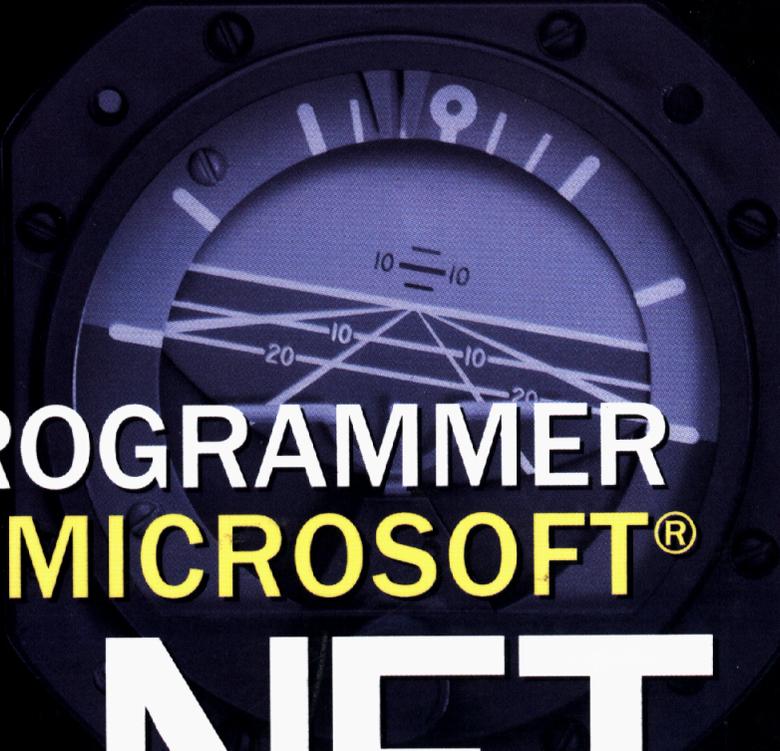


Développeur

Microsoft



PROGRAMMER
MICROSOFT®
.NET
FRAMEWORK

BIBLIOTHEQUE DU CERIST

Jeffrey Richter

Microsoft®
.net

Wintellect

ISI 3014

BIBLIOTHEQUE DU CERIST

PROGRAMMER

MICROSOFT®

.NET FRAMEWORK

BIBLIOTHEQUE DU CERIST

Développeur

Microsoft®



BIBLIOTHEQUE NUCERIST

PROGRAMMER

MICROSOFT®

NET FRAMEWORK

Jeffrey Richter

Adapté de l'anglais par :
Frédéric Rutkowski
avec la collaboration de
Frank Deshayes

Microsoft®
.net

Wintellect®

Les programmes figurant dans ce livre, et éventuellement sur la disquette ou le CD-ROM d'accompagnement, sont fournis gracieusement sous forme de code source, à titre d'illustration. Ils sont fournis en l'état sans garantie aucune quant à leur fonctionnement une fois compilés, assemblés ou interprétés dans le cadre d'une utilisation professionnelle ou commerciale. Ils peuvent nécessiter des adaptations et modifications dépendant de la configuration utilisée. Microsoft Press ne pourra en aucun cas être tenu responsable des préjudices ou dommages de quelque nature que ce soit pouvant résulter de l'utilisation de ces programmes.

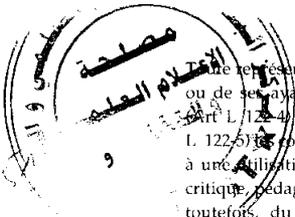
Tous les efforts ont été faits pour fournir dans ce livre une information complète et exacte à la date de la parution. Néanmoins, Microsoft Press n'assume de responsabilités ni pour son utilisation, ni pour les contrefaçons de brevets ou atteintes aux droits de tierces personnes qui pourraient résulter de cette utilisation.

Visual Basic, Visual C++, Visual C#, Visual Studio et Windows sont soit des marques déposées, soit des marques de Microsoft Corporation aux États-Unis ou/et d'autres pays.

Copyright 2002 by Microsoft Corporation.

Original English language edition Copyright © 2002 by Jeffrey Richter. All rights published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U.S.A.

Titre U.S. : APPLIED MICROSOFT. NET FRAMEWORK PROGRAMMING
ISBN U.S. : 0 7356 1422 9



Une représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite selon le Code de la propriété intellectuelle (Art L 122-4) et constitue une contrefaçon réprimée par le Code pénal. • Seules sont autorisées (Art L 122-5) les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective, ainsi que les analyses et courtes citations justifiées par le caractère critique, pédagogique ou d'information de l'œuvre à laquelle elles sont incorporées, sous réserve, toutefois, du respect des dispositions des articles L 122-10 à L 122-12 du même Code, relatives à la reproduction par reprographie.

Édition et diffusion : Dunod

Distribution : Vivendi Universal Publishing Services

Traducteur : Frédéric Rutkowski avec la collaboration de Frank Deshayes

Mise en page : IID

ISBN : 2 10 006732 X

Sommaire

	Introduction	1
Partie I	Aspects fondamentaux de Microsoft .NET Framework	
1	Architecture de la plate-forme de développement .NET Framework	3
	Compilation du code source en modules managés	3
	Combinaison de modules managés en assemblages	7
	Chargement du CLR (Common Language Runtime)	9
	Exécution du code de l'assemblage	12
	IL et la vérification	19
	La bibliothèque de classes .NET Framework	21
	Le système de type commun (CTS)	25
	La spécification du langage commun	27
	Interopérabilité avec le code non managé	32
2	Construction, packaging, déploiement et administration des applications et des types	37
	Objectifs du déploiement de .NET Framework	38
	Construction des types en un module	39
	Combinaison de modules en assemblage	48
	Ajout d'assemblages à un projet avec l'environnement de développement intégré de Visual Studio .NET	55
	Utilisation de l'éditeur d'assemblage (AL.exe)	56
	Ajout de fichiers de ressources dans un assemblage	58
	Informations relatives à la version de l'assemblage	59
	Numéros de version	63
	Culture	64

	Déploiement d'applications simples (assemblages déployés à titre privé)	66
	Contrôles d'administration simples	67
3	Assemblages partagés	73
	Deux types d'assemblages, deux types de déploiement	74
	Attribuer un nom fort à un assemblage	75
	Le cache global des assemblages	81
	La structure interne du GAC	86
	Construction d'un assemblage qui référence un assemblage à nom fort	88
	Les assemblages à noms forts résistent à la falsification	91
	Signature retardée	92
	Déploiement à titre privé d'assemblages à noms forts	97
	Exécution côte à côte	98
	Comment le runtime résout-il les références aux types ?	100
	Contrôles d'administration avancés (configuration)	103
	Contrôles de stratégie d'éditeur	108
	Réparation d'une application défectueuse	111
Partie II	Types et CLR	
4	Aspects fondamentaux des types	117
	Tous les types sont dérivés de <code>System.Object</code>	117
	Réalisation de casts entre les types	119
	Opérations de cast avec les opérateurs <code>is</code> et <code>as</code> du langage C#	122
	Espaces de noms et assemblages	124
5	Types primitifs, références et valeurs	129
	Types primitifs des langages de programmation	129
	Opérations checked et unchecked sur les types primitifs	133
	Types références et types valeurs	137
	Compartimentation et décompartimentation des types valeurs	144

6	Opérations communes sur les objets	157
	Égalité et identité des objets	157
	Mise en œuvre de la méthode <code>Equals</code> pour un type de référence dont les classes de base ne remplacent pas la méthode <code>Equals</code> du type <code>Object</code>	158
	Mise en œuvre de la méthode <code>Equals</code> pour un type de référence dont une ou plusieurs classes de base remplacent la méthode <code>Equals</code> du type <code>Object</code>	160
	Mise en œuvre de la méthode <code>Equals</code> pour un type de valeur	162
	Résumé de la mise en œuvre de la méthode <code>Equals</code> et les opérateurs <code>==/!=</code>	164
	Identité	165
	Codes de hachage des objets	166
	Clonage d'objets	169
Partie III	Conception de types	
7	Membres de types et accessibilité	175
	Membres des types	175
	Modificateurs d'accessibilité et attributs prédéfinis	178
	Attributs prédéfinis pour un type	180
	Attributs prédéfinis pour un champ	180
	Attributs prédéfinis pour une méthode	181
8	Constantes et champs	183
	Constantes	183
	Champs	184
9	Méthodes	187
	Constructeurs d'instances	187
	Constructeurs de types	193
	Méthodes de surcharge d'opérateur	197

	Opérateurs et interopérabilité des langages de programmation	200
	Méthodes d'opérateurs de conversion	203
	Fournir des paramètres par référence à une méthode	207
	Fourniture d'un nombre variable de paramètres à une méthode	213
	Comment les méthodes virtuelles sont-elles invoquées ?	215
	Méthode virtuelle de gestion de version	216
10	Propriétés	223
	Propriétés sans paramètres	223
	Propriétés avec paramètres	228
11	Événements	235
	Conception d'un type fournissant un événement	237
	Conception d'un type à l'écoute d'un événement	241
	Contrôle explicite des enregistrements d'événements	244
	Conception d'un type qui définit de nombreux événements	246
	Conception du type <code>EventHandlerSet</code>	251
Partie IV	Types essentiels	
12	Manipulation de texte	257
	Caractères	257
	Le type <code>System.String</code>	260
	Construction de chaînes de caractères	261
	Les chaînes de caractères sont immuables	263
	Comparaison de chaînes de caractères	264
	Internalisation de chaînes de caractères	270
	Mise en commun d'une chaîne de caractères	275
	Examiner les caractères d'une chaîne de caractères	275
	Autres opérations sur les chaînes de caractères	279
	Construction dynamique et efficace d'une chaîne de caractères	279

Construction d'un objet <code>StringBuilder</code>	280
Membres de <code>StringBuilder</code>	281
Obtenir une représentation en chaîne de caractères d'un objet	284
Formats et cultures spécifiques	285
Mise en forme de plusieurs objets en une seule chaîne de caractères	289
Fournir votre outil personnalisé de mise en forme	291
Analyse syntaxique d'une chaîne de caractères pour obtenir un objet	294
Encodages : conversion entre des caractères et des octets	299
Encodage/décodage de flux de caractères et d'octets	306
Encodage et décodage de chaînes de caractères fondées sur 64 bits	307
13 Types énumérés et drapeaux de bits	309
Types énumérés	309
Drapeaux de bits	315
14 Tableaux	319
Tous les tableaux sont implicitement dérivés de <code>System.Array</code>	322
Conversion de tableaux	325
Passage et retour des tableaux	327
Création de tableaux qui ont une borne inférieure différente de zéro	328
Accès rapide aux tableaux	329
Redimensionnement d'un tableau	333
15 Interfaces	335
Interfaces et héritages	335
Conception d'une application qui prend en charge les plug-ins	341
Modification des champs d'un type de valeur compartimenté avec les interfaces	343

	Mise en œuvre de plusieurs interfaces contenant la même méthode	346
	Mises en œuvre explicites de membre d'interface	348
16	Attributs personnalisés	355
	Utilisation des attributs personnalisés	355
	Définir votre attribut	359
	Constructeur d'attribut et type de données de champ/propriété	363
	Détecter l'utilisation d'un attribut personnalisé	364
	Comparaison de deux instances d'attribut	369
	Attributs pseudo-personnalisés	372
17	Délégations	375
	Première approche des délégations	375
	Rappel de méthodes statiques avec des délégations	378
	Rappel de méthodes d'instances avec des délégations	380
	Démystification des délégations	380
	Historique des délégations : <code>System.Delegate</code> et <code>System.MulticastDelegate</code>	385
	Comparaison de l'égalité de délégations	387
	Chaînes de délégations	387
	Prise en charge des chaînes de délégations par le langage C#	393
	Obtenir un meilleur contrôle sur l'invocation d'une chaîne de délégations	393
	Délégations et réflexion	396
Partie V	Types managés	
18	Exceptions	403
	Évolution de la gestion des exceptions	404
	Les mécanismes de la gestion des exceptions	406
	Le bloc <code>try</code>	407
	Le bloc <code>catch</code>	407
	Le bloc <code>finally</code>	410

Qu'est-ce qu'une exception exactement ?	411
La classe <code>System.Exception</code>	415
Les classes d'exceptions définies dans la FCL	417
Définir votre propre classe d'exceptions	420
Comment employer correctement les exceptions	424
N'ayez pas trop de blocs <code>finally</code>	425
N'interceptez pas tout	426
Remise en état élégante après une exception	428
Annulation d'une opération partiellement effectuée lorsqu'une exception non récupérable se produit	429
Cacher les détails de l'implémentation	430
Ce qui ne va pas dans la FCL	433
Considérations sur les performances	435
Filtres d'interception	437
Exceptions non gérées	440
Contrôler ce que fait le CLR lorsqu'une exception non gérée se produit	445
Exceptions non gérées et Windows Forms	447
Exceptions non gérées et Web Forms ASP.NET	448
Exceptions non gérées et Services Web XML ASP.NET	449
Trace de la pile des exceptions	449
Traces de la pile distante	452
Mise au point des exceptions	453
Indiquer le type de code à déboguer à Visual Studio	457

19 Gestion automatique de la mémoire **461**

Comprendre les fondements du travail avec une plate-forme dont la mémoire est nettoyée	461
Algorithme de nettoyage de la mémoire	465
Finalisation	469
Quand la méthode <code>Finalize</code> est-elle invoquée ?	476
Mécanismes internes de la finalisation	478

Modèle de mise à disposition : forcer le nettoyage d'un objet	481
Utilisation d'un type qui met en œuvre le modèle de mise à disposition	488
Déclaration <code>using</code> du langage C#	492
Un problème de dépendance intéressant	494
Références faibles	496
Mécanismes internes des références faibles	497
Résurrection	499
Conception d'une réserve d'objets en utilisant la résurrection	501
Générations	504
Contrôle du nettoyeur de la mémoire par la programmation	509
Autres problèmes de performance du nettoyeur de la mémoire	512
Allocations sans synchronisation	513
Nettoyages parallèles évolutifs	514
Nettoyage simultané	514
Grands objets	516
Surveillance des nettoyages de la mémoire	516
20 Hébergement du CLR, AppDomains et réflexion	519
Métadonnée : le fondement de .NET Framework	519
Hébergement du CLR	520
AppDomains	522
Accès aux objets au travers des frontières d'une AppDomain	525
Événements des AppDomains	527
Les applications et la façon dont elles hébergent le CLR et gèrent les AppDomains	528
« Yukon »	529
L'essentiel de la réflexion	530
Réflexion sur les types d'un assemblage	532
Réflexion sur les assemblages d'une AppDomain	535

Réflexion sur les membres d'un type : liaison	535
Chargement explicite d'assemblages	537
Chargement d'assemblages en tant que « fichiers de données »	540
Construction d'une hiérarchie de types dérivés de <code>Exception</code>	541
Déchargement explicite des assemblages : déchargement d'une <code>AppDomain</code>	544
Obtenir une référence d'un objet <code>System.Type</code>	546
Réflexion sur les membres d'un type	551
Création d'une instance d'un type	553
Invoquer une méthode d'un type	555
Lié une fois, invoqué plusieurs fois	561
Réflexion sur les interfaces d'un type	566
Performances de la réflexion	568
 Index	 571