

Query Management Facility



Initiation à QMF pour Windows

Version 7

Query Management Facility



Initiation à QMF pour Windows

Version 7

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Annexe. Remarques» à la page 151.

Première édition – septembre 2000

Réf. US : SC27-0723-00

LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT". IBM DECLINE TOUTE RESPONSABILITE, EXPRESSE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE QUALITE MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

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

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

Vous pouvez également consulter les serveurs Internet suivants :

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

*Compagnie IBM France
Direction Qualité
Tour Descartes
92066 Paris-La Défense Cedex 50*

© Copyright IBM France 2000. Tous droits réservés.

© Copyright International Business Machines Corporation 1997, 2000. All rights reserved.

Table des matières

Bibliothèque QMF	vii	Enregistrement de requêtes SQL sur le serveur de base de données	14
Avis aux lecteurs canadiens	ix	Ouverture de requêtes SQL enregistrées sur le serveur de base de données	14
Chapitre 1. Introduction	1	Impression de requêtes SQL.	15
Serveurs de base de données	1	Prévisualisation d'une requête	15
Bases de données DB2	1	Impression de requêtes SQL.	15
Nom d'utilisateur et nom technique	1		
Définition du nom de serveur	1	Chapitre 3. Utilisation des requêtes assistées	17
Sécurité de la base de données	2	Création de requêtes simples	17
Connexion	2	Ouverture de nouvelles requêtes assistées	17
Correction des mots de passe	3	Boutons d'action associés aux requêtes assistées	17
Modification des mots de passe	3	Ajout de tables à des requêtes assistées	18
Définition des identifiants comptables	4	Exécution de requêtes assistées.	19
Gestion	4	Création de requêtes complexes	19
Affichage des limites de ressources.	4	Ajout de colonnes à des requêtes assistées	19
Définition d'un nombre limite de lignes	5	Utilisation de conditions de tri	20
Personnalisation de la barre d'outils	6	Ajout de conditions de tri	20
Ajout de boutons à la barre d'outils	6	Utilisation de conditions de lignes	21
Déplacement de boutons sur la barre d'outils	6	Ajout de conditions de lignes	21
Suppression de boutons de la barre d'outils	7	Utilisation de plusieurs tables dans les requêtes assistées	22
		Création de conditions de jonction dans une requête assistée	23
Chapitre 2. Utilisation des requêtes SQL	9	Requêtes assistées et requêtes SQL	23
Requêtes SQL	9	Affichage des instructions SQL associées aux requêtes assistées	23
Création de requêtes SQL	9	Conversion de requêtes assistées en requêtes SQL.	23
Exécution de requêtes SQL sur un serveur de base de données	9	Utilisation des variables de substitution dans les requêtes assistées	24
Navigation entre la fenêtre des résultats et la fenêtre SQL.	9	Enregistrement de requêtes assistées	24
Utilisation des polices.	10	Enregistrement de requêtes assistées dans des fichiers	24
Sélection de la police d'affichage d'une requête.	10	Ouverture de fichiers de requêtes assistées	24
Requêtes multiples.	10	Enregistrement de requêtes assistées sur le serveur de base de données.	25
Affichage simultané de plusieurs requêtes	10	Ouverture de requêtes assistées enregistrées sur le serveur de base de données	25
Création de requêtes	11	Impression de requêtes assistées	26
Création de requêtes SQL	11	Prévisualisation des requêtes assistées	26
Variables de substitution dans des requêtes SQL.	12		
Exécution de requêtes SQL contenant des variables de substitution	13		
Sauvegarde et ouverture des requêtes SQL	13		
Enregistrement des requêtes SQL dans des fichiers.	13		
Ouverture de fichiers de requêtes SQL	14		

Chapitre 4. Utilisation des résultats de requête	27
Tri et classement des résultats de requête	27
Sélection de colonnes et de lignes	27
Redimensionnement de colonnes et de lignes	27
Ajustement automatique des colonnes et des lignes	28
Tri des résultats de requête	28
Reclassement des colonnes	28
Formatage des résultats de requête	28
Sélection de la police d’affichage des résultats d’une requête	29
Formatage des résultats numériques de requête	29
Conversion du formatage des résultats de requête en format	29
Regroupement et agrégation des résultats de requête	29
Reroupage des résultats de requête	29
Résumé des résultats de requête	30
Enregistrement des résultats de requête et formatage	30
Enregistrement des résultats de requête dans une table	30
Enregistrement des résultats de requête sous forme de fichiers	30
Impression des résultats de requête	30
Prévisualisation des résultats de requête	30
Impression des résultats de requête	31
Chapitre 5. Utilisation des rapports	33
Formats	33
Présentation des formats	33
Création d’un rapport à l’aide d’un format	34
Modification d’un format	34
Création d’un format	35
Étape 1 : Création d’un format	35
Étape 2 : Changement de l’ordre des colonnes	35
Étape 3 : Modification des en-têtes de colonnes	35
Étape 4 : Modification du format des colonnes	36
Étape 5 : Ajout d’informations récapitulatives	36
Étape 6 : Ajout d’en-têtes et de bas de page	36
Enregistrement des formats	37
Enregistrement d’un format dans un fichier	37

Ouverture de fichiers de formats enregistrés	37
Enregistrement de formats sur le serveur de base de données	37
Ouverture de formats enregistrés sur le serveur de base de données	38
Impression de rapports	38
Exportation de rapports	39

Chapitre 6. Utilisation des procédures	41
Exécution de procédures	41
Création d’une nouvelle procédure linéaire	41
Création d’une nouvelle procédure logique	41
Exécution d’une procédure sur un serveur de base de données	42
Enregistrement de procédures	42
Enregistrement des procédures dans des fichiers	42
Ouverture d’un fichier de procédure	42
Enregistrement d’une procédure sur le serveur de base de données	42
Ouverture de procédures enregistrées sur le serveur de base de données	43
Impression de procédures	43
Prévisualisation d’une procédure	43
Impression d’une procédure	44

Chapitre 7. Utilisation des listes	45
Objets	45
Création de listes d’objets	45
Commandes de la fenêtre Liste	46
Création de listes	46
Ajout d’objets aux listes	47
Suppression d’objets des listes	47
Sauvegarde de listes dans des fichiers	47
Ouverture de fichiers de listes	47

Chapitre 8. Utilisation de fichiers de tâches	49
Fichiers de tâches	49
Création de fichiers de tâches	49
Exécution de fichiers de tâches	49
Ajustement automatique des colonnes et des lignes	49
Tri des résultats de requête	50
Reclassement des colonnes	50
Formatage des résultats de requête	50
Sélection de la police d’affichage des résultats d’une requête	50
Formatage des résultats numériques de requête	51

Conversion du formatage des résultats de requête en format	51
Regroupement et agrégation des résultats de requête.	51
Regroupement des résultats de requête	51
Résumé des résultats de requête	51
Enregistrement des résultats de requête et formatage.	51
Enregistrement des résultats de requête dans une table	52
Enregistrement des résultats de requête sous forme de fichiers.	52
Impression des résultats de requête	52
Prévisualisation des résultats de requête	52
Impression des résultats de requête	52

Chapitre 9. Utilisation des requêtes statiques. 55

Requêtes statiques	55
Création des requêtes statiques.	55
Remplacement des variables de substitution par des variables hôtes	56
Exécution d'une requête statique	57

Chapitre 10. Utilisation de l'éditeur de tables 59

Editeur de tables	59
Recherche de lignes au moyen de l'éditeur de tables	59
Ajout d'une ligne	60
Modification d'une ligne	60
Suppression d'une ligne	61
Modification de tables à partir de la vue des résultats de requête	61
Suppression d'une ligne à partir de la vue des résultats de requête	61
Mise à jour des colonnes à partir de la vue des résultats de requête	61
Formats DB2.	61

Chapitre 11. Distribution de données 63

Exportation des données	63
Exportation de données dans des fichiers	63
Importation de données	64
Enregistrement de données sur un serveur de base de données	65
Utilisation de la commande Envoyer à	65
Utilisation de l'extension Microsoft Excel	66
Utilisation des exemples d'applications	66

Chapitre 12. Utilisation du Centre d'édition de rapports QMF 67

Initiation	67
Fenêtre du Centre d'édition de rapports QMF	67
Connexion au serveur.	68
Utilisation des rapports et objets	69
Exécution de rapports.	70
Utilisation des dossiers et des favoris	70
Ajout de rapports aux favoris	71

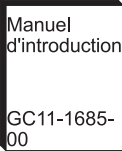
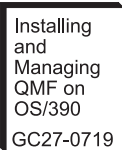













Chapitre 13. Utilisation de l'API QMF pour Windows. 73

Contrôle de QMF pour Windows par le biais de l'API	73
Blocage d'appels	74
Connexion à la base de données	74
Aide-mémoire sur les API	75
AddDecimalHostVariable()	75
AddHostVariable()	76
BindDecimalHostVariable()	77
BindHostVariable()	78
BindSection().	79
CancelBind()	79
ChangePassword()	80
ClearList().	80
Close().	81
Commit()	81
CompleteQuery()	82
CopyToClipboard().	82
DeleteQMFObject().	83
EndBind().	84
Execute()	84
ExecuteEx()	85
ExecuteStored Procedure()	86
ExecuteStored ProcedureEx()	87
Export()	89
ExportForm()	91
ExportReport()	92
FastSaveData()	94
FetchNextRow().	95
FetchNextRowEx()	96
FetchNextRows()	96
FetchNextRowsEx()	98
FlushQMFCache()	98
GetColumnCount().	99
GetColumnDataValue()	99
GetColumnHeader()	100
GetColumnHeaderEx()	100
GetColumnHeadings()	101

GetColumnValue()	102	InitializeServer()	130
GetColumnValueEx()	103	InitializeStaticQuery()	132
GetDefaultServerName()	103	IsStatic()	132
GetGlobalVariable()	103	Open()	132
GetHostVariableNames()	104	Prepare()	133
GetHostVariableTypes()	104	PrintReport()	134
GetLastErrorString()	105	ReinitializeServer()	134
GetLastErrorType()	105	Rollback()	135
GetLastSQLCode()	107	RunProc()	135
GetLastSQLError()	107	SaveData()	136
GetLastSQLState()	108	SaveQMFPProc()	138
GetOption()	109	SaveQMFPQuery()	139
GetOptionEx()	110	SetBindOption()	139
GetProcText()	111	SetBindOwner()	141
GetProcVariables()	111	SetBusyWindowButton()	142
GetQMFObjectInfo()	112	SetBusyWindowMessage()	143
GetQMFObjectInfoEx()	114	SetBusyWindowMode()	143
GetQMFObjectList()	116	SetBusyWindowTitle()	144
GetQMFObjectListEx()	117	SetGlobalVariable()	144
GetQMFPProcText()	118	SetHostVariable()	145
GetQMFPQueryText()	118	SetOption()	146
GetQueryText()	119	SetParent()	147
GetQueryVerb()	119	SetProcVariable()	147
GetResourceLimit()	120	SetVariable()	148
GetResourceLimitEx()	124	ShowBusyWindow()	149
GetRowCount()	125	StartBind()	149
GetServerList()	125		
GetServerListEx()	126	Annexe. Remarques	151
GetStoredProcedureResultSets()	127	Marques	154
GetVariables()	127		
GetVariablesEx()	128	Index	155
InitializeProc()	129		
InitializeQuery()	130		

Bibliothèque QMF

Pour commander des exemplaires de ces publications, prenez contact avec votre partenaire commercial IBM.

Evaluation	 <p>Manuel d'introduction</p> <p>GC11-1685- 00</p>			
Installation, planification, administration et diagnostic	 <p>Installing and Managing QMF on OS/390</p> <p>GC27-0719</p>	 <p>Installing and Managing QMF on VM/ESA</p> <p>GC27-0720</p>	 <p>Installing and Managing QMF on VSE/ESA</p> <p>GC27-0721</p>	 <p>Installing and Managing QMF for Windows</p> <p>GC27-0722</p>
	 <p>QMF Messages and Codes</p> <p>GC27-0717</p>	 <p>QMF High Performance Option User's Guide for OS/390</p> <p>SC27-0724</p>		
Utilisation de QMF	 <p>Manuel d'utilisation</p> <p>SC11-1687- 00</p>	 <p>QMF Reference</p> <p>SC27-0715</p>	 <p>Initiation à QMF pour Windows</p> <p>SC11-1694- 00</p>	
Développement d'applications	 <p>Developing QMF Applications</p> <p>SC27-0718</p>			
Bibliothèques électroniques				
	SK2T-0730 OS/390, VM, & VSE	SK2T-6700 OS/390 seulement	SK2T-2067 VM seulement	SK2T-0060 VSE seulement

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.

OS/2 et Windows - Paramètres canadiens








Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire

correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

Chapitre 1. Introduction

Ce chapitre présente QMF pour Windows et décrit certaines tâches de base nécessaires à l'exécution du produit.

Serveurs de base de données

Les requêtes, les formats, les procédures et les tables sont exécutés et enregistrés sur un serveur de base de données.

Bases de données DB2

QMF pour Windows se connecte à une large gamme de bases de données DB2.

- DB2 UDB pour OS/390, DB2 pour OS/390 et DB2 pour MVS
- Serveur DB2 pour VSE & VM et SQLDS
- DB2 Universal Database et DB2 Common Server
- DB2 Parallel Edition
- DataJoiner

La licence QMF pour Windows détermine les produits DB2 que vous pouvez installer et connecter à votre copie du programme QMF pour Windows.

Nom d'utilisateur et nom technique

Selon le type et la version de DB2, une base de données est désignée par un nom BDR (base de données relationnelles), un nom d'emplacement ou tout autre nom technique.

Avec QMF pour Windows, l'administrateur peut attribuer à une base de données un nom d'utilisateur plus facile à mémoriser, par exemple, Base de données des achats à la place de DB2P_01_PURCH.

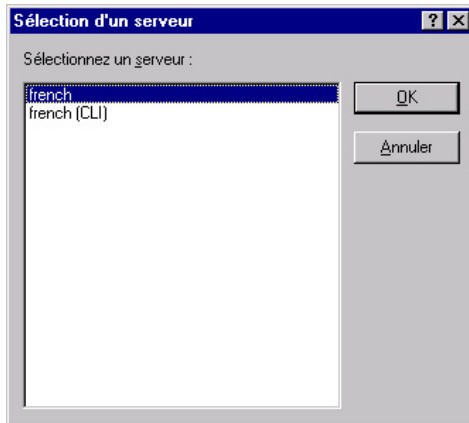
QMF pour Windows fait référence à un serveur de base de données ou à une base de données DB2 en utilisant le terme "serveur".

Définition du nom de serveur

Pour vous permettre d'interroger une base de données, QMF pour Windows doit savoir où celle-ci est stockée.

1. Dans le menu **Fichier**, cliquez sur **Nouvelle requête SQL**. Un nouveau document de requête SQL s'ouvre.

2. Dans le menu **Requête**, cliquez sur **Sélectionner un serveur**. La boîte de dialogue Sélection d'un serveur s'affiche.



3. Dans la liste des serveurs disponibles, sélectionnez celui que vous voulez interroger et cliquez sur **OK**. Lorsque vous ouvrez la session QMF pour Windows suivante, vous êtes automatiquement reconnecté au même serveur.

Sécurité de la base de données

Vous devez fournir un ID utilisateur et un mot de passe pour pouvoir vous connecter à un serveur.

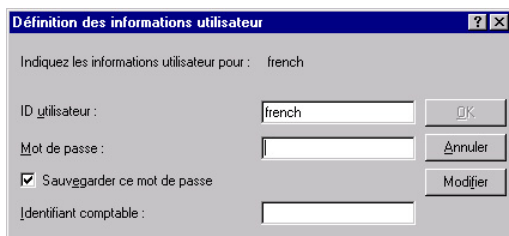
Connexion

Vous devez fournir un ID utilisateur et un mot de passe valides pour le serveur de base de données auquel vous tentez d'accéder. Cet ID utilisateur et ce mot de passe ne correspondent pas nécessairement à l'ID utilisateur et au mot de passe qui sont employés localement ou sur le réseau.

Si vous utilisez Windows, vous avez la possibilité de conserver les mots de passe du serveur d'une session QMF pour Windows. Si vous êtes actuellement connecté à Windows, la boîte de dialogue Définition des informations utilisateur présente une case à cocher supplémentaire appelée **Sauvegarder ce mot de passe**. Si vous cochez cette case, le mot de passe que vous entrez pour ce serveur sera stocké dans la liste des mots de passe de Windows. Par la suite, lorsque vous vous connecterez à Windows, QMF pour Windows pourra retrouver automatiquement ce mot de passe pour vous éviter de le spécifier. Par contre, si vous n'êtes pas connecté lorsque vous exécutez QMF pour Windows, ou si vous êtes connecté sous un autre nom d'utilisateur, QMF pour Windows demandera l'ID utilisateur et le mot de passe.

Remarque : Si vous choisissez d'enregistrer un mot de passe, tout utilisateur pouvant se connecter à votre compte Windows aura accès à vos serveurs en utilisant votre ID utilisateur (serveur) et votre mot de passe.

1. A partir du menu **Requête**, cliquez sur **Définir les infos utilisateur**. La boîte de dialogue Définition des informations utilisateur s'ouvre.



2. Entrez l'ID utilisateur et le mot de passe dans les zones appropriées.

Remarque : La distinction majuscule/minuscule s'applique à l'ID utilisateur et au mot de passe. Par exemple, si votre ID utilisateur ou votre mot de passe est défini en majuscules, vous devez l'entrer en majuscules. Certains types de serveur de base de données appliquent cette distinction dans l'ID utilisateur et le mot de passe, tandis que d'autres l'ignorent.

3. Cochez la case **Sauvegarder ce mot de passe** si vous voulez enregistrer l'ID utilisateur et le mot de passe.
4. Cliquez sur **OK**. QMF pour Windows stocke ces informations en vue de l'accès au serveur.

Correction des mots de passe

Si vous avez entré un mot de passe incorrect, vous pouvez corriger l'erreur en rouvrant la boîte de dialogue Définition des informations utilisateur.

1. A partir du menu **Requête**, cliquez sur **Définir les infos utilisateur**. La boîte de dialogue Définition des informations utilisateur s'ouvre.
2. Tapez de nouveau le mot de passe et cliquez sur **OK**. Le mot de passe est corrigé.

Modification des mots de passe

Vous pouvez changer votre mot de passe sur le serveur de base de données à partir de QMF pour Windows. Actuellement, cette fonction est uniquement prise en charge par DB2 pour OS/390 version 5 et ultérieure.

1. A partir du menu **Requête**, cliquez sur **Définir les infos utilisateur**. La boîte de dialogue Définition des informations utilisateur s'ouvre.
2. Cliquez sur **Changer**. Les zones Nouveau mot de passe et Confirmation du nouveau mot de passe s'affichent.

3. Entrez le nouveau mot de passe dans les zones **Nouveau mot de passe** et **Confirmer nouveau mot de passe** et cliquez sur **OK**. Le mot de passe est remplacé.

Définition des identifiants comptables

Le serveur de base de données utilise des identifiants comptables pour contrôler l'utilisation du système. Adressez-vous à l'administrateur de la base de données pour savoir si votre système fait appel à ce type de chaîne.

1. A partir du menu **Requête**, cliquez sur **Définir les infos utilisateur**. La boîte de dialogue Définition des informations utilisateur s'ouvre.
2. Dans la zone **Identifiant comptable**, entrez l'identifiant de votre choix et cliquez sur **OK**. QMF pour Windows stocke ces informations en vue de l'accès au serveur.

Gestion

Le gestionnaire de QMF pour Windows est toujours exécuté en arrière-plan, contrôlant l'utilisation des ressources de la base de données et du système. Il impose également des restrictions quant au type et à la taille des requêtes que vous pouvez exécuter.

Affichage des limites de ressources

Dans le menu **Affichage**, cliquez sur **Limites de ressources**. La boîte de dialogue correspondante s'ouvre. Toutes les informations de cette boîte de dialogue sont accessibles en lecture uniquement. C'est l'administrateur système qui définit ces limites.



Les types de limites et de contrôles pouvant être appliqués sont les suivants :

- Délais
- Limites
- Instructions SQL
- Options
- Enregistrement des données
- Edition de liens
- Suivi des objets

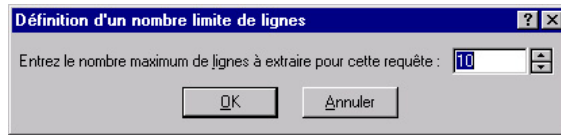
Définition d'un nombre limite de lignes

Vous pouvez indiquer le nombre maximum de lignes pouvant être extraites pour une requête. Une fois cette limite atteinte, QMF pour Windows annule la requête. Le nombre maximal de lignes autorisé dans le groupe de limites de ressources est prioritaire sur ce paramètre.

Entrez 0 si vous ne voulez pas spécifier de limite dans cette zone.

Les lignes que QMF pour Windows a déjà extraites au-delà de cette limite sont conservées et peuvent être affichées.

1. Dans le menu **Requête**, cliquez sur **Définir un nombre limite de lignes**. La boîte de dialogue Définition d'un nombre limite de lignes s'affiche.



2. Entrez le nombre maximum de lignes à extraire pour cette requête et cliquez sur **OK**. Le nombre limite de lignes est appliqué à la prochaine exécution de la requête.

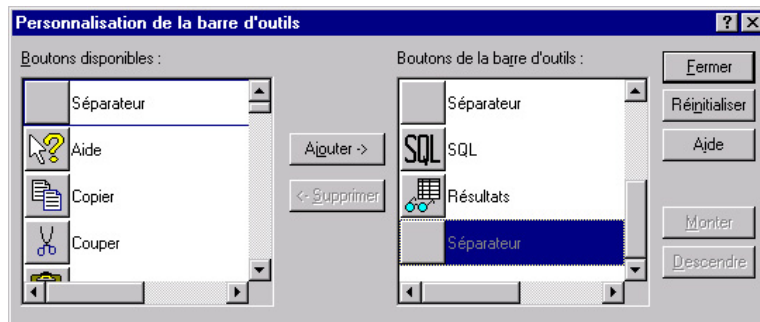
Personnalisation de la barre d'outils

Vous pouvez personnaliser la barre d'outils pour afficher uniquement les boutons de votre choix.

Ajout de boutons à la barre d'outils

Vous avez la possibilité d'ajouter des boutons à la barre d'outils QMF pour Windows existante. Ces boutons représentent des fonctions dont tous les utilisateurs n'ont pas forcément besoin, mais qui peuvent être incluses dans la barre d'outils.

1. Cliquez deux fois sur la zone grise entourant la barre d'outils. La boîte de dialogue Personnalisation de la barre d'outils s'ouvre.



2. Dans la colonne **Boutons disponibles**, sélectionnez le bouton à ajouter et cliquez sur **Ajouter**. Le bouton est ajouté à la barre d'outils.
3. Une fois que vous avez ajouté tous les boutons, cliquez sur **Fermer**. La boîte de dialogue se referme et les nouveaux boutons sont ajoutés à la barre d'outils.

Déplacement de boutons sur la barre d'outils

Vous avez la possibilité de réordonner les boutons de la barre d'outils QMF pour Windows.

1. Cliquez deux fois sur la zone grise entourant la barre d'outils. La boîte de dialogue Personnalisation de la barre d'outils s'ouvre.
2. Dans la colonne **Boutons disponibles**, sélectionnez le bouton à déplacer.
3. Utilisez les boutons **Monter** et **Descendre** pour déplacer le bouton dans la barre d'outils.
4. Une fois que vous avez déplacé tous les boutons, cliquez sur **Fermer**. La boîte de dialogue se referme et les boutons s'affichent dans leur nouvel emplacement.

Suppression de boutons de la barre d'outils

Vous avez la possibilité de supprimer des boutons de la barre d'outils QMF pour Windows.

1. Cliquez deux fois sur la zone grise entourant la barre d'outils. La boîte de dialogue Personnalisation de la barre d'outils s'ouvre.
2. Dans la colonne Boutons de la barre d'outils, sélectionnez le bouton à supprimer et cliquez sur **Supprimer**. Ce bouton est supprimé de la barre d'outils.
3. Une fois que vous avez supprimé les boutons, cliquez sur **Fermer**. La boîte de dialogue se referme et les boutons sont supprimés de la barre d'outils.

Chapitre 2. Utilisation des requêtes SQL

SQL (Structured Query Language) est l'interface de base qui met en contact un utilisateur et une base de données. Les requêtes sont formulées dans le langage SQL et traitées par la base de données. Les utilisateurs peuvent écrire des requêtes QMF pour Windows en langage SQL, ou créer des requêtes au moyen de la souris.

Requêtes SQL

Pour créer des requêtes SQL (Structured Query Language), vous devez connaître les commandes et la syntaxe SQL. Les utilisateurs qui ne sont pas familiarisés avec SQL doivent essayer de créer des requêtes assistées.

Création de requêtes SQL

Cliquez sur le bouton **Nouvelle requête SQL** dans la barre d'outils.



Un nouveau document de requête s'ouvre.

Exécution de requêtes SQL sur un serveur de base de données

1. Ouvrez un nouveau document de requête et entrez une requête, ou bien ouvrez un fichier de requête existant, ou encore ouvrez une requête dans la base de données.
2. Cliquez sur le bouton **Exécuter requête** dans la barre d'outils.



La requête s'exécute et ses résultats s'affichent.

Navigation entre la fenêtre des résultats et la fenêtre SQL

Vous pouvez afficher les résultats d'une requête ou l'instruction SQL elle-même.

Dans la fenêtre SQL d'une requête exécutée, cliquez sur le bouton **Résultats** de la barre d'outils.



Les résultats de la requête s'affichent.

Autre possibilité :

Cliquez sur le bouton **SQL** dans la fenêtre des résultats d'une requête.



L'instruction SQL s'affiche.

Utilisation des polices

Vous pouvez changer la police utilisée pour afficher des requêtes. Les polices disponibles varient en fonction de l'installation effectuée sur l'ordinateur. Pour de plus amples informations sur l'ajout de polices, reportez-vous à l'aide du système d'exploitation.

Remarque : Si vous enregistrez la requête après avoir sélectionné une nouvelle police d'affichage de requête, elle s'affiche avec la nouvelle police.

Sélection de la police d'affichage d'une requête

1. A partir de la fenêtre SQL, cliquez sur **Définir la police** dans le menu **Requête**. La boîte de dialogue Police s'ouvre.
2. Sélectionnez la police que vous voulez utiliser pour afficher le texte de la requête, puis cliquez sur **OK**. La requête s'affiche à nouveau dans la nouvelle police.

Remarque : Cliquez sur **Définir par défaut** afin d'utiliser la police sélectionnée comme police par défaut pour toutes les nouvelles requêtes.

Requêtes multiples

Vous pouvez ouvrir plusieurs documents de requête en même temps. Vous avez aussi la possibilité d'exécuter plusieurs requêtes simultanément. Cette fonction permet de générer plusieurs rapports, ou de couper et de coller le texte SQL d'une requête dans une autre requête.

Affichage simultané de plusieurs requêtes

1. Ouvrez au moins deux documents de requête.

2. Dans le menu **Fenêtre**, sélectionnez une des commandes suivantes :

Commande	Résultat
Cascade	Affiche les requêtes empilées en cascade.
Empilement vertical	Affiche les fenêtres de requête empilées verticalement.
Empilement horizontal	Affiche les fenêtres de requête empilées côte à côte.

Les fenêtres de requête sont disposées selon l'option sélectionnée.

Création de requêtes

La commande Créer une requête permet de créer des documents de requête SQL. Il suffit de spécifier un ou plusieurs noms de table et le type d'instruction SQL de votre choix pour que QMF pour Windows crée automatiquement une instruction SQL faisant référence aux noms et types de données des colonnes de la table.

Création de requêtes SQL

1. Dans le menu **Fichier**, cliquez sur **Créer une requête**. La boîte de dialogue Création d'une requête s'ouvre.

Création d'une requête [?] [X]

Serveur : french [v] [OK]

Type de requête

Sélection Mise à jour Insertion [Annuler]

Tables des requêtes

Défendeur : [] [Ajouter]

Nom : [] [Ajouter à partir de la liste...]

Identificateur : []

Tables :

Défendeur	Nom	Identificateur
-----------	-----	----------------

[Remonter] [Descendre] [Supprimer]

2. Sélectionnez le type de requête à créer.

Type de requête	Résultat
Sélection	Extrait des lignes d'une ou de plusieurs tables.
Mise à jour	Modifie les informations contenues dans une table.
Insertion	Ajoute des lignes à une table.

3. Indiquez le détenteur et le nom de la table à interroger.

Remarque : Vous pouvez utiliser des modèles pour sélectionner des noms dans une liste de tables répondant aux critères de recherche.

- Le caractère de pourcentage (%) est utilisé pour remplacer une chaîne de longueur indéfinie, contenant n'importe quel caractère. Par exemple, entrez A% pour répertorier toutes les tables dont le nom commence par la lettre A.
- Le caractère de soulignement (_) est utilisé pour remplacer un seul caractère. Par exemple, entrez _A% pour répertorier toutes les tables dont le détenteur a un nom comportant la lettre A en deuxième position.

Après avoir indiqué un modèle, cliquez sur **Ajouter de la liste** et sélectionnez une table dans la liste obtenue.

4. Entrez un identificateur unique pour la table.

5. Cliquez sur **Ajouter**. La table est ajoutée à la requête.

6. Une fois que vous avez ajouté la table à interroger, cliquez sur **OK**. Une requête SQL portant sur les tables sélectionnées est créée et affichée.

Variables de substitution dans des requêtes SQL

Grâce aux variables de substitution, vous pouvez utiliser la même requête pour extraire des informations différentes, en fournissant des valeurs distinctes à chaque exécution de la requête. Pour ce faire, il n'est pas nécessaire de réécrire la requête. Il suffit d'indiquer des valeurs différentes pour les variables de substitution lorsque vous exécutez la requête.

Une variable de substitution est un texte que vous insérez dans une requête. Elle doit commencer par une perluète (&) et peut contenir au maximum 18 caractères comprenant des lettres alphabétiques, des chiffres ou un des caractères spéciaux suivants : ^ ! \$ % & ' { } ? @ # % \ ou _ . Par exemple, les valeurs suivantes sont des variables de substitution valides :

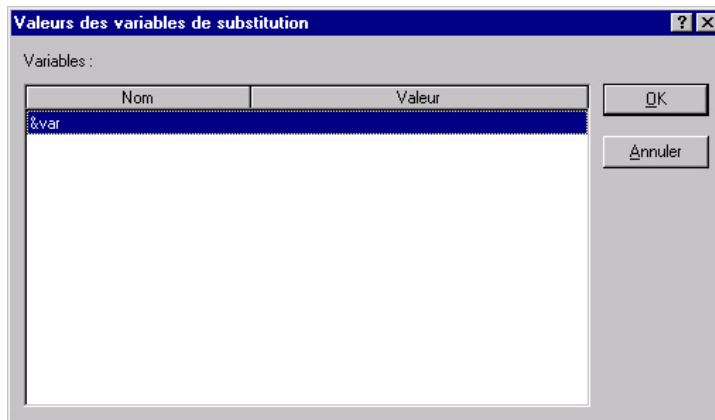
```
&VARIABLE1  
&ID_DEPARTEMENT
```


L'emplacement d'une variable dans une requête est indifférent ; elle peut remplacer tout ce qui peut être écrit dans une requête (à l'exception d'un commentaire). Par exemple, vous pouvez utiliser une variable pour remplacer un nom de colonne, un critère de recherche, une sous-requête ou une valeur donnée.

Exécution de requêtes SQL contenant des variables de substitution

1. Ouvrez un nouveau document de requête et entrez l'instruction SQL suivante :

```
SELECT * FROM Q.PERS WHERE DEPT  
>= &DEPT_MIN
```
2. Exécutez la requête. La boîte de dialogue Valeurs des variables de substitution s'ouvre.



3. Dans la zone **Valeur**, entrez la valeur 50 et cliquez sur **OK**. La requête est exécutée et les résultats s'affichent.

Entraînez-vous à utiliser les variables de substitution en remplaçant les valeurs des clauses SELECT et FROM. Comparez les résultats obtenus en fonction des valeurs entrées.

Sauvegarde et ouverture des requêtes SQL

Vous pouvez sauvegarder les requêtes sur votre PC, sur un serveur de fichiers ou sur un serveur de base de données.

Enregistrement des requêtes SQL dans des fichiers

1. A partir d'une requête ouverte, cliquez sur le bouton **Sauvegarder** dans la barre d'outils.



Si la requête a déjà été enregistrée, elle est enregistrée à nouveau. Si elle n'a jamais été enregistrée, la boîte de dialogue Sauvegarder sous s'ouvre.

2. Entrez le nom du fichier où vous voulez stocker la requête, puis cliquez sur **OK**. La requête est enregistrée.

Ouverture de fichiers de requêtes SQL

1. Cliquez sur le bouton **Ouvrir** dans la barre d'outils.



La boîte de dialogue Ouvrir s'affiche.

2. Sélectionnez le fichier à ouvrir et cliquez sur **OK**. La requête sélectionnée s'ouvre dans un nouveau document de requête.

Enregistrement de requêtes SQL sur le serveur de base de données

Les requêtes enregistrées sur le serveur peuvent être rendues accessibles aux autres utilisateurs. Si vous voulez partager vos requêtes avec d'autres utilisateurs, enregistrez-les sur le serveur de base de données.

1. A partir d'une requête ouverte, cliquez sur le bouton **Sauvegarder sur le serveur** dans la barre d'outils.



La boîte de dialogue correspondante s'ouvre.

2. Indiquez un détenteur et un nom, puis spécifiez si la requête enregistrée doit être partagée avec d'autres utilisateurs et cliquez sur **OK**. La requête est enregistrée sur le serveur.

Si une requête portant ce nom existe déjà, vous êtes invité à la remplacer.

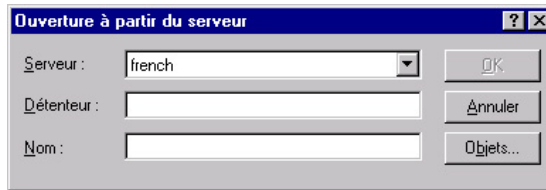
Ouverture de requêtes SQL enregistrées sur le serveur de base de données

Vous pouvez ouvrir des requêtes enregistrées sur le serveur de base de données.

1. Cliquez sur le bouton **Ouvrir depuis serveur** dans la barre d'outils.



La boîte de dialogue Ouverture à partir du serveur s'affiche.



2. Indiquez un serveur, un détenteur et un nom, puis cliquez sur **OK**. La requête SQL s'ouvre.

Impression de requêtes SQL

Vous pouvez prévisualiser et imprimer vos requêtes SQL.

Prévisualisation d'une requête

1. Ouvrez une requête et activez la fenêtre SQL. L'instruction SQL s'affiche.
2. Dans le menu **Fichier**, cliquez sur **Mise en page**. La boîte de dialogue correspondante s'ouvre.
3. Modifiez à votre convenance la présentation de la page et cliquez sur **OK**.
4. Cliquez sur le bouton **Aperçu avant Impression** dans la barre d'outils.



Un aperçu de la requête imprimée s'affiche.

Impression de requêtes SQL

1. Ouvrez une requête et activez la fenêtre SQL. L'instruction SQL s'affiche.
2. Dans le menu **Fichier**, cliquez sur **Mise en page**. La boîte de dialogue correspondante s'ouvre.
3. Modifiez à votre convenance la présentation de la page et cliquez sur **OK**.
4. Cliquez sur le bouton **Imprimer** dans la barre d'outils.



La requête est imprimée.

Chapitre 3. Utilisation des requêtes assistées

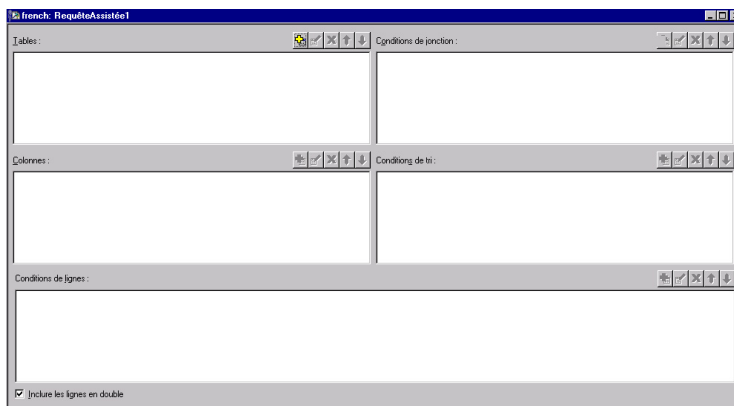
Les requêtes assistées vous permettent de créer facilement une requête en sélectionnant des options dans des menus et des listes. Une fois que vous avez créé une requête assistée, vous pouvez l'enregistrer ou la convertir en requête SQL.

Création de requêtes simples

Vous pouvez facilement élaborer des requêtes simples au moyen de l'interface de création de requêtes assistées.



Ouverture de nouvelles requêtes assistées



- Dans le menu **Fichier**, cliquez sur **Nouvelle requête assistée**. Un nouveau document de requête assistée s'ouvre.



Boutons d'action associés aux requêtes assistées

Vous pouvez modifier les requêtes assistées au moyen des boutons d'action. Une série de boutons apparaît au-dessus de la section concernée.

Bouton d'action associé à une requête assistée	Aspect	Résultat
Ajouter		Cliquez sur ce bouton pour ajouter un élément à la requête assistée.
Modifier		Cliquez sur ce bouton pour modifier l'élément en surbrillance dans la requête.

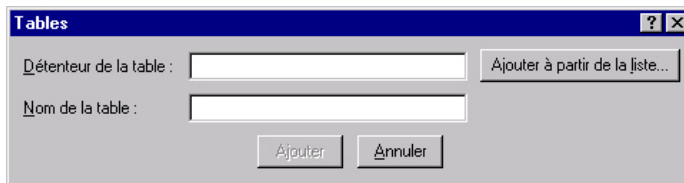
Supprimer		Cliquez sur ce bouton pour supprimer l'élément sélectionné.
Remonter et Descendre		Cliquez sur ce bouton pour déplacer l'élément sélectionné vers le haut et vers le bas dans la requête assistée.

Ajout de tables à des requêtes assistées

1. Dans la section Tables du document de requête assistée, cliquez sur le bouton **Ajouter**.



La boîte de dialogue Tables s'ouvre.



- 2.

Indiquez le détenteur et le nom de la table à ajouter, puis cliquez sur **Ajouter**. La table est ajoutée à la requête.

Remarque : Vous pouvez utiliser des modèles pour sélectionner des objets dans une liste d'objets répondant aux critères de recherche.

- Le caractère de pourcentage (%) est utilisé pour remplacer une chaîne de longueur indéfinie, contenant n'importe quel caractère. Par exemple, entrez A% pour répertorier toutes les tables dont le nom commence par la lettre A.
- Le caractère de soulignement (_) est utilisé pour remplacer un seul caractère. Par exemple, entrez _A% pour répertorier toutes les tables dont le détenteur a un nom comportant la lettre A en deuxième position.

Après avoir indiqué un modèle, cliquez sur **Ajouter de la liste** et sélectionnez une table dans la liste obtenue.

3. Ajoutez d'autres conditions de tables à la requête, puis cliquez sur **Fermer**. Le document de requête assistée s'affiche avec les nouvelles tables.

Exécution de requêtes assistées

Une requête assistée s'exécute de la même façon qu'une requête SQL. Cliquez sur le bouton Exécuter requête dans la barre d'outils.



La requête assistée s'exécute.

Création de requêtes complexes

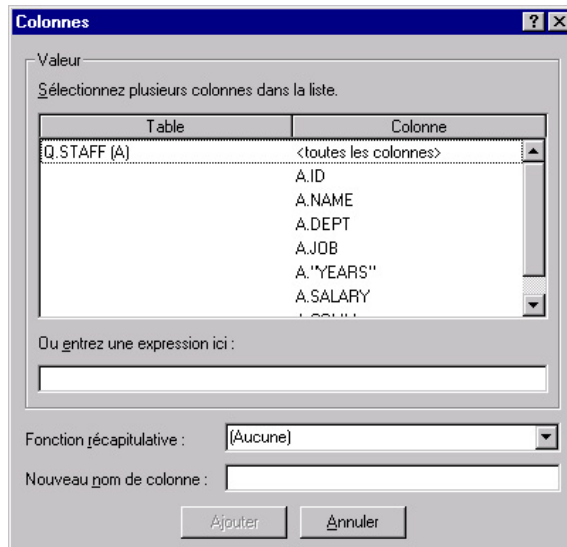
Vous pouvez également élaborer des requêtes plus complexes au moyen de l'interface de création de requêtes assistées.

Ajout de colonnes à des requêtes assistées

1. Dans la section Colonnes du document de requête assistée, cliquez sur le bouton Ajouter.



La boîte de dialogue Colonnes s'ouvre.



2. Sélectionnez la colonne à ajouter, puis cliquez sur **Ajouter**. La colonne est ajoutée à la requête assistée.
3. Ajoutez d'autres colonnes à la requête, puis cliquez sur Fermer. Le document de requête assistée s'affiche avec les nouvelles colonnes.

Remarque : Vous pouvez appliquer une fonction récapitulative à la colonne en effectuant une sélection dans la zone correspondante. Les fonctions de synthèse suivantes sont disponibles : MOYENNE, NOMBRE, MAXIMUM, MINIMUM et SOMME.

Remarque : Vous pouvez renommer une colonne dans la requête en entrant un nouveau nom dans la zone **Nouveau nom de colonne**.

Utilisation de conditions de tri

Les conditions de tri permettent d'indiquer la façon dont vous souhaitez trier les lignes d'une requête. Le tri peut être effectué dans un ordre croissant (de A à Z) ou décroissant (de Z à A).

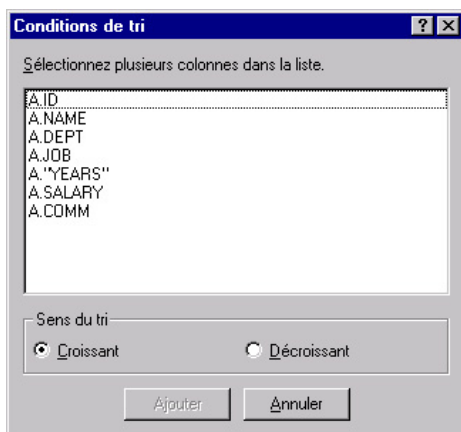
Si vous triez les lignes sur plusieurs colonnes, la première colonne est ordonnée en premier, la deuxième est triée dans le même ordre que la première, et ainsi de suite.

Ajout de conditions de tri

1. Dans la section Conditions de tri du document de requête assistée, cliquez sur le bouton **Ajouter**.



La boîte de dialogue Conditions de tri s'affiche.



2. Sélectionnez la colonne que vous souhaitez trier et l'ordre du tri, puis cliquez sur Ajouter. La condition de tri est ajoutée à la requête assistée.
3. Ajoutez d'autres conditions de tri à la requête, puis cliquez sur **Fermer**. Le document de requête assistée s'affiche avec les nouvelles conditions de tri.

Utilisation de conditions de lignes

Il arrive souvent que vous ayez besoin d'afficher uniquement certaines lignes d'une table. Pour sélectionner les lignes à afficher, il suffit d'ajouter des conditions de lignes. Si vous n'utilisez pas de condition de ligne, toutes les lignes de la table s'affichent.

Les conditions de lignes suivantes sont disponibles :

- Egal à
- Inférieur à
- Inférieur ou égal à
- Supérieur à
- Supérieur ou égal à
- Entre
- Commencant par
- Se terminant par
- Contenant
- Nul

Les conditions de lignes sont contrôlées par les opérateurs suivants :

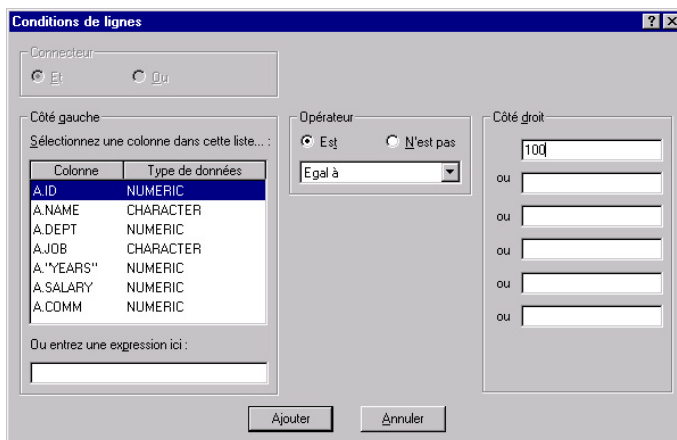
- est
- n'est pas

Ajout de conditions de lignes

1. Dans la section Conditions de lignes du document de requête assistée, cliquez sur le bouton **Ajouter**.



La boîte de dialogue Conditions de lignes s'affiche.



2. Sélectionnez les parties de l'instruction conditionnelle, puis cliquez sur **Ajouter**.

Partie de la condition de ligne	Fonction
Côté gauche	Sélectionnez la colonne que vous souhaitez examiner.
Opérateur	Déterminez la relation entre les parties gauche et droite de la ligne.
Côté droit	Entrez la condition que vous souhaitez vérifier.

La condition de ligne est ajoutée à la requête assistée.

3. Ajoutez d'autres conditions de lignes à la requête, puis cliquez sur **Fermer**. Le document de requête assistée s'affiche avec les nouvelles conditions de lignes.

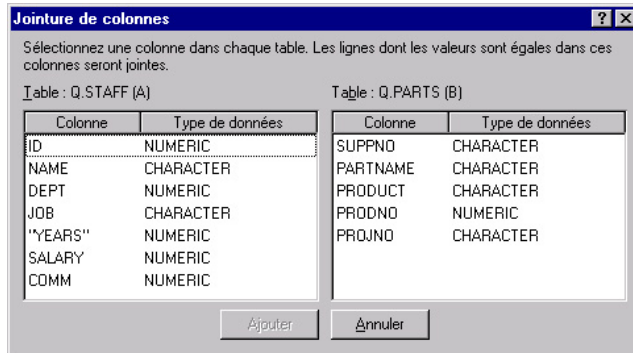
Utilisation de plusieurs tables dans les requêtes assistées

Vous pouvez inclure des informations provenant de plusieurs tables dans une requête assistée.

Vous devez mettre les deux tables en relation en spécifiant une ou plusieurs colonnes de jonction dans chacune d'elles. Seules les lignes qui proviennent de tables contenant des colonnes de jonction identiques sont incluses dans les résultats. Les types de données des colonnes d'une condition de jonction doivent concorder. Une fois que vous avez spécifié une relation entre deux colonnes, QMF pour Windows mémorise cette relation et la propose dans les requêtes ultérieures, facilitant et améliorant ainsi la création d'autres requêtes.

Création de conditions de jonction dans une requête assistée

1. Dans la section Tables de la fenêtre Requête assistée, cliquez sur le bouton **Ajouter** pour ajouter au moins deux tables. Si ces tables n'ont jamais été jointes, la boîte de dialogue Jointure de tables s'affiche. Si les tables ont déjà été jointes, QMF pour Windows propose la condition de jonction utilisée précédemment.



2. Sélectionnez dans chaque table une colonne contenant des types de données identiques, puis cliquez sur **Ajouter**. La nouvelle condition de jonction apparaît dans la requête assistée.

Requêtes assistées et requêtes SQL

Vous pouvez utiliser l'interface de création des requêtes assistées pour vous familiariser avec SQL.

Affichage des instructions SQL associées aux requêtes assistées

A partir de la fenêtre Requête assistée, cliquez sur le bouton **SQL** dans la barre d'outils.



L'instruction SQL équivalente à la requête assistée s'affiche. Il est impossible de modifier cette instruction dans cette vue.

Conversion de requêtes assistées en requêtes SQL

Vous pouvez convertir une requête assistée en un nouveau document de requête SQL. La nouvelle requête SQL peut alors être modifiée, enregistrée, imprimée, et exécutée. Dans le menu **Requête**, cliquez sur **Convertir en requête SQL**. La requête est convertie en un nouveau document de requête SQL.

Utilisation des variables de substitution dans les requêtes assistées

Les variables de substitution peuvent être utilisées dans une requête assistée de la même manière que dans une requête SQL. Voir la section "Utilisation des Variables de substitution dans des requêtes SQL".

Par exemple, vous pouvez employer des variables de substitution dans :

- une condition de ligne
DEPT Est supérieur ou égal à
&DeptMin
- une spécification de colonne
&NumEntré

Enregistrement de requêtes assistées

Vous pouvez sauvegarder les requêtes assistées dans des fichiers sur votre PC, sur un serveur de fichiers ou sur un serveur de base de données.

Enregistrement de requêtes assistées dans des fichiers

1. A partir d'une requête assistée ouverte, cliquez sur le bouton **Sauvegarder** dans la barre d'outils.



Remarque : Si la requête a déjà été enregistrée, elle est enregistrée à nouveau. Si elle n'a jamais été enregistrée, la boîte de dialogue Sauvegarder sous s'ouvre.

2. Entrez le nom du fichier où vous voulez stocker la requête assistée et cliquez sur **OK**. La requête est enregistrée.

Ouverture de fichiers de requêtes assistées

1. Cliquez sur le bouton **Ouvrir** dans la barre d'outils.



La boîte de dialogue Ouvrir s'affiche.

2. Sélectionnez le fichier à ouvrir et cliquez sur **OK**. La requête assistée sélectionnée s'ouvre dans un nouveau document de requête.

Enregistrement de requêtes assistées sur le serveur de base de données

1. A partir d'une requête assistée ouverte, cliquez sur le bouton **Sauvegarder sur le serveur** dans la barre d'outils.



La boîte de dialogue correspondante s'ouvre.



2. Indiquez un détenteur et un nom, puis spécifiez si la requête enregistrée doit être partagée avec d'autres utilisateurs et cliquez sur **OK**. La requête est enregistrée sur le serveur.

Si une requête portant ce nom existe déjà, vous êtes invité à la remplacer.

Ouverture de requêtes assistées enregistrées sur le serveur de base de données

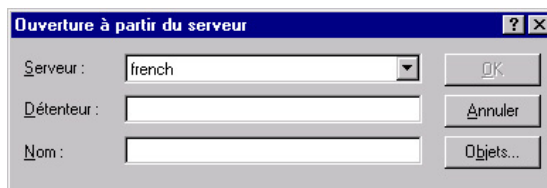
Vous pouvez ouvrir des requêtes assistées qui ont été enregistrées sur le serveur de base de données.

- 1.

Cliquez sur le bouton **Ouvrir depuis serveur** dans la barre d'outils.



La boîte de dialogue Ouverture à partir du serveur s'affiche.



2. Indiquez un serveur, un détenteur et un nom, puis cliquez sur **OK**. La requête assistée s'ouvre.

Impression de requêtes assistées

Vous pouvez imprimer une requête assistée. Vous avez également la possibilité d'imprimer le texte SQL d'une requête assistée. Voir la section «Impression de requêtes SQL» à la page 15.

Prévisualisation des requêtes assistées

Vous pouvez prévisualiser les résultats ou le texte d'une requête assistée avant de l'imprimer.

1. Ouvrez une requête et activez la vue en mode assisté. La requête s'affiche.
2. Dans le menu **Fichier**, cliquez sur **Mise en page**. La boîte de dialogue correspondante s'ouvre.
3. Modifiez à votre convenance la présentation de la page et cliquez sur **OK**.
4. Cliquez sur le bouton **Aperçu avant Impression** dans la barre d'outils.



Un aperçu de la requête imprimée s'affiche.

Chapitre 4. Utilisation des résultats de requête

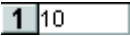



Vous pouvez directement formater, regrouper et agréger les résultats de la requête. Ce formatage peut être enregistré avec la requête ou être exporté dans un format.

Tri et classement des résultats de requête

Les utilisateurs peuvent sélectionner, redimensionner, réordonner, et trier les résultats d'une requête.

Sélection de colonnes et de lignes

Une fois que vous avez exécuté une requête, vous pouvez utiliser les contrôles de la fenêtre des résultats pour modifier et sélectionner des informations.

Boutons de sélection de colonnes et de lignes	Aspect	Fonction
Bouton de sélection de ligne		Cliquez sur ce bouton pour sélectionner toutes les données d'une ligne.
Bouton de sélection de colonne		Cliquez sur ce bouton pour sélectionner toutes les données d'une colonne.
Cellule		Cliquez directement sur la cellule pour la sélectionner.
Boutons de défilement vers le haut et vers le bas		Cliquez sur ces boutons pour faire défiler une liste de résultats de requête jusqu'au début ou jusqu'à la fin.

Redimensionnement de colonnes et de lignes

Vous pouvez changer l'aspect d'une liste de résultats de requête en redimensionnant les colonnes et les lignes.

1. Sélectionnez au moyen de la souris la ligne noire séparant deux colonnes ou deux lignes.
2. Faites glisser la ligne de séparation d'un côté à l'autre, ou vers le haut et vers le bas, pour redimensionner la colonne ou la ligne.

Remarque : Si vous enregistrez la requête après avoir redimensionné les lignes et les colonnes, elle s'affiche avec la nouvelle mise en forme.

Ajustement automatique des colonnes et des lignes

Vous pouvez redimensionner automatiquement des colonnes et des lignes pour les ajuster aux données qu'elles contiennent.

Sélectionnez toute une colonne ou ligne en cliquant deux fois sur la ligne noire la séparant de l'objet adjacent. La colonne est automatiquement redimensionnée en fonction des données qu'elle contient.

Remarque : Si vous enregistrez la requête après avoir redimensionné les lignes et les colonnes, elle s'affiche avec la nouvelle mise en forme.

Tri des résultats de requête

Une fois que vous avez exécuté une requête, vous pouvez trier les résultats de chaque colonne par ordre alphabétique.

Dans la fenêtre des résultats d'une requête, sélectionnez une colonne et cliquez sur le bouton **Trier dans l'ordre croissant** dans le menu résultat.

Les résultats de la requête sont triés par ordre croissant.

Autre possibilité :

Dans la fenêtre des résultats d'une requête, sélectionnez une colonne et cliquez sur le bouton **Trier dans l'ordre Décroissant** dans le menu résultats.

Les résultats de la requête sont triés par ordre décroissant.

Remarque : Pour trier les colonnes sélectionnées de façon plus complexe, sélectionnez **Trier** dans le menu **Résultats**.

Reclassement des colonnes

Vous pouvez modifier l'ordre des colonnes dans les résultats de la requête.

Dans la fenêtre des résultats d'une requête, sélectionnez une colonne et déplacez-la.

La colonne apparaît reclassée.

Formatage des résultats de requête

Vous pouvez changer la police utilisée pour afficher des requêtes et des résultats de requête. Les polices disponibles varient en fonction de l'installation effectuée sur l'ordinateur. Pour de plus amples informations sur l'ajout de polices, reportez-vous à l'aide du système d'exploitation.

Remarque : Si vous enregistrez la requête après avoir sélectionné une nouvelle police d’affichage de requête ou de résultat de requête, elle s’affiche avec la nouvelle police.

Sélection de la police d’affichage des résultats d’une requête

1. A partir de la fenêtre des résultats, cliquez sur **Définir les polices** dans le menu **Résultats**. La boîte de dialogue Police s’ouvre.
2. Sélectionnez la police et la taille d’affichage des résultats de la requête et cliquez sur **OK**. Les résultats de la requête sont affichés dans le format sélectionné.

Remarque : Cliquez sur **Définir par défaut** afin d’utiliser la police sélectionnée comme police par défaut pour tous les résultats de requête.

Formatage des résultats numériques de requête

1. Dans la fenêtre Résultats, sélectionnez une colonne contenant des valeurs numériques et sélectionnez **Format** dans le menu **Résultats**. La boîte de dialogue Format s’ouvre.
2. Sélectionnez le formatage à appliquer et cliquez sur **OK**. Les valeurs sont formatées selon la sélection.

Remarque : Cliquez sur **Définir par défaut** afin d’utiliser la police sélectionnée comme police par défaut pour tous les résultats de requête.

Conversion du formatage des résultats de requête en format

Vous pouvez convertir le formatage des résultats de requête en format.

1. Dans le menu **Résultats**, sélectionnez **Afficher un rapport**.
La boîte de dialogue Sélection d’un format s’ouvre.
2. Sélectionnez **A partir de la requête** et cliquez sur **OK**.
Le formatage des résultats de la requête est converti en format et s’ouvre dans une nouvelle fenêtre de format.

Regroupement et agrégation des résultats de requête

Les résultats de requête peuvent être regroupés, agrégés et résumés.

Reroupeement des résultats de requête

Vous pouvez regrouper les résultats de requête avec ou sans information récapitulative.

1. Sélectionnez la colonne à regrouper.
2. Dans le menu **Résultats**, sélectionnez le type de regroupement à appliquer.
La colonne est regroupée selon la sélection.

Résumé des résultats de requête

Vous pouvez résumer les résultats de requête par colonne.

1. Sélectionnez la colonne à regrouper.
2. Dans le menu **Résultats**, sélectionnez le type de résumé à appliquer.
La colonne est résumée selon la sélection.

Enregistrement des résultats de requête et formatage

Vous pouvez enregistrer les résultats de requête et enregistrer le formatage en format.

Enregistrement des résultats de requête dans une table

Vous pouvez enregistrer les résultats de requête sous forme de table sur un serveur de base de données.

1. Dans le menu **Résultats**, sélectionnez **Sauvegarder dans la base de données**.

La boîte de dialogue correspondante s'ouvre.

2. Indiquez un serveur, un détenteur et un nom de table, puis cliquez sur **OK**.

Les résultats de requête sont enregistrés sous forme de table dans la base de données.

Enregistrement des résultats de requête sous forme de fichiers

Vous pouvez enregistrer les résultats de requête sous forme de fichier sur votre PC ou sur un serveur de fichiers.

1. Dans le menu **Résultats**, sélectionnez **Sauvegarder dans le fichier**.

La boîte de dialogue Exportation des données s'ouvre.

2. Indiquez l'emplacement de sauvegarde du fichier, toutes les options d'exportation, puis cliquez sur **OK**.

Les résultats de requête sont enregistrés dans un fichier.

Impression des résultats de requête

Vous pouvez prévisualiser et imprimer vos résultats de requête.

Prévisualisation des résultats de requête

1. Ouvrez et exécutez une requête. Les résultats de la requête s'affichent.
2. Dans le menu **Fichier**, cliquez sur **Mise en page**. La boîte de dialogue correspondante s'ouvre.
3. Modifiez à votre convenance la présentation de la page et cliquez sur **OK**.

4. Cliquez sur le bouton **Aperçu avant Impression** dans la barre d'outils.



Un aperçu des résultats imprimés s'affiche.

Impression des résultats de requête

1. Ouvrez une requête et activez la fenêtre des résultats. Les résultats de la requête s'affichent.
2. Dans le menu **Fichier**, cliquez sur **Mise en page**. La boîte de dialogue correspondante s'ouvre.
3. Modifiez à votre convenance la présentation de la page et cliquez sur **OK**.
4. Cliquez sur le bouton **Imprimer** dans la barre d'outils.



Les résultats de la requête sont imprimés.

Chapitre 5. Utilisation des rapports

Vous pouvez créer des rapports en combinant les résultats d'une requête avec le formatage contenu dans un format.

Formats

Les formats sont des instructions formatées servant à créer, afficher et imprimer des rapports.

Présentation des formats

Les formats comprennent divers composants. Tous ces composants peuvent être modifiés dans un document de format.

Principal

Principaux composants d'un format, y compris les en-têtes, les bas de page et les ruptures.

Ruptures

Caractéristiques, contenu et position des lignes de sous-total (six au maximum) dans un rapport.

Calculs

Expressions de calcul d'un rapport.

Remarque : IBM ObjectREXX doit être installé sur votre machine pour que vous puissiez utiliser la fonction de calcul de format.

Colonnes

Aspect et mise en forme des colonnes du rapport. Les caractéristiques définissables comprennent l'ordre des colonnes, leur format, leur syntaxe, leur retrait et leur largeur.

Conditions

Contraintes de mise en forme conditionnelles. Par exemple, vous pouvez définir le format afin d'afficher uniquement les lignes présentant certaines caractéristiques.

Détails

En-têtes et texte détaillés d'un rapport. Cette commande permet de combiner ou de remplacer des données tabulaires par du texte libre, afin de créer des lettres types ou des étiquettes d'adresse.

Final Contenu et position du texte final du rapport. Par exemple, vous pouvez choisir d'inclure un texte final et des données récapitulatives à la fin du rapport.

HTML

Contenu et position des caractéristiques de mise en forme et des étiquettes dans les rapports HTML.

Options

Diverses options définissant l'aspect du rapport.

Page Contenu et position de l'en-tête et du bas de page du rapport.

Création d'un rapport à l'aide d'un format

Vous pouvez créer des rapports en combinant les résultats d'une requête avec les options de mise en forme contenues dans un format. En répétant ce processus, vous avez la possibilité de créer plusieurs rapports à partir d'un même ensemble de résultats de requête.

1. A partir d'une fenêtre de résultats de requête, cliquez sur le bouton **Afficher un rapport**.



La boîte de dialogue Sélection d'un format s'ouvre.



2. Selon le type de format sélectionné dans la boîte de dialogue, vous pouvez être invité à entrer d'autres données. Fournissez les informations requises (emplacement du fichier, détenteur et nom, ou titre du document), puis cliquez sur **OK**. Le rapport est créé à partir du format sélectionné et des résultats de la requête en cours.

Modification d'un format

La fenêtre Format fournit plusieurs options permettant de modifier et de créer des formats.

A partir d'un format ouvert, affichez le menu Format. Ce menu contient toutes les options nécessaires pour modifier et créer un format. Vous pouvez également modifier n'importe lequel de ces composants en cliquant sur le bouton correspondant dans la barre d'outils.

Création d'un format

Les procédures ci-dessous utilisent des exemples de données de la table Q.PERS. Essayez d'autres paramètres pour créer des formats personnalisés.

Etape 1 : Création d'un format

1. Exécutez la requête SQL suivante pour extraire les données à afficher dans le rapport :

```
SELECT * FROM Q.PERS ORDER BY DEPT,  
NOM
```

Les résultats correspondants s'affichent.

2. Cliquez sur le bouton **Afficher un rapport** dans la barre d'outils. La boîte de dialogue Sélection d'un format s'ouvre.
3. Indiquez que vous voulez utiliser le format par défaut et cliquez sur **OK**. QMF pour Windows affiche le rapport par défaut. Pour modifier le format par défaut, cliquez sur un des boutons des composants de format dans la barre d'outils. A chaque composant est associé un bouton de la barre d'outils des formats.

Etape 2 : Changement de l'ordre des colonnes

Nous voulons que la colonne NOM soit la première du rapport et que la colonne ID soit la deuxième. L'ordre des colonnes est spécifié dans le composant Colonnes du format.

1. Cliquez sur l'option **Colonnes** dans le menu **Format** pour afficher le bouton Colonne dans la boîte de dialogue Formats.
2. Changez la position d'une colonne en remplaçant le numéro de séquence existant. Pour que la colonne NOM soit la première du rapport, remplacez son numéro de séquence (valeur figurant dans la liste Séquence) par 1.
3. Pour que la colonne ID soit la deuxième du rapport, remplacez son numéro de séquence par 2 et cliquez sur **OK**. QMF pour Windows affiche le rapport avec les colonnes disposées dans le nouvel ordre dans la fenêtre Format.

Etape 3 : Modification des en-têtes de colonnes

Nous voulons qu'EMPLOYE soit l'en-tête de la première colonne et COMMISSION l'en-tête de la dernière colonne. Le texte d'en-tête des colonnes est spécifié dans le composant Colonnes du format.

1. Cliquez sur l'option **Colonnes** dans le menu **Format** pour afficher le bouton Colonne dans la boîte de dialogue Formats.
2. Modifiez l'en-tête de colonne en remplaçant le texte de l'en-tête existant. Remplacez l'en-tête de la première colonne par EMPLOYE, et l'en-tête de la dernière colonne par COMMISSION, puis cliquez sur **OK**. QMF pour Windows affiche le rapport avec les nouveaux en-têtes de colonne dans la fenêtre Format.

Etape 4 : Modification du format des colonnes

Nous voulons que la colonne SALAIRE s'affiche avec le symbole monétaire approprié. Le format d'une colonne est déterminé par son code d'édition, qui est indiqué dans le composant Colonnes du format.

1. Cliquez sur l'option **Colonnes** dans le menu **Format** pour afficher le bouton Colonne dans la boîte de dialogue Formats.
2. Remplacez le code d'édition de la colonne SALAIRE par la valeur D2, puis cliquez sur **OK**. QMF pour Windows affiche le rapport dans la fenêtre Format, avec la colonne SALAIRE comportant le symbole monétaire approprié.

Etape 5 : Ajout d'informations récapitulatives

Nous voulons diviser le rapport en sections, avec une section pour chaque département. En outre, nous voulons afficher les valeurs SALAIRE total et COMMISSION de chaque département à la fin de chaque section. Pour ce faire, il est nécessaire de spécifier le mode d'utilisation de chaque colonne du rapport. L'utilisation d'une colonne, qui est déterminée par son code usage, est spécifiée dans le composant Colonnes du format.

1. Cliquez sur l'option **Colonnes** dans le menu **Format** pour afficher le bouton Colonne dans la boîte de dialogue Formats.
2. Pour diviser le rapport en définissant une section par département, remplacez le code usage de DEPT par RUPT1. Les codes usage commençant par le mot RUPT génèrent une rupture de section pour la colonne spécifiée. Le chiffre qui suit le mot RUPT détermine le niveau de rupture ; il est possible de définir jusqu'à six niveaux de rupture dans un rapport.
3. Pour spécifier que nous voulons inclure les valeurs SALAIRE total et COMMISSION pour chaque DEPT, remplacez le code usage de SALAIRE et COMMISSION par SOMME.
4. Le rapport sera plus compréhensible si nous ajoutons des informations descriptives à la fin de chaque rupture de section. Pour ce faire, cliquez sur **Ruptures** dans le menu **Format**.
5. Vous spécifiez le texte de bas de page de rupture dans la boîte de dialogue Ruptures de format. Attribuez la valeur Total du département à la première ligne de bas de page de rupture et cliquez sur **OK**. QMF pour Windows affiche la fenêtre Format.

Etape 6 : Ajout d'en-têtes et de bas de page

Nous voulons ajouter un en-tête et un bas de page au rapport. Les en-têtes et les bas de page sont spécifiés dans le composant Page du format.

1. Cliquez sur l'option **Page...** dans le menu **Format** pour afficher le bouton Page dans la boîte de dialogue Formats.

2. La partie supérieure de cette boîte de dialogue sert à spécifier l'en-tête de page. Attribuez la valeur Rapport du département à la première ligne de l'en-tête de page et la valeur Salaire total et commissions à la seconde ligne. Choisissez le mode d'alignement de l'en-tête.
3. La partie inférieure de la boîte de dialogue sert à spécifier le bas de page. Attribuez la valeur Fin de page à la première ligne du bas de page. Choisissez le mode d'alignement du bas de page et cliquez sur **OK**. QMF pour Windows affiche la fenêtre Format.

Enregistrement des formats

Vous pouvez enregistrer les formats sur votre PC, sur un serveur de fichiers ou sur un serveur de base de données.

Enregistrement d'un format dans un fichier

1. A partir d'un format ouvert, cliquez sur le bouton **Sauvegarder**.
2. Si le format a déjà été sauvegardé, cliquez sur **Sauvegarder**. S'il n'a jamais été sauvegardé, la boîte de dialogue Sauvegarder sous s'ouvre.
3. Entrez le nom du fichier où vous voulez stocker le format et cliquez sur **OK**. Le format est enregistré.

Ouverture de fichiers de formats enregistrés

1. Cliquez sur le bouton **Ouvrir** dans la barre d'outils.



La boîte de dialogue Ouvrir s'affiche.

2. Sélectionnez le fichier à ouvrir et cliquez sur **OK**. Le format sélectionné s'ouvre dans un nouveau document de format.

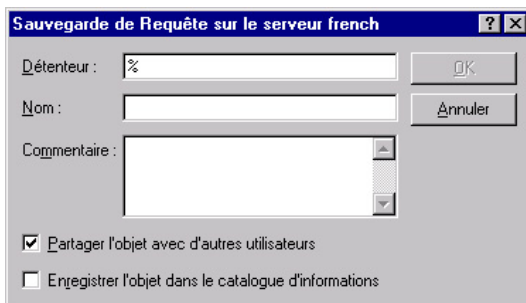
Enregistrement de formats sur le serveur de base de données

Les formats enregistrés sur le serveur peuvent être rendus accessibles aux autres utilisateurs. Si vous voulez partager des formats avec d'autres utilisateurs, enregistrez-les sur le serveur de base de données.

1. A partir d'un format ouvert, cliquez sur le bouton **Sauvegarder sur le serveur** dans la barre d'outils.



La boîte de dialogue correspondante s'ouvre.



2. Indiquez un détenteur et un nom, puis spécifiez si le format enregistré doit être partagé avec d'autres utilisateurs et cliquez sur **OK**. Le format est enregistré sur le serveur.

Si un format portant ce nom existe déjà, vous êtes invité à le remplacer.

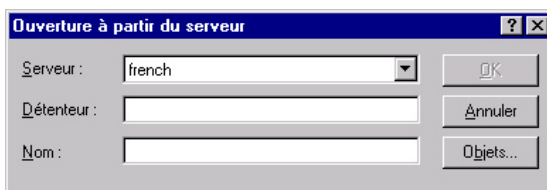
Ouverture de formats enregistrés sur le serveur de base de données

Vous pouvez ouvrir des formats qui ont été enregistrés sur le serveur de base de données.

1. Cliquez sur le bouton **Ouvrir à partir du serveur** dans la barre d'outils.



La boîte de dialogue Ouverture à partir du serveur s'affiche.



2. Indiquez un serveur, un détenteur et un nom, puis cliquez sur **OK**. Le format s'ouvre.

Impression de rapports

Vous pouvez imprimer des rapports.

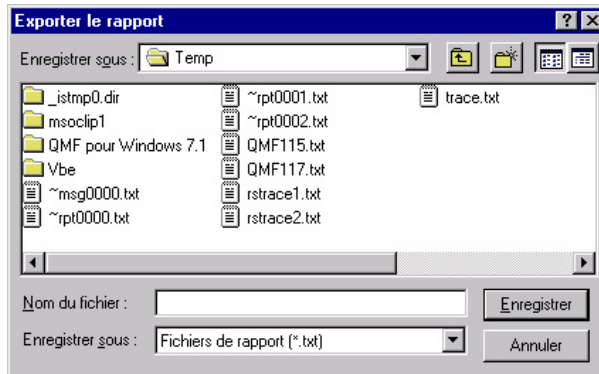
1. Ouvrez un format et cliquez sur **Mise en page**.
2. Modifiez à votre convenance la présentation de la page et cliquez sur **OK**.
3. Cliquez sur **Imprimer le rapport** dans le menu **Fichier**.

Le rapport est imprimé.

Exportation de rapports

Vous pouvez exporter un rapport dans un fichier.

1. Ouvrez un format et cliquez sur **Mise en page**.
2. Modifiez à votre convenance la présentation de la page et cliquez sur **OK**.
3. Cliquez sur **Exporter le rapport** dans le menu **Fichier**. La boîte de dialogue correspondante s'ouvre.



4. Entrez le nom du fichier où vous voulez stocker le rapport et cliquez sur **OK**. Le rapport est exporté.

Chapitre 6. Utilisation des procédures

Les procédures linéaires vous permettent d'exécuter des requêtes, de générer des rapports, de modifier des données et de mettre en oeuvre d'autres fonctions au moyen d'une seule commande. Pour obtenir une liste complète de toutes les commandes de procédure prises en charge par QMF pour Windows, reportez-vous à l'aide en ligne associée.

Les procédures logiques, ou procédures REXX, sont semblables aux procédures linéaires à cela près qu'elles contiennent le langage de programmation IBM Object REXX d'IBM ainsi que des commandes de procédure. Object REXX doit être installé localement pour que vous puissiez utiliser des procédures logiques.

Exécution de procédures

Les procédures sont utilisées pour exécuter plusieurs fonctions au moyen d'une seule commande.

Création d'une nouvelle procédure linéaire

Dans le menu **Fichier**, cliquez sur **Nouvelle procédure**.

Un nouveau document de procédure s'ouvre.

Création d'une nouvelle procédure logique

1. Dans le menu **Fichier**, cliquez sur **Nouvelle procédure**.
Un nouveau document de procédure s'ouvre.
2. Taper une ligne de commentaire REXX comme première ligne de la procédure. Les lignes de commentaire REXX commencent par /* et finissent par */
3. Tapez les commandes de procédure QMF que vous voulez voir apparaître dans la procédure. Les commandes QMF doivent être introduites en majuscules et mises entre guillemets.
4. Tapez les commandes de procédure REXX que vous voulez voir apparaître dans la procédure.

Remarque : Les commandes REXX sont exécutées localement, et pas sur le serveur de base de données. Object REXX doit être installé localement.

Exécution d'une procédure sur un serveur de base de données

1. Ouvrez un nouveau document de procédure et entrez une série de commandes, ou bien ouvrez une procédure existante à partir d'un fichier ou du serveur de base de données.
2. Cliquez sur le bouton **Exécuter la procédure** dans la barre d'outils.



La procédure est exécutée.

Enregistrement de procédures

Vous pouvez enregistrer les procédures sur votre PC, sur un serveur de fichiers ou sur un serveur de base de données.

Enregistrement des procédures dans des fichiers

1. A partir d'une procédure ouverte, cliquez sur le bouton **Sauvegarder** dans la barre d'outils.



Si la procédure a déjà été enregistrée, elle l'est à nouveau. Si elle n'a jamais été enregistrée, la boîte de dialogue Sauvegarder sous s'ouvre.

2. Entrez le nom du fichier où vous voulez stocker la procédure et cliquez sur **OK**. La procédure est enregistrée.

Ouverture d'un fichier de procédure

1. Cliquez sur le bouton **Ouvrir** dans la barre d'outils.



La boîte de dialogue Ouvrir apparaît.

2. Sélectionnez le fichier à ouvrir et cliquez sur **OK**. La procédure sélectionnée s'ouvre dans un nouveau document de procédure.

Enregistrement d'une procédure sur le serveur de base de données

1. A partir d'une procédure ouverte, cliquez sur le bouton **Sauvegarder sur le serveur** dans la barre d'outils.



La boîte de dialogue correspondante s'ouvre.



- Indiquez un détenteur et un nom, puis spécifiez si la procédure enregistrée doit être partagée avec d'autres utilisateurs et cliquez sur **OK**. La procédure est enregistrée sur le serveur.

Si une procédure portant ce nom existe déjà, vous êtes invité à la remplacer.

Ouverture de procédures enregistrées sur le serveur de base de données

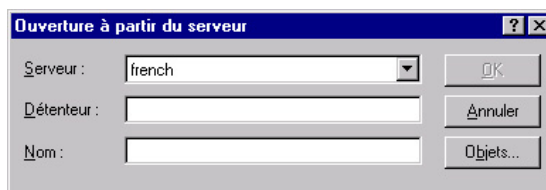
Vous pouvez ouvrir des procédures qui ont été enregistrées sur le serveur de base de données.

-

Cliquez sur le bouton **Ouvrir depuis serveur** dans la barre d'outils.



La boîte de dialogue Ouverture à partir du serveur s'affiche.



- Indiquez un serveur, un détenteur et un nom, puis cliquez sur **OK**. La procédure s'ouvre.

Impression de procédures

Vous pouvez imprimer le texte d'une procédure.

Prévisualisation d'une procédure

- Ouvrez une procédure. Les commandes de la procédure s'affichent.

2. Dans le menu **Fichier**, cliquez sur **Mise en page**. La boîte de dialogue correspondante s'ouvre.
3. Modifiez à votre convenance la présentation de la page et cliquez sur **OK**.
4. Cliquez sur le bouton **Aperçu avant impression** dans la barre d'outils.



Un aperçu de la procédure imprimée s'affiche.

Impression d'une procédure

1. Ouvrir une procédure. Les commandes de la procédure s'affichent.
2. Dans le menu **Fichier**, cliquez sur **Mise en page**. La boîte de dialogue correspondante s'ouvre.
3. Modifiez à votre convenance la présentation de la page et cliquez sur **OK**.
4. Cliquez sur le bouton **Imprimer** dans la barre d'outils.



La procédure est imprimée.

Chapitre 7. Utilisation des listes

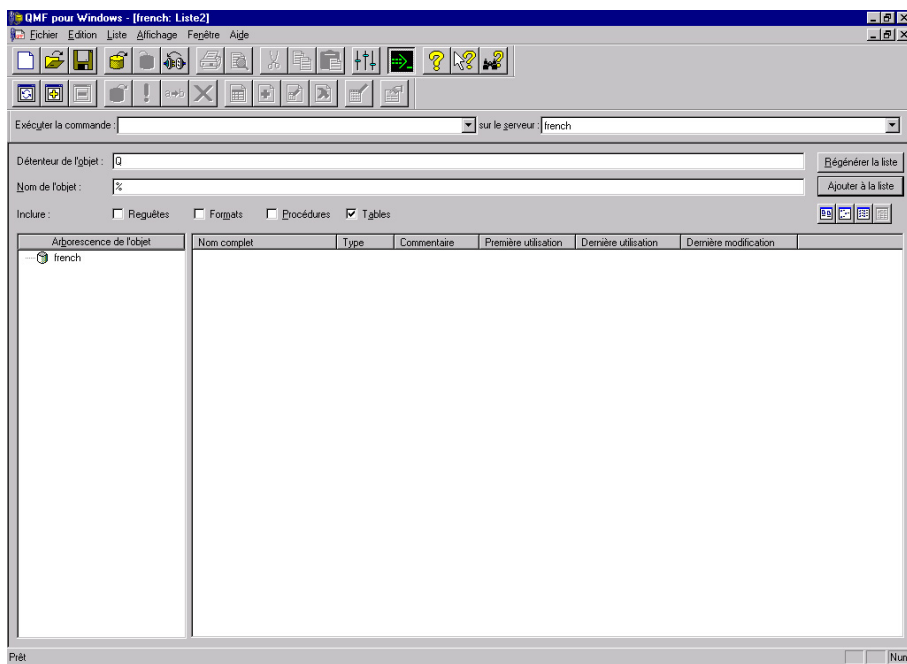
Les listes vous permettent de visualiser facilement les collectes d'objets QMF.

Objets

QMF pour Windows reconnaît quatre types d'objets : requêtes, formats, procédures et tables. Vous pouvez utiliser la fenêtre Liste pour afficher les objets en fonction de leur nom, de leur détenteur et de leur type.

Création de listes d'objets

1. Dans le menu **Fichier**, cliquez sur **Nouvelle liste**.
La fenêtre Liste apparaît.



2. Indiquez un détenteur et un nom.

Remarque : Vous pouvez utiliser des modèles pour sélectionner des objets dans une liste d'objets répondant aux critères de recherche.

- Le caractère de pourcentage (%) est utilisé pour remplacer une chaîne de longueur indéfinie, contenant n'importe quel caractère. Par exemple, entrez A% pour répertorier toutes les tables dont le nom commence par la lettre A.
 - Le caractère de soulignement (_) est utilisé pour remplacer un seul caractère. Par exemple, entrez _A% pour répertorier toutes les tables dont le détenteur a un nom comportant la lettre A en deuxième position.
3. Sélectionnez le type d'objet recherché.
 4. Cliquez sur **Régénérer la liste**. Une liste d'objets enregistrés sur le serveur de base de données et répondant à vos critères de recherche s'affiche.

Commandes de la fenêtre Liste

En cliquant avec le bouton droit de la souris sur un objet de la fenêtre Liste, vous affichez une liste de commandes identiques à celles du menu Liste.

Afficher

Ouvre l'objet sélectionné pour que vous puissiez le visualiser. Cette commande est disponible pour les requêtes, les formats, les procédures et les tables.

Exécuter

Exécute l'objet sélectionné. Cette commande est disponible pour les requêtes et les procédures.

Créer Crée une requête en fonction de la table sélectionnée. Vous pouvez choisir de créer une requête SQL SELECT, une requête SQL UPDATE, une requête SQL INSERT ou une requête assistée. Cette commande est disponible pour les tables.

Editer Ouvre l'objet sélectionné pour que vous puissiez le modifier. Cette commande est disponible pour les tables.

Propriétés

Affiche les propriétés de l'objet sélectionné, y compris les commentaires, les attributs et les données historiques relatives à son utilisation. Cette commande est disponible pour les requêtes, les formats, les procédures et les tables.

Création de listes

Vous pouvez créer des listes qui vont être utilisées comme des collectes d'objets. Par exemple, vous avez la possibilité de créer une liste de tous les formats, requêtes, procédures et tables liés à un inventaire afin de regrouper ces données. Une fois cette opération effectuée, vous pouvez supprimer ou ajouter des objets et enregistrer la liste en vue d'une utilisation ultérieure.

Ajout d'objets aux listes

Vous pouvez ajouter des objets aux listes.

Dans une liste ouverte, indiquez l'utilisateur et le nom des objets à ajouter et cliquez sur **Ajouter à la Liste** dans la barre d'outils.



Les objets correspondant à l'utilisateur et au nom sont ajoutés à la liste.

Suppression d'objets des listes

Vous pouvez supprimer des objets indépendants des listes.

A partir d'une liste ouverte, cliquez sur le bouton **Supprimer** dans la barre d'outils.



L'objet disparaît de la liste, mais il n'est pas supprimé.

Sauvegarde de listes dans des fichiers

1. A partir d'une liste ouverte, cliquez sur le bouton **Sauvegarder** dans la barre d'outils.



Si la liste a déjà été sauvegardée, elle l'est à nouveau. Si elle n'a jamais été sauvegardée, la boîte de dialogue Sauvegarder sous s'ouvre.

2. Entrez le nom du fichier où vous voulez stocker la liste et cliquez sur **OK**. La liste est enregistrée.

Ouverture de fichiers de listes

1. Cliquez sur le bouton **Ouvrir** dans la barre d'outils.



La boîte de dialogue Ouvrir s'affiche.

2. Sélectionnez le fichier à ouvrir et cliquez sur **OK**. La liste sélectionnée s'ouvre dans un document de liste.

Chapitre 8. Utilisation de fichiers de tâches

Vous pouvez programmer et exécuter des procédures en utilisant les fichiers de tâches. Les fichiers de tâches utilisent le programmeur Windows pour exécuter des procédures en fonction de l'heure et de la date introduites.

Fichiers de tâches

Vous pouvez créer des fichiers de tâches et les stocker localement sur le serveur de base de données.

Création de fichiers de tâches

1. Dans le menu **Fichier**, cliquez sur **Nouvelle tâche**.

Un nouveau document de tâche s'ouvre.

Exécution de fichiers de tâches

Vous pouvez exécuter des fichiers de tâches enregistrés localement.

1. Ouvrez un fichier de tâches.
2. Cliquez sur **Exécuter la tâche** dans la barre d'outils.



3. Faites glisser la ligne de séparation d'un côté à l'autre, ou vers le haut et vers le bas, pour redimensionner la colonne ou la ligne.

Remarque : Si vous enregistrez la requête après avoir redimensionné les lignes et les colonnes, elle s'affiche avec la nouvelle mise en forme.

Ajustement automatique des colonnes et des lignes

Vous pouvez redimensionner automatiquement des colonnes et des lignes pour les ajuster aux données qu'elles contiennent.

Sélectionnez toute une colonne ou ligne en cliquant deux fois sur la ligne noire la séparant de l'objet adjacent. La colonne est automatiquement redimensionnée en fonction des données qu'elle contient.

Remarque : Si vous enregistrez la requête après avoir redimensionné les lignes et les colonnes, elle s'affiche avec la nouvelle mise en forme.

Tri des résultats de requête

Une fois que vous avez exécuté une requête, vous pouvez trier les résultats de chaque colonne par ordre alphabétique.

Dans la fenêtre des résultats d'une requête, sélectionnez une colonne et cliquez sur le bouton **Trier dans l'ordre croissant** dans le menu Résultats.

Les résultats de la requête sont triés par ordre croissant.

Autre possibilité :

Dans la fenêtre des résultats de requête, sélectionnez une colonne et cliquez sur le bouton **Trier dans l'ordre Décroissant** dans le menu Résultats.

Les résultats de la requête sont triés par ordre décroissant.

Remarque : Pour trier les colonnes sélectionnées de façon plus complexe, sélectionnez **Trier** dans le menu Résultats.

Reclassement des colonnes

Vous pouvez modifier l'ordre des colonnes dans les résultats de la requête.

Dans la fenêtre des résultats d'une requête, sélectionnez une colonne et déplacez-la.

La colonne apparaît reclassée.

Formatage des résultats de requête

Vous pouvez changer la police utilisée pour afficher des requêtes et des résultats de requête. Les polices disponibles varient en fonction de l'installation effectuée sur l'ordinateur. Pour de plus amples informations sur l'ajout de polices, reportez-vous à l'aide du système d'exploitation.

Remarque : Si vous enregistrez la requête après avoir sélectionné une nouvelle police d'affichage de résultats de requête, elle s'affiche avec la nouvelle police.

Sélection de la police d'affichage des résultats d'une requête

1. A partir de la fenêtre des résultats, cliquez sur **Définir la police** dans le menu Résultats. La boîte de dialogue Police s'ouvre.
2. Sélectionnez la police et la taille d'affichage des résultats de la requête et cliquez sur **OK**. Les résultats de la requête sont affichés dans le format sélectionné.

Remarque : Cliquez sur **Définir par défaut** afin d'utiliser la police sélectionnée comme police par défaut pour tous les résultats de requête.

Formatage des résultats numériques de requête

1. Dans la fenêtre Résultats, sélectionnez une colonne contenant des valeurs numériques et sélectionnez **Format** dans le menu **Résultats**. La boîte de dialogue Format s'ouvre.
2. Sélectionnez le formatage à appliquer et cliquez sur **OK**. Les valeurs

Remarque : Cliquez sur **Définir par défaut** afin d'utiliser la police sélectionnée comme police par défaut pour tous les résultats de requête.

Conversion du formatage des résultats de requête en format

Vous pouvez convertir le formatage des résultats de requête en format.

1. Dans le menu **Résultats**, sélectionnez **Afficher un rapport**.
La boîte de dialogue Sélection d'un format s'ouvre.
2. Sélectionnez **Requête Formatet** cliquez sur **OK**.
Le formatage des résultats de la requête est converti en format et s'ouvre dans une nouvelle fenêtre de format.

Regroupement et agrégation des résultats de requête

Les résultats de requête peuvent être regroupés, agrégés et résumés.

Regroupement des résultats de requête

Vous pouvez regrouper les résultats de requête avec ou sans information récapitulative.

1. Sélectionnez la colonne à regrouper.
2. Dans le menu **Résultats**, sélectionnez le type de regroupement à appliquer.
La colonne est regroupée selon la sélection.

Résumé des résultats de requête

Vous pouvez résumer les résultats de requête par colonne.

1. Sélectionnez la colonne à regrouper.
2. Dans le menu **Résultats**, sélectionnez le type de résumé à appliquer.
La colonne est résumée selon la sélection.

Enregistrement des résultats de requête et formatage

Vous pouvez sauvegarder les résultats de requête et sauvegarder le formatage en format.

Enregistrement des résultats de requête dans une table

Vous pouvez enregistrer les résultats de requête sous forme de table dans un serveur de base de données.

1. Dans le menu **Résultats**, sélectionnez **Sauvegarder dans Base de données**.
La boîte de dialogue correspondante s'ouvre.
2. Indiquez un utilisateur et un nom de table et cliquez sur **OK**.
Les résultats de requête sont enregistrés sous forme de table dans la base de données.

Enregistrement des résultats de requête sous forme de fichiers

Vous pouvez enregistrer les résultats de requête sous forme de fichier sur votre PC ou sur un serveur de fichier.

1. Dans le menu **Résultats**, sélectionnez **Sauvegarder dans le fichier**.
La boîte de dialogue Exportation des données s'ouvre.
2. Indiquez l'emplacement de sauvegarde du fichier, toutes les options d'exportation, puis cliquez sur **OK**.
Les résultats de requête sont enregistrés dans un fichier.

Impression des résultats de requête

Vous pouvez prévisualiser et imprimer vos résultats de requête.

Prévisualisation des résultats de requête

1. Ouvrez et exécutez une requête. Les résultats correspondants s'affichent.
2. Dans le menu **Fichier**, cliquez sur **Mise en page**. La boîte de dialogue correspondante s'ouvre.
3. Modifiez à votre convenance la présentation de la page et cliquez sur **OK**.
4. Cliquez sur le bouton **Aperçu avant impression** dans la barre d'outils.



Un aperçu des résultats imprimés s'affiche.

Impression des résultats de requête

1. Ouvrez une requête et activez la fenêtre des résultats. Les résultats de la requête s'affichent.
2. Dans le menu **Fichier**, cliquez sur **Mise en page**. La boîte de dialogue correspondante s'affiche.
3. Modifiez à votre convenance la présentation de la page et cliquez sur **OK**.

4. Cliquez sur le bouton **Imprimer** dans la barre d'outils.



Les résultats de la requête sont imprimés.

Chapitre 9. Utilisation des requêtes statiques

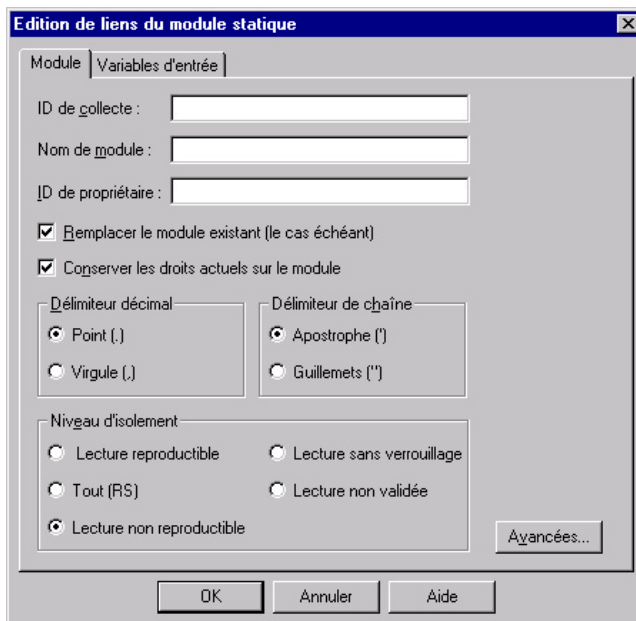
Une requête statique est une requête SQL qui a été précédemment transmise à un serveur de base de données et liée sous forme de module. Lors de l'exécution d'une requête statique, le serveur de base de données utilise le texte SQL lié dans le module, au lieu de celui qui figure dans la fenêtre de requête. Les requêtes statiques exploitent mieux les ressources que les requêtes dynamiques, mais ne peuvent pas être modifiées.

Requêtes statiques

Les requêtes statiques sont créées à partir de requêtes assistées et de requêtes SQL existantes.

Création des requêtes statiques

1. Dans le menu **Requête**, cliquez sur **Edition de liens du module statique**. La boîte de dialogue correspondante s'ouvre.



2. Sélectionnez l'onglet **Module**, entrez un ID de collecte et un nom de module, puis modifiez les autres options de votre choix.
3. Si la requête contient des variables de substitution, sélectionnez l'onglet des variables. Remplacez les variables de substitution par des variables hôtes.

4. Cliquez sur **OK**. La requête statique est liée.

Remarque : Une fois que vous avez édité les liens d'une requête, vous devez enregistrer cette requête dans un fichier ou sur le serveur de base de données.

Remplacement des variables de substitution par des variables hôtes

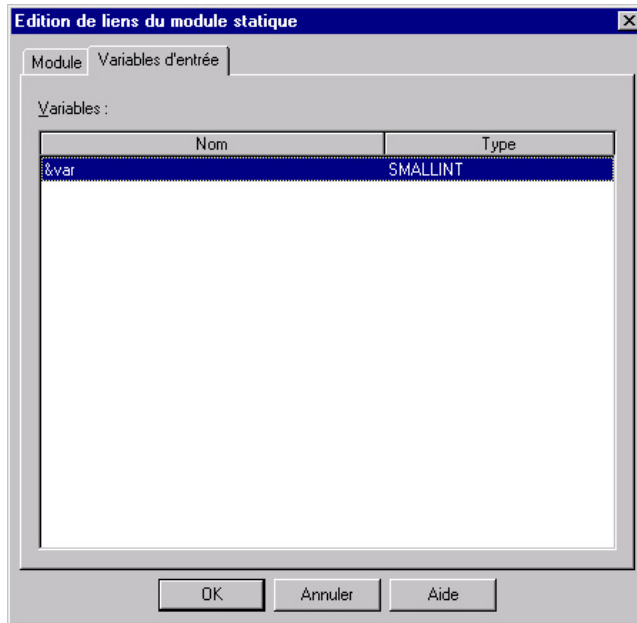
Pour éditer les liens d'un module, vous devez spécifier une variable hôte à utiliser à la place de chaque variable de substitution dans le texte SQL. Cependant, il n'est pas toujours possible de remplacer directement une variable de substitution par une variable hôte. Les variables de substitution assurent une substitution directe dans le texte de la requête avant l'envoi de ce texte au serveur de base de données. Les variables hôte sont envoyées au serveur avec la requête. Pour savoir où et comment les variables hôtes peuvent être employées dans les requêtes, reportez-vous à la documentation du serveur de base de données.

Une fois que vous avez spécifié une relation entre une variable de substitution et une variable hôte, QMF pour Windows mémorise cette relation et la propose dans les requêtes ultérieures, facilitant ainsi l'édition de liens de module.

Les types de données valides pour les variables hôtes sont les suivants :

- CHAR(n)
- VARCHAR(n)
- INTEGER
- SMALLINT
- FLOAT
- DECIMAL(p,s)
- DATE
- TIME
- TIMESTAMP

1. Dans la boîte de dialogue Edition de liens du module statique, sélectionnez l'onglet Variables d'entrée.



2. Entrez un type de variable pour chaque variable hôte et cliquez sur **OK**.
Les variables de substitution sont converties en variables hôtes.

Exécution d'une requête statique

Vous pouvez exécuter des requêtes statiques comme toute autre requête. Pour plus de détails, reportez-vous à la section «Requêtes SQL» à la page 9.

Chapitre 10. Utilisation de l'éditeur de tables

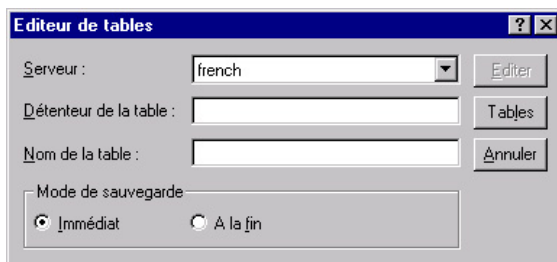
L'éditeur de tables permet de rechercher, d'ajouter, de modifier et de supprimer des données dans les tables, sans avoir besoin d'écrire des formulations SQL.

Editeur de tables

L'éditeur de tables vous donne la possibilité de modifier et de rechercher des données.

Recherche de lignes au moyen de l'éditeur de tables

1. Dans le menu **Fichier**, cliquez sur **Editeur de tables**. La boîte de dialogue correspondante s'ouvre.



2. Indiquez une table.

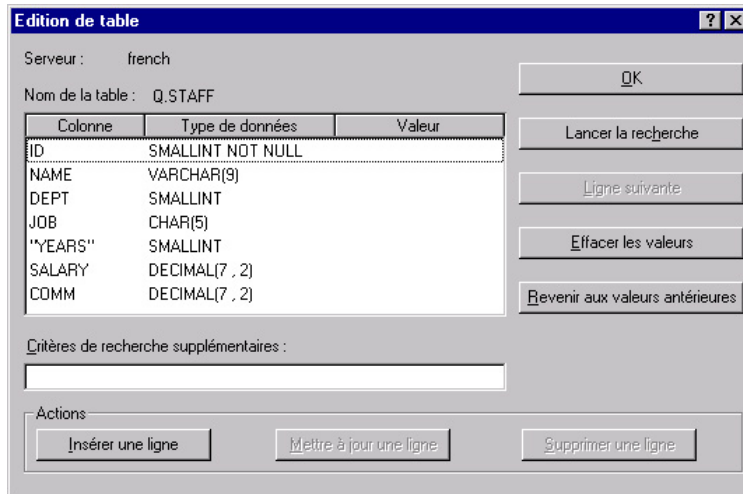
Remarque : Vous pouvez utiliser des modèles pour sélectionner des noms dans une liste de tables répondant aux critères de recherche.

- Le caractère de pourcentage (%) est utilisé pour remplacer une chaîne de longueur indéfinie, contenant n'importe quel caractère. Par exemple, entrez A% pour répertorier toutes les tables dont le nom commence par la lettre A.
- Le caractère de soulignement (_) est utilisé pour remplacer un seul caractère. Par exemple, entrez _A% pour répertorier toutes les tables dont le détenteur a un nom comportant la lettre A en deuxième position.

Après avoir indiqué un modèle, cliquez sur **Tables** et sélectionnez une table dans la liste obtenue.

3. Choisissez un mode de sauvegarde.
 - Immédiat - La table sera mise à jour sur le serveur de base de données après chaque modification.

- A la fin - La table sera mise à jour sur le serveur de base de données une fois que vous aurez entré toutes les modifications. Les autres utilisateurs ne pourront pas apporter de changement à la table pendant que vous la modifiez.



Edition de table [?] [X]

Serveur : french

Nom de la table : Q.STAFF

Colonne	Type de données	Valeur
ID	SMALLINT NOT NULL	
NAME	VARCHAR(9)	
DEPT	SMALLINT	
JOB	CHAR(5)	
"YEARS"	SMALLINT	
SALARY	DECIMAL(7, 2)	
COMM	DECIMAL(7, 2)	

Critères de recherche supplémentaires :

Actions

Insérer une ligne Mettre à jour une ligne Supprimer une ligne

OK

Lancer la recherche

Ligne suivante

Effacer les valeurs

Revenir aux valeurs antérieures

4. Cliquez sur **Editer**. La boîte de dialogue Edition de table s'ouvre.
5. Entrez les valeurs sur lesquelles doit porter la recherche dans la colonne Valeur, ou indiquez des critères de recherche dans la zone Critères de recherche supplémentaires pour spécifier des conditions de recherche plus complexes. Vous pouvez entrer n'importe quel prédicat SQL valide dans cette zone.
6. Cliquez sur **Lancer la recherche**. La première ligne répondant aux critères apparaît dans la colonne Valeur.

Ajout d'une ligne

1. Dans la boîte de dialogue Edition de table, entrez les informations correspondant à la nouvelle sauvegarde.
2. Cliquez sur **Insérer une ligne**. La nouvelle ligne est ajoutée à la table.
3. Cliquez sur **OK**. Les modifications sont enregistrées.

Modification d'une ligne

1. Dans la boîte de dialogue Edition de table, recherchez la ligne à modifier.
2. Cliquez sur **Ligne suivante** pour afficher la ligne à modifier.
3. Modifiez les données contenues dans la colonne Valeur et cliquez sur **Mettre à jour une ligne**. La ligne est modifiée.
4. Cliquez sur **OK**. Les modifications sont enregistrées.

Suppression d'une ligne

1. Dans la boîte de dialogue Edition de table, recherchez la ligne à supprimer.
2. Cliquez sur **Ligne suivante** pour afficher la ligne à supprimer.
3. Cliquez sur **Supprimer une ligne**. La ligne est supprimée.
4. Cliquez sur **OK**. Les modifications sont enregistrées.

Modification de tables à partir de la vue des résultats de requête

Vous pouvez modifier des tables directement à partir de la vue des résultats de requête.

Suppression d'une ligne à partir de la vue des résultats de requête

Vous pouvez supprimer individuellement les lignes des tables à partir de la vue des résultats de requête.

Dans la vue des résultats de requête, sélectionnez une ligne et cliquez sur **Supprimer** dans le menu **Edition**. La ligne est supprimée.

Mise à jour des colonnes à partir de la vue des résultats de requête

Vous pouvez mettre à jour individuellement le contenu de chaque colonne à partir de la vue des résultats de requête.

Dans la vue des résultats de requête, cliquez deux fois sur une cellule, spécifiez une nouvelle valeur et appuyez sur Entrée. La table est mise à jour.

Formats DB2

Si le composant utilisateur de Formats DB2 est installé, vous pouvez l'utiliser comme éditeur de table pour les tables ne comprenant pas de données LOB. Pour de plus amples informations sur les Formats DB2, consultez notre Centre de Ressources pour formats DB2 sur le site www.rocketsoftware.com/db2forms.

Chapitre 11. Distribution de données

Vous pouvez exporter des données dans d'autres bases et d'autres applications.

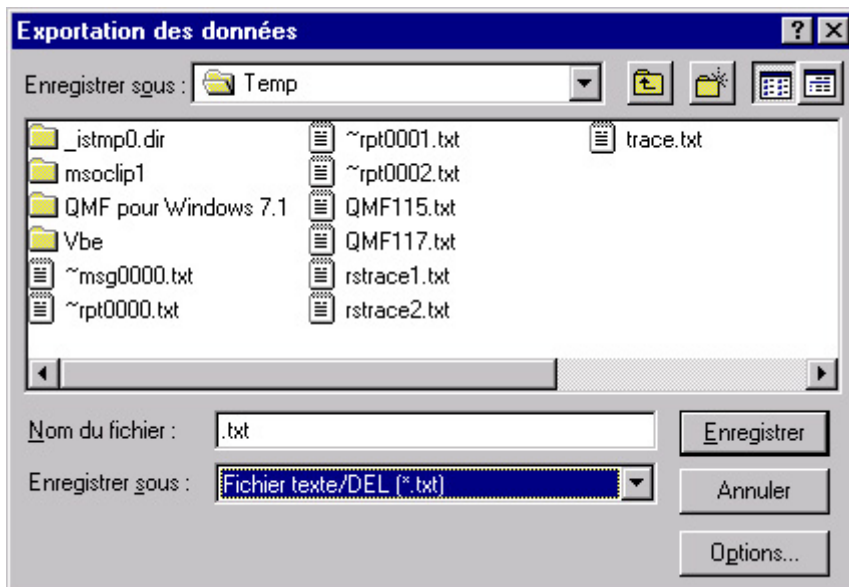
Exportation des données

Vous pouvez exporter des données à partir de QMF pour Windows dans d'autres applications en effectuant les opérations suivantes :

- exportation des données dans un fichier texte, CSV, IXF ou HTML ;
- enregistrement des résultats de requête dans une table ;
- insertion des résultats de requête dans une feuille de calcul Microsoft Excel.

Exportation de données dans des fichiers

1. Dans la fenêtre des résultats des requêtes, cliquez sur **Exporter des données** dans le menu **Fichier**. La boîte de dialogue Exportation des données s'ouvre.



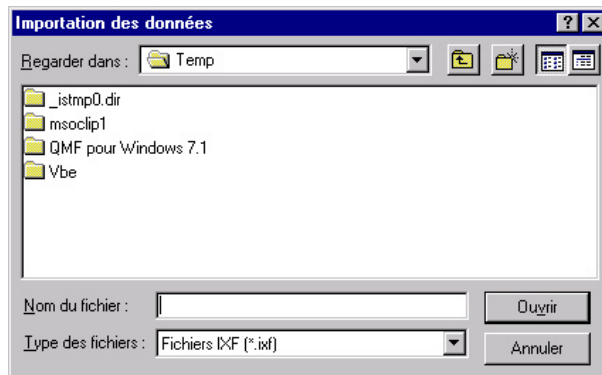
2. Sélectionnez un type de fichier de sortie et cliquez sur le bouton Options. Selon le type de fichier de sortie choisi, la boîte de dialogue Options d'exportation au format Texte/DEL, Options d'exportation au format HTML, Options d'exportation au format IXF ou Options d'exportation CSV s'affiche.

- Vous pouvez créer un fichier texte avec une extension .TXT. Il s'agit d'un fichier ASCII standard pouvant comporter des délimiteurs de chaîne et de colonne (comme indiqué dans la boîte de dialogue Options d'exportation au format Texte/DEL).
 - Vous pouvez créer un fichier HTML comportant une extension .HTM. Ce type de fichier peut être affiché par n'importe quel navigateur Web. Tous les codes HTML sont automatiquement générés dans le fichier qui est prêt à être publié sur un site Web Internet ou Intranet. Les options choisies dans la boîte de dialogue Options d'exportation au format HTML déterminent l'aspect des données exportées.
 - Vous pouvez créer un fichier .IXF. L'exportation au format IXF préserve toutes les informations de la base de données, y compris les en-têtes de colonnes et les types de données. Elle est généralement utilisée pour transférer des informations d'une base de données dans une autre.
 - Vous pouvez créer un fichier .CSV. L'exportation au format CSV est similaire à l'exportation au format texte. La virgule est utilisée comme délimiteur de colonne. Ce format est le plus souvent employé dans les tableurs.
3. Sélectionnez des options pour le type de fichier d'exportation sélectionné et cliquez sur **OK**. La boîte de dialogue Options se referme.
 4. Cliquez sur **OK** dans la boîte de dialogue Exportation des données. Les données sont exportées.

Importation de données

Vous pouvez importer des données enregistrées dans un fichier IXF. Une fois qu'elles ont été importées dans une fenêtre de requête, vous pouvez les enregistrer sur un serveur de base de données, les exporter dans un nouveau fichier ou les utiliser dans des rapports. Les fichiers PCIXF et System/370 IXF en mode caractère sont pris en charge.

1. Dans le menu **Fichier**, cliquez sur **Importer des données**. La boîte de dialogue correspondante s'ouvre.

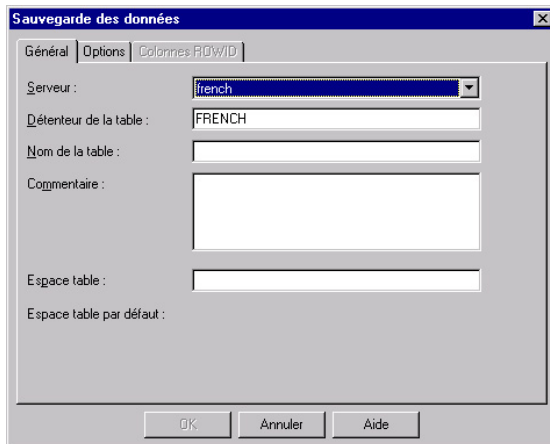


2. Sélectionnez le fichier à importer et cliquez sur **OK**. Les données importées s'affichent dans une nouvelle fenêtre de requête.

Enregistrement de données sur un serveur de base de données

Les résultats de requête importés peuvent être enregistrés dans une table de base de données.

1. Dans la fenêtre des résultats de requête importés, cliquez sur **Sauvegarder les données** dans le menu **Fichier**. La boîte de dialogue correspondante s'ouvre.



2. Choisissez un serveur de base de données et indiquez un détenteur et un nom de table. Sélectionnez ensuite les autres options de votre choix et cliquez sur **OK**. Les données sont enregistrées.

Utilisation de la commande Envoyer à

QMF pour Windows comprend une commande Envoyer à et un client de messagerie de base. Vous pouvez utiliser la commande Envoyer à et les fichiers de tâches pour programmer les requêtes et distribuer leurs résultats.

1. Dans le menu **Fichier**, sélectionnez **Envoyer à** et **Destinataire du courrier Internet**. La boîte de dialogue Message s'ouvre.
2. Indiquez un destinataire, un sujet, le texte du message, et cliquez sur **Suivant**. La boîte de dialogue Fichiers joints s'ouvre.
3. Ajoutez ou supprimez des fichiers au message et cliquez sur **Suivant**. La boîte de dialogue Envoi d'un message s'ouvre.
4. Indiquez le nom de votre serveur et cliquez sur **Terminer**. Le message est envoyé.

Utilisation de l'extension Microsoft Excel

QMF pour Windows inclut une extension pour Microsoft Excel 7.0 ou les versions ultérieures. Ces extensions permettent de lancer QMF pour Windows à partir d'Excel et de renvoyer les résultats de requête directement dans une feuille de calcul. Elles sont installées automatiquement si vous choisissez l'option d'installation "Standard", ou si vous sélectionnez l'option d'installation "Personnalisée" avec l'option Extension Microsoft Excel.

1. Cliquez sur le bouton **QMF pour Windows** dans la barre d'outils Excel.



QMF pour Windows s'ouvre.

2. A partir de QMF pour Windows, sélectionnez et exécutez une requête. Les résultats correspondants s'affichent.
3. Sélectionnez les données que vous voulez renvoyer dans Excel.
4. Dans le menu **Fichier**, cliquez sur **Retourner les données à Microsoft Excel**. Excel active et affiche la boîte de dialogue Extension QMF pour Windows.
5. Entrez la plage de destination des données et cliquez sur **OK**. Les données sont ajoutées à la feuille de calcul.

Utilisation des exemples d'applications

Plusieurs exemples d'applications et solutions d'intégration sont disponibles pour QMF pour Windows. Pour en savoir plus, consultez le site Web IBM à l'adresse <http://www.ibm.com/qmf/>.

Chapitre 12. Utilisation du Centre d'édition de rapports QMF

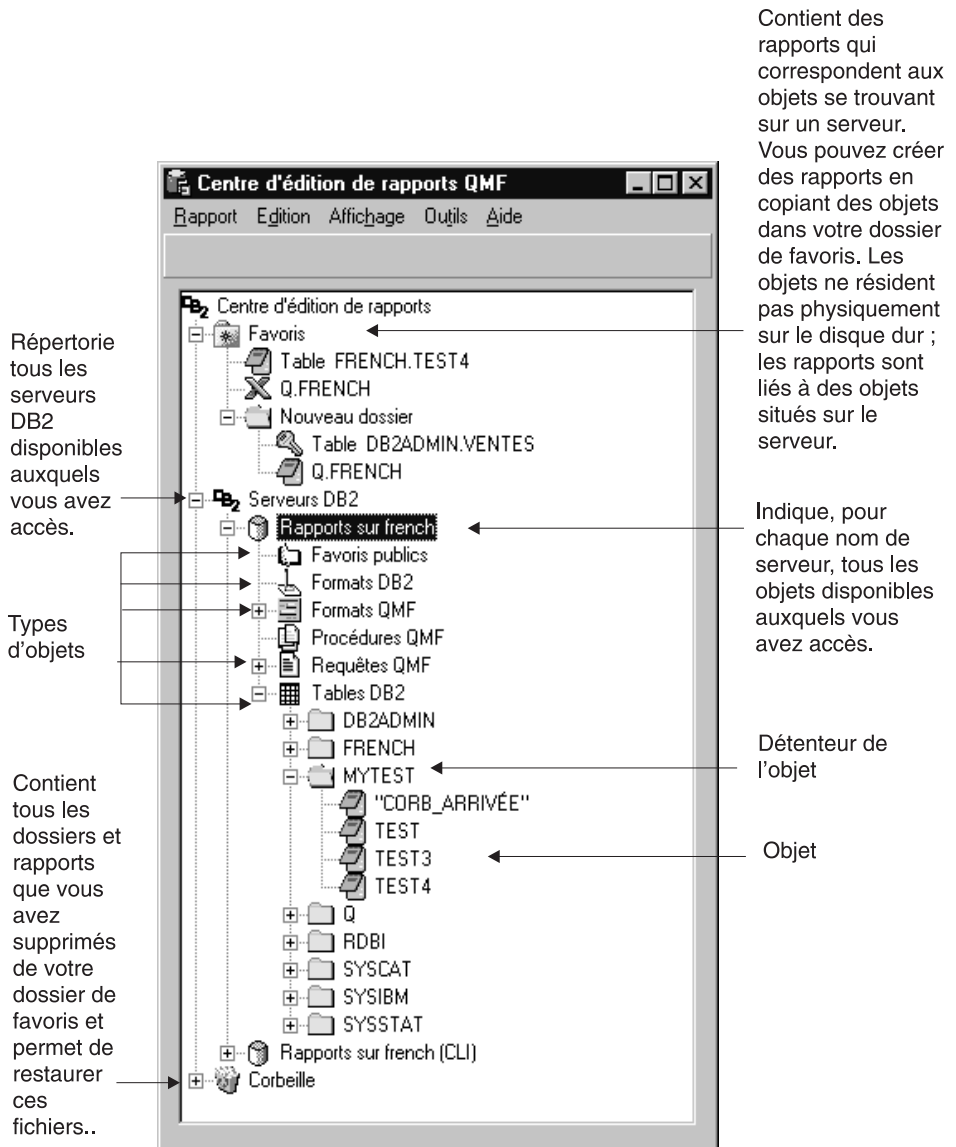
Le Centre d'édition de rapports QMF vous permet de créer des rapports personnalisés au moyen de formats, de procédures, de tables et de requêtes QMF. En ayant un accès rapide à ces objets, vous pouvez indiquer vos préférences de format de données et créer des rapports personnalisés qui peuvent être consultés et modifiés dans de nombreuses applications.

Initiation

- Cliquez avec le bouton droit de la souris sur un objet ou un dossier afin d'activer les mêmes options que celles disponibles dans les menus de la barre d'outils.
- Cliquez sur le symbole plus (+) placé à côté du dossier pour ouvrir le premier niveau du contenu. Maintenez la touche MAJ enfoncée tout en cliquant sur le symbole plus (+) afin d'ouvrir tous les niveaux du dossier.

Fenêtre du Centre d'édition de rapports QMF

La fenêtre du Centre d'édition de rapports QMF contient une structure arborescente des favoris disponibles, des serveurs DB2, des favoris publics, des objets ainsi qu'une corbeille.

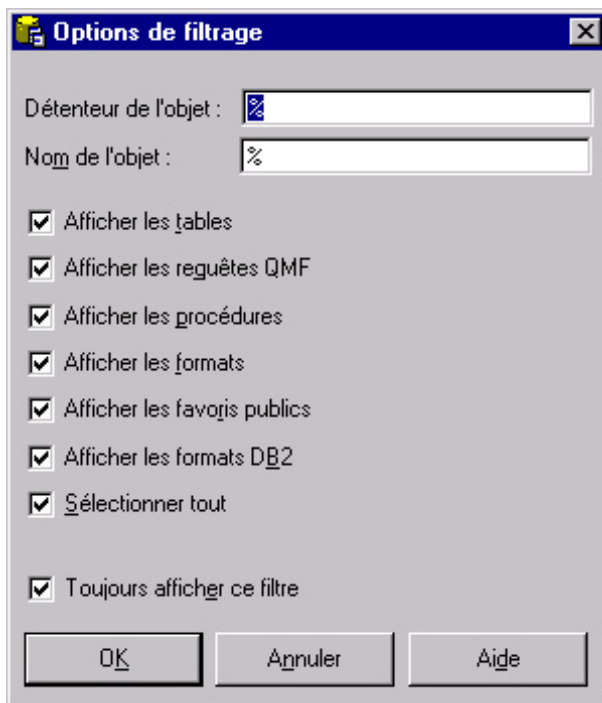


Les objets affichés dans cette fenêtre comprennent une icône représentant le type d'application à laquelle la sortie objet est associée.

Connexion au serveur

1. Si aucun nom de serveur n'apparaît sous Serveurs DB2, cliquez sur le symbole plus (+).

2. Cliquez sur le symbole plus (+) placé à côté d'un serveur. La boîte de dialogue Options de filtrage s'ouvre.



3. Sélectionnez les types d'objet que vous voulez voir apparaître, puis cliquez sur **OK**. Les objets disponibles sur le serveur s'affichent, classés par type d'objet.

Utilisation des rapports et objets

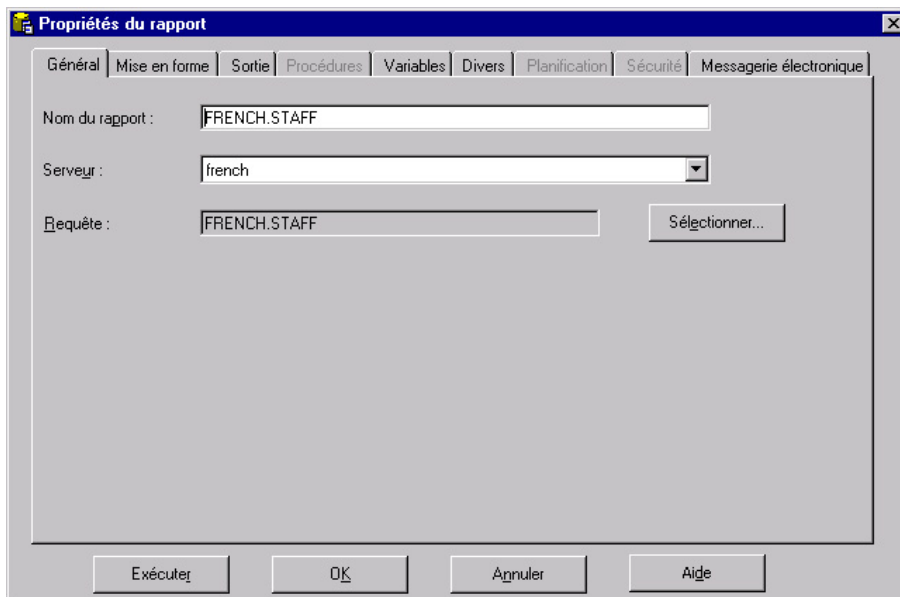
Les rapports correspondent à des objets QMF. Tous les articles contenus dans vos dossiers Favoris personnels et publics sont considérés comme des rapports ; vous pouvez modifier le formatage et afficher des options pour ces articles. Les articles contenus dans ces dossiers Favoris sont reliés aux objets QMF se trouvant sur le serveur. Vous ne modifiez pas vraiment un objet QMF, vous modifiez le lien vers cet objet qu'on appelle rapport. Etant donné que les rapports correspondent à des objets, les caractéristiques des objets s'appliquent également aux rapports.

Vous pouvez créer des rapports à partir d'objets se trouvant sur le serveur. Ils ne seront cependant pas enregistrés sur le serveur. Cette fonctionnalité vous permet de créer rapidement des rapports à usage unique. Après avoir créé un rapport à partir d'objets se trouvant sur le serveur, vous pouvez toutefois l'enregistrer dans un dossier Favoris.

Exécution de rapports

Vous pouvez exécuter des rapports à partir des dossiers Favoris ou à partir des objets se trouvant sur le serveur.

1. Après avoir sélectionné le rapport ou l'objet, cliquez sur *Propriétés* dans le menu Rapport. La boîte de dialogue Propriétés du rapport s'ouvre.



2. Définissez les propriétés si vous le désirez.
3. Cliquez sur **Exécuter**. Le rapport est terminé et si vous sélectionnez l'option *Afficher le rapport après l'édition* dans la boîte de dialogue sorties Propriétés du rapport : Sortie, le rapport s'affiche dans l'application sélectionnée.

Vous pouvez également exécuter rapidement un rapport en suivant une des méthodes suivantes :

- Sélectionnez le rapport, choisissez *Exécuter* dans le menu Rapport.
- Cliquez avec le bouton droit de la souris sur le rapport, puis sur *Exécuter*.
- Cliquez deux fois sur le nom du rapport.

Utilisation des dossiers et des favoris

Les dossiers servent à regrouper les rapports et les objets QMF ; le nom du dossier dépend du nom du détenteur de l'objet. Vous pouvez utiliser les dossiers comme vous utilisez les rapports, par exemple pour exécuter des rapports et définir des caractéristiques de rapport. Si vous effectuez ces opérations sur un dossier, les opérations s'appliqueront dans chacun des

rapports contenus dans ce dossier. Par exemple, si vous voulez exécuter successivement chacun des rapports contenus dans un dossier, sélectionnez le dossier, puis cliquez sur **Exécuter** dans le menu Rapport.

Le Centre d'édition de rapports QMF contient deux dossiers de niveau supérieur dans lesquels vous pouvez stocker des rapports. Les dossiers contiennent des rapports liés à des objets se trouvant sur les serveurs ; les objets eux-mêmes ne se trouvent pas dans les dossiers Favoris. Votre dossier Favoris personnels est local (sur votre PC), vous êtes donc le seul utilisateur à pouvoir accéder à ce dossier et à son contenu. Le dossier Favoris Publics se trouve sur le serveur et tous les utilisateurs autorisés peuvent y avoir accès. Vous pouvez avoir accès à différents dossiers Favoris Publics, en fonctions des limites de vos ressources, il n'y aura cependant toujours qu'un seul dossier Favoris Publics par serveur.

Lorsque vous copiez des objets QMF dans des dossiers Favoris, les dossiers sont automatiquement renommés afin d'inclure le type d'objet et le nom du détenteur. Lorsque vous copiez un type d'objet complet (c'est-à-dire un dossier ou des objets de même type) d'un serveur (par exemple toutes les requêtes), le nom du serveur est également inclus dans le nom du nouveau dossier.

Ajout de rapports aux favoris

Vous pouvez ajouter n'importe quel objet ou rapport provenant d'un serveur au dossier de vos favoris personnels ou au dossier des favoris publics d'un serveur (à condition que vous ayez reçu la permission de votre administrateur de système).

Pour ajouter des rapports aux favoris personnels :

Après avoir sélectionné le rapport ou l'objet, choisissez *Ajouter aux favoris* dans le menu Rapport, ou déplacez le rapport ou l'objet dans le dossier de vos favoris personnels. Un rapport est ajouté au début de votre dossier Favoris personnels avec la convention d'attribution de nom suivante : ObjecttypeOWNERNAME.OBJECTNAME.

Pour ajouter des rapports aux favoris publics :

Déplacez l'objet QMF ou le rapport dans le dossier Favoris Publics d'un serveur. Vous pouvez ajouter des rapports provenant de votre dossier Favoris personnels ou de tout autre serveur.

Remarque : Lorsque vous ajoutez des rapports aux favoris publics ou que vous modifiez des rapports dans ce dossier, vous devez sélectionner *Sauvegarder les modifications dans les favoris publics* dans le menu Rapport avant d'enregistrer les mises à jour sur le serveur.

Pour obtenir de plus amples informations sur l'utilisation du Centre d'édition de rapports QMF, veuillez consulter le système d'aide en ligne.

Chapitre 13. Utilisation de l'API QMF pour Windows

Vous pouvez créer des applications personnalisées en utilisant l'interface de programmation d'application de QMF pour Windows.

Contrôle de QMF pour Windows par le biais de l'API

Les étapes suivantes fournissent un aperçu du mode de fonctionnement de l'API pour le contrôle de QMF pour Windows.

1. Créez une instance de l'objet API QMF pour Windows. Si vous utilisez Microsoft Visual Basic, ajoutez une référence à la bibliothèque de types QMF pour Windows, qmfwin.tlb. Utilisez ensuite l'instruction Dim :

```
Dim QMFWin As New QMFWin
```

Ou l'instruction CreateObject :

```
Dim QMFWin As Object
```

```
Set QMFWin = CreateObject ("QMFWin.Interface")
```

Remarque : Si vous utilisez un environnement de développement différent, veuillez vous reporter à la documentation associée au produit pour réaliser cette opération.

2. Sélectionnez le serveur DB2 que vous souhaitez mettre en service et appelez InitializeServer() pour initialiser une connexion à la base de données.

Remarque : Vous ne pouvez pas initialiser un serveur tant qu'un ID utilisateur et un mot de passe ne sont pas validés par DB2. Vous pouvez configurer QMF pour Windows pour qu'il invite à entrer l'ID utilisateur et le mot de passe, ou bien inviter à les entrer dans votre application et les transmettre comme paramètres dans l'appel de la fonction InitializeServer().

3. Sélectionnez la recherche à exécuter avec InitializeQuery(). Si la requête contient des variables, utilisez la fonction SetVariable() pour définir les valeurs des variables.
4. Ouvrez ou exécutez la requête. La fonction Open() permet d'ouvrir le pointeur de la requête pour les instructions SELECT, tandis que la fonction Execute() permet d'exécuter SQL pour les autres instructions.
5. Si la requête est une instruction SELECT, vous pouvez extraire les lignes de données en appelant de manière récurrente la fonction FetchNextRow(). Pour extraire plusieurs lignes simultanément, utilisez la fonction

FetchNextRows() ou faites appel à CompleteQuery() pour demander à QMF pour Windows d'extraire toutes les lignes.

6. Si la requête est une instruction SELECT, fermez la requête à l'aide de la fonction Close().
7. Mettez fin à l'unité de travail en appelant soit Commit(), soit Rollback().

Blocage d'appels

Toutes les fonctions API QMF pour Windows sont synchrones. Cela implique un blocage, ou l'absence de retour, tant que la requête sur la base de données n'a pas abouti. Cette mise en oeuvre est souhaitable, car elle simplifie la programmation dans l'application cliente. Il faut néanmoins noter que si votre application cliente n'est pas multitâche, elle sera incapable de répondre aux entrées utilisateur ou d'effectuer des régénérations d'écran tant qu'elle attendra le retour de l'une des fonctions de l'API de QMF pour Windows.

L'API de QMF pour Windows ne gère qu'un appel de fonction à la fois de la part d'un client. Si votre application cliente est multitâche, vous devez :

- attendre la fin de l'appel d'une fonction avant d'en lancer une autre, ou
- créer plusieurs instances de l'API QMF pour Windows (une par processus utilisant l'API).

Connexion à la base de données

Chaque instance de l'objet API de QMF pour Windows crée et utilise une connexion unique à la base de données pour toute activité de base de données soumise à une annulation ou une validation ultérieure, comme l'ouverture d'une requête, l'extraction de données ou l'exécution d'instructions SQL.

Si vous créez plusieurs requêtes dans une même instance d'un objet de l'API QMF pour Windows en appelant plusieurs fois InitializeQuery(), toutes les requêtes partagent la même connexion.

L'API de QMF pour Windows ne gère qu'un appel de fonction à la fois de la part d'un client. Si votre application cliente est multitâche, vous devez appeler :

- DeleteQMFObject()
- GetQMFObjectInfo()
- GetQMFObjectInfoEx()
- GetQMFObjectList()
- GetQMFObjectListEx()
- GetQMFQueryText()
- SaveQMFQuery()

QMF for Windows crée et utilise une seconde connexion à la base de données de façon à traiter les tâches administratives sur la base de données (extraction

d'informations QMF par exemple). Cette deuxième connexion est nécessaire pour gérer un mécanisme cohérent d'annulation et de validation pour les applications clientes.

Ces connexions à la base de données sont gérées automatiquement par l'objet API de QMF pour Windows. Cependant, si l'administrateur système a limité le nombre de connexions autorisées, tenez compte du fait que chaque instance de l'objet QMF pour Windows est susceptible d'utiliser deux connexions.

Aide-mémoire sur les API

Cet aide-mémoire énumère toutes les commandes disponibles pour la création d'applications en utilisant l'API de QMF pour Windows.

AddDecimalHostVariable()

short AddDecimalHostVariable(long *IDRequête*, short *Type*, short *Précision*, short *Echelle*, const VARIANT& *Valeur*)

Description

Cette fonction affecte les données de *Valeur* à une variable de l'instruction SQL statique initialisée avec *IDRequête*. Vous appelez cette fonction pour chaque variable dans l'instruction. QMF pour Windows ne tente pas de faire coïncider les valeurs avec les variables ; vous devez dès lors appeler cette fonction dans le même ordre que les variables dans l'instruction SQL.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeStaticQuery().
<i>Type</i>	Le type de données SQL de la valeur à transmettre au serveur de base de données. Cette valeur influence la conversion de la <i>Valeur</i> d'un type de données VARIABLE à la valeur effectivement transmise. La seule valeur admise pour AddDecimalHostVariable() est 484 (RSDT_DECIMAL).
<i>Précision</i>	La précision de la valeur décimale.
<i>Echelle</i>	L'échelle de la valeur décimale.
<i>Valeur</i>	La valeur de données à substituer dans l'instruction. Pour spécifier une valeur nulle, le type de la variante doit être positionné sur VT_EMPTY.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou GetLastErrorType() pour obtenir des informations complémentaires à propos de l'erreur.

AddHostVariable()

short AddHostVariable(long *IDRequête*, short *Type*, const VARIANT& *Value*)

Description

Cette fonction affecte les données de Valeur à une variable de l'instruction SQL statique initialisée avec *IDRequête*. Vous appelez cette fonction pour chaque variable dans l'instruction. QMF pour Windows ne cherchant pas à associer les valeurs aux variables, vous devez appeler cette fonction en respectant l'ordre des variables dans l'instruction SQL.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeStaticQuery().
<i>Type</i>	Le type de données SQL de la valeur à transmettre au serveur de la base de données. Cette valeur influence la conversion de la <i>Valeur</i> d'un type de données VARIANTE à la valeur effectivement transmise.
<i>Valeur</i>	La valeur de données à substituer dans l'instruction. Pour spécifier une valeur nulle, le type de la variante doit être positionné sur VT_EMPTY.

Les valeurs valables pour le *Type* comprennent :

Valeur	Signification
384 (RSDT_DATE)	Date
388 (RSDT_TIME)	Heure
392 (RSDT_TIMESTAMP)	Horodatage
448 (RSDT_VARCHAR)	Longueur de la chaîne de caractères de la variable
452 (RSDT_CHAR)	Chaîne de caractères
464 (RSDT_VARGRAPHIC)	Longueur graphique de la variable
468 (RSDT_GRAPHIC)	Graphique
480 (RSDT_FLOAT)	Chiffre de la décimale flottante
496 (RSDT_INTEGER)	Nombre entier de 4 octets
500 (RSDT_SMALLINT)	Nombre entier de 2 octets

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()` pour obtenir des informations complémentaires à propos de l'erreur.

BindDecimalHostVariable()

short BindDecimalHostVariable(BSTR *NomCollecte*, BSTR *NomModule*, short *NuméroSection*, short *Numéro*, BSTR *Nom*, short *TypeDonnées*, short *Précision*, short *Echelle*)

Description

Cette fonction relie une variable dans la section indiquée. Mentionnez ":H" dans le texte SQL comme marqueur destiné à une variable hôte. La fonction `BindDecimalHostVariable()` permet de spécifier des informations propres à une variable, ce que vous devez faire pour chacune des variables décimales hôte du texte SQL.

Paramètres

Nom	Description
NomCollecte	L'ID de collecte du module dont vous voulez éditer les liens.
NomModule	Le nom du module dont vous voulez éditer les liens.
NuméroSection	Le numéro de section de l'instruction dans la collecte et le module dont vous voulez éditer les liens.
Nombre	L'identificateur de la variable dont vous voulez éditer les liens. La première variable de l'instruction SQL est la variable 0, etc.
Nom	Exploité par le serveur de base de données à des fins de diagnostic. Cette valeur n'est ni validée, ni requise par QMF pour Windows.
TypeDonnées	Le type de données SQL de la variable. La seule valeur admise pour <code>BindDecimalHostVariable()</code> est 484 (RSDT_DECIMAL).
Précision	La précision de la valeur décimale.
Echelle	L'échelle de la valeur décimale.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

BindHostVariable()

short BindHostVariable(BSTR *NomCollecte*, BSTR *NomModule*, short *NuméroSection*, short *Numéro*, BSTR *Nom*, short *TypeDonnées*, short *Longueur*)

Description

Cette fonction relie une variable dans la section indiquée. Mentionnez ":H" dans le texte SQL comme marqueur destiné à une variable hôte. La fonction BindHostVariable() permet de spécifier des informations propres à une variable, ce que vous devez faire pour chacune des variables décimales hôte du texte SQL.

Paramètres

Nom	Description
<i>NomCollecte</i>	L'ID de collecte du module dont vous voulez éditer les liens.
<i>NomModule</i>	Le nom du module dont vous voulez éditer les liens.
<i>NuméroSection</i>	Le numéro de section de l'instruction dans la collecte et le module dont vous voulez éditer les liens.
<i>Numéro</i>	L'identificateur de la variable dont vous voulez éditer les liens. La première variable de l'instruction SQL est la variable 0, etc.
Nom	Exploité par le serveur de base de données à des fins de diagnostic. Cette valeur n'est ni validée, ni requise par QMF pour Windows.
<i>TypeDonnées</i>	Le type de données SQL de la variable.
Longueur	La longueur de la variable.

Les valeurs valables pour le *TypeDonnées* comprennent :

Valeur	Signification
384 (RSDT_DATE)	Date
388 (RSDT_TIME)	Heure
392 (RSDT_TIMESTAMP)	Horodatage
448 (RSDT_VARCHAR)	Longueur de la chaîne de caractères de la variable
452 (RSDT_CHAR)	Chaîne de caractères
464 (RSDT_VARGRAPHIC)	Longueur graphique de la variable
468 (RSDT_GRAPHIC)	Graphique
480 (RSDT_FLOAT)	Chiffre de la décimale flottante
484 (RSDT_DECIMAL)	Décimale
496 (RSDT_INTEGER)	Nombre entier de 4 octets
500 (RSDT_SMALLINT)	Nombre entier de 2 octets

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

BindSection()

short BindSection(BSTR *NomCollecte*, BSTR *NomModule*, short *NuméroSection*, BSTR *TexteSQL*)

Description

Cette fonction détermine le texte SQL à utiliser dans le numéro de section spécifié de la collecte et du module lors de l'édition de liens.

Paramètres

Nom	Description
NomCollecte	L'ID de collecte du module dont vous voulez éditer les liens.
NomModule	Le nom du module dont vous voulez éditer les liens.
NuméroSection	Le numéro de section de l'instruction dans la collecte et le module dont vous voulez éditer les liens.
<i>TexteSQL</i>	Le texte SQL de l'instruction dont vous voulez éditer les liens.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

CancelBind()

short CancelBind(BSTR *NomCollecte*, BSTR *NomModule*)

Description

Cette fonction permet d'annuler une opération d'édition de liens déjà initialisée. Toutes les informations relatives au module indiqué sont fournies.

Paramètres

Nom	Description
NomCollecte	L'ID de collecte du module dont vous voulez éditer les liens.
NomModule	Le nom du module dont vous voulez éditer les liens.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() pour obtenir des informations complémentaires à propos de l'erreur.

ChangePassword()

short ChangePassword(BSTR *NouveauMotDePasse*)

Description

Cette fonction modifie le mot de passe pour l'ID utilisateur précédemment spécifié dans l'appel InitializeServer().

Remarque : Tous les types de serveurs de base de données ne permettent pas de modifier les mots de passe. Si le serveur spécifié lors de l'appel InitializeServer() ne permet pas de modifier les mots de passe, un message d'erreur est envoyé et le mot de passe n'est pas modifié.

Paramètres

Nom	Description
NouveauMotDePasse	Le nouveau mot de passe.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() pour obtenir des informations complémentaires à propos de l'erreur.

ClearList()

short ClearList(short *Type*)

Description

Cette fonction réinitialise la liste interne spécifiée par le *Type* de paramètre.

Paramètres

Nom	Description
<i>Type</i>	Soit la valeur RSL_SERVER, soit SL_QUERY.

Valeur renvoyée

Zéro en cas de succès, RS_ERROR_OUTOFRANGE en cas d'échec.

Sujets connexes

Open()

Close()

short Close(long *IDRequête*)

Description

Cette fonction ferme une requête et annule la *IDRequête*. Si un pointeur est ouvert pour la requête, le pointeur est fermé, libérant la base de données pour d'autres utilisateurs. Cette fonction ne met pas un terme à la connexion au serveur de la base de données. Etant donné que la connexion demeure ouverte, aucune annulation ou validation n'est effectuée.

Remarque : Le nom de cette fonction s'oppose au mot clé Fermer de Microsoft Access 2.0. Si vous utilisez MS Access 2.0, placez des accolades [] autour de ce nom de fonction.

Paramètres

Nom	Description
IDRequête	L'ID de la requête telle que générée à partir d'InitializeQuery().

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou GetLastErrorType(), GetLastErrorCode(), GetLastError() ou GetLastErrorSQLState() pour obtenir des informations complémentaires à propos de l'erreur.

Sujets connexes

Execute()

Open()

Commit()

short Close(long *IDRequête*)

Description

Cette fonction valide toutes les modifications effectuées dans l'unité de travail, met fin à celle-ci, ferme les pointeurs ouverts et invalide tous les ID de requête en attente.

Remarque : Le nom de cette fonction s'oppose au mot clé Sauvegarder de Microsoft Access 2.0. Si vous utilisez MS Access 2.0, placez des accolades [] autour de ce nom de fonction.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() pour obtenir des informations complémentaires à propos de l'erreur.

Sujets connexes

Rollback()

CompleteQuery()

short CompleteQuery(long *IDRequête*)

Description

Cette fonction extrait toutes les lignes d'un ensemble de résultats et les sauvegarde dans QMF pour Windows. Si un pointeur est ouvert pour la requête, le pointeur est fermé, libérant la base de données pour d'autres utilisateurs. Vous pouvez utiliser FetchNextRow() ou FetchNextRows() pour extraire les lignes. Appelez Close() lorsque vous en avez terminé avec cette requête.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeQuery().

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() pour obtenir des informations complémentaires à propos de l'erreur.

CopyToClipboard()

short CopyToClipboard(long *IDRequête*, long *PremièreLigne*, long *PremièreColonne*, long *DernièreLigne*, long *DernièreColonne*, BOOL *InclureEnTêtesColonnes*, [VARIANT *DateTimeFormat*])

Description

Cette fonction copie la fourchette de lignes et de colonnes précisée vers le Presse-papier. Si vous n'avez pas encore extrait les données de ligne de toutes les lignes que vous souhaitez copier dans le presse-papiers, appelez préalablement la fonction CompleteQuery(). Un message d'erreur s'affiche si vous tentez de copier des lignes qui n'ont pas été extraites de la base de données.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeQuery().
<i>PremièreLigne</i>	La première ligne que vous voulez inclure dans la copie.
<i>PremièreColonne</i>	La première colonne que vous voulez inclure dans la copie.
<i>DernièreLigne</i>	La dernière ligne que vous voulez inclure dans la copie ou -1 si toutes les lignes sont reprises.
<i>DernièreColonne</i>	La dernière colonne que vous voulez inclure dans la copie ou -1 si toutes les colonnes sont reprises.
<i>InclureEnTêtesColonnes</i>	Utilisez une valeur autre que zéro pour reprendre les en-têtes de colonne dans la première ligne et zéro pour les exclure.
<i>FormatDateHeure</i>	Facultativement, le format à utiliser pour les valeurs de date et d'heures. Les valeurs valides sont 0 (format ISO), 1 (format américain), 2 (format EUR), 3 (format JIS) ou 4 (format Panneau de configuration Windows). La valeur par défaut est 4.

Remarque : La valeur de la première ligne dans un ensemble de résultats est 0, et la valeur de la dernière ligne est inférieure d'une unité au nombre total de lignes. La valeur de la première colonne dans un ensemble de résultats est 0, et la valeur de la dernière colonne est inférieure d'une unité au nombre total de colonnes.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou GetLastErrorType() pour obtenir des informations complémentaires à propos de l'erreur. Si l'ensemble de résultats est vide ou qu'aucune ligne n'a été extraite de la base de données, une valeur autre que zéro est renvoyée, sauf si *PremièreLigne*=0 et *DernièreLigne*=1. Dans ce cas, la valeur zéro est renvoyée et une chaîne vide est copiée dans le presse-papiers.

DeleteQMFObject()

short DeleteQMFObject(BSTR *DétenteurEtNom*)

Description

Cette fonction supprime un objet QMF (requête, format, procédure ou table).

Paramètres

Nom	Description
-----	-------------

<i>DétenteurEtNom</i>	Une chaîne contenant le propriétaire et le nom, séparés par un point, de l'objet que vous voulez supprimer. Exemple John.Query2
-----------------------	------------------------------------------------------------------------------------------------------------------------------------

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

EndBind()

short EndBind(BSTR *NomCollecte*, BSTR *NomModule*)

Description

Cette fonction termine le processus d'édition de liens sur un module SQL statique. Cette fonction demande à QMF pour Windows d'envoyer la totalité des informations sur le module concerné à la base de données en vue de son traitement.

Paramètres

Nom	Description
<i>NomCollecte</i>	Le nom de collecte utilisé lors de la précédente demande de <code>StartBind()</code> .
<i>NomModule</i>	Le nom de module utilisé lors de la précédente demande de <code>StartBind()</code> .

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

Execute()

short Execute(long *IDRequête*)

Description

Cette fonction exécute une instruction SQL contenant une commande SQL autre que SELECT. Utilisez `Execute()` lorsque l'instruction ne génère aucun résultat. Pour les instructions générant effectivement des résultats, utiliser `ExecuteEx()`. Pour les instructions contenant la commande SELECT, faites appel à `Open()` et non à `Execute()` ou `ExecuteEx()`. Pour déterminer l'instruction utilisée par une requête, sollicitez `GetQueryVerb()`.

Remarque : Le nom de cette fonction s'oppose au mot clé Exécuter de Microsoft Access 2.0. Si vous utilisez MS Access 2.0, placez des accolades [] autour de ce nom de fonction.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeQuery().

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou GetLastErrorType(), GetLastErrorCode(), GetLastErrorError() ou GetLastErrorSQLState() pour obtenir des informations complémentaires à propos de l'erreur.

Sujets connexes

Execute()

Open()

ExecuteEx()

short ExecuteEx(long *IDRequête*, VARIANT* *Résultat*)

Description

Cette fonction exécute une instruction SQL qui utilise une instruction SQL autre que SELECT. Utilisez ExecuteEx() lorsque l'instruction génère des résultats, par exemple avec une instruction SELECT INTO. Pour les instructions qui ne génèrent pas de résultats, utilisez Execute(). Pour les instructions contenant la commande SELECT, faites appel à Open() et non Execute() ou ExecuteEx(). Pour déterminer l'instruction utilisée par une requête, sollicitez GetQueryVerb().

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeQuery().
<i>Résultats</i>	<p>Un pointeur dirigé vers VARIANT dans lequel le résultat est stocké. Le résultat est un tableau (variant type VT_ARRAY VT_VARIANT) contenant une valeur pour chaque colonne dans le résultat.</p> <p>Chaque valeur est spécifiée soit dans son type de donnée initiale, soit dans le type de donnée variant le plus proche. Les types de résultats supportés sont les suivants : chaîne (type de variante VT_BSTR), flottant (type de variante VT_R4), double (type de variante VT_R8), short (type de variante VT_I2), long (type de variante VT_I4) et binaire (type de variante VT_UI1 VT_ARRAY).</p> <p>Vous devez correctement initialiser VARIANT avant d'appeler cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent solliciter VariantInit().</p>

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() pour obtenir des informations complémentaires à propos de l'erreur.

ExecuteStored Procedure()

```
short ExecuteStoredProcedure(long IDRequête,[VARIANTE vaCommitOK],  
[VARIANTE vaMaxResultSets], [VARIANTE vaColumnNames], [VARIANTE  
vaColumnLabels], [VARIANTE vaColumnComments])
```

Description

Cette fonction exécute une instruction SQL contenant la commande SQL CALL pour exécuter une procédure sauvegardée au niveau du serveur de base de donnée. Utilisez ExecuteStoredProcedure() lorsque la procédure sauvegardée ne génère aucun résultat (au lieu de ou en plus des ensembles de résultats). Pour les procédures sauvegardées générant effectivement des résultats, utiliser ExecuteStoredProcedureEx().

Pour initialiser une procédure sauvegardée en vue de son exécution avec ExecuteStoredProcedure(), appelez tout d'abord InitializeQuery() en spécifiant une instruction SQL contenant la commande CALL. Le nom de la procédure sauvegardée doit être spécifié comme un littéral dans l'instruction CALL. Tous

les paramètres spécifiés dans l'instruction CALL (constants ou autres) sont ignorés. Utilisez de préférence AddHostVariable() pour indiquer les variables d'entrée et de sortie.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeQuery(). Le texte SQL de la requête devrait préciser une instruction CALL.
<i>vaCommitOK</i>	Une valeur booléenne en option spécifiant si la procédure stockée peut sauvegarder l'unité de travail ou si cette opération doit être réservée. La valeur par défaut est True.
<i>vaMaxResultSets</i>	Une valeur numérique en option spécifiant le nombre maximal d'ensembles de résultats que la procédure stockée sera autorisée à générer. Spécifier la valeur zéro si vous ne souhaitez pas que la procédure stockée génère le moindre ensemble de résultats ou si le serveur de base de données ne supporte pas la génération d'ensembles de résultats à partir de procédures stockées sur DRDA.
<i>vaColumnNames</i>	Une valeur booléenne en option spécifiant si la base de données doit générer les noms de colonne dans chaque ensemble de résultats généré.
<i>vaColumnLabels</i>	Une valeur booléenne en option spécifiant si la base de données doit générer les labels de colonne dans chaque ensemble de résultats généré.
<i>vaColumnComments</i>	Une valeur booléenne en option spécifiant si la base de données doit générer les commentaires de colonne dans chaque ensemble de résultats généré.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() pour obtenir des informations complémentaires à propos de l'erreur.

ExecuteStored ProcedureEx()

```
short ExecuteStoredProcedureEx(long IDRequête, VARIANTE*
Résultat, [VARIANTE vaCommitOK], [VARIANTE vaMaxResultSets], [VARIANT
vaColumnNames], [VARIANTE vaColumnLabels], [VARIANTE
vaColumnComments])
```

Description

Cette fonction exécute une instruction SQL contenant la commande SQL CALL, pour exécuter une procédure sauvegardée au niveau du serveur de base de donnée. Utilisez `ExecuteStoredProcEx()` lorsque la procédure sauvegardée génère des résultats (au lieu de ou en plus des ensembles de résultats). Pour les procédures sauvegardées générant effectivement des résultats, utiliser `ExecuteStoredProcEx()`.

Pour initialiser une procédure sauvegardée en vue de son exécution avec `ExecuteStoredProc()`, appelez tout d'abord `InitializeQuery()` en spécifiant une instruction SQL CALL. Le nom de la procédure sauvegardée doit être spécifié comme un littéral dans l'instruction CALL. Tous les paramètres spécifiés dans l'instruction CALL (constants ou autres) sont ignorés. Utilisez de préférence `AddHostVariable()` pour indiquer les variables d'entrée et de sortie.

Si la procédure sauvegardée génère des ensembles de résultats, appelez `GetStoredProcResultSets()` pour retrouver les ID de requête pour les ensembles de résultats.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d' <code>InitializeQuery()</code> . Le texte SQL de la requête devrait préciser une instruction CALL.
<i>Résultats</i>	<p>Un pointeur dirigé vers VARIANT dans lequel le résultat est sauvegardé. Le résultat est un tableau (variant type VT_ARRAY VT_VARIANT) contenant une valeur pour chaque colonne dans le résultat.</p> <p>Chaque valeur est spécifiée soit dans son type de donnée initiale, soit dans le type de donnée variante le plus proche. Les types de résultats supportés sont les suivants : chaîne (type de variante VT_BSTR), flottant (type de variante VT_R4), double (type de variante VT_R8), short (type de variante VT_I2), long (type de variante VT_I4) et binaire (type de variante VT_UI1 VT_ARRAY).</p> <p>Vous devez correctement initialiser VARIANT avant d'appeler cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent solliciter <code>VariantInit()</code>.</p>
<i>vaCommitOK</i>	Une valeur booléenne facultative indiquant si la procédure stockée peut valider l'unité de travail ou si l'opération doit être restreinte. La valeur par défaut est True.

<i>vaMaxResultSets</i>	Une valeur numérique en option spécifiant le nombre maximal d'ensembles de résultats que la procédure sauvegardée sera autorisée à générer. Spécifier la valeur zéro si vous ne souhaitez pas que la procédure sauvegardée génère le moindre ensemble de résultats ou si le serveur de base de données ne supporte pas la génération d'ensembles de résultats à partir de procédures stockées sur DRDA.
<i>vaColumnNames</i>	Une valeur booléenne en option spécifiant si la base de données doit générer les noms de colonne dans chaque ensemble de résultats généré.
<i>vaColumnLabels</i>	Une valeur booléenne en option spécifiant si la base de données doit générer les labels de colonne dans chaque ensemble de résultats généré.
<i>vaColumnComments</i>	Une valeur booléenne en option spécifiant si la base de données doit générer les commentaires de colonne dans chaque ensemble de résultats généré.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

Export()

`shortExport(long IDRequête, long PremièreLigne, long PremièreColonne, long DernièreLigne, long DernièreColonne, short Format, short StringDelimétrer, short ColumnDelimétrer, BOOL InclureEnTêtesColonnes, BSTR FileName, [VARIANT DateTimeFormat])`

Description

Cette fonction copie la fourchette de lignes et de colonnes précisée vers le Presse-papiers. Si vous n'avez pas encore extrait les données de toutes les lignes que vous souhaitez copier dans le presse-papiers, appelez préalablement la fonction `CompleteQuery()`. Un message d'erreur s'affiche si vous tentez de copier des lignes qui n'ont pas été extraites de la base de données.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d' <code>InitializeQuery()</code> .
<i>PremièreLigne</i>	La première ligne que vous voulez inclure dans l'exportation.

PremièreColonne	La première colonne que vous voulez inclure dans l'exportation.
DernièreLigne	La dernière ligne que vous voulez inclure dans la copie ou -1 si toutes les lignes sont reprises.
DernièreColonne	La dernière colonne que vous voulez inclure dans la copie ou -1 si toutes les colonnes sont reprises.
InclureEnTêtesColonnes	Utilisez une valeur autre que zéro pour reprendre les en-têtes de colonne dans la première ligne et zéro pour les exclure.
Filename	Une chaîne contenant le nom du fichier vers lequel vous souhaitez écrire l'exportation.
<i>FormatDateHeure</i>	Facultativement, le format à utiliser pour les valeurs de date et d'heures. Les valeurs valides sont 0 (format ISO), 1 (format américain), 2 (format EUR), 3 (format JIS) ou 4 (format Panneau de configuration Windows). La valeur par défaut est 4.

Remarque : La valeur de la première ligne dans un ensemble de résultats est 0, et la valeur de la dernière ligne est inférieure d'une unité au nombre total de lignes. La valeur de la première colonne dans un ensemble de résultats est 0, et la valeur de la dernière colonne est inférieure d'une unité au nombre total de colonnes.

Nom	Description
<i>Format</i>	Spécifie le format de sortie.

Valeur	Signification
0 (RSEF_TEXT)	Le fichier de sortie est écrit en format texte clair.
1 (RSEF_HTML)	Le fichier de sortie est rédigé en format HTML et les données sont organisées dans une table HTML.
2 (RSEF_CSV)	Le fichier de sortie est écrit en format CSV (valeurs séparées par une virgule).
3 (RSEF_PCIXF)	Le fichier de sortie est rédigé en format PC/ IXF.
4 (RSEF_S370IXF)	Le fichier de sortie est rédigé en format System/370 IXF.

Nom	Description
<i>Séparateur de chaîne</i>	Spécifie le séparateur de chaîne. Ce paramètre est ignoré si le <i>Format</i> est RSEF_HTML.

Valeur	Signification
0 (RSSD_NONE)	Pas de séparateur de chaîne utilisé.
1 (RSSD_SINGLEQUOTE)	Les chaînes sont séparées par un caractère de quotation unique (').
2 (RSSD_DOUBLEQUOTE)	Les chaînes sont séparées par un caractère de quotation double (").

Nom	Description
<i>Séparateur de colonne</i>	Spécifie le séparateur de colonne. Ce paramètre est ignoré si le <i>Format</i> est RSEF_HTML.

Valeur	Signification
0 (RSCD_SPACE)	Les colonnes sont séparées par un caractère d'espace (' ').
1 (RSCD_TAB)	Les colonnes sont séparées par un caractère de tabulation ('\t').
2 (RSCD_COMMA)	Les colonnes sont séparées par une virgule (',').

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()` pour obtenir des informations complémentaires à propos de l'erreur. Si l'ensemble de résultats est vide ou qu'aucune ligne n'a été extraite de la base de données, une valeur autre que zéro est renvoyée, sauf si *PremièreLigne*=0 et *DernièreLigne*=1. Dans ce cas, la valeur zéro est renvoyée et une chaîne vide est copiée dans le presse-papiers.

Sujets connexes

`CopyToClipboard()`

ExportForm()

short `ExportForm(BSTR DétenteurEtNom, BSTR FileName)`

Description

Cette fonction exporte le format QMF spécifié dans le fichier précisé.

Paramètres

Nom	Description
DétenteurEtNom	Une chaîne contenant le propriétaire et le nom, séparés par un point, du format que vous voulez exporter. Exemple John.Query2
FileName	Une chaîne contenant le nom du fichier vers lequel vous souhaitez écrire le format exporté.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLException()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

Sujets connexes

`PrintReport()`

ExportReport()

`short ExportReport(long IDRequête, short SourceType, BSTR Source, BSTR OutputFileName, short PageLength, short PageWidth, BOOL IncludeDateTime, BOOL IncludePageNumbers, [VARIANT Format], [VARIANTE UseFormPageSetup])`

Description

Cette fonction génère un rapport pour la requête spécifiée et l'écrit dans un fichier. Vous spécifiez le formattage et la mise en page pour le rapport dans un format QMF. Le fichier de sortie est un fichier texte ASCII dont chaque ligne est séparée par deux retours de chariot et des caractères interligne et dont chaque page est séparée par un caractère de présentation de feuille. Il est recommandé de visualiser le fichier de sortie en utilisant une police à pitch déterminé.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d' <code>InitializeQuery()</code> .
<i>Source</i>	Le nom (soit un nom de fichier, soit un nom.propriétaire) du format que vous souhaitez utiliser.
<i>OutputFileName</i>	Le nom du fichier vers lequel vous souhaitez écrire le rapport.
<i>PageLength</i>	Le nombre de lignes sur chaque page du rapport. Une <i>PageLength</i> de -1 spécifie une sortie permanente (pas de rupture de page, à moins que la largeur du rapport n'excède <i>PageWidth</i>).

IncludeDateTime	La valeur autre que zéro spécifie que la date et l'heure sont indiquées au pied de chaque page. La valeur zéro spécifie que la date et l'heure ne sont pas indiquées.
IncludePageNumbers	La valeur autre que zéro spécifie que les numéros de page sont indiqués au pied de chaque page. La valeur zéro spécifie que les numéros de page ne sont pas indiqués.
Format	Spécifie de manière facultative le format du rapport exporté. Si la valeur est de zéro, le format est le texte clair, en précisant que la sortie doit être exactement celle générée par le format (texte ou HTML, en fonction du type de format). Si la valeur est autre que zéro, le format est HTML, en spécifiant que la sortie doit être HTML. Avec les formats non HTML, la sortie est convertie en HTML en ajoutant des descripteurs HTML au début et à la fin de la sortie. La valeur par défaut est zéro.
<i>FormatDateHeure</i>	Facultativement, le format à utiliser pour les valeurs de date et d'heures. Les valeurs valides sont 0 (format ISO), 1 (format américain), 2 (format EUR), 3 (format JIS) ou 4 (format Panneau de configuration Windows). La valeur par défaut est 4.
Format	Le format du fichier de sortie.
UseFormPageSetup	Facultativement, si la valeur autre que zéro indique que les paramètres <i>PageLength</i> , <i>PageWidth</i> , <i>IncludeDateTime</i> et <i>IncludePageNumbers</i> devraient être ignorés et que les valeurs de ces paramètres devraient de préférence être prises à partir des valeurs sauvegardées avec le format spécifié. La valeur par défaut est zéro.

Valeur	Signification
0 (RSF_DEFAULT)	Utilisez le format par défaut. <i>FormName</i> doit être une chaîne vide.
1 (RSF_DATABASE)	Utilisez un format extrait de la base de données. Spécifiez le propriétaire du format et son nom (Nom.propriétaire) dans le paramètre <i>FormName</i> . Pour utiliser un format situé sur un serveur de base de données autre, utilisez tout d'abord <code>ExportForm()</code> pour exporter le format vers un fichier et spécifiez ensuite un <i>SourceType</i> du RSF_FILE.
2 (RSF_FILE)	Utilisez un format extrait d'un fichier. Spécifiez le nom du fichier dans le paramètre <i>FormName</i> .

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou

GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() pour obtenir des informations complémentaires à propos de l'erreur.

Sujets connexes

ExportForm()

FastSaveData()

short FastSaveData(long *IDRequête*, BOOL *Replace*, BSTR *Tablename*, BSTR *TableSpaceName*, [VARIANTE *Comment*])

Description

Cette fonction génère un rapport pour la requête spécifiée et l'écrit dans un fichier. Vous spécifiez le formatage et la mise en page pour le rapport dans un format QMF. Le fichier de sortie est un fichier texte ASCII dont chaque ligne est séparée par deux retours de chariot et des caractères interligne et dont chaque page est séparée par un caractère de présentation de feuille. Il est recommandé de visualiser le fichier de sortie en utilisant une police à pitch déterminé.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeQuery().
<i>Replace</i>	Utilisez une valeur autre que zéro si vous souhaitez que les valeurs indiquées remplacent n'importe quelle donnée existante dans la table. Utilisez la valeur zéro si vous souhaitez que les valeurs indiquées soient annexées à n'importe quelle donnée existante dans la table.
<i>TableName</i>	Le nom de la table dans laquelle les données seront stockées. Si la table n'existe pas, QMF pour Windows la crée.
<i>TableSpaceName</i>	Le nom de l'ensemble des tables dans lequel la table existe ou est créée. Si <i>TableSpaceName</i> est une chaîne NULLE ou vide, QMF pour Windows utilise l'ensemble des tables par défaut. Si vous avez configuré QMF pour Windows pour toujours utiliser l'ensemble des tables par défaut, ce paramètre est ignoré. Cf. RSR_SDDIFFERENTTS dans la description de GetResourceLimit().
Commentaires	Facultativement, une chaîne reprenant des commentaires relatifs à la table dans laquelle les données sont sauvegardées.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou

GetLastErrorType(), GetLastErrorCode(), GetLastErrorError() ou GetLastErrorState() pour obtenir des informations complémentaires à propos de l'erreur.

FetchNextRow()

short FetchNextRow(long *IDRequête*, VARIANTE* *Row*)

Description

Cette fonction extrait la ligne de données suivante de la base de données.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeQuery().
Ligne	<p>Un pointeur dirigé vers VARIANT dans lequel le résultat est sauvegardé. Le résultat est un tableau (type de variante VT_ARRAY / VT_VARIANT) reprenant une valeur pour chaque colonne dans le résultat. Appelez GetColumnCount() pour déterminer le nombre de valeurs dans le tableau.</p> <p>Chaque valeur est spécifiée soit dans son type de donnée initiale, soit dans le type de donnée variante le plus proche. Les types de résultats supportés sont les suivants : chaîne (type de variante VT_BSTR), flottant (type de variante VT_R4), double (type de variante VT_R8), short (type de variante VT_I2), long (type de variante VT_I4) et binaire (type de variante VT_UI1 VT_ARRAY).</p> <p>Lorsque la fin de l'ensemble de résultats a été atteinte (il n'y a plus de ligne à extraire) ou lorsqu'un ensemble de résultats est vide, le résultat est vierge (type de variante VT_EMPTY) au lieu d'un tableau.</p> <p>Vous devez correctement initialiser VARIANT avant d'appeler cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent solliciter VariantInit().</p>

Remarque : En raison d'un bogue dans Microsoft Excel 7.0 et Microsoft Access 7.0 (voire dans d'autres produits 32-bit de Microsoft utilisant Visual Basic pour Applications), les données de chaîne dans les variables Variant reçues de QMF pour Windows pourraient ne pas être traduites à partir d'Unicode (utilisé par OLE) vers ANSI (utilisé par VBA). Lorsque cela se produit, seul le premier caractère de la chaîne est affiché. Pour résoudre ce problème, paramétrez la variable à une valeur égale à une chaîne vide avant d'appeler la fonction QMF pour Windows qui utilise cette variable.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Lorsque la fin de la série de résultats est atteinte, la valeur renvoyée est égale à -1. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

Sujets connexes

`FetchNextRows()`

FetchNextRowEx()

`short FetchNextRowEx(long IDRequête)`

Description

Cette fonction extrait la ligne de données suivante de la base de données. Vous pouvez utiliser cette fonction dans des environnements qui ne supportent pas les tableaux VARIANT, comme Microsoft Access 2.0. Utilisez cette fonction avec `GetColumnValue()` pour extraire les données dans chaque colonne de la ligne actuelle.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d' <code>InitializeQuery()</code> .

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Lorsque la fin de la série de résultats est atteinte, la valeur renvoyée est égale à -1. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

Sujets connexes

`FetchNextRowsEx()`

FetchNextRows()

`short FetchNextRows(long IDRequête, VARIANTE* Rows, long* NombreLignes)`

Description

Cette fonction extrait la *NombreLignes* de données suivante de la base de données.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeQuery().
Ligne	<p>Un pointeur dirigé vers VARIANT dans lequel le résultat est sauvegardé. Le résultat est un tableau en deux dimensions (type de variante VT_ARRAY VT_VARIANT) contenant une valeur pour chaque colonne dans chaque ligne. Appelez GetColumnCount() pour déterminer le nombre de colonnes dans le tableau. Les dimensions du tableau sont [<i>NombreLignes</i>][<i>NombreColonnes</i>], même si le nombre de lignes non extraites dans l'ensemble de résultats est inférieur à la valeur <i>NombreLignes</i> (dans ce cas, le tableau contient des entrées supplémentaires, inutilisées).</p> <p>Chaque valeur est spécifiée soit dans son type de donnée initiale, soit dans le type de donnée variante le plus proche. Les types de résultats supportés sont les suivants : chaîne (type de variante VT_BSTR), flottant (type de variante VT_R4), double (type de variante VT_R8), short (type de variante VT_I2), long (type de variante VT_I4) et binaire (type de variante VT_UI1 VT_ARRAY).</p> <p>Lorsque la fin de la série de résultats a été atteinte (il n'y a plus de ligne à extraire) ou lorsqu'une série de résultats est vide, le résultat est vierge (type de variante VT_EMPTY) au lieu d'un tableau</p> <p>Vous devez correctement initialiser VARIANT avant d'appeler cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent solliciter VariantInit().</p>
NombreLignes	Un pointeur vers un long contenant le nombre de ligne à extraire. Si le nombre de lignes non extraites dans la série de résultats est inférieur à <i>NombreLignes</i> , le <i>NombreLignes</i> est réinitialisé au nombre réel de lignes reprises dans les résultats.

Remarque : En raison d'un bogue dans Microsoft Excel 7.0 et Microsoft Access 7.0 (voire dans d'autres produits 32-bit de Microsoft utilisant Visual Basic pour Applications), les données de chaîne dans les variables Variante reçues de QMF pour Windows pourraient ne pas être traduites à partir d'Unicode (utilisé par OLE) vers ANSI (utilisé par VBA). Lorsque cela se produit, seul le premier caractère de la chaîne est affiché. Pour résoudre ce problème, paramétrez la variable à une valeur égale à une chaîne vide avant d'appeler la fonction QMF pour Windows qui utilise cette variable.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Lorsque la fin de la série de résultats est atteinte, la valeur renvoyée est égale à -1. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

Sujets connexes

`FetchNextRow()`

FetchNextRowsEx()

`short FetchNextRowsEx(long IDRequête, long* NombreLignes)`

Description

Cette fonction extrait la *NombreLignes* de données suivante de la base de données. Vous pouvez utiliser cette fonction dans des environnements qui ne supportent pas les tableaux VARIANT, comme Microsoft Access 2.0. Utilisez cette fonction avec `GetColumnValueEx()` pour extraire les données dans chaque colonne d'une ligne donnée.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d' <code>InitializeQuery()</code> .
NomLignes	Un pointeur vers un long contenant le nombre de ligne à extraire. Si le nombre de lignes non extraites dans la série de résultats est inférieur à <i>NombreLignes</i> , le <i>NombreLignes</i> est réinitialisé au nombre réel de lignes reprises dans les résultats.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Lorsque la fin de la série de résultats est atteinte, la valeur renvoyée est égale à -1. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

Sujets connexes

`FetchNextRowEx()`

FlushQMFCache()

`FlushQMFCache()` annulé

Description

Cette fonction demande à QMF pour Windows de vider sa mémoire cache des informations QMF, sans tenir compte de son contenu. La prochaine fois que

QMF pour Windows a besoin d'informations QMF, il les obtient à partir de la base de données. Normalement, QMF pour Windows met en mémoire cache les informations obtenues à partir de la base de données afin de réduire le trafic de la base de données et d'améliorer les performances. Vous appelez cette fonction avant de solliciter `GetQMFObjectInfo()`, `GetQMFQueryText()`, ou `GetQMFObjectList()` afin de vous assurer que l'information générée est actualisée.

Valeur renvoyée

Aucune.

GetColumnCount()

`long GetColumnCount(long IDRequête)`

Description

Cette fonction génère le nombre de colonnes dans la série de résultats.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d' <code>InitializeQuery()</code> .

Valeur renvoyée

Le nombre de colonnes dans chaque ligne en cas de succès. En cas d'échec, 0 ou -1. Si la valeur renvoyée est égale à 0 ou -1, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState` pour obtenir des informations complémentaires à propos de l'erreur.

GetColumnDataValue()

`short GetColumnDataValue(long IDRequête, long Index)`

Description

Cette fonction génère la valeur de données pour la colonne spécifiée dans *Index* pour la ligne actuelle de données. Après avoir appelé cette fonction, la propriété de *Value* peut être interrogée pour connaître la valeur générée. Vous utilisez cette fonction avec `FetchNextRowEx()` pour avoir accès aux données dans une seule ligne de données.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d' <code>InitializeQuery()</code> .
<i>Index</i>	L'index de base zéro de la ligne de données à extraire.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` pour obtenir des informations complémentaires à propos de l'erreur.

GetColumnHeader()

BSTR GetColumnHeader(long *IDRequête*, long *Index*, short* *Résultats*)

Description

Cette fonction génère l'en-tête de colonne (nom de colonne) associée à l'index *Index*.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d' <code>InitializeQuery()</code> .
<i>Index</i>	L'index de base zéro de la ligne de données à extraire.
<i>Résultat</i>	Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si le résultat est différent de zéro, vous pouvez solliciter <code>GetLastErrorString()</code> ou <code>GetLastErrorType()</code> , <code>GetLastSQLCode()</code> , <code>GetLastSQLError()</code> ou <code>GetLastSQLState()</code> pour obtenir des informations complémentaires à propos de l'erreur.

Remarque : Les en-têtes de colonne ne sont pas disponibles pour des instructions SQL statiques. Pour des ID de requête générées à partir d'`InitializeStaticQuery()`, `GetColumnHeader` génère une chaîne du formulaire "Coln", dans laquelle "n" est le numéro de colonne.

Valeur renvoyée

La chaîne générée représente le nom de colonne tel que spécifié dans le paramètre *Index*.

GetColumnHeaderEx()

short GetColumnHeaderEx(long *IDRequête*, long *Index*)

Description

Cette fonction génère l'en-tête de colonne (nom de colonne) associée à l'index *Index*. Après avoir appelé cette fonction, la propriété de *Value* peut être interrogée pour connaître la valeur générée.

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeQuery().
Index	L'index de base zéro de la ligne de données à extraire.

Remarque : Les en-têtes de colonne ne sont pas disponibles pour des instructions SQL statiques. Pour des ID de requête générées à partir d'InitializeStaticQuery(), GetColumnHeader génère une chaîne du formulaire "Coln", dans laquelle "n" est le numéro de colonne.

Valeur renvoyée

Zéro en cas de succès, valeur différente de zéro en cas d'échec. Si la valeur générée est zéro, rechercher la propriété de la *Value* pour la chaîne représentant le nom de colonne. Si la valeur renvoyée est différente de zéro, vous pouvez solliciter GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() pour obtenir des informations complémentaires à propos de l'erreur.

GetColumnHeadings()

short GetColumnHeadings(long *IDRequête*, VARIANT* *Headings*)

Description

Cette fonction génère les en-têtes de colonne (également appelés noms de colonne)

Paramètres

Nom	Description
<i>IDRequête</i>	L'ID de la requête telle que générée à partir d'InitializeQuery().
En-têtes	Un pointeur dirigé vers une VARIANTE dans laquelle le résultat est sauvegardé. Le résultat est un ensemble de chaînes (type de variante VT_ARRAY VT_BSTR) contenant une chaîne pour chaque en-tête de colonne. Vous devez correctement initialiser la VARIANTE avant d'appeler cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent solliciter VariantInit().

Remarque : A cause d'un bug en Microsoft Excel 7.0 et Microsoft Access 7.0 (et éventuellement d'autres produits Microsoft 32-bit utilisant Visual Basic dans leurs Applications), les chaînes de données en variables Variant provenant de QMF pour Windows peuvent ne pas être converties d'Unicode (utilisé par OLE) en ANSI (utilisé

par VBA). Lorsque cela se produit, seul le premier caractère de la chaîne s'affiche. Pour remédier à ce problème, veillez à ce que la variable soit équivalente à une chaîne vide avant de faire appel à la fonction QMF pour Windows qui utilise cette variable.

Remarque : Les en-têtes de colonne ne sont pas disponibles pour les instructions SQL statiques. Pour les IDs requête provenant d'InitializeStaticQuery(), GetColumnHeadings renvoie les chaînes "Col1", "Col2", etc.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), ou GetLastErrorSQLState() pour avoir plus de renseignements sur l'erreur.

GetColumnValue()

short GetColumnValue(long *IDRequête*, long *Index*, VARIANT* *Valeur*)

Description

Cette fonction renvoie la valeur des données pour la colonne spécifiée en *Index* pour la série de données courante. Vous utilisez cette fonction avec FetchNextRowEx() pour avoir accès aux données dans une seule chaîne de données.

Paramètres

Nom	Description
<i>ID Requête</i>	L'ID requête comme elle est renvoyée par InitializeQuery().
<i>Index</i>	L'index basé sur zéro de la chaîne de données à extraire.
<i>Valeur</i>	Pointeur vers VARIANT dans lequel vous voulez stocker les résultats. Le résultat est une valeur de données basée sur le type de variante. Vous devez initialiser correctement VARIANT avant de lancer cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent appeler VariantInit().

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), ou GetLastErrorSQLState() pour avoir plus de renseignements sur l'erreur.

GetColumnValueEx()

Short GetColumnValueEx(long *IDRequête*, long *RowIndex*, VARIANT* *Valeur*)

Description

Cette fonction renvoie la valeur des données pour la colonne spécifiée en *ColIndex* pour la série de données courante. Vous utilisez cette fonction avec *FetchNextRowsEx()* pour avoir accès aux données dans une seule chaîne de données.

Paramètres

Nom	Description
<i>ID Requête</i>	L'ID requête comme renvoyée par <i>InitializeQuery()</i> .
<i>RowIndex</i>	L'index basé sur zéro de la chaîne de données à extraire.
<i>ColIndex</i>	L'index basé sur zéro de la chaîne de données à extraire.
<i>Valeur</i>	Pointeur sur VARIANT dans lequel vous voulez stocker les résultats. Vous pouvez interroger la variante résultante pour retrouver le type de données pour des utilisations ultérieures. Vous devez initialiser correctement VARIANT avant de lancer cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent appeler <i>VariantInit()</i> .

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à *GetLastErrorString()* ou *GetLastErrorType()*, *GetLastSQLCode()*, *GetLastSQLError()*, ou *GetLastSQLState()* pour avoir plus de renseignements sur l'erreur.

GetDefaultServerName()

BSTR GetDefaultServerName()

Description

Cette fonction renvoie une chaîne comprenant le nom du serveur par défaut.

Valeur renvoyée

Une chaîne qui spécifie le nom du serveur par défaut.

GetGlobalVariable()

BSTR GetGlobalVariable(BSTR *Nom*)

Description

Cette fonction sert à extraire la valeur de la variable globale spécifiée.

Paramètres

Nom	Description
Nom	Une chaîne contenant le nom de la variable que vous voulez déterminer.

Valeur renvoyée

Une chaîne contenant la valeur de la variable globale, ou NULL si la variable n'a pas de valeur ou si une erreur a eu lieu.

GetHostVariableNames()

short GetHostVariableNames(long *IDRequête*, VARIANT* *Noms*)

Description

Cette fonction fournit une liste de noms de toutes les variables hôtes dont référence est faite dans la requête en question. La requête doit être une requête statique faisant référence aux variables hôtes (soit celles stockées avec la requête QMF, soit celles créées par `AddHostVariable()`).

Paramètres

Nom	Description
<i>ID Requête</i>	L'ID requête comme renvoyée par <code>InitializeQuery()</code> .
Noms	Pointeur sur VARIANT dans lequel vous voulez stocker les résultats.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` pour avoir plus de renseignements sur l'erreur.

GetHostVariableTypes()

short GetHostVariableTypes(long *IDRequête*, VARIANT* *Noms*)

Description

Cette fonction fournit une liste de types de données de toutes les variables hôtes dont référence est faite dans la requête en question. La requête doit être une requête statique faisant référence aux variables hôtes (soit celles stockées avec la requête QMF, soit celles créées par `AddHostVariable()`). Voir `AddHostVariable()` pour obtenir une liste des types de données qui peuvent être fournis.

Paramètres

Nom	Description
<i>ID Requête</i>	L'ID requête comme renvoyée par InitializeQuery().
Types	Pointeur sur VARIANT dans lequel vous voulez stocker les résultats.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() pour avoir plus de renseignements sur l'erreur.

GetLastErrorString()

BSTR GetLastErrorString()

Description

Cette fonction fournit une chaîne contenant des informations sur l'erreur la plus récente. Si vous faites appel à cette fonction après avoir utilisé une fonction avec succès (sans erreur), cette fonction vous fournira des informations sur la dernière erreur rencontrée lors d'un précédent appel à une fonction. Pour éviter toute confusion, faites toujours immédiatement appel à cette fonction après avoir fait appel à une fonction ayant donné un message d'erreur.

Valeur renvoyée

Une chaîne contenant des renseignements sur une erreur. Si aucune erreur n'a eu lieu depuis la création de l'objet QMF API, le message NULL apparaît.

Thèmes liés

GetLastErrorType()
GetLastSQLCode()
GetLastSQLError()
GetLastSQLState()

GetLastErrorType()

short GetLastErrorType()

Description

Cette fonction fournit le type de l'erreur la plus récente. Si vous faites appel à cette fonction après avoir utilisé une fonction avec succès (sans erreur), cette fonction vous fournira des informations sur la dernière erreur rencontrée lors d'un précédent appel à une fonction. Pour éviter toute confusion, faites toujours immédiatement appel à cette fonction après avoir fait appel à une fonction ayant donné un message d'erreur.

Valeur renvoyée

La valeur renvoyée indique le type d'erreur :

Valeur	Signification
0 (RS_ERROR_NONE)	Il n'y a pas eu d'erreur depuis la création de l'objet API QMF pour Windows.
1 (RS_ERROR_SQL)	Une erreur SQL a eu lieu. Si une erreur est survenue durant l'appel d'une fonction ayant pour objet <i>ID Requête</i> , choisissez <i>Close()</i> pour fermer cette requête. Vous ne revenez pas en arrière. Vous pouvez continuer à utiliser l'objet API QMF pour Windows, même s'il est possible que vous rencontriez des erreurs supplémentaires.
2 (RS_ERROR_USER_CANCEL)	Un utilisateur a annulé l'opération, certainement en cliquant sur Annuler dans la fenêtre ouverte. QMF pour Windows revient implicitement en arrière (interrompant toutes les IDs requête en cours) et cela provoque l'interruption de la connexion à la base de données. Vous devez appeler <i>InitializeServer()</i> ou <i>ReinitializeServer()</i> si vous voulez continuer.
3 (RS_ERROR_FATAL_GOV)	Une erreur gouverneur fatale a eu lieu. Une possibilité est qu'API QMF pour Windows s'est arrêté car le temps d'arrêt maximum autorisé est dépassé. QMF pour Windows revient implicitement en arrière (interrompant toutes les IDs requête en cours) et cela provoque l'interruption de la connexion à la base de données. Vous devez appeler <i>InitializeServer()</i> ou <i>ReinitializeServer()</i> si vous voulez continuer.
4 (RS_ERROR_NONFATAL_GOV)	Une erreur gouverneur non-fatale a eu lieu. Soit le nombre de lignes maximum autorisé a été dépassé, soit l'instruction SQL n'est pas acceptée. Si une erreur est survenue durant l'appel d'une fonction ayant pour objet <i>ID Requête</i> , choisissez <i>Close()</i> pour fermer cette requête. Vous ne revenez pas en arrière et la connexion à la base de données n'est en rien affectée de sorte que vous pouvez poursuivre l'utilisation de l'objet API QMF pour Windows.
5 (RS_ERROR_OTHER)	Une erreur générale a eu lieu. Vous ne revenez pas en arrière. Vous pouvez continuer à utiliser l'objet API QMF pour Windows, même s'il est possible que vous rencontriez des erreurs supplémentaires.

Thèmes liés

GetLastErrorString()
GetLastSQLCode()
GetLastSQLError()
GetLastSQLState()

GetLastSQLCode()

long GetLastSQLCode()

Description

Cette fonction fournit le code SQL de l'erreur la plus récente. Si vous faites appel à cette fonction après avoir utilisé une fonction avec succès (sans erreur), cette fonction vous fournira des informations sur la dernière erreur rencontrée lors d'un précédent appel à une fonction. Pour éviter toute confusion, faites toujours immédiatement appel à cette fonction après avoir fait appel à une fonction ayant donné un message d'erreur.

Valeur renvoyée

Cette fonction fournit le code SQL de l'erreur la plus récente. Si aucune erreur n'a eu lieu depuis la création de l'objet API QMF pour Windows, ou que l'erreur la plus récente n'était pas une erreur SQL, zéro s'affiche.

Thèmes liés

GetLastErrorString()
GetLastErrorType()
GetLastSQLError()
GetLastSQLState()

GetLastSQLError()

VARIANT GetLastSQLError()

Description

Cette fonction fournit des informations détaillées sur l'erreur SQL la plus récente. Si vous faites appel à cette fonction après avoir utilisé une fonction avec succès (sans erreur), cette fonction vous fournira des informations sur la dernière erreur rencontrée lors d'un précédent appel à une fonction. Pour éviter toute confusion, faites toujours immédiatement appel à cette fonction après avoir fait appel à une fonction ayant donné un message d'erreur.

Valeur renvoyée

Une liste (type variant VT_ARRAY | VT_VARIANT) contenant des informations concernant l'erreur. Si aucune erreur n'a eu lieu depuis la création de l'objet API QMF pour Windows, ou que l'erreur la plus récente n'était pas une erreur SQL, vide (type variant VT_EMPTY) s'affiche. La liste a le format suivant :

Elément	Type	Contenus
0	long (VT_I4)	Code
1	chaîne (VT_BSTR)	Etat
2	chaîne (VT_BSTR)	ErrProc
3	chaîne (VT_BSTR)	NomRDB
4	long (VT_I4)	ErrD1
5	long (VT_I4)	ErrD2
6	long (VT_I4)	ErrD3
7	long (VT_I4)	ErrD4
8	long (VT_I4)	ErrD5
9	long (VT_I4)	ErrD6
10	chaîne (VT_BSTR)	Warn0
11	chaîne (VT_BSTR)	Warn1
12	chaîne (VT_BSTR)	Warn2
13	chaîne (VT_BSTR)	Warn3
14	chaîne (VT_BSTR)	Warn4
15	chaîne (VT_BSTR)	Warn5
16	chaîne (VT_BSTR)	Warn6
17	chaîne (VT_BSTR)	Warn7
18	chaîne (VT_BSTR)	Warn8
19	chaîne (VT_BSTR)	Warn9
20	chaîne (VT_BSTR)	WarnA
21	chaîne (VT_BSTR)	MessageTokens

Thèmes liés

GetLastErrorString()
GetLastErrorType()
GetLastSQLCode()
GetLastSQLState()

GetLastSQLState()

BSTR GetLastSQLState()

Description

Cette fonction fournit l'état SQL de l'erreur la plus récente. Si vous faites appel à cette fonction après avoir utilisé une fonction avec succès (sans erreur), cette fonction vous fournira des informations sur la dernière erreur rencontrée lors d'un précédent appel à une fonction. Pour éviter toute confusion, faites toujours immédiatement appel à cette fonction après avoir fait appel à une fonction ayant donné un message d'erreur.

Valeur renvoyée

Une chaîne comprenant le code SQL de l'erreur la plus récente. Si aucune erreur n'a eu lieu depuis la création de l'objet QMF API, le message NULL apparaît.

Thèmes liés

GetLastErrorString()
GetLastErrorType()
GetLastSQLCode()
GetLastSQLError()

GetOption()

short GetOption(short *Option*, VARIANT* *Valeur*)

Description

Va à la valeur option spécifiée dans QMF pour Windows.

Paramètres

Nom	Description
<i>Option</i>	Spécifie quelle option rechercher.

Valeur	Signification
0 (RSO_SERVER_DEFINITION_FILE)	Nom du fichier de définition du serveur.
1 (RSO_CPIC_DLL)	Nom du fichier DLL du fournisseur CPI-C.
2 (RSO_CPIC_TIMEOUT_WARNING)	CPI-C avertissement de dépassement (en secondes). Cette limite n'est pas utilisée par API QMF pour Windows.
3 (RSO_CPIC_TIMEOUT_CANCEL)	CPI-C annulation du dépassement (en secondes).
4 (RSO_TCP_TIMEOUT_WARNING)	TCP avertissement de dépassement (en secondes). Cette limite n'est pas utilisée par API QMF pour Windows.
5 (RSO_TCP_TIMEOUT_CANCEL)	TCP annulation du dépassement (en secondes).
6 (RSO_DISPLAY_NULLS_STRING)	La chaîne à utiliser pour afficher des valeurs nulles.

7 (RSO_ENTER_NULLS_STRING)	La chaîne à utiliser pour afficher des valeurs nulles.
8 (RSO_ENTER_DEFAULTS_STRING)	La chaîne à utiliser pour afficher des valeurs par défaut.
9 (RSO_TRACE_FILE_1)	Retrouver nom du fichier 1.
10 (RSO_TRACE_FILE_2)	Retrouver nom du fichier 2.
11 (RSO_TCP_TRACE_LEVEL)	Niveau de recherche TCP.
12 (RSO_CPIC_TRACE_LEVEL)	Niveau de recherche CPI-C.
13 (RSO_DDM_TRACE_LEVEL)	Niveau de recherche DDM.
Valeur	Pointeur sur VARIANT dans lequel le résultat est stocké. Le résultat est une série (variant type VT_ARRAY VT_VARIANT) contenant une valeur pour chaque colonne dans le résultat. Faites appel à GetColumnCount() pour déterminer le nombre de valeurs dans la série. Vous devez initialiser correctement VARIANT avant de lancer cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent appeler VariantInit().

Remarque : A cause d'un bug en Microsoft Excel 7.0 et Microsoft Access 7.0 (et éventuellement d'autres produits Microsoft 32-bit utilisant Visual Basic dans leurs Applications), les chaînes de données en variables Variant provenant de QMF pour Windows peuvent ne pas être converties d'Unicode (utilisé par OLE) en ANSI (utilisé par VBA). Lorsque cela se produit, seul le premier caractère de la chaîne s'affiche. Pour remédier à ce problème, veillez à ce que la variable soit équivalente à une chaîne vide avant de faire appel à la fonction QMF pour Windows qui utilise cette variable.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType() pour avoir plus de renseignements sur l'erreur.

Thèmes liés

SetOption()

GetOptionEx()

short GetOptionEx(short *Option*)

Description

Va à la valeur option spécifiée dans QMF pour Windows. Lorsque la valeur de l'option est affichée, vous devez interroger la propriété *Option* pour les données.

Paramètres

Nom	Description
Option	Les valeurs option sont les mêmes que celles de l'appel <code>GetOption()</code> .

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()` pour avoir plus de renseignements sur l'erreur.

Thèmes liés

`GetOption()`
`SetOption()`

GetProcText()

BSTR `GetProcText(long ProcID)`

Description

Cette fonction affiche le texte exécuté pour la procédure spécifiée, après substitution de variable. Vous devez utiliser `SetProcVariable()` pour déterminer la valeur de n'importe quelle variable utilisée dans la procédure avant de faire appel à cette fonction.

Paramètres

Nom	Description
ProcID	L'ID de la procédure comme renvoyée par <code>InitializeProc()</code> .

Valeur renvoyée

Si le résultat est positif, une chaîne contenant le texte de la procédure est renvoyé. Si le résultat est négatif, NULL est renvoyé. Si la valeur renvoyée est NULL, vous pouvez appeler `GetLastErrorString()` ou `GetLastErrorType()` pour obtenir de plus amples renseignements sur l'erreur.

GetProcVariables()

short `GetProcVariables(long ProcID, VARIANT* Variables)`

Description

Va à la valeur option spécifiée dans QMF pour Windows.

Paramètres

Nom	Description
ProcID	L'ID de la procédure comme renvoyée par InitializeProc().
Valeur	Pointeur sur VARIANT dans lequel le résultat est stocké. Le résultat est une série de chaînes (variant type VT_ARRAY VT_BSTR) contenant chacune le nom d'une variable. S'il n'y a pas de variable dans la procédure, le résultat est vide (variant type VT_EMPTY). Vous devez initialiser correctement VARIANT avant de lancer cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent appeler VariantInit().

Remarque : A cause d'un bug en Microsoft Excel 7.0 et Microsoft Access 7.0 (et éventuellement d'autres produits Microsoft 32-bit utilisant Visual Basic dans leurs Applications), les chaînes de données en variables Variant provenant de QMF pour Windows peuvent ne pas être converties d'Unicode (utilisé par OLE) en ANSI (utilisé par VBA). Lorsque cela se produit, seul le premier caractère de la chaîne s'affiche. Pour remédier à ce problème, veillez à ce que la variable soit équivalente à une chaîne vide avant de faire appel à la fonction QMF pour Windows qui utilise cette variable.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. S'il n'y a pas de variables dans la procédure, la valeur renvoyée est RS_NO_ERROR_NO_DATA (-1). Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType() pour avoir plus de renseignements sur l'erreur.

GetQMFObjectInfo()

short GetQMFObjectInfo(BSTR *DétenteurEtNom*, short *Type*, short *Heure*, VARIANT* *Valeur*)

Description

Cette fonction renvoie des informations concernant un objet QMF (soit un format, soit une requête). L'information renvoyée est spécifiée par les paramètres *Type* et *Heure*.

Paramètres

Nom	Description
DétenteurEtNom	Une chaîne contenant le détenteur et le nom, séparés par un point, de l'objet pour lequel vous voulez obtenir des informations. Par exemple, John.Requête2

Valeur	Pointeur sur VARIANT dans lequel le résultat est stocké. Pour RSI_TIMEUSED, RSI_TIMESRUN, RSI_TIMESCANCELLED, et RSI_LEVEL, le résultat est short (type variant VT_I2). Pour RSI_RESTRICTED le résultat est un booléen (variant type VT_BOOL). Pour tous les autres, le résultat est une chaîne (variant type VT_BSTR). Vous devez initialiser correctement VARIANT avant de lancer cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent appeler VariantInit().
--------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Remarque : A cause d'un bug en Microsoft Excel 7.0 et Microsoft Access 7.0 (et éventuellement d'autres produits Microsoft 32-bit utilisant Visual Basic dans leurs Applications), les chaînes de données en variables Variant provenant de QMF pour Windows peuvent ne pas être converties d'Unicode (utilisé par OLE) en ANSI (utilisé par VBA). Lorsque cela se produit, seul le premier caractère de la chaîne s'affiche. Pour remédier à ce problème, veillez à ce que la variable soit équivalente à une chaîne vide avant de faire appel à la fonction QMF pour Windows qui utilise cette variable.

<i>Type</i>	Spécifiez le type d'information à obtenir.
-------------	--------------------------------------------

Valeur	Signification
0 (RSI_COMMENT)	Commentaire
1 (RSI_LEVEL)	Niveau
2 (RSI_TYPE)	Type
3 (RSI_SUBTYPE)	Sous-type
4 (RSI_RESTRICTED)	Réduit
5 (RSI_MODEL)	Modèle
6 (RSI_TIMESUSED)	Temps utilisé.
7 (RSI_TIMESRUN)	Nombre d'exécutions.
8 (RSI_TIMESCANCELLED)	Nombre d'annulations.
9 (RSI_DATE)	Date de première utilisation, de dernière utilisation et de dernière modification.
10 (RSI_TIME)	Heure de première utilisation, de dernière utilisation et de dernière modification.
11 (RSI_USERID)	ID utilisateur : première utilisation, dernière utilisation, dernière modification.
12 (RSI_SQLID)	ID SQL : première utilisation, dernière utilisation, dernière modification.

13 (RSI_ENVIRONMENT)	Environnement : première utilisation, dernière utilisation, dernière modification.
14 (RSI_MODE)	Mode : première utilisation, dernière utilisation, dernière modification.
15 (RSI_COMMAND)	Commande : première utilisation, dernière utilisation, dernière modification.
<i>Heure</i>	Spécifie : première utilisation, dernière utilisation, dernière modification.
Valeur	Signification
0 (RST_FIRSTUSED)	Première utilisation.
1 (RST_LASTUSED)	Dernière utilisation.
2 (RST_LASTMODIFIED)	Dernière modification.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, ou `GetLastSQLState()` pour avoir plus de renseignements sur l'erreur.

GetQMFObjectInfoEx()

short GetQMFObjectInfoEx(BSTR *DétenteurEtNom*, short *Type*, short *Heure*)

Description

Cette fonction renvoie des informations concernant un objet QMF. L'information renvoyée est spécifiée par les paramètres *Type* et *Heure*. Après avoir fait appel à cette fonction, la propriété de *QMFObjectInfo* peut être interrogée pour connaître la valeur renvoyée.

Paramètres

Nom	Description
DétenteurEtNom	Une chaîne contenant le détenteur et le nom, séparés par un point, de l'objet pour lequel vous voulez obtenir des informations. Par exemple, John.Requête2
Type	Spécifiez le type d'information à obtenir.

Valeur	Signification
0 (RSI_COMMENT)	Commentaire
1 (RSI_LEVEL)	Niveau
2 (RSI_TYPE)	Type
3 (RSI_SUBTYPE)	Sous-type
4 (RSI_RESTRICTED)	Réduit
5 (RSI_MODEL)	Modèle
6 (RSI_TIMESUSED)	Temps utilisé.
7 (RSI_TIMESRUN)	Nombre d'exécutions.
8 (RSI_TIMESCANCELLED)	Nombre d'annulations.
9 (RSI_DATE)	Date de première utilisation, de dernière utilisation et de dernière modification.
10 (RSI_TIME)	Heure de première utilisation, de dernière utilisation et de dernière modification.
11 (RSI_USERID)	ID utilisateur : première utilisation, dernière utilisation, dernière modification.
12 (RSI_SQLID)	ID SQL : première utilisation, dernière utilisation, dernière modification.
13 (RSI_ENVIRONMENT)	Environnement : première utilisation, dernière utilisation, dernière modification.
14 (RSI_MODE)	Mode : première utilisation, dernière utilisation, dernière modification.
15 (RSI_COMMAND)	Commande : première utilisation, dernière utilisation, dernière modification.

<i>Heure</i>	Spécifie: première utilisation, dernière utilisation, dernière modification.
--------------	------------------------------------------------------------------------------

Valeur	Signification
0 (RST_FIRSTUSED)	Première utilisation.
1 (RST_LASTUSED)	Dernière utilisation.
2 (RST_LASTMODIFIED)	Dernière modification.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, ou `GetLastSQLState()` pour avoir plus de renseignements sur l'erreur.

GetQMFOBJECTLIST()

short GetQMFOBJECTLIST(BSTR *Détenteur*, BSTR *Nom*, short *Type*, VARIANT* *Liste*)

Description

Cette fonction renvoie une liste des noms de tous les objets QMF correspondant aux modèles spécifiés dans les paramètres *Détenteur* et *Nom*.

Paramètres

Nom	Description
Détenteur	Une chaîne contenant le détenteur des objets que vous voulez inclure à la liste de retour.
Nom	Une chaîne contenant le détenteur des objets que vous voulez inclure à la liste de retour.
Liste	Pointeur sur VARIANT dans lequel le résultat est stocké. Le résultat est une série de chaînes (variant type VT_ARRAY VT_BSTR) chacune du format <i>Détenteur.Nom</i> . Si aucune requête QMF pour Windows correspondante n'est trouvée, le résultat est vide (variant type VT_EMPTY). Vous devez initialiser correctement VARIANT avant de lancer cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent appeler <code>VariantInit()</code> .

Remarque : A cause d'un bug en Microsoft Excel 7.0 et Microsoft Access 7.0 (et éventuellement d'autres produits Microsoft 32-bit utilisant Visual Basic dans leurs Applications), les chaînes de données en variables Variant provenant de QMF pour Windows peuvent ne pas être converties d'Unicode (utilisé par OLE) en ANSI (utilisé par VBA). Lorsque cela se produit, seul le premier caractère de la chaîne s'affiche. Pour remédier à ce problème, veillez à ce que la variable soit équivalente à une chaîne vide avant de faire appel à la fonction QMF pour Windows qui utilise cette variable.

<i>Type</i>	Spécifie les types d'objets QMF que vous voulez inclure à la liste. Ces valeurs peuvent être ajoutées ensembles pour spécifier des types multiples d'objets.
-------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------

Valeur	Signification
2048 (RSQ_MASK_QUERIES)	Inclure requête QMF dans la liste.
1024 (RSQ_MASK_FORMS)	Inclure formats QMF dans la liste.
512 (RSQ_MASK_PROCS)	Inclure procédures QMF dans la liste.
256 (RSQ_MASK_TABLES)	Inclure tables dans la liste.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si aucun objet QMF correspondant n'est trouvé, la valeur renvoyée est zéro. Si la valeur renvoyée est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, ou `GetLastSQLState()` pour avoir plus de renseignements sur l'erreur.

GetQMFOBJECTListEx()

short `GetQMFOBJECTListEx(BSTR Détenteur, BSTR Nom, short Index)`

Description

Cette fonction renvoie le nom de l'objet QMF correspondant aux modèles spécifiés dans les paramètres *Détenteur* et *Nom* dont référence est faite dans le paramètre *Index*. Après avoir fait appel à cette fonction, la propriété de *Valeur* peut être interrogée pour connaître la valeur renvoyée.

Paramètres

Nom	Description
Détenteur	Une chaîne contenant le détenteur des objets que vous voulez inclure à la liste de retour.
Nom	Une chaîne contenant le nom des objets que vous voulez inclure à la liste de retour.
Index	L'index de la liste des objets QMF correspondant au modèle.
<i>Type</i>	Spécifie les types d'objets QMF que vous voulez inclure à la liste. Ces valeurs peuvent être ajoutées ensembles pour spécifier des types multiples d'objets.

Valeur	Signification
2048 (RSQ_MASK_QUERIES)	Inclure requêtes QMF dans la liste.
1024 (RSQ_MASK_FORMS)	Inclure formats QMF dans la liste.
512 (RSQ_MASK_PROCS)	Inclure procédures QMF dans la liste.
256 (RSQ_MASK_TABLES)	Inclure tables dans la liste.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si aucun objet QMF correspondant n'est trouvé, la valeur renvoyée est RS_ERROR_OUTOFRANGE. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), ou GetLastErrorSQLState() pour avoir plus de renseignements sur l'erreur.

GetQMFPProcText()

BSTR GetQMFPProcText(BSTR *DétenteurEtNom*)

Description

Cette fonction affiche le texte exécuté pour la procédure spécifiée, après substitution de variable. Vous devez utiliser SetProcVariable() pour déterminer la valeur de n'importe quelle variable utilisée dans la procédure avant de faire appel à cette fonction.

Paramètres

Nom	Description
DétenteurEtNom	Une chaîne contenant le détenteur et le nom, séparés par un point, de l'objet que vous voulez effacer. Par exemple, John.Proc2

Valeur renvoyée

Une chaîne contenant le texte pour la procédure extraite, ou NULL si la procédure n'a pu être extraite. Si la valeur renvoyée est NULL, vous pouvez appeler GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError, ou GetLastErrorSQLState() pour obtenir de plus amples renseignements concernant l'erreur.

GetQMFPQueryText()

BSTR GetQMFPQueryText(BSTR *DétenteurEtNom*)

Description

Cette fonction extrait le texte SQL stocké dans la requête spécifiée.

Paramètres

Nom	Description
DétenteurEtNom	Une chaîne contenant le détenteur et le nom, séparés par un point, de l'objet que vous voulez effacer. Par exemple, John.Requête2

Valeur renvoyée

Une chaîne contenant le texte pour la procédure extraite , ou NULL si la procédure n'a pu être extraite. Si la valeur renvoyée est NULL, vous pouvez appeler GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError, ou GetLastErrorSQLState() pour obtenir de plus amples renseignements concernant l'erreur.

GetQueryText()

BSTR GetQueryText(long *IDRequête*)

Description

Cette fonction affiche le texte SQL exécuté pour la procédure spécifiée, après substitution de variable. Vous devez utiliser SetVariable() pour déterminer la valeur de n'importe quelle variable utilisée dans la procédure avant de faire appel à cette fonction.

Paramètres

Nom	Description
<i>ID Requête</i>	L'ID requête comme renvoyée par InitializeQuery().

Remarque : Le texte de requête n'est pas disponible pour les instructions SQL statiques. Pour les IDs requête provenant d'InitializeStaticQuery(), GetQueryText() renvoie une chaîne vide.

Valeur renvoyée

Si le résultat est positif, une chaîne contenant le texte SQL est renvoyé. Si le résultat est négatif, NULL est renvoyé. Si la valeur renvoyée est NULL, vous pouvez appeler GetLastErrorString() ou GetLastErrorType() pour obtenir de plus amples renseignements sur l'erreur.

GetQueryVerb()

BSTR GetQueryVerb(long *IDRequête*)

Description

Cette fonction renvoie une chaîne contenant l'instruction SQL utilisée dans la requête.

Paramètres

Nom	Description
<i>ID Requête</i>	L'ID requête comme renvoyée par InitializeQuery().

Remarque : L'instruction de requête n'est pas disponible pour les instructions SQL statiques. Pour les IDs requête provenant d'InitializeStaticQuery(), GetQueryVerb() renvoie une chaîne vide.

Valeur renvoyée

Si le résultat est positif, une chaîne contenant l'instruction SQL est renvoyée. Si le résultat est négatif, NULL est renvoyé. Si la valeur renvoyée est NULL, vous pouvez appeler GetLastErrorString() ou GetLastErrorType() pour obtenir de plus amples renseignements sur l'erreur.

GetResourceLimit()

short GetResourceLimit(short Ressource, long* Valeur)

Description

Cette fonction reçoit la limite de ressource demandée. Vous devez appeler InitializeServer() avant de faire appel à cette fonction, étant donné que les limites de ressource sont gérées par serveur.

Paramètres

Nom	Description
Ressource	Les valeurs ressource comprennent :
Valeur	Signification
0 (RSR_IDLE_CONNECTION_TIMEOUT)	Ralentir dépassement de connexion (en secondes).
1 (RSR_IDLE_QUERY_TIMEOUT_CANCEL)	Ralentir dépassement de requête (en secondes).
2 (RSR_IDLE_QUERY_TIMEOUT_WARNING)	Ralentir dépassement de requête (en secondes). C'est la limite d'avertissement ; elle n'est pas d'application pour API QMF pour Windows.
3 (RSR_SERVER_RESPONSE_TIMEOUT_CANCEL)	Dépassement serveur (en secondes).
4 (RSR_SERVER_RESPONSE_TIMEOUT_WARNING)	Dépassement serveur (en secondes). C'est la limite d'avertissement ; elle n'est pas d'application pour API QMF pour Windows.
5 (RSR_MAX_ROWS_TO_FETCH_CANCEL)	Nombre maximum de lignes à rechercher.

6 (RSR_MAX_ROWS_TO_FETCH_WARNING)	Nombre maximum de lignes à rechercher. C'est la limite d'avertissement ; elle n'est pas d'application pour API QMF pour Windows.
7 (RSR_MAX_BYTES_TO_FETCH_CANCEL)	Nombre maximum d'octets à rechercher.
8 (RSR_MAX_BYTES_TO_FETCH_WARNING)	Nombre maximum d'octets à rechercher. C'est la limite d'avertissement ; elle n'est pas d'application pour API QMF pour Windows.
9 (RSR_MAX_CONNECTIONS)	Nombre maximum de connexions autorisées vers le serveur de base de données.
10 (RSR_ALLOW_SERVER_ACCESS_UI)	L'accès vers le serveur de base de données est-il autorisé à partir de l'interface QMF pour Windows ?
11 (RSR_ALLOW_SERVER_ACCESS_API)	L'accès vers le serveur de base de données est-il autorisé à partir d'API QMF pour Windows ?
12 (RSR_FETCH_ALL_ROWS)	Recherche automatiquement toutes les lignes ?
13 (RSR_CONFIRM_UPDATES)	Confirmer les mises à jour du serveur de base de données ? Cette option n'a pas d'incidence sur API QMF pour Windows; les mises à jour de base de données ne sont pas confirmées pour API QMF pour Windows.
14 (RSR_SUMMARY_TRACKING)	Le suivi d'objet récapitulatif est-il activé ?
15 (RSR_DETAILED_TRACKING)	Le suivi détaillé d'objet est-il possible ?
16 (RSR_SQL_TRACKING)	Le suivi de texte SQL est-il possible ?

17 (RSR_ADHOC_TRACKING)	Le suivi ad hoc d'objet est-il possible ?
18 (RSR_ALLOW_ACQUIRE)	L'instruction SQL ACQUIRE est-elle autorisée ?
19 (RSR_ALLOW_ALTER)	L'instruction SQL ALTER est-elle autorisée ?
20 (RSR_ALLOW_COMMENT)	L'instruction SQL COMMENT est-elle autorisée ?
21 (RSR_ALLOW_CREATE)	L'instruction SQL CREATE est-elle autorisée ?
22 (RSR_ALLOW_DELETE)	L'instruction SQL DELETE est-elle autorisée ?
23 (RSR_ALLOW_DROP)	L'instruction SQL DROP est-elle autorisée ?
24 (RSR_ALLOW_EXPLAIN)	L'instruction SQL EXPLAIN est-elle autorisée ?
25 (RSR_ALLOW_GRANT)	L'instruction SQL GRANT est-elle autorisée ?
26 (RSR_ALLOW_INSERT)	L'instruction SQL INSERT est-elle autorisée ?
27 (RSR_ALLOW_LABEL)	L'instruction SQL LABEL est-elle autorisée ?
28 (RSR_ALLOW_LOCK)	L'instruction SQL LOCK est-elle autorisée ?
29 (RSR_ALLOW_REVOKE)	L'instruction SQL REVOKE est-elle autorisée ?
30 (RSR_ALLOW_SELECT)	L'instruction SQL SELECT est-elle autorisée ?
31 (RSR_ALLOW_SET)	L'instruction SQL SET est-elle autorisée ?
32 (RSR_ALLOW_SIGNAL)	L'instruction SQL SIGNAL est-elle autorisée ?
33 (RSR_ALLOW_UPDATE)	L'instruction SQL UPDATE est-elle autorisée ?
34 (RSR_ALLOW_CALL)	L'instruction SQL CALL est-elle autorisée ?

35 (RSR_ALLOW_SAVE_DATA)	La commande Sauvegarde des données est-elle autorisée ?
36 (RSR_SAVE_DATA_TABLE_SPACE_NAME)	Le nom de collecte par défaut des liens des modules.
37 (RSR_SAVE_DATA_TABLE_SPACE_NAME_OVERRIDE)	L'utilisateur peut-il ignorer le nom de l'espace table par défaut pour la commande Sauvegarde des données ?
38 (RSR_ALLOW_BIND_PACKAGE)	Autoriser l'édition des liens des modules ?
39 (RSR_DEF_COLLECTION)	Le nom de collecte par défaut des liens des modules.
40 (RSR_DEF_COLLECTION_OVERRIDE)	L'utilisateur peut-il ignorer le nom de collecte par défaut des liens des modules ?
41 (RSR_DEF_ISOLATION_LEVEL)	Le niveau d'isolement par défaut des liens des modules.
42 (RSR_DEF_ISOLATION_LEVEL_OVERRIDE)	L'utilisateur peut-il ignorer le niveau d'isolement par défaut des liens des modules ?
43 (RSR_ALLOW_TABLE_EDIT)	Autoriser l'utilisation de l'éditeur de table ?
44 (RSR_ALLOW_EXPORT)	Autoriser l'exportation de données ?
45 (RSR_ALLOW_SAVED_QUERIES_ONLY)	L'utilisateur n'est-il autorisé qu'à exécuter des requête enregistrées ?
46 (RSR_ALLOW_DROP_PACKAGE)	Autoriser l'interruption de modules ?
47 (RSR_QUERY_ISOLATION_LEVEL)	Niveau d'isolement à utiliser lors de l'exécution de requêtes.

48 (RSR_ACCOUNT_STRING)	La chaîne spécifiant les informations de compte à fournir lors de la connexion à la base de données.
49 (RSR_ACCOUNT_OVERRIDE)	L'utilisateur peut-il ignorer la chaîne spécifiant les informations de compte à fournir lors de la connexion à la base de données ?
Valeur	Pointeur sur un entier long dans lequel est stocké le résultat. Le résultat est la valeur de la limite ressource demandée. Pour les valeurs booléennes, le résultat est différent de zéro pour vrai, et zéro pour faux. Pour RSR_SAVE_DATA_TABLE_SPACE_NAME, RSR_DEF_COLLECTION, et RSR_ACCOUNT_STRING, -1 est renvoyé et la propriété <i>ResourceLimit</i> peut être consultée pour connaître la valeur de la chaîne renvoyée.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorMessage()` ou `GetLastErrorType()` pour avoir plus de renseignements sur l'erreur.

GetResourceLimitEx()

short GetResourceLimitEx(short *Ressource*)

Description

Cette fonction reçoit la limite de ressource demandée. Vous devez appeler `InitializeServer()` avant de faire appel à cette fonction, étant donné que les limites de ressource sont gérées par serveur. Après avoir fait appel à cette fonction, consultez la propriété *ResourceLimit* pour connaître le résultat.

Paramètres

Nom	Description
Ressource	Les valeurs ressource sont les mêmes que celles de l'appel <code>GetResourceLimit()</code> .

Remarque : L'instruction de requête n'est pas disponible pour les instructions SQL statiques. Pour les IDs requête provenant d'`InitializeStaticQuery()`, `GetQueryVerb()` renvoie une chaîne vide.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()` pour avoir plus de renseignements sur l'erreur.

GetRowCount()

`long GetRowCount(long IDRequête)`

Description

Cette fonction renvoie le nombre de lignes courantes dans la mémoire tampon interne de QMF pour Windows'. Il se peut qu'il soit plus important que le nombre de lignes extraites avec `FetchNextRow()` ou `FetchNextRows()`, étant donné que QMF pour Windows enregistre les données provenant de la base de données.

Cette fonction renvoie le nombre de lignes déjà extraites de la base de données. Si vous voulez extraire la totalité des lignes comprises dans le résultat, vous pouvez :

- Appeler `CompleteQuery()` et rechercher toutes les lignes grâce à `FetchNextRow()` ou `FetchNextRows()`.
- Spécifier `FetchAllRows = TRUE` lorsque vous choisissez `Open()`.

Paramètres

Nom	Description
<i>ID Requête</i>	L'ID requête comme renvoyée par <code>InitializeQuery()</code> .

Valeur renvoyée

Le nombre de lignes si l'action est positive (0 si aucune ligne n'a été extraite), ou -1 si l'action est négative. Si 1, vous pouvez appeler `GetLastErrorString()` ou `GetLastErrorType()` pour obtenir de plus amples informations sur l'erreur.

GetServerList()

`short GetServerList(VARIANT* Liste)`

Description

Cette fonction renvoie une chaîne contenant le nom des serveurs de base de données définis dans QMF pour Windows' Fichier de Définition Serveur (SDF). Vous devez définir un serveur de base de données dans le fichier SDF si vous voulez y accéder en utilisant API QMF pour Windows.

Paramètres

Nom	Description
Liste	Pointeur sur VARIANT dans lequel le résultat est stocké. Le résultat est une série de chaînes (variant type VT_ARRAY VT_BSTR) contenant chacune le nom d'un serveur de base de données. Si vous n'avez pas défini de serveur de base de données, le résultat est vide (variant type VT_EMPTY). Vous devez initialiser correctement VARIANT avant de lancer cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent appeler VariantInit().

Remarque : A cause d'un bug en Microsoft Excel 7.0 et Microsoft Access 7.0 (et éventuellement d'autres produits Microsoft 32-bit utilisant Visual Basic dans leurs Applications), les chaînes de données en variables Variant provenant de QMF pour Windows peuvent ne pas être converties d'Unicode (utilisé par OLE) en ANSI (utilisé par VBA). Lorsque cela se produit, seul le premier caractère de la chaîne s'affiche. Pour remédier à ce problème, veillez à ce que la variable soit équivalente à une chaîne vide avant de faire appel à la fonction QMF pour Windows qui utilise cette variable.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si vous n'avez pas défini de serveur de base de données, la valeur renvoyée est zéro. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType() pour avoir plus de renseignements sur l'erreur.

GetServerListEx()

short GetServerListEx(short *Index*)

Description

Cette fonction recherche le nom du serveur dont référence est faite dans le paramètre *Index*. Après avoir fait appel à cette fonction, la propriété de *Valeur* peut être interrogée pour connaître la valeur renvoyée.

Paramètres

Nom	Description
Index	Un index dans la liste des serveurs.

Valeur renvoyée

Zéro si c'est positif, RS_OUTOFRANGE lorsque l'index est plus important que le nombre de serveurs disponibles, valeur autre que zéro si c'est négatif. Si vous n'avez pas défini de serveur de base de données, la valeur renvoyée est

RS_OUTOFRANGE. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()` pour avoir plus de renseignements sur l'erreur.

GetStoredProcedureResultSets()

short GetStoredProcedureResultSets(long *IDRequête*, VARIANT* *Résultats*)

Description

Cette fonction recherche les IDs requête pour les résultats renvoyés par la procédure stockée avec l'*IDRequête* originale. Chacune des IDs requête renvoyées peut être utilisée avec `FetchNextRow()` ou `FetchNextRows()` pour extraire les lignes du résultat, et avec `Close()` lorsque chaque résultat est terminé.

Paramètres

Nom	Description
ID Requête	L'ID de la requête originale comme renvoyée par <code>InitializeQuery()</code> .
ResultSets	Pointeur sur VARIANT dans lequel les résultats sont stockés. Le résultat est une série d'entiers longs (variant type VT_ARRAY VT_I4), chacun des entiers étant l'ID requête pour les résultats correspondants. Si la procédure stockée ne renvoie aucun résultat, le résultat est vide (variant type VT_EMPTY). Vous devez initialiser correctement VARIANT avant de lancer cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent appeler <code>VariantInit()</code> .

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, ou `GetLastSQLState()` pour avoir plus de renseignements sur l'erreur.

GetVariables()

short GetVariables(long *IDRequête*, VARIANT* *Variables*)

Description

Cette fonction fournit une liste des noms des variables dans le texte SQL de la requête. Vous devez conférer des valeurs à ces variables en appelant `SetVariable()` avant d'exécuter la requête en utilisant soit `Open()` soit `Execute()`.

Paramètres

Nom	Description
IDRequête	L'ID requête comme elle est renvoyée par InitializeQuery().
Variables	Pointeur sur VARIANT dans lequel le résultat est stocké. Le résultat est une série de chaînes (variant type VT_ARRAY VT_BSTR) contenant chacune le nom d'une variable. S'il n'y a pas de variable dans l'instruction SQL, le résultat est vide (variant type VT_EMPTY). Vous devez initialiser correctement VARIANT avant de lancer cette fonction. Visual Basic le fait automatiquement. Les programmeurs Visual C++ doivent appeler VariantInit().

Remarque : A cause d'un bug en Microsoft Excel 7.0 et Microsoft Access 7.0 (et éventuellement d'autres produits Microsoft 32-bit utilisant Visual Basic dans leurs Applications), les chaînes de données en variables Variant provenant de QMF pour Windows peuvent ne pas être converties d'Unicode (utilisé par OLE) en ANSI (utilisé par VBA). Lorsque cela se produit, seul le premier caractère de la chaîne s'affiche. Pour remédier à ce problème, veillez à ce que la variable soit équivalente à une chaîne vide avant de faire appel à la fonction QMF pour Windows qui utilise cette variable.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. S'il n'y a pas de variable dans l'instruction SQL, la valeur renvoyée est RS_ERROR_NO_DATA (-1). Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType() pour avoir plus de renseignements sur l'erreur.

GetVariablesEx()

short GetVariablesEx(long *IDRequête*, short *Index*)

Description

Cette fonction renvoie le nom de la variable dans le texte SQL de la requête dont référence est faite dans le paramètre *Index*. Après avoir fait appel à cette fonction, la propriété de *Valeur* peut être interrogée pour connaître la valeur renvoyée. Vous devez conférer des valeurs à cette variable (et à toutes les autres dans le texte SQL) en appelant SetVariable() avant d'exécuter la requête en utilisant soit Open() ou Execute().

Paramètres

Nom	Description
ID Requête	L'ID de la requête comme renvoyée par InitializeQuery().
Variables	Un index dans la liste interne des variables. Interroger la propriété <i>Valeur</i> pour la chaîne correspondant à l'index. S'il n'y a pas de variable dans l'instruction SQL, la valeur renvoyée est RS_ERROR_NO_DATA (-1).

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. S'il n'y a pas de variable dans l'instruction SQL, la valeur renvoyée est RS_ERROR_NO_DATA (-1). Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType() pour avoir plus de renseignements sur l'erreur.

InitializeProc()

long InitializeProc(short *SourceType*, BSTR *Source*)

Description

Cette fonction détermine le texte que vous voulez utiliser dans une procédure. Le texte peut être pris comme paramètre de cette fonction, lu à partir d'un fichier texte ou obtenu à partir d'une procédure existante.

Paramètres

Nom	Description
SourceType	Spécifie la source pour le texte de la procédure.

Valeur	Signification
0 (RSS_STRING)	Le texte est contenu dans le paramètre <i>Source</i> .
2 (RSS_FILE)	Le texte est contenu dans le fichier texte dont le nom est spécifié dans le paramètre <i>Source</i> .
3 (RSS_QMFPROC)	Le texte est contenu dans la procédure dont le détenteur et le nom sont spécifiés dans le paramètre <i>Source</i> .

<i>Source</i>	Une chaîne contenant le texte, le détenteur et le nom (Détendeur.Nom) de la procédure, ou le nom d'un fichier contenant le texte de la procédure.
---------------	---------------------------------------------------------------------------------------------------------------------------------------------------

Valeur renvoyée

Si c'est positif, l'ID de la procédure (ProcID). Si le résultat est négatif, -1. Vous devez utiliser cette valeur dans tous les appels d'interface qui requièrent le paramètre *ProcID*.

InitializeQuery()

long InitializeQuery(short *SourceType*, BSTR *Source*)

Description

Cette fonction détermine le texte que vous voulez utiliser dans une requête. Le texte SQL peut être pris comme paramètre de cette fonction, lu à partir d'un fichier texte ou obtenu à partir d'une requête existante. Appelez Close() lorsque vous avez terminé la requête.

Paramètres

Nom	Description
SourceType	Spécifie la source pour le texte de l'instruction SQL.

Valeur	Signification
0 (RSS_STRING)	Le texte SQL est contenu dans le paramètre <i>Source</i> .
1 (RSS_QMFQUERY)	Le texte SQL est contenu dans la requête dont le détenteur et le nom sont spécifiés dans le paramètre <i>Source</i> .
2 (RSS_FILE)	Le texte SQL est contenu dans le fichier texte dont le nom est spécifié dans le paramètre <i>Source</i> .

Valeur renvoyée

Si c'est positif, l'ID de la requête. Si le résultat est négatif, -1. Vous devez utiliser cette valeur dans tous les appels d'interface qui requièrent le paramètre *Query*.

InitializeServer()

short InitializeServer(BSTR *Nom Serveur*, BSTR *IDUtilisateur*, BSTR *MotdePasse*, BOOL *ForceDialog*, [VARIANT *Compte*], [VARIANT *SuppressDialog*])

Description

Cette fonction initialise une connexion au serveur de base de données. Vous devez appeler cette fonction avant de faire appel à n'importe quelle autre fonction en API QMF pour Windows. Vous pouvez appeler plusieurs fois cette fonction. Cependant, si vous appelez cette fonction et ne terminez pas en appelant Commit() ou Rollback() vous revenez implicitement en arrière.

Paramètres

Nom	Description
ServerName	Une chaîne contenant le nom du serveur de base de données que vous voulez utiliser. Ce nom doit correspondre à un des noms définis dans le fichier de définition de serveur de QMF pour Windows. Appelez GetServerList() pour obtenir une liste des serveurs valides.
UserID	Une chaîne contenant l'ID utilisateur que vous voulez utiliser. Si l'ID Utilisateur équivaut à NULL ou à une chaîne vide, QMF pour Windows tente d'utiliser l'ID utilisateur de la requête la plus récente, si disponible. Dans le cas contraire, QMF pour Windows affiche la boîte de dialogue Information Utilisateur pour obtenir un ID utilisateur et un mot de passe.
Mot de passe	Une chaîne contenant le mot de passe pour l'ID utilisateur spécifié. Si un mot de passe équivaut à NULL ou à une chaîne vide, QMF pour Windows tente d'utiliser un mot de passe mémorisé, si disponible (cela nécessite Windows pour Workgroups). Dans le cas contraire, QMF pour Windows affiche la boîte de dialogue Information Utilisateur pour obtenir un ID utilisateur et un mot de passe.
ForceDialog	Une valeur autre que zéro indique que QMF pour Windows affiche la boîte de dialogue Information Utilisateur sans vérifier qu'un ID utilisateur ou qu'un mot de passe est spécifié. Cela donne à l'utilisateur l'opportunité de changer l'information avant son utilisation. Zéro indique que QMF pour Windows n'affiche la boîte de dialogue Information Utilisateur que lorsque cela s'avère nécessaire.
Compte	Occasionnellement, une chaîne spécifiant des informations sur le compte à communiquer au le serveur lors de la connexion. Le serveur peut utiliser cette information dans un système de comptabilité des travaux.
SuppressDialog	Une valeur autre que zéro indique que QMF pour Windows n'affiche pas la boîte de dialogue Information Utilisateur, même si aucun ID utilisateur ou mot de passe n'a été spécifié. Cette option est utile lorsqu'elle fonctionne dans un environnement dans lequel il n'y a pas d'utilisateur pour répondre à la boîte de dialogue Information Utilisateur, par exemple sur un serveur web.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), ou GetLastSQLState() pour avoir plus de renseignements sur l'erreur.

Thèmes liés

SetParent()

InitializeStaticQuery()

long InitializeStaticQuery(BSTR *NomCollecte*, BSTR *NomModule*, BSTR *ConsistencyToken*, short *NombreSection*)

Description

Cette fonction spécifie la section d'un module que vous voulez exécuter en requête statique.

Paramètres

Nom	Description
NomCollecte	Le nom d'une collecte précédemment liée.
NomModule	Le nom d'un module précédemment lié.
ConsistencyToken	le signe utilisé par la collecte et le module ci-dessus.
NuméroSection	Le numéro de section de l'instruction dans la collecte et le module que vous voulez exécuter.

Valeur renvoyée

Si c'est positif, l'ID de la requête. Si le résultat est négatif, -1. Vous devez utiliser cette valeur dans tous les appels d'interface qui requièrent le paramètre *IDRequête*.

IsStatic()

BOOL IsStatic(long *IDRequête*)

Description

Cette fonction détermine si l'ID requête spécifié se réfère ou non à une requête statique ou dynamique.

Paramètres

Nom	Description
IDRequête	L'ID requête comme renvoyée par InitializeQuery() ou InitializeStaticQuery().

Valeur renvoyée

Renvoie une valeur autre que zéro si c'est positif et *ID requête* se réfère à une requête statique, autrement zéro.

Open()

short Open(long *IDRequête*, long *RowLimit*, BOOL *FetchAllRows*)

Description

Utilisez cette fonction pour exécuter une requête utilisant l'instruction SELECT, en ouvrant un pointeur dans la base de données pour la requête. Utilisez FetchNextRow() ou FetchNextRows() pour obtenir les données pour la requête, et appelez Close() lorsque vous aurez terminé. Si QMF pour Windows est configuré pour saisir automatiquement toutes les lignes (voir RSR_FETCHALLROWS dans la description de GetResourceLimit()) ou si le paramètre FetchAllRows a une autre valeur que zéro, QMF pour Windows saisit toutes les lignes du résultat dans sa mémoire interne avant de les renvoyer à l'appel.

Remarque : Le nom de cette fonction coïncide avec l'option Ouvrir de Microsoft Access 2.0. Si vous utilisez MS Access 2.0, mettez le nom de la fonction entre crochets [].

Remarque : N'utilisez cette fonction que dans les instructions SQL SELECT. Pour n'importe quelle autre instruction, par exemple SET, appelez Execute() à la place. Pour déterminer l'instruction utilisée dans une requête, appelez GetQueryVerb().

Paramètres

Nom	Description
<i>ID Requête</i>	L'ID requête comme renvoyée par InitializeQuery().
RowLimit	Un nombre indiquant le nombre maximum de lignes à extraire de la base de données. Zéro indique qu'il n'y a pas de limite sauf la limite de lignes fixée par le programme d'Administration de QMF pour Windows.
FetchAllRows	Une valeur booléenne qui indique si toutes les lignes du résultat sont reprises automatiquement ou non dans la mémoire interne de QMF pour Windows. Si la valeur est différente de zéro, toutes les lignes sont automatiquement reprises, le pointeur est fermé et la base de données est libre pour d'autres utilisateurs. C'est le même procédé qu'en appelant CompleteQuery().

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), ou GetLastSQLState() pour avoir plus de renseignements sur l'erreur.

Prepare()

short Prepare(long *IDRequête*)

Description

Cette fonction prépare la requête spécifiée dans *IDRequête*. L'instruction est examinée par le serveur de base de données, qui recherche l'existence d'objet, les autorisations requises, etc. Si la requête est une instruction SELECT, l'information concernant les colonnes renvoyée par l'instruction est disponible après avoir sélectionné Prepare().

Paramètres

Nom	Description
<i>ID Requête</i>	L'ID requête comme renvoyée par InitializeQuery().

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType(), GetLastErrorCode(), GetLastErrorError(), ou GetLastErrorSQLState() pour avoir plus de renseignements sur l'erreur.

Thèmes liés

Execute()
Open()

PrintReport()

short PrintReport(long *IDRequête*, short *SourceType*, BSTR *Source*, BSTR *OutputFileName*, short *PageLength*, short *PageWidth*, BOOL *IncludeDateTime*, BOOL *IncludePageNumbers*, [VARIANT *Format*], [VARIANT *UseFormPageSetup*])

Description

PrintReport() est synonyme d'ExportReport().

ReinitializeServer()

short ReinitializeServer()

Description

Cette fonction réinitialise la connexion à un serveur de base de données. Normalement, vous ne devez appeler cette fonction que si l'une ou l'autre fonctions API QMF pour Windows renvoie un message d'erreur. Appeler cette fonction provoque un retour implicite en arrière, ce qui provoque la fermeture de tout pointeur ouvert et l'annulation de tout ID de requête courant.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType(), GetLastErrorCode(), GetLastErrorError(), ou GetLastErrorSQLState() pour avoir plus de renseignements sur l'erreur.

Rollback()

short Rollback()

Description

Cette fonction annule tous les changements apportés dans l'unité de travail courante, termine l'unité de travail courante, ferme tout pointeur ouvert et annule tout ID de requête courant.

Remarque : Le nom de cette fonction coïncide avec l'option Revenir de Microsoft Access 2.0. Si vous utilisez MS Access 2.0, mettez le nom de la fonction entre crochets [].

Remarque : Le retour en arrière n'affecte que les changements SQL apportés en utilisant Open() ou Execute(). Le retour en arrière n'affecte pas les changements apportés par d'autres fonctions API QMF pour Windows, comme FastSaveData(), SaveData(), ou DeleteQMFObject().

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString(), GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), ou GetLastSQLState() pour avoir plus de renseignements sur l'erreur.

Thèmes liés

Commit()

RunProc()

short RunProc(long ProcID)

Description

Cette fonction exécute la procédure spécifiée. La procédure s'exécute jusqu'à ce qu'elle soit terminée ou jusqu'à ce qu'une erreur se produise. Vous ne pouvez pas avoir accès aux résultats de la procédure (par exemple, les données d'une requête exécutée) avec cette interface de programmation. Cependant, tout fichier exporté ou toute donnée enregistrée par la procédure sont disponibles après l'exécution.

Paramètres

Nom	Description
ProcID	L'ID de la procédure comme renvoyée par InitializeProc().

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, ou `GetLastSQLState()` pour avoir plus de renseignements sur l'erreur.

SaveData()

short SaveData(long *IDRequête*, long *PremièreLigne*, long *PremièreColonne*, long *DernièreLigne*, long *DernièreColonne*, BOOL *Replace*, BSTR *TableName*, BSTR *TableSpaceName*, BSTR *ServerName*, BSTR *UserID*, BSTR *Password*, BOOL *ForceDialog*, [VARIANT *Account*], [VARIANT *Comment*], [VARIANT *CommitScope*])

Description

Cette fonction copie l'ensemble de lignes et de colonnes indiquées dans l'espace table spécifié. Vous devez appeler `CompleteQuery()` avant de faire appel à cette fonction si vous n'avez pas extrait les données de toutes les lignes que vous voulez enregistrer dans la table. Si vous essayez de sauvegarder des lignes qui n'ont pas encore été extraites de la base de données, l'opération échoue. Si la table indiquée existe déjà, la présentation des nouvelles données doit être identique (même nombre et même type de colonnes) à celle de la table existante.

Cette fonction opère dans une autre unité de travail que les autres fonctions API et ses résultats sont directement renvoyés. Le fait d'appeler `Commit()` ou `Rollback()` n'a aucune incidence sur les changements apportés lors de l'utilisation de cette fonction.

Paramètres

Nom	Description
<i>ID Requête</i>	L'ID requête comme elle est renvoyée par <code>InitializeQuery()</code> .
<i>PremièreLigne</i>	La première ligne que vous voulez inclure dans la sauvegarde. La valeur d'une première ligne de résultats est 0.
<i>PremièreColonne</i>	La première colonne que vous voulez inclure dans la sauvegarde. La valeur de la première colonne de résultats est zero.
<i>DernièreLigne</i>	La dernière ligne que vous voulez inclure dans la sauvegarde, ou -1 si toutes les lignes sont reprises. La valeur de la dernière ligne des résultats est égale au nombre total de lignes moins une.
<i>DernièreColonne</i>	La dernière colonne que vous voulez inclure dans la sauvegarde, ou -1 si toutes les colonnes sont reprises. La valeur de la dernière colonne des résultats est égale au nombre total de colonnes moins une.

Replace	Une valeur autre que zéro indique que les données spécifiées remplacent n'importe quelle donnée existante dans la table. Zéro indique que la donnée spécifiée est ajoutée à n'importe quelle donnée existante dans la table.
TableName	Le nom de la table dans laquelle les données sont enregistrées. Si la table n'existe pas, elle est créée.
TableSpaceName	Le nom de l'espace table dans lequel la table existe ou est créée. Si <i>TableSpaceName</i> équivaut à NULL ou à une chaîne vide, l'espace table par défaut est utilisé. Si vous avez configuré QMF de telle sorte que Windows utilise toujours l'espace table par défaut (voir <i>RSR_SDDIFFERENTTS</i> dans la description de <i>GetResourceLimit()</i>), ce paramètre n'est pas retenu.
ServerName	Le nom du serveur de base de données dans lequel la table est stockée. Si <i>ServerName</i> équivaut à NULL ou à une chaîne vide, le serveur que vous indiquez dans l'appel vers <i>InitializeServer()</i> est utilisé, et <i>UserID</i> , <i>Password</i> , <i>ForceDialog</i> , et <i>Account</i> sont ignorés.
IDUtilisateur	Si vous spécifiez un serveur différent dans <i>ServerName</i> , <i>UserID</i> est l'ID utilisateur utilisé pour ce serveur. Si vous ne spécifiez pas d'ID utilisateur, QMF pour Windows utilise le dernier ID utilisateur spécifié pour ce serveur, si disponible, ou affiche la boîte de dialogue si aucun ID n'est disponible. Ce paramètre est ignoré si <i>ServerName</i> équivaut à NULL ou à une chaîne vide.
Mot de passe	Si vous spécifiez un serveur différent <i>ServerName</i> , <i>Password</i> sera le mot de passe utilisé pour ce serveur. Si vous ne spécifiez pas de mot de passe, QMF pour Windows utilisera le dernier mot de passe spécifié pour ce serveur, si disponible, ou affiche la boîte de dialogue si aucun mot de passe n'est disponible. Ce paramètre est ignoré si <i>ServerName</i> équivaut à NULL ou à une chaîne vide.
ForceDialog	Si vous spécifiez un serveur différent dans <i>ServerName</i> , une valeur différente de zéro force QMF pour Windows à afficher une boîte de dialogue exigeant une information d'ouverture de session, et ce même si un ID utilisateur ou un mot de passe ont été spécifiés ou sont disponibles. Zéro indique que QMF pour Windows n'affiche la boîte de dialogue Information Utilisateur que lorsque cela s'avère nécessaire. Ce paramètre est ignoré si <i>ServerName</i> équivaut à NULL ou à une chaîne vide.
Compte	Si vous spécifiez un serveur différent dans <i>ServerName</i> , occasionnellement, une chaîne contenant des informations sur le compte à transmettre au serveur lors de la connexion. Le serveur peut utiliser cette information dans un système de comptabilité des travaux. Ce paramètre est ignoré si <i>ServerName</i> équivaut à NULL ou à une chaîne vide.
Commentaire	Occasionnellement, une chaîne spécifiant un commentaire pour la table dans laquelle les données sont enregistrées.

<i>CommitScope</i>	Occasionnellement, le nombre de lignes à insérer à la fois dans la table avant d'enregistrer l'unité de travail. La valeur zéro indique que toutes les lignes doivent être insérées avant la validation. La valeur 10, par exemple, indique qu'une opération de validation doit être effectuée après chaque insertion de 10 lignes.
--------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, ou `GetLastSQLState()` pour avoir plus de renseignements sur l'erreur. Si le résultat est vide ou si aucune ligne n'a été extraite de la base de données, une valeur autre que zéro est renvoyée sauf si `PremièreLigne=0`, et `DernièreLigne=-1`. Dans ce cas, zéro est renvoyé et une table vide est créée.

SaveQMFPProc()

short SaveQMFPProc(BSTR *DétenteurEtNom*, BSTR *Text*, BSTR *Comment*, BOOL *Replace*, BOOL *Share*)

Description

Cette fonction enregistre une procédure sur un serveur de base de données.

Paramètres

Nom	Description
<i>DétenteurEtNom</i>	Une chaîne contenant le détenteur et le nom, séparés par un point, de la procédure que vous voulez enregistrer. Par exemple, <code>John.Proc2</code>
Texte	Une chaîne contenant le texte que vous voulez enregistrer dans la procédure.
Commentaire	Une chaîne contenant n'importe quel commentaire que vous voulez enregistrer avec la procédure. S'il n'y a pas de commentaire, faites passer ce paramètre pour une chaîne vide ou pour NULL .
Remplacer	Une valeur autre que zéro remplace une procédure existante portant le même nom. Zéro interrompt l'opération s'il existe une procédure portant le même nom.
Partager	Une valeur autre que zéro partage la procédure avec d'autres utilisateurs. Zéro ne partage pas la procédure avec d'autres utilisateurs.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, ou `GetLastSQLState()` pour avoir plus de renseignements sur l'erreur.

SaveQMFQuery()

short SaveQMFQuery(BSTR *DétenteurEtNom*, BSTR *Text*, BSTR *Comment*, BOOL *Replace*, BOOL *Share*)

Description

Cette fonction enregistre une requête sur un serveur de base de données.

Paramètres

Nom	Description
<i>DétenteurEtNom</i>	Une chaîne contenant le détenteur et le nom, séparés par un point, de la requête que vous voulez enregistrer. Par exemple, John.Requête2
Texte	Une chaîne contenant le texte que vous voulez enregistrer dans la requête.
Commentaire	Une chaîne contenant n'importe quel commentaire que vous voulez enregistrer avec la requête. S'il n'y a pas de commentaire, faites passer ce paramètre pour une chaîne vide ou pour NULL .
Remplacer	Une valeur autre que zéro remplace une procédure existante portant le même nom. Zéro interrompt l'opération s'il existe une requête portant le même nom.
Partager	Une valeur autre que zéro partage la requête avec d'autres utilisateurs. Zéro ne partage pas la requête avec d'autres utilisateurs.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, ou `GetLastSQLState()` pour avoir plus de renseignements sur l'erreur.

SetBindOption()

short SetBindOption(BSTR *NomCollecte*, BSTR *NomModule*, short *Option*, short *Value*)

Description

Cette fonction détermine les options pour la collecte et le module avant de faire appel à EndBind().

Paramètres

Nom	Description
NomCollecte	L'ID collecte du module pour lequel vous voulez déterminer l'option.
NomModule	Le nom du module pour lequel vous voulez déterminer l'option.
Option	Une des options reprises dans la liste ci-dessous.
Valeur	Une valeur autre que zéro remplace une requête existante portant le même nom. Zéro interrompt l'opération s'il existe une requête portant le même nom.
Partager	Une des valeurs reprises dans la liste ci-dessous pour l'option spécifiée.

Les significations et valeurs des différentes options sont les suivantes :

Option	Signification	Description
DDM_PKGRPLOPT(0x211C)	Indicateur spécifiant s'il faut remplacer ou non un module existant en lui attribuant le même ID collecte et nom.	DDM_PKGRPLALW (0x241F) Oui DDM_PKGRPLNA (0x2420) Non
DDM_STTDECDEL(0x2121)	le limiteur utilisé comme point des décimales dans les instructions SQL du module.	DDM_DECDELPRD (0x243C) Point DDM_DECDELCMA (0x243D) Virgule
DDM_STTSTRDEL(0x2120)	le limiteur utilisé pour les valeurs de chaîne dans les instructions SQL du module.	DDM_STRDELAP (0x2426) Apostrophe DDM_STRDELQ (0x2427) Double point
DDM_PKGISOLVL(0x2124)	Le niveau d'isolement pour le module.	DDM_ISOLVLALL (0x2443) Tous DDM_ISOLVLCHG (0x2441) Changer DDM_ISOLVLCS (0x2442) Stabilité du pointeur DDM_ISOLVLNC (0x2445) Pas de validation DDM_ISOLVLR (0x2444) Lecture répétée

DDM_PKGATHOPT(0x211E)	Indicateur spécifiant s'il faut ou pas garder les autorisations existantes sur le module.	DDM_PKGATHKP (0x2425) Garder DDM_PKGATHRVK (0x2424) Annuler
DDM_QRYBLKCTL(0x2132)	La méthode à utiliser lors de la recherche de ligne de données pour des requêtes dans le module.	DDM_FIXROWPRC (0x2418) Ligne à la fois DDM_LMTBLKPRC (0x2417) Bloc à la fois
DDM_RDBRLSOPT(0x2129)	Quand extraire des ressources de base de données acquises lors de l'exécution du module.	DDM_RDBRLSCMM (0x2438) Enregistrement DDM_RDBRLSCNV (0x2439) Libération de la conversation
DDM_STTDATFMT(0x2122)	Format pour valeurs date extraites.	DDM_ISODATFMT (0x2429) ISO DDM_USADATFMT (0x242A) US DDM_EURDATFMT (0x242B) Européen DDM_JISDATFMT (0x242C) Normes de l'industrie japonaise
DDM_STTTIMFMT(0x2123)	Format pour valeurs heure extraites.	DDM_ISOTIMFMT (0x242E) ISO DDM_USATIMFMT (0x242F) US DDM_EURTIMFMT (0x2430) Européen DDM_JISTIMFMT (0x2431) Normes de l'industrie japonaise

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()` pour avoir plus de renseignements sur l'erreur.

SetBindOwner()

short SetBindOwner(BSTR *NomCollecte*, BSTR *NomModule*, BSTR *OwnerID*)

Description

Cette fonction vous permet de spécifier un détenteur différent de votre ID utilisateur pour le module que vous liez. Ceci peut être utilisé dès lors que l'ID utilisateur ne possède pas les droits nécessaires à l'édition du module, contrairement au détenteur indiqué.

Paramètres

Nom	Description
NomCollecte	L'ID collecte du module pour lequel vous voulez déterminer le détenteur.
NomModule	Le nom du module pour lequel vous voulez déterminer le détenteur.
Commentaire	Une chaîne contenant n'importe quel commentaire que vous voulez enregistrer avec la requête. S'il n'y a pas de commentaire, faites passer ce paramètre pour une chaîne vide ou pour NULL.
OwnerID	L'ID détenteur demandé pour le module que vous liez.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, ou `GetLastSQLState()` pour avoir plus de renseignements sur l'erreur.

SetBusyWindowButton()

annuler `SetBusyWindowButton(BSTR Text)`

Description

Cette fonction spécifie le texte affiché sur le bouton annuler de la fenêtre ouverte.

Paramètres

Nom	Description
Texte	Une chaîne spécifiant le texte affiché sur le bouton annuler de la fenêtre ouverte. La valeur par défaut est "Annuler". Si vous spécifiez une chaîne vide, le bouton est caché. Quel que soit le texte que vous spécifiez, le bouton annule, ou ferme, toujours la fenêtre.

Valeur renvoyée

Rien.

Thèmes liés

`SetBusyWindowMessage()`

`SetBusyWindowMode()`

`SetBusyWindowTitle()`

`ShowBusyWindow()`

SetBusyWindowMessage()

annuler SetBusyWindowMessage(*BSTR Message*)

Description

Cette fonction spécifie le texte affiché dans le champ message de la fenêtre ouverte.

Paramètres

Nom	Description
Message	Une chaîne spécifiant le texte affiché dans le champ message de la fenêtre ouverte.

Valeur renvoyée

Rien.

Thèmes liés

SetBusyWindowButton()

SetBusyWindowMode()

SetBusyWindowTitle()

ShowBusyWindow()

SetBusyWindowMode()

annuler SetBusyWindowMode(*short Mode*)

Description

Cette fonction détermine si QMF pour Windows affiche ou pas la fenêtre ouverte. La fenêtre ouverte est utile pour donner un retour d'information à l'utilisateur et pour permettre à l'utilisateur d'annuler une action en cours dans la base de données. Vos changements prendront effet la prochaine fois que QMF pour Windows exécutera une opération qui provoquera l'affichage ou le masquage de la fenêtre ouverte.

Paramètres

Nom	Description
Mode	Spécifie quand QMF pour Windows affiche la fenêtre ouverte.

Valeur	Signification
0 (RSM_NEVER)	La fenêtre ne s'affiche pas. C'est le défaut.
1 (RSM_WHENBUSY)	La fenêtre s'affiche lorsque QMF pour Windows communique avec la base de données. QMF pour Windows affiche automatiquement cette fenêtre de façon appropriée.

2 (RSM_CLIENTCONTROLLED)	La fenêtre s'affiche après avoir appelé ShowBusyWindow(TRUE), et après avoir appelé ShowBusyWindow(FALSE). Le client détermine quand s'affiche la fenêtre.
-----------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType() pour avoir plus de renseignements sur l'erreur.

Thèmes liés

SetBusyWindowButton()
SetBusyWindowMessage()
SetBusyWindowTitle()
SetParent()
ShowBusyWindow()

SetBusyWindowTitle()

annuler SetBusyWindowTitle(BSTR *Title*)

Description

Cette fonction spécifie le texte affiché dans la barre de titre de la fenêtre ouverte.

Paramètres

Nom	Description
Titre	Une chaîne spécifiant le texte affiché dans la barre de titre de la fenêtre ouverte.

Valeur renvoyée

Rien.

Thèmes liés

SetBusyWindowButton()
SetBusyWindowMode()
SetBusyWindowMessage()
ShowBusyWindow()

SetGlobalVariable()

short SetGlobalVariable(BSTR *Name*, BSTR *Value*)

Description

Cette fonction attribue une valeur à la variable globale spécifiée. Cette valeur est disponible pour une utilisation dans des requêtes, des formats et des procédures.

Paramètres

Nom	Description
Nom	Une chaîne contenant le nom de la variable que vous voulez déterminer.
Valeur	Une chaîne contenant la valeur que vous voulez attribuer à la variable spécifiée.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()` pour avoir plus de renseignements sur l'erreur.

SetHostVariable()

short `SetHostVariable(long IDRequête, VARIANT Index, VARIANT Value)`

Description

Cette fonction attribue une valeur à la variable hôte spécifiée dont référence est faite dans la requête. La requête doit être une requête statique faisant référence aux variables hôtes (soit celles stockées avec la requête QMF, soit celles créées par `AddHostVariable()`). *Index* peut spécifier soit l'index numérique de la variable hôte, ou le nom de la variable hôte.

Paramètres

Nom	Description
ID requête	L'ID requête comme renvoyée par <code>InitializeStaticQuery()</code> .
Index	Soit un nombre (variant type <code>VT_I2</code>) spécifiant l'index de la variable hôte dans la requête, ou une chaîne (variant type <code>VT_BSTR</code>) spécifiant le nom de la variable hôte.
Valeur	La valeur de la variable hôte. Pour spécifier une valeur nulle, le type de variant doit être fixé à <code>VT_EMPTY</code> .

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()` pour avoir plus de renseignements sur l'erreur.

SetOption()

short SetOption(short *Mode*, VARIANT *Value*)

Description

Cette fonction détermine la valeur de l'option spécifiée dans QMF pour Windows. Pour certaines options, les changements ne prennent pas effet avant d'avoir redémarré QMF pour Windows. Dans des conditions normales, vous ne redémarrez pas QMF pour Windows avant d'avoir détruit toutes les récurrences de l'objet d'API QMF pour Windows.

Paramètres

Nom	Description
Option	Spécifie quelle option déterminer.

Valeur	Signification
0 (RSO_SERVER_DEFINITION_FILE)	Nom du fichier de définition du serveur.
1 (RSO_CPIC_DLL)	Nom du fichier DLL du fournisseur CPI-C.
2 (RSO_CPIC_TIMEOUT_WARNING)	CPI-C avertissement de dépassement (en secondes). Cette limite n'est pas utilisée par API QMF pour Windows.
3 (RSO_CPIC_TIMEOUT_CANCEL)	CPI-C annulation du dépassement (en secondes).
4 (RSO_TCP_TIMEOUT_WARNING)	TCP avertissement de dépassement (en secondes). Cette limite n'est pas utilisée par API QMF pour Windows.
5 (RSO_TCP_TIMEOUT_CANCEL)	TCP annulation du dépassement (en secondes).
6 (RSO_DISPLAY_NULLS_STRING)	La chaîne utilisée pour afficher des valeurs nulles.
7 (RSO_ENTER_NULLS_STRING)	La chaîne utilisée pour introduire des valeurs nulles.
8 (RSO_ENTER_DEFAULTS_STRING)	La chaîne utilisée pour introduire des valeurs par défaut.
9 (RSO_TRACE_FILE_1)	Retrouver nom du fichier 1.
10 (RSO_TRACE_FILE_2)	Retrouver nom du fichier 2.
11 (RSO_TCP_TRACE_LEVEL)	Niveau de recherche TCP.
12 (RSO_CPIC_TRACE_LEVEL)	Niveau de recherche CPI-C.
13 (RSO_DDM_TRACE_LEVEL)	Niveau de recherche DDM.

Nom	Description
Valeur	La valeur à attribuer à l'option.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()` pour avoir plus de renseignements sur l'erreur.

Thèmes liés

`GetOption()`

SetParent()

`short SetParent(long ParentWnd)`

Description

Cette fonction détermine la fenêtre-parent pour les dialogues. Normalement, lorsque QMF pour Windows affiche un dialogue (dans la fenêtre ouverte ou dans la boîte de dialogue Information Utilisateur), il est centré sur la fenêtre principale de QMF pour Windows et suit le même modèle. Cette fonction vous permet de forcer les boîtes de dialogue de QMF pour Windows à être centrées sur votre fenêtre d'application cliente et de suivre le même modèle.

Paramètres

Nom	Description
<code>ParentWnd</code>	Le <code>HWND</code> de la nouvelle fenêtre-parent. Choisissez <code>NULL</code> pour utiliser la fenêtre principale de QMF pour Windows comme fenêtre-parent.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()` pour avoir plus de renseignements sur l'erreur.

Thèmes liés

`ShowBusyWindow()`

SetProcVariable()

`short SetProcVariable(long ProcID, BSTR Name, BSTR Value)`

Description

Cette fonction attribue une valeur à la variable spécifiée. Cette valeur remplace la variable avant d'exécuter la procédure. Si votre procédure contient

une ou plusieurs variables, vous devez appeler cette fonction pour déterminer les valeurs des variables avant d'appeler RunProc().

Paramètres

Nom	Description
ProcID	L'ID de la procédure comme renvoyée par InitializeProc().
Nom	Une chaîne contenant le nom de la variable que vous voulez déterminer.
Valeur	Une chaîne contenant la valeur que vous voulez attribuer à la variable spécifiée.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType() pour avoir plus de renseignements sur l'erreur.

SetVariable()

short SetVariable(long *IDRequête*, BSTR *Name*, BSTR *Value*)

Description

Cette fonction attribue une valeur à la variable spécifiée. Cette valeur remplace la variable avant d'exécuter l'instruction SQL. Si votre instruction SQL contient une ou plusieurs variables, vous devez appeler cette fonction pour déterminer les valeurs des variables avant d'appeler soit Open() soit Execute().

Paramètres

Nom	Description
ID requête	L'ID requête comme fournie par InitializeQuery().
Nom	Une chaîne contenant le nom de la variable que vous voulez déterminer.
Valeur	Une chaîne contenant la valeur que vous voulez attribuer à la variable spécifiée.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à GetLastErrorString() ou GetLastErrorType() pour avoir plus de renseignements sur l'erreur.

ShowBusyWindow()

annuler ShowBusyWindow(BOOL *Show*)

Description

Cette fonction commande à QMF pour Windows d'afficher ou de cacher la fenêtre ouverte. La fenêtre ouverte est utile pour donner un retour d'information à l'utilisateur et pour permettre à l'utilisateur d'annuler une action en cours dans la base de données. Cette fonction ne fonctionne que si vous appelez SetBusyWindowMode() dans un mode RSM_CLIENTCONTROLLED. Si vous définissez une fenêtre parent via la fonction SetParent(), la fenêtre est modale par rapport à la fenêtre spécifiée.

Paramètres

Nom	Description
Afficher	Une valeur autre que zéro affiche la fenêtre ouverte; zéro cache la fenêtre ouverte. Si la valeur est différente de zéro, la fenêtre ouverte s'affiche jusqu'à ce que vous choisissiez ShowBusyWindow() avec <i>Afficher</i> mis à zéro.

Valeur renvoyée

Rien.

StartBind()

short StartBind(BSTR *NomCollecte*, BSTR *NomModule*, BSTR *ConsistencyToken*)

Description

Cette fonction lance l'édition de liens d'un module dans la base de données.

Paramètres

Nom	Description
NomCollecte	L'ID collecte désiré pour le module.
NomModule	Le nom désiré pour le module.
ConsistencyToken	Une chaîne de 16 caractères contenant la représentation hexadécimale d'un signe de 8 octets utilisé pour assurer la logique entre le module lié dans la base de données et une application utilisant ce module. Lorsqu'une section est exécutée dans le module, vous devez lui donner cette même valeur.

Valeur renvoyée

Valeur zéro si ça a fonctionné, valeur autre que zéro si ça n'a pas fonctionné. Si la valeur est différente de zéro, vous pouvez faire appel à `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, ou `GetLastSQLState()` pour avoir plus de renseignements sur l'erreur.

Thèmes liés

`EndBind()`

`CancelBind()`

Annexe. Remarques

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

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

IBM EMEA Director of Licensing
IBM Europe Middle-East Africa
Tour Descartes
92066 Paris-La Défense Cedex 50
France

Pour le Canada, veuillez adresser votre courrier à :

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

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales :LE PRESENT DOCUMENT EST LIVRE «EN L'ETAT». IBM DECLINE TOUTE RESPONSABILITE, EXPLICITE OU IMPLICITE, RELATIVE AUX

INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE VALEUR MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Il est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut modifier sans préavis les programmes et les logiciels qu'il décrit.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

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

IBM Corporation
J74/G4
555 Bailey Avenue
P.O. Box 49023
San Jose, CA 95161-9023
U.S.A.

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

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux termes du Contrat sur les produits et services IBM, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Tous les tarifs indiqués sont les prix de vente actuels suggérés par IBM et sont susceptibles d'être modifiés sans préavis. Les tarifs appliqués peuvent varier selon les revendeurs.

Ces informations sont fournies uniquement à titre de planification. Elles sont susceptibles d'être modifiées avant la mise à disposition des produits décrits.

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

LICENCE SOUS COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquelles ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Marques

Les termes suivants sont des marques d'IBM Corporation dans certains pays :

ACF/VTAM	IBMLink
Advanced Peer-to-Peer Networking	IMS
AIX	Language Environment
AIX/6000	MVS/ESA
AS/400	MVS/XA
CICS	OfficeVision/VM
CICS/ESA	OS/2
CICS/MVS	OS/390
CICS/VSE	PL/I
COBOL/370	PROFS
DATABASE 2	QMF
DataJoiner.DB2	RACF
DB2 Universal Database	S/390
Distributed Relational Database Architecture	SQL/DS
DRDA	Virtual Machine/Enterprise Systems Architecture
DXTGDDM	Visual Basic
IBM	VM/XA
	VM/ESA
	VSE/ESAVTAM

Java et toutes les marques et logos incluant Java, ainsi que Solaris, sont des marques de Sun Microsystems, Inc. dans certains pays.

Lotus et 1-2-3 sont des marques de Lotus Development Corporation dans certains pays.

Microsoft, Windows et Windows NT sont des marques de Microsoft Corporation dans certains pays.

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

Index

A

- AddDecimalHostVariable() 75
- AddHostVariable() 76
- Affichage
 - instructions SQL dans les requêtes assistées 23
 - résultats 9
 - SQL 9
- Affichage des objets 46
- Aide-mémoire sur les API 75
- ajout
 - Conditions de lignes 21
 - conditions de tri 20
- ajout d'objets aux listes 47
- ajout d'une ligne
 - éditeur de tables 60
- Aperçu avant impression
 - requêtes assistées 26

B

- Barre d'outils
 - ajout de boutons 6
 - déplacement de boutons 6
 - personnalisation 6
 - suppression de boutons 7
- Base de données
 - sécurité 2
- BindDecimalHostVariable() 77
- BindHostVariable() 78
- BindSection() 79
- Blocage d'appels 74
- Boutons d'action
 - requête assistée 17

C

- Calculs
 - Formats 33
- CancelBind() 79
- ChangePassword() 80
- ClearList() 80
- Close() 81
- colonnes
 - ajout à des requêtes assistées 19
 - Formats 33
- colonnes, reclassement 28, 50
- colonnes, sélection 27
- Commencant par (condition de ligne) 21
- Commit() 81

- CompleteQuery() 82
- conditions
 - Formats 33
- Conditions de jonction
 - création dans des requêtes assistées 23
- conditions de lignes
 - ajout 21
 - contenant 21
 - égal à 21
 - entre 21
 - inférieur à 21
 - inférieur ou égal à 21
 - nul 21
 - se terminant par 21
 - supérieur ou égal à 21
 - utilisation 21
- Conditions de lignes
 - commencant par 21
 - supérieur à 21
- conditions de tri
 - ajout 20
 - utilisation 20
- Connexion 2
- Connexion à la base de données 74
- Contenant (condition de ligne) 21
- conversion du formatage des résultats de requête en format. 29, 51
- CopyToClipboard() 82
- Courrier internet 65
- création
 - requêtes statiques 55
- Création
 - requêtes 11
 - requêtes assistées 17
 - requêtes SQL 9
- Création d'objets 46
- création d'une procédure linéaire 41
- création de fichiers de tâches. 49

D

- DeleteQMFObject() 83
- détails
 - Formats 33
- Documents de requêtes multiples 10

E

- éditeur de tables 59
 - ajout d'une ligne 60
 - modification d'une ligne 60, 61
 - recherche de lignes 59
- Egal à (condition de ligne) 21
- EndBind() 84
- Enregistrement
 - formats dans des fichiers 37, 42
 - formats sur le serveur de base de données 37, 42
 - requêtes assistées dans des fichiers 24
 - requêtes assistées sur le serveur de base de données 25
 - requêtes SQL dans des fichiers 13
 - requêtes SQL sur le serveur de base de données 14
- enregistrement des résultats de requête 52
- enregistrement des résultats de requête sous forme de fichiers 30, 52
- enregistrer les résultats de requête 30
- Entre (condition de ligne) 21
- Envoyer à 65
- Est (opérateur de condition de ligne) 21
- Excel
 - extension 66
- Execute() 84
- ExecuteEx() 85
- ExecuteStoredProcedure() 86
- ExecuteStoredProcedureEx() 87
- exécution
 - requêtes statiques 57
- Exécution
 - requête SQL sur un serveur de base de données 9
 - requêtes assistées 19
- Exécution d'objets 46
- Exemples d'applications 66
- Export() 89
- exportation
 - rapports 39
- Exportation de données dans d'autres tables 65

Exportation de données (*suite*)
dans des fichiers 63
sur un serveur de base de
données 65
ExportForm() 91
ExportReport() 92
extension
Excel 66

F

FastSaveData() 94
Fenêtre des résultats 9
FetchNextRow() 95
FetchNextRowEx() 96
FetchNextRows() 96
FetchNextRowsEx() 98
Fichiers
exportation de données dans 63
fichiers de tâches, créer 49
final
Formats 33
FlushQMFCache() 98
Format
principal 33
formatage des résultats numériques
de requête 29, 51
formats
ouverture d'un fichier
enregistré 37, 42
sauvegarde dans des fichiers 42
sauvegarde sur le serveur de base
de données 37, 42
Formats
Calculs 33
colonnes 33
conditions 33
création d'un rapport 34
détails 33
final 33
HTML 34
options 34
page 34
ruptures 33
sauvegarde dans des fichiers 37
formats DB2 61

G

Gestion 4
GetColumnCount() 99
GetColumnDataValue() 99
GetColumnHeader() 100
GetColumnHeaderEx() 100
GetColumnHeadings() 101
GetColumnValue() 102
GetColumnValueEx() 103
GetDefaultServerName() 103

GetGlobalVariable() 103
GetHostVariableNames() 104
GetHostVariableTypes() 104
GetLastErrorString() 105
GetLastErrorType() 105
GetLastSQLCode() 107
GetLastSQLError() 107
GetLastSQLState() 108
GetOption() 109
GetOptionEx() 110
GetProcText() 111
GetProcVariables() 111
GetQMFOBJECTInfo() 112
GetQMFOBJECTInfoEx() 114
GetQMFOBJECTList() 116
GetQMFOBJECTListEx() 117
GetQMFPProcText() 118
GetQMFPQueryText() 118, 119
GetQueryVerb() 119
GetResourceLimit() 120
GetResourceLimitEx() 124
GetRowCount() 125
GetServerList() 125
GetServerListEx() 126
GetStoredProcedureResultSets() 127
GetVariables() 127
GetVariablesEx() 128

H

HTML
Formats 34

I

Identifiants comptables 4
impression
procédures 44
rapports 38
Impression
requêtes SQL 15
impression des résultats de
requête 31, 52
Inférieur à (condition de ligne) 21
Inférieur ou égal à (condition de
ligne) 21
InitializeProc() 129
InitializeQuery() 130
InitializeServer() 130
InitializeStaticQuery() 132
IsStatic() 132

L

lignes, sélection 27
Listes
objets 45
ouverture de fichiers
enregistrés 47

Listes, ajout d'objets 47
Listes, supprimer objets 47

M

Modification d'objets 46
modification d'une ligne
éditeur de tables 60, 61
Modification des mots de passe 3
Mots de passe
correction 3

N

N'est pas (opérateur de condition de
ligne) 21
Nul (condition de ligne) 21

O

objets
Listes 45
Open() 132
Opérateur de condition de ligne
est 21
n'est pas 21
options
Formats 34
Ouverture
fichiers de requêtes assistées 24
fichiers SQL enregistrés 14
formats enregistrés sur le serveur
de base de données 38
procédures sur le serveur de base
de données 43
requêtes assistées sur le serveur
de base de données 25
requêtes SQL sur le serveur de
base de données 14

P

page
Formats 34
Polices
affichage des requêtes 10
affichage des résultats 29, 50
Prepare() 133
prévisualisation
procédures imprimées 43
rapports 34
Prévisualisation
requêtes imprimées 15
prévisualisation des résultats de
requête 30, 52
principal
Formats 33
PrintReport() 134
procédure logique 41
procédure REXX 41

procédures
impression 44

R

rapports

création d'un rapport à l'aide de
formats 34
exportation 39
impression 38
prévisualisation 34

Recherche

éditeur de tables 59

reclassement des colonnes 28

Reclassement des colonnes 50

Redimensionnement de colonnes et
de lignes 27

regroupement des résultats de
requête 29, 51

ReinitializeServer() 134

Remarques 151

requêtes

création 11

Requêtes

création de requêtes
complexes 19

requêtes assistées

affichage des instructions
SQL 23

ajout de colonnes 19

conversion en requêtes SQL 23

création de conditions de
jonction 23

ouverture de fichiers

enregistrés 24

sauvegarde dans des fichiers 24

sauvegarde sur le serveur de base
de données 25

utilisation de plusieurs tables 22

utilisation de SQL 23

utilisation des variables de
substitution 24

Requêtes assistées

affichage des instructions
SQL 23

ajout de colonnes 19

ajout de tables 18

boutons d'action 17

conversion en requêtes SQL 23

création 17

création de conditions de
jonction 23

exécution 19

sauvegarde dans des fichiers 24

utilisation de plusieurs tables 22

utilisation de SQL 23

Requêtes complexes

création 19

Requêtes multiples

affichage simultané 10

requêtes SQL

aperçu avant impression 15

exécution sur un serveur de base
de données 9

Impression 15

ouverture d'un nouveau

document 9

ouverture de fichiers

enregistrés 14

sauvegarde dans des fichiers 13

sauvegarde sur le serveur de base
de données 14

requêtes statiques

création 55

exécution 57

utilisation des variables de

substitution 55

résultats de la requête,

formatage 29, 51

résultats de la requête,

regroupement 51

résultats de la requête, tri 28, 50

résultats de requête,

enregistrement 52

résultats de requête, enregistrement

sous forme de fichiers. 30, 52

résultats de requête, enregistrer 30

résultats de requête,

prévisualisation 30, 52

résultats de requête,

regroupement 29

résultats de requête, résumé. 30, 51

résultats de requête, impression 31,

52

résumé des résultats de requête 30,

51

Rollback() 135

RunProc() 135

ruptures

Formats 33

S

SaveData() 136

SaveQMFProc() 138

SaveQMFQuery() 139

se terminant par 21

sélection de colonnes et de

lignes 27

Serveur

définition 1

Serveur de base de données

exportation de données dans 65

SetBindOption() 139

SetBindOwner() 141

SetBusyWindowButton() 142

SetBusyWindowMessage() 143

SetBusyWindowMode() 143

SetBusyWindowTitle() 144

SetGlobalVariable() 144

SetHostVariable() 145

SetOption() 146

SetParent() 147

SetProcVariable() 147

SetVariable() 148

ShowBusyWindow() 149

SQL

utilisation dans des requêtes

assistées 23

StartBind() 149

Supérieur à (condition de ligne) 21

Supérieur ou égal à (condition de
ligne) 21

suppression d'objets des listes 47

T

Tables

ajout à des requêtes assistées 18

exportation de données dans 65

Tables multiples

dans les requêtes assistées 22

tri des résultats de requête 28, 50

V

Variables de substitution

exécution de requêtes SQL 13

remplacement par des variables

hôtes 55

utilisation dans des requêtes

assistées 24

utilisation dans des requêtes

SQL 12

utilisation dans des requêtes

statiques 55

Variables hôtes

utilisation dans des requêtes

statiques 55



SC11-1694-00



Spine information:



QMF

Initiation à QMF pour Windows

Version 7