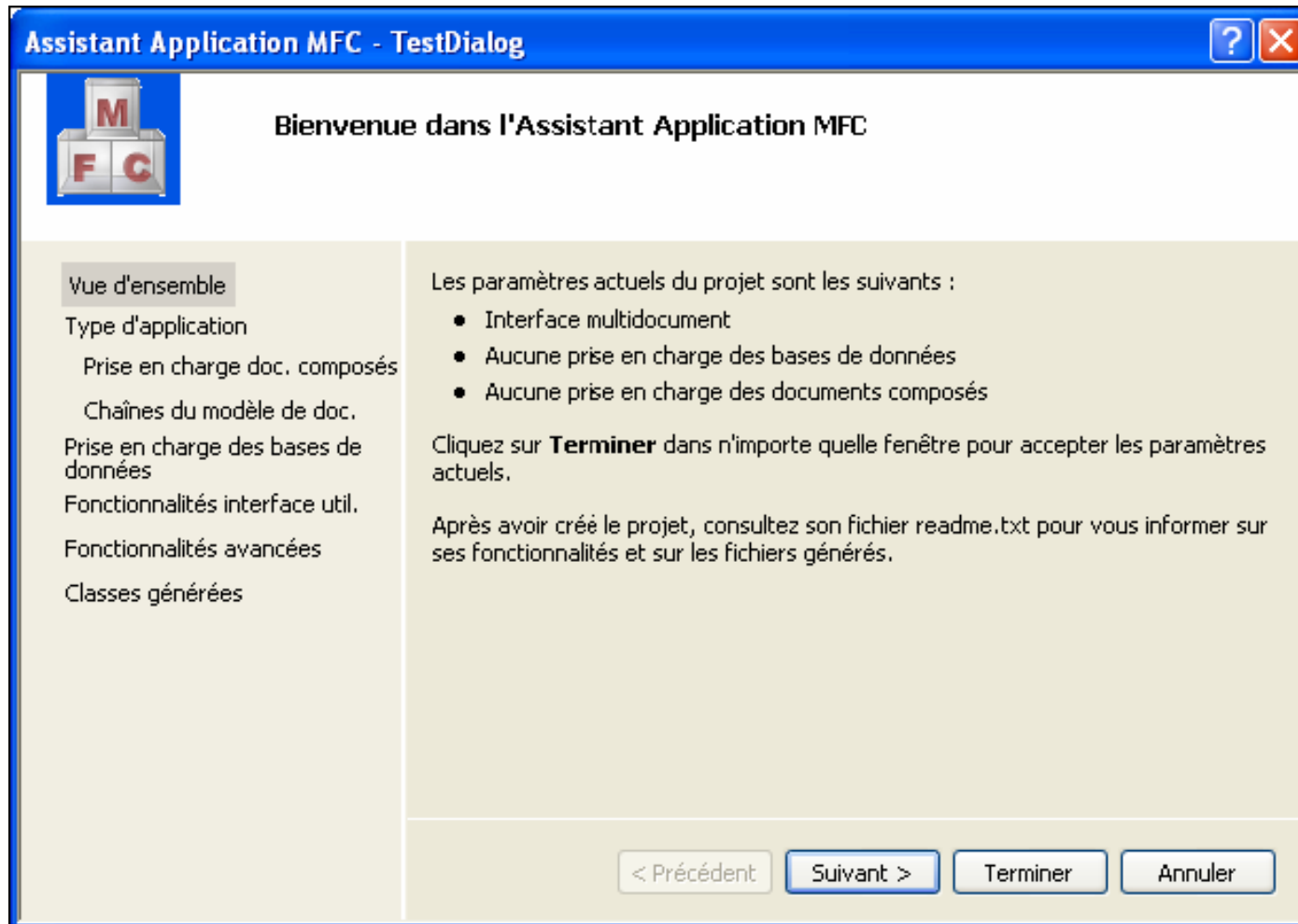
Une application boîte de dialogue :

- Créer un projet :
 - Avec Visual 2005 sélectionnez le menu File new project sélectionnez le template MFC Application, remplissez le nom et le répertoire de destination du projet.

Gestion d'un carnet de rendez-vous



Gestion d'un carnet de rendez-vous

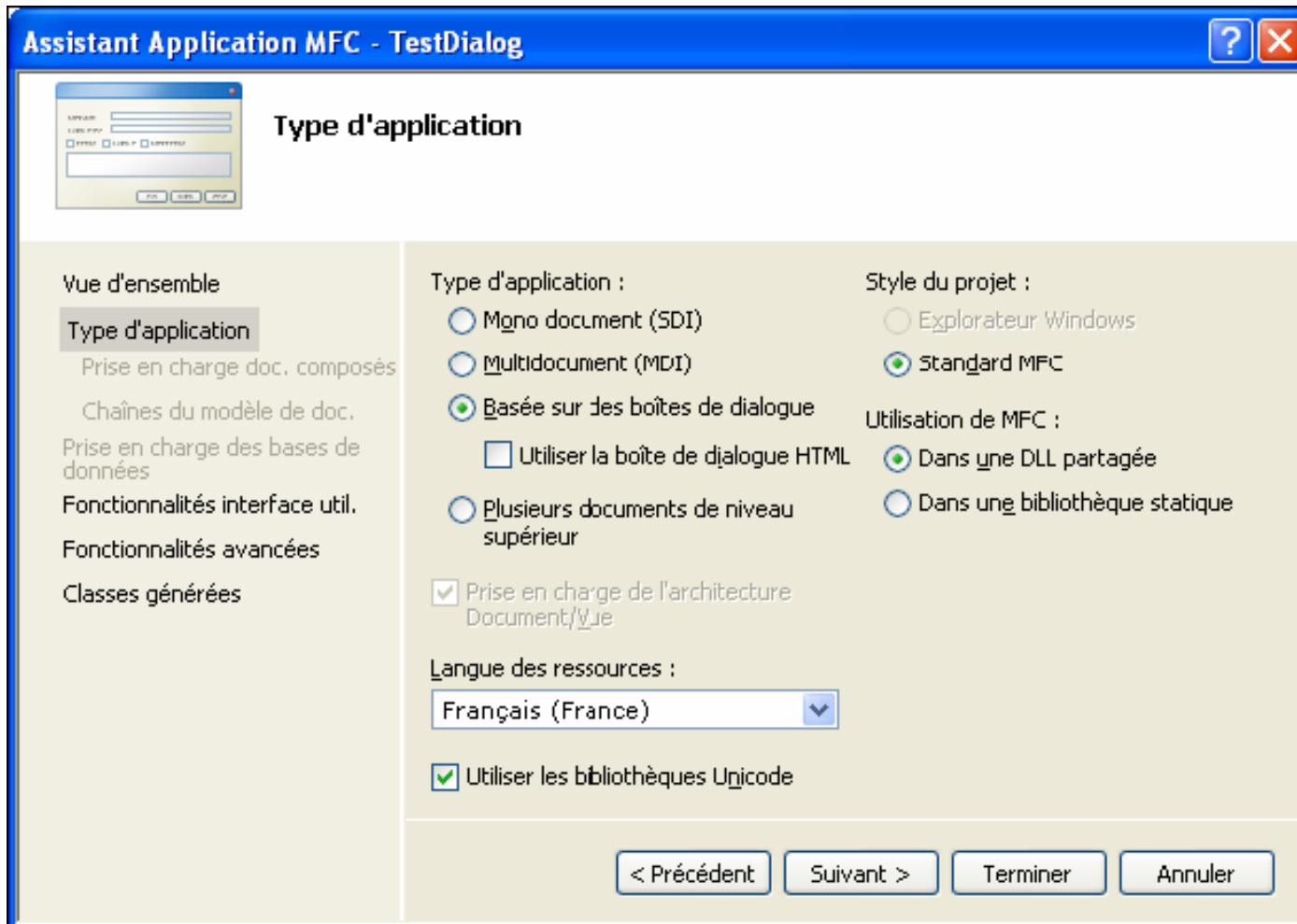




Gestion d'un carnet de rendez-vous

- Sélectionnez la ligne Type d'application
- Puis les options :
 - Basée sur des boîtes de dialogue
 - Langue de ressources : Français.

Gestion d'un carnet de rendez-vous



The screenshot shows the 'Assistant Application MFC - TestDialog' window, specifically the 'Type d'application' step. The window has a blue title bar with a question mark and a close button. On the left, there is a 'Vue d'ensemble' (Overview) pane with a tree view containing: 'Type d'application' (selected), 'Prise en charge doc. composés', 'Chaînes du modèle de doc.', 'Prise en charge des bases de données', 'Fonctionnalités interface util.', 'Fonctionnalités avancées', and 'Classes générées'. The main area is divided into three sections: 'Type d'application :', 'Style du projet :', and 'Utilisation de MFC :'. The 'Type d'application' section has radio buttons for 'Mono document (SDI)', 'Multidocument (MDI)', 'Basée sur des boîtes de dialogue' (selected), and 'Plusieurs documents de niveau supérieur'. There is also a checkbox for 'Utiliser la boîte de dialogue HTML'. The 'Style du projet' section has radio buttons for 'Explorateur Windows' and 'Standard MFC' (selected). The 'Utilisation de MFC' section has radio buttons for 'Dans une DLL partagée' (selected) and 'Dans une bibliothèque statique'. At the bottom, there is a 'Langue des ressources' dropdown set to 'Français (France)' and a checked checkbox for 'Utiliser les bibliothèques Unicode'. Navigation buttons at the bottom right are '< Précédent', 'Suivant >', 'Terminer', and 'Annuler'.

Assistant Application MFC - TestDialog

Type d'application

Vue d'ensemble

- Type d'application
- Prise en charge doc. composés
- Chaînes du modèle de doc.
- Prise en charge des bases de données
- Fonctionnalités interface util.
- Fonctionnalités avancées
- Classes générées

Type d'application :

- ☐ Mono document (SDI)
- ☐ Multidocument (MDI)
- ☒ Basée sur des boîtes de dialogue
 - ☐ Utiliser la boîte de dialogue HTML
- ☐ Plusieurs documents de niveau supérieur

☒ Prise en charge de l'architecture Document/View

Style du projet :

- ☐ Explorateur Windows
- ☒ Standard MFC

Utilisation de MFC :

- ☒ Dans une DLL partagée
- ☐ Dans une bibliothèque statique

Langue des ressources :

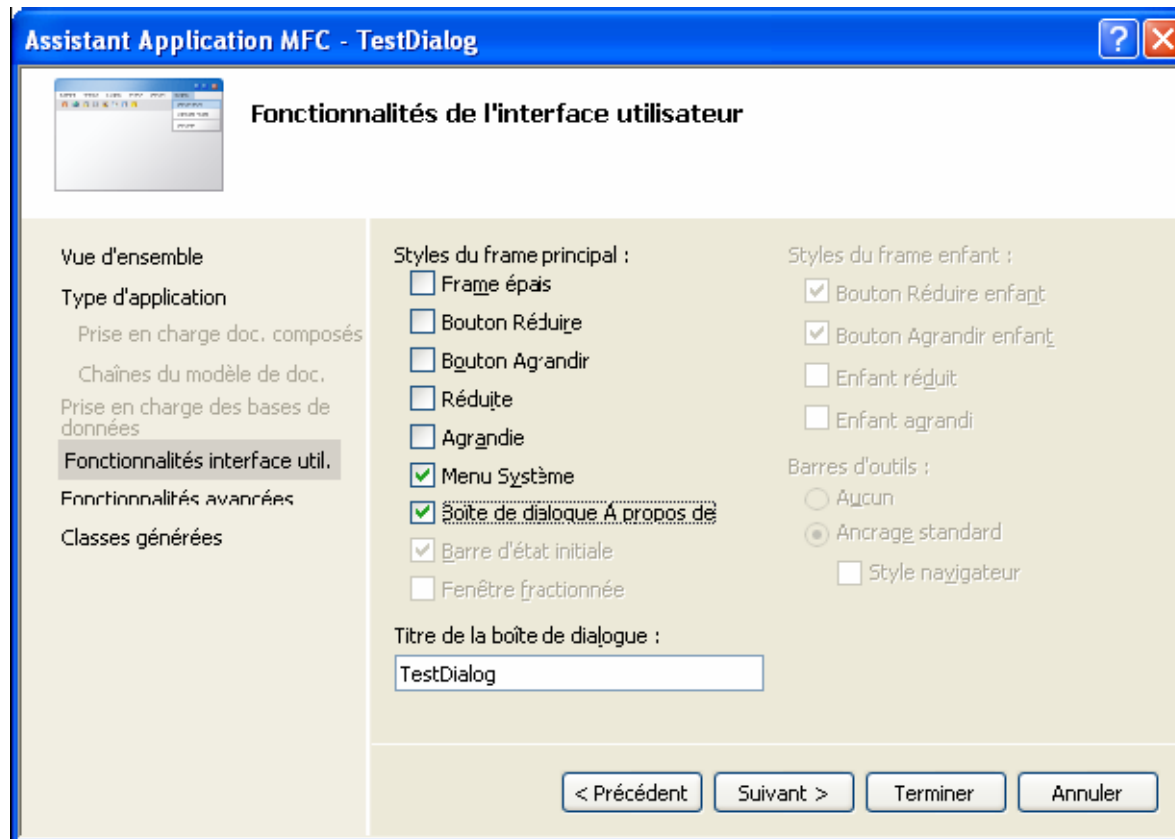
Français (France)

☒ Utiliser les bibliothèques Unicode

< Précédent Suivant > Terminer Annuler

Gestion d'un carnet de rendez-vous

- L'onglet suivant concerne les options de l'interface utilisateur :



Gestion d'un carnet de rendez-vous

Assistant Application MFC - TestDialog

Classes générées

Vue d'ensemble
Type d'application
Prise en charge doc. composés
Chaînes du modèle de doc.
Prise en charge des bases de données
Fonctionnalités interface util.
Fonctionnalités avancées
Classes générées

Classes générées :

- CTestDialogApp
- CTestDialogDlg

Nom de la classe : CTestDialogApp

Fichier .h : TestDialog.h

Classe de base : CWinApp

Fichier .cpp : TestDialog.cpp

< Précédent Suivant > Terminer Annuler



Gestion d'un carnet de rendez-vous

```
// InitCommonControlsEx() est requis sur Windows XP si le manifeste de l'application
// spécifie l'utilisation de ComCtl32.dll version 6 ou ultérieure pour activer les
// styles visuels. Dans le cas contraire, la création de fenêtres échouera.
INITCOMMONCONTROLSEX InitCtrls;
InitCtrls.dwSize = sizeof(InitCtrls);
// À définir pour inclure toutes les classes de contrôles communs à utiliser
// dans votre application.
InitCtrls.dwICC = ICC_WIN95_CLASSES;
InitCommonControlsEx(&InitCtrls);
CWinApp::InitInstance();
AfxEnableControlContainer();
// Initialisation standard
// Si vous n'utilisez pas ces fonctionnalités et que vous souhaitez réduire la taille
// de votre exécutable final, vous devez supprimer ci-dessous
// les routines d'initialisation spécifiques dont vous n'avez pas besoin.
// Changez la clé de Registre sous laquelle nos paramètres sont enregistrés
// TODO : modifiez cette chaîne avec des informations appropriées,
// telles que le nom de votre société ou organisation
```

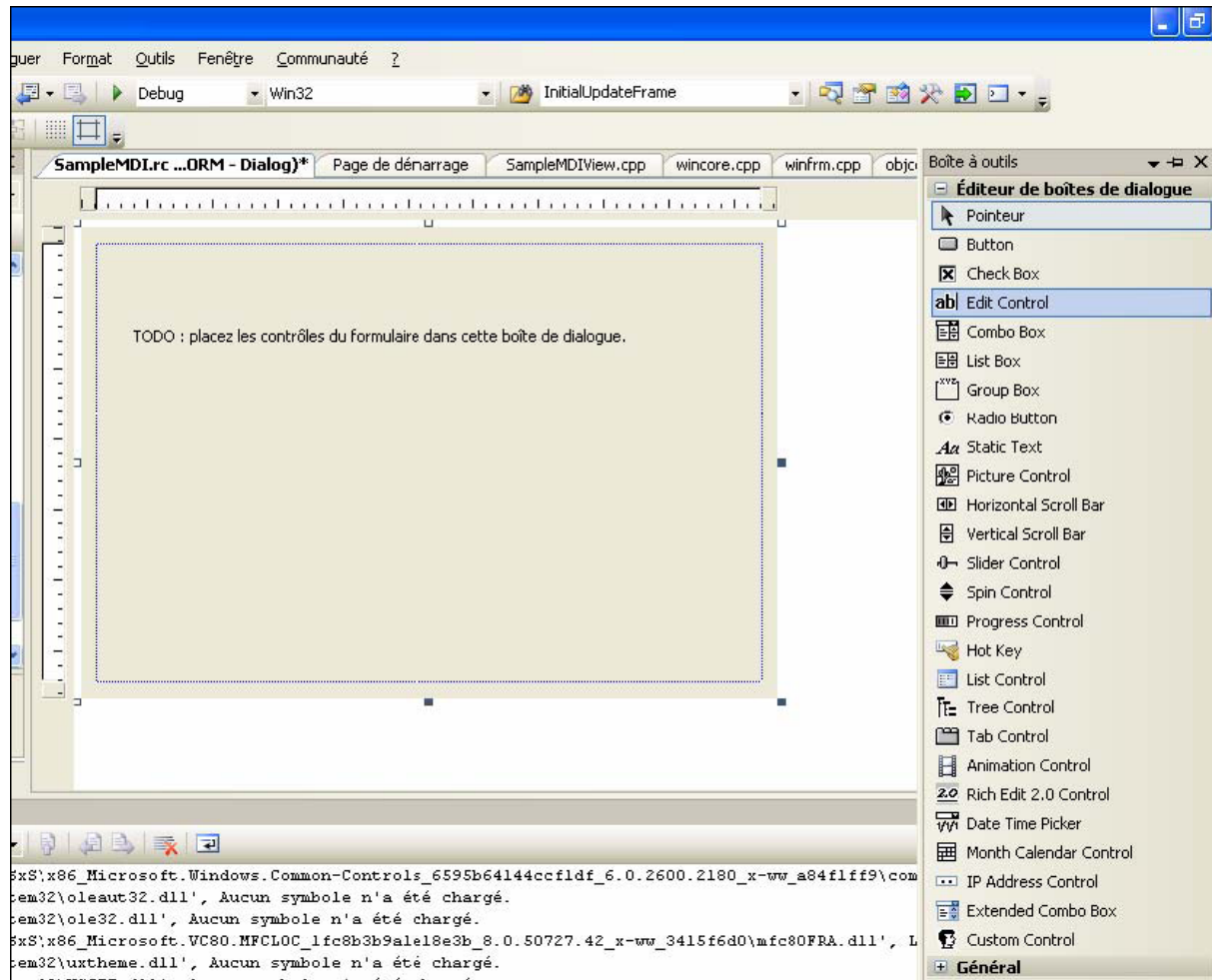


Gestion d'un carnet de rendez-vous

```
SetRegistryKey(_T("Applications locales générées par AppWizard"));
CTestDialogDlg dlg;
m_pMainWnd = &dlg;
INT_PTR nResponse = dlg.DoModal();
if (nResponse == IDOK)
{
    // TODO : placez ici le code définissant le comportement lorsque la boîte de dialogue est
    // fermée avec OK
}
else if (nResponse == IDCANCEL)
{
    // TODO : placez ici le code définissant le comportement lorsque la boîte de dialogue est
    // fermée avec Annuler
}
// Lorsque la boîte de dialogue est fermée, retourner FALSE afin de quitter
// l'application, plutôt que de démarrer la pompe de messages de l'application.
return FALSE;
```

Gestion d'un carnet de rendez-vous

Ajouter des éléments à la boîte de dialogue



Gestion d'un carnet de rendez-vous

