

Développeur

Microsoft

BIBLIOTHEQUE DU CERIST

MANUEL DE RÉFÉRENCE
MICROSOFT®

ADO.NET

Microsoft®
.net

David Sceppa

IST 3041

Développeur

Microsoft®

BIBLIOTHEQUE DU CERIST

MANUEL DE RÉFÉRENCE MICROSOFT® ADO.NET

David Sceppa

Adapté de l'anglais par :
Marc Israël

Microsoft®
.net

Envoyer les modifications à la base de données	39
La barre d'état des composants	43
L'Assistant Formulaire de données comme point de départ	45
Questions devant être posées plus fréquemment	46

Partie II **Se connecter : utiliser un fournisseur de données .NET**

3	Se connecter à votre base de données	49
	Utiliser les objets <i>Connection</i>	50
	Créer des objets <i>Connection</i>	54
	Chaînes de connexion	55
	Ouvrir et fermer des connexions	60
	Pool de connexions	61
	Supprimer des connexions	65
	Utiliser les connexions pour créer d'autres objets	65
	Récupérer les informations de schéma de la base de données	68
	Fonctionnalité de Visual Studio .NET en mode conception	70
	Travailler avec les connexions dans l'Explorateur de serveurs	70
	Ajouter une connexion à l'Explorateur de serveurs	72
	Ajouter des connexions à votre application	75
	Utiliser la nouvelle connexion au moment de l'exécution	77
	Où est le code ?	78
	Référence de l'objet <i>OleDbConnection</i>	80
	Propriétés de l'objet <i>OleDbConnection</i>	80
	Méthodes de l'objet <i>OleDbConnection</i>	84
	Événements de l'objet <i>OleDbConnection</i>	90
	Questions devant être posées plus fréquemment	94
4	Interroger votre base de données	97
	Utiliser les objets <i>Command</i> dans le code	97
	Créer un objet <i>Command</i>	97
	Exécuter une requête ne retournant pas de données	98

Utiliser un objet <i>DataReader</i> pour examiner les résultats d'une requête	101
Exécuter une requête qui renvoie une seule valeur	110
Exécuter une requête paramétrée	111
Appeler une procédure stockée	113
Récupérer des données à partir des paramètres de sortie	115
Exécuter une requête dans une transaction	117
Créer des commandes dans Visual Studio .NET	118
Faire glisser depuis la boîte à outils	118
Définir une connexion	119
Utiliser le Générateur de requêtes	120
Utiliser votre nouvel objet <i>Command</i> dans le code	122
Faire glisser depuis l'Explorateur de serveurs	124
Référence des objets <i>Command</i> , <i>DataReader</i> et <i>Parameter</i>	125
Propriétés de l'objet <i>OleDbCommand</i>	126
Méthodes de l'objet <i>OleDbCommand</i>	129
Propriétés de l'objet <i>OleDbDataReader</i>	134
Méthodes de l'objet <i>OleDbDataReader</i>	135
Créer des objets <i>Parameter</i>	141
Propriétés de l'objet <i>OleDbParameter</i>	141
Questions devant être posées plus fréquemment	145
5 Récupérer des données avec des objets <i>DataAdapter</i>	157
Qu'est ce qu'un objet <i>DataAdapter</i> ?	158
En quoi un <i>DataAdapter</i> est-il différent des autres objets ?	158
Anatomie d'un <i>DataAdapter</i>	161
Créer et utiliser des objets <i>DataAdapter</i>	164
Créer un <i>DataAdapter</i>	164
Récupérer les résultats d'une requête	166
Mettre en correspondance les résultats de votre requête avec votre <i>DataSet</i>	172
Travailler avec des lots de requêtes	174
Récupérer des lignes à partir d'une procédure stockée	175
Récupérer les informations de schéma	176

Créer des objets <i>DataAdapter</i> avec Visual Studio .NET	177
Faire glisser un <i>DataAdapter</i> depuis la boîte à outils	178
Utiliser l'Assistant Configuration d'adaptateur de données	178
Faire glisser depuis l'Explorateur de serveurs	183
Prévisualiser les résultats de votre <i>DataAdapter</i>	184
Examiner le code généré par l'assistant	185
Référence du <i>DataAdapter</i>	185
Propriétés de l'objet <i>DataAdapter</i>	185
Méthodes de l'objet <i>DataAdapter</i>	191
Événements de l'objet <i>DataAdapter</i>	198
Questions devant être posées plus fréquemment	201

Partie III **Travailler avec des données déconnectées : le *DataSet* ADO.NET**

6	Travailler avec des objets <i>DataSet</i>	205
	Fonctionnalités de l'objet <i>DataSet</i>	205
	Travailler avec des données déconnectées	206
	Parcourir, trier, rechercher et filtrer	206
	Travailler avec des données hiérarchiques	206
	Modifications en cache	207
	Intégration de XML	207
	Fonctionnalités uniformes	207
	Utiliser les objets <i>DataSet</i>	208
	Créer un objet <i>DataSet</i>	208
	Examiner la structure créée par l'appel de <i>DataAdapter.Fill</i>	209
	Examiner les données retournées par un <i>DataAdapter</i>	211
	Valider les données dans votre <i>DataSet</i>	214
	Créer des objets <i>DataTable</i> en code	219
	Modifier le contenu d'une <i>DataTable</i>	235
	Travailler avec les objets <i>DataSet</i> dans Visual Studio .NET	246
	Générer un <i>DataSet</i> à partir d'un objet <i>DataAdapter</i>	247
	Créer un nouveau <i>DataSet</i> à partir de rien	250
	Créer un <i>DataSet</i> non typé	254

Référence des objets <i>DataSet</i> , <i>DataTable</i> , <i>DataColumn</i> , <i>DataRow</i> , <i>UniqueConstraint</i> et <i>ForeignKeyConstraint</i>	258
Propriétés de l'objet <i>DataSet</i>	258
Méthodes de l'objet <i>DataSet</i>	263
Événement de l'objet <i>DataSet</i>	267
Propriétés de l'objet <i>DataTable</i>	267
Méthodes de l'objet <i>DataTable</i>	272
Événements de l'objet <i>DataTable</i>	277
Propriétés de l'objet <i>DataColumn</i>	278
Propriétés de l'objet <i>DataRow</i>	285
Méthodes de l'objet <i>DataRow</i>	286
Propriétés de l'objet <i>UniqueConstraint</i>	293
Propriétés de l'objet <i>ForeignKeyConstraint</i>	294
Questions devant être posées plus fréquemment	296
7 Travailler avec des données relationnelles	299
Un bref tour d'horizon de l'accès aux données relationnelles	300
Requêtes avec jointures	300
Requêtes séparées	302
Objets ADO <i>Recordset</i> hiérarchiques	302
Objets ADO.NET <i>DataRelation</i>	303
Travailler avec des objets <i>DataRelation</i> dans du code	304
Créer des objets <i>DataRelation</i>	305
Localiser des données liées	308
Utiliser des objets <i>DataRelation</i> pour valider vos données	311
Objets <i>DataRelation</i> auto-référencés	315
Relations plusieurs-à-plusieurs	317
Utiliser les objets <i>DataRelation</i> dans des objets <i>DataColumn</i> basés sur des expressions	320
Modifications en cascade	323
Se déprendre des requêtes avec jointures	325
Créer des objets <i>DataRelation</i> dans Visual Studio .NET	325
Ajouter une <i>DataRelation</i> à un <i>DataSet</i> fortement typé	325
Ajouter une <i>DataRelation</i> à un <i>DataSet</i> non typé	327

Référence de l'objet <i>DataRelation</i>	328
Propriétés de l'objet <i>DataRelation</i>	328
Questions devant être posées plus fréquemment	332
8 Trier, rechercher et filtrer	337
Utiliser les fonctions de recherche et de filtrage de l'objet <i>DataTable</i>	337
Localiser une ligne à partir de la valeur de sa clé primaire	337
Lancer des recherches plus dynamiques	340
Lancer des recherches avec des jokers	341
Travailler avec des délimiteurs	341
Utiliser les méthodes <i>Select</i> supplémentaires	344
Qu'est-ce qu'un objet <i>DataRowView</i> ?	346
Les objets <i>DataRowView</i> retournent les données d'une <i>DataTable</i>	346
Les objets <i>DataRowView</i> ne sont pas des requêtes SQL	347
Travailler avec les objets <i>DataRowView</i> dans le code	348
Créer des objets <i>DataRowView</i>	348
Utiliser la propriété <i>RowStateFilter</i>	349
Utiliser l'objet <i>DataRowView</i>	350
Examiner toutes les lignes de données disponibles dans une <i>DataRowView</i>	351
Rechercher des données dans une <i>DataRowView</i>	352
Modifier les objets <i>DataRowView</i>	354
Créer des objets <i>DataRowView</i> dans Visual Studio .NET	356
Ajouter un nouvel objet <i>DataRowView</i> à votre concepteur	356
Définir les propriétés de votre objet <i>DataRowView</i>	356
Références de l'objet <i>DataRowView</i>	357
Propriétés de l'objet <i>DataRowView</i>	357
Méthodes de l'objet <i>DataRowView</i>	361
L'événement <i>ListChanged</i> de l'objet <i>DataRowView</i>	365
Propriétés de l'objet <i>DataRowView</i>	366
Méthodes de l'objet <i>DataRowView</i>	367
Questions devant être posées plus fréquemment	369

9	Travailler avec des objets <i>DataSet</i> fortement typés	375
	Créer des objets <i>DataSet</i> fortement typés	376
	La manière dure	376
	La manière simple	379
	Utiliser des objets <i>DataSet</i> fortement typés	382
	Ajouter une ligne	382
	Trouver une ligne	384
	Editer une ligne	385
	Travailler avec des données Null	386
	Travailler avec des données hiérarchiques	387
	Autres fonctionnalités des objets <i>DataSet</i> , <i>DataTable</i> et <i>DataRow</i>	389
	Quand utiliser des objets <i>DataSet</i> fortement typés	389
	Composants logiciels et couteaux suisses	389
	Avantage à la conception	390
	Avantage à l'exécution	391
	Considérations complémentaires	394
	Choisir votre chemin	397
	Questions devant être posées plus fréquemment	397
10	Soumettre les mises à jour à votre base de données	401
	Une leçon d'histoire	404
	Avantages de soumettre les mises à jour avec les objets ADO <i>Recordset</i>	405
	Inconvénients de soumettre les mises à jour avec les objets ADO <i>Recordset</i>	406
	Utiliser les objets ADO.NET <i>Command</i> pour soumettre les mises à jour	407
	Utiliser les objets ADO.NET <i>DataAdapter</i> pour soumettre les mises à jour	419
	Configurer manuellement les objets <i>DataAdapter</i>	419
	Introduire les paramètres liés	419
	Utiliser des procédures stockées pour soumettre les mises à jour	424

Fournir votre propre code de mise à jour	430
Utiliser l'objet <i>CommandBuilder</i> pour générer le code de mise à jour	431
Comment le <i>CommandBuilder</i> génère le code de mise à jour	432
Avantages et inconvénients du <i>CommandBuilder</i>	434
Utiliser l'Assistant de configuration d'adaptateur de données pour générer le code de mise à jour	434
Examiner la structure du <i>DataAdapter</i>	435
Options de construction du code de mise à jour	436
Utiliser des procédures stockées pour soumettre les mises à jour	437
Avantages et inconvénients de l'assistant	441
Autres questions liées à la mise à jour	441
Options d'accès concurrentiel optimiste	441
Travailler avec les valeurs Null	446
Soumettre les mises à jour dans des transactions	447
Utiliser la collection <i>TableMappings</i>	451
La meilleure méthode de mise à jour	454
Référence de l'objet <i>OleDbCommandBuilder</i>	455
Propriétés de l'objet <i>OleDbCommandBuilder</i>	455
Méthodes de l'objet <i>OleDbCommandBuilder</i>	456
Questions devant être posées plus fréquemment	458
11 Scénarios avancés de mise à jour	467
Actualiser une ligne après avoir soumis une mise à jour	468
Récupérer la valeur nouvellement générée pour une colonne timestamp après avoir soumis une mise à jour	469
Utiliser des lots de requêtes pour récupérer les données après avoir soumis une mise à jour	469
Récupérer les nouvelles données avec des paramètres de sortie	471

Utiliser l'événement <i>RowUpdated</i> de l'objet <i>DataAdapter</i> pour récupérer les données après avoir envoyé une mise à jour	472
L'application exemple Timestamp	474
Récupérer les nouvelles valeurs de compteur	474
Travailler avec SQL Server	475
Travailler avec Access 2000	478
Travailler avec les séquences Oracle	478
Application exemple récupérant les valeurs de compteurs	481
Utiliser l'instruction NOCOUNT de SQL Server	482
Soumettre des modifications hiérarchiques	485
Soumettre les insertions et les suppressions en attente	485
Travailler avec les compteurs et les données relationnelles	488
Isoler et réintégrer les modifications	490
Économiser la bande passante avec la méthode <i>GetChanges</i>	491
Gérer élégamment les tentatives échouées de mise à jour	503
Prévoir les conflits	503
Informier l'utilisateur de l'échec	504
Récupérer le contenu de lignes en conflit	506
Si vous ne réussissez pas du premier coup...	508
L'application exemple de conflits	509
Travailler avec les transactions distribuées	510
Coordinateurs de transactions et gestionnaires de ressources	512
Transactions distribuées dans le .NET Framework	513
Support des bases de données des transactions distribuées	514
Construire vos composants	514
L'application exemple de transactions distribuées	519
Autres avantages de l'utilisation des services de composants	519

Utiliser ADO.NET pour gérer les scénarios avancés de mise à jour	519
Questions devant être posées plus fréquemment	520
12 Travailler avec des données XML	523
Comblar le fossé entre le XML et l'accès aux données	523
Lire et écrire des données XML	524
Les méthodes XML de l'objet <i>DataSet</i>	524
Schémas inférant	528
Propriétés ADO.NET ayant un effet sur le schéma de vos documents XML	529
Mettre en cache les modifications et les documents XML	531
<i>DataSet + XmlDocument = XmlDataDocument</i>	534
Utiliser l'objet <i>XmlDataDocument</i>	534
Accéder à votre <i>DataSet</i> comme document XML	534
Mettre en cache les modifications apportées au document XML	536
Récupérer des données XML à partir de SQL Server 2000	537
Travailler avec les requêtes SELECT... FOR XML	538
Le fournisseur de données .NET SQL XML	541
Utiliser un objet <i>SqlXmlCommand</i> pour charger des données dans un <i>XmlDocument</i>	542
Utiliser un objet <i>SqlXmlAdapter</i> pour charger des données dans un <i>DataSet</i>	543
Travailler avec les requêtes modèles	544
Travailler avec les requêtes XPath	547
Appliquer une transformation XSLT	550
Envoyer des mises à jour	550
Un exemple ADO.NET et XML simple	555
Deux chemins, une seule destination	556
ADO.NET et XML : un couple heureux	556
Questions devant être posées plus fréquemment	557

Partie IV **Construire des applications efficaces avec ADO.NET**

13	Construire des applications Windows efficaces	561
	Construire rapidement une interface utilisateur avec la liaison de données	562
	Etape 1 : Créer votre DataAdapter et votre DataSet	563
	Etape 2 : Ajouter des zones de texte liées	564
	Etape 3 : Récupérer les données	567
	Etape 4 : Ajouter des boutons de navigation	568
	Etape 5 : Ajouter les boutons Ajouter et Supprimer	571
	Etape 6 : Envoyer les modifications	572
	Etape 7 : Ajouter les boutons Edition, Mettre à jour et Annuler	573
	Etape 8 : Visualiser les données enfants	576
	Etape 9 : Lier une deuxième fenêtre à la même source de données	581
	Etape 10 : Améliorer l'interface utilisateur	583
	Etape 11 : Si vous voulez bien faire les choses	587
	Résumé sur la liaison de données	588
	Considérations sur la conception d'application	588
	Ne récupérer que les données dont vous avez besoin	588
	Stratégies de mise à jour	589
	Stratégies de connexion	593
	Travailler avec des données BLOB	595
	Interfaces utilisateur construites avec la puissance d'ADO.NET	601
	Questions devant être posées plus fréquemment	601
14	Construire des applications Web efficaces	605
	Une introduction aux applications Web	605
	ASP.NET facilite la construction d'applications Web	605

Le bon et le mauvais du sans état	606
Se connecter à votre base de données	607
Travailler avec des connexions approuvées	607
Travailler avec des bases de données Access	609
Afficher des données sur votre page Web	610
Utiliser DataBinder.Eval	611
Lier les contrôles DataGrid aux résultats des requêtes	613
Mettre les données en cache entre deux allers-retours	615
L'approche sans état	615
Cacher des données sur le client	616
Maintenir l'état dans votre serveur Web	619
Maintenir l'état dans votre base de données	621
Guides sur l'état	622
Pagination	623
Fonctionnalités de pagination de la DataGrid Web	624
Fonctionnalités de pagination de la méthode	
Fill du DataAdapter	627
Construire des requêtes qui paginent les données	628
L'exemple PagingOptions	629
Éditer des données sur une page web	629
Utiliser la DataGrid pour simplifier l'édition des données	630
Gérer les événements d'édition de la DataGrid	631
Envoyer les mises à jour à la base de données	632
L'exemple ShoppingCart	633
Questions devant être posées plus fréquemment	634

Partie V Annexes

Annexe A Travailler avec d'autres fournisseurs de données .NET	641
Le fournisseur de données .NET SQL Client	641
Paramètres nommés et marqueurs de paramètres	641
Se connecter à une base de données SQL Server avec un SqlConnection	642

Récupérer les résultats d'une requête avec un SqlDataAdapter	643
Utiliser les objets SqlCommand et SqlDataReader	644
Les méthodes GetSql<TypeDonnées> et l'espace de noms SqlTypes	645
Appeler des procédures stockées	647
Récupérer des informations de schéma de base de données	648
Le fournisseur de données .NET ODBC	649
Se connecter à votre base de données avec une OdbcConnection	650
Travailler avec des requêtes paramétrées	651
Récupérer les résultats d'une requête avec un OdbcDataAdapter	651
Examiner les résultats d'une requête avec un OdbcDataReader	652
Appeler une procédure stockée	653
Récupérer les informations de schéma de la base de données	654
Le fournisseur de données .NET Oracle	654
Se connecter à votre base de données Oracle avec une OracleConnection	655
Travailler avec des requêtes paramétrées	655
Récupérer les résultats d'une requête avec un OracleDataAdapter	656
Examiner les résultats d'une requête avec un OracleDataReader	656
Types de données spécifiques à Oracle	657
Appeler une procédure stockée	659
Récupérer des données à partir des curseurs REF d'Oracle	660
Récupérer les informations de schéma de la base de données	661

Questions communes à tous les fournisseurs	662
Écrire du code portable de fournisseur en fournisseur	662
Choisir le bon type de données du fournisseur de données .NET	666

Annexe B Les Outils 669

Outil de requêtage ad hoc ADO.NET	669
Se connecter à votre base de données	670
Les fournisseurs de données .NET	670
Exécuter des requêtes	671
Examiner les informations de schéma de requête	672
Soumettre les mises à jour	675
Paramètres de l'application	675
Le Constructeur de DataAdapter ADO.NET	675
Indiquer votre logique de mise à jour	676
Contrôle de navigation ADO.NET	678
Ajouter le Contrôle de Navigation ADO.NET à la boîte à outils de Visual Studio .NET	679
Définir les propriétés de données du Contrôle de Navigation ADO.NET	679
Définir les autres propriétés du Contrôle de Navigation ADO.NET	679

Index 681