

Query Management Facility



Erste Schritte mit QMF für Windows

Version 7

Query Management Facility



Erste Schritte mit QMF für Windows

Version 7

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter „Anhang. Bemerkungen“ auf Seite 159 gelesen werden.

Fünfte Ausgabe (September 2000)

Diese Veröffentlichung ist eine Übersetzung des Handbuchs

Query Management Facility Getting Started with QMF for Windows,
IBM Form SC27-0723-00,

herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2000
© Copyright IBM Deutschland Informationssysteme GmbH 2000

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW NLS Center
Kst. 2877
September 2000

Inhaltsverzeichnis

QMF-Bibliothek vii

Kapitel 1. Einführung 1

Datenbank-Server 1
DB2-Datenbank-Familie 1
Benutzername oder technischer Name —
Begriffserklärung 1
Server-Namen festlegen 2
Datenbanksicherheit. 3
Anmelden 3
Kennwörter korrigieren 4
Kennwörter ändern 4
Kontonamen angeben 5
Steuerung mit dem Governor 5
Ressourcengrenzwerte anzeigen 5
Eigenen Grenzwert für die Zeilenanzahl festlegen 6
Symbolleiste anpassen 7
Knöpfe zur Symbolleiste hinzufügen 7
Knöpfe in der Symbolleiste verschieben 8
Knöpfe aus der Symbolleiste entfernen 8

Kapitel 2. Mit SQL-Abfragen arbeiten 9

SQL-Abfragen. 9
Neue SQL-Abfragen erstellen 9
SQL-Abfragen auf einem Datenbank-Server ausführen 9
Zwischen der Ergebnissicht und der SQL-Sicht umschalten 9
Mit Schriftarten arbeiten 10
Schriftart zur Anzeige der Abfrage auswählen. 10
Mehrfachabfragen 10
Mehrere Abfragen gleichzeitig anzeigen. 10
Abfragen zeigen (erstellen) 11
Neue SQL-Abfragen erstellen 11
Substitutionsvariablen in SQL-Abfragen. 12
SQL-Abfragen mit Substitutionsvariablen ausführen. 13
SQL-Abfragen speichern und öffnen 14
SQL-Abfragen in Dateien speichern 14
Gespeicherte SQL-Abfragedateien öffnen 14
SQL-Abfragen auf dem Datenbank-Server speichern 14

Gespeicherte SQL-Abfragen auf dem Datenbank-Server öffnen. 15
SQL-Abfragen drucken 16
Abfrage vorab anzeigen 16
SQL-Abfragen drucken 16

Kapitel 3. Mit menügesteuerten Abfragen arbeiten 17

Einfache Abfragen erstellen 17
Neue menügesteuerte Abfragen öffnen 17
Aktionsknöpfe für menügesteuerte Abfragen 18
Menügesteuerten Abfragen Tabellen hinzufügen 19
Menügesteuerte Abfragen ausführen. 19
Komplexe Abfragen erstellen 20
Menügesteuerten Abfragen Spalten hinzufügen 20
Mit Sortierbedingungen arbeiten 21
Sortierbedingungen hinzufügen 21
Mit Zeilenbedingungen arbeiten 22
Zeilenbedingungen hinzufügen 22
Mehrere Tabellen in menügesteuerten Abfragen verwenden 23
Verknüpfungsbedingungen für menügesteuerte Abfragen erstellen 24
Menügesteuerte Abfragen und SQL 24
SQL für menügesteuerte Abfragen anzeigen 24
Menügesteuerte Abfragen in SQL konvertieren 24
Substitutionsvariablen in menügesteuerten Abfragen verwenden 25
Menügesteuerte Abfragen speichern 25
Menügesteuerte Abfragen in Dateien speichern 25
Menügesteuerte Abfragen öffnen, die in Dateien gespeichert sind 25
Menügesteuerte Abfragen auf dem Datenbank-Server speichern. 26
Gespeicherte menügesteuerte Abfragen auf dem Datenbank-Server öffnen 26
Menügesteuerte Abfragen drucken 27
Menügesteuerte Abfragen vorab anzeigen 27

Kapitel 4. Mit Abfrageergebnissen arbeiten	29
Abfrageergebnisse sortieren und in der Größe ändern	29
Spalten und Zeilen auswählen	29
Spalten und Zeilen in der Größe ändern	29
Automatisches Anpassen von Zeilen und Spalten.	30
Abfrageergebnisse sortieren	30
Spalten neu anordnen	30
Abfrageergebnisse formatieren	31
Schriftart zur Anzeige der Abfrageergebnisse auswählen	31
Numerische Abfrageergebnisse formatieren	31
Formatierungen von Abfrageergebnissen in ein Format konvertieren	32
Gruppierungs- und Berechnungsfunktionen für Abfrageergebnisse ausführen	32
Abfrageergebnisse gruppieren	32
Abfrageergebnisse in Übersichten zusammenfassen.	32
Abfrageergebnisse speichern und Formatierungen ausführen	33
Abfrageergebnisse als Tabelle speichern .	33
Abfrageergebnisse in Dateien speichern .	33
Abfrageergebnisse drucken	34
Druckvorschau der Abfrageergebnisse vorab anzeigen	34
Abfrageergebnisse drucken	34
Kapitel 5. Mit Berichten arbeiten	35
Formate	35
Grundlegendes zu Formaten	35
Bericht mit Hilfe eines Formats erstellen	36
Format bearbeiten	36
Format erstellen.	37
Schritt 1: Format erstellen	37
Schritt 2: Spaltenreihenfolge ändern . . .	37
Schritt 3: Spaltenüberschriften ändern .	37
Schritt 4: Spaltenformatierung ändern .	38
Schritt 5: Übersichtsinformationen hinzufügen	38
Schritt 6: Seitenkopfzeilen und -fußzeilen hinzufügen	39
Formate speichern	39
Format in einer Datei speichern	39
Gespeicherte Formatdateien öffnen	39
Formate auf dem Datenbank-Server speichern	40
Formate auf dem Datenbank-Server öffnen	41
Berichte drucken	41

Berichte exportieren	42
--------------------------------	----

Kapitel 6. Mit Prozeduren arbeiten 43

Prozeduren ausführen.	43
Neue lineare Prozedur erstellen	43
Neue Prozedur mit Logik erstellen	43
Prozedur auf einem Datenbank-Server ausführen	43
Prozeduren speichern	44
Prozeduren in Dateien speichern	44
Gespeicherte Prozedurdatei öffnen. . . .	44
Prozedur auf dem Datenbank-Server speichern	44
Gespeicherte Prozeduren auf dem Datenbank-Server öffnen.	45
Prozeduren drucken	46
Prozedur vorab anzeigen.	46
Prozedur drucken	46

Kapitel 7. Mit Listen arbeiten. 47

Objekte	47
Objekte auflisten	47
Befehle im Fenster 'Liste'.	48
Listen erstellen	49
Objekte zu Listen hinzufügen	49
Objekte in Listen löschen.	49
Listen in Dateien speichern	50
Gespeicherte SQL-Listendateien öffnen . .	50

Kapitel 8. Mit Job-Dateien arbeiten. 51

Job-Dateien	51
Job-Dateien erstellen	51
Job-Dateien ausführen.	51
Automatisches Anpassen von Zeilen und Spalten.	51
Abfrageergebnisse sortieren	52
Spalten neu anordnen.	52
Abfrageergebnisse formatieren	52
Schriftart zur Anzeige der Abfrageergebnisse auswählen	52
Numerische Abfrageergebnisse formatieren	53
Formatierungen von Abfrageergebnissen in ein Format konvertieren	53
Gruppierungs- und Berechnungsfunktionen für Abfrageergebnisse ausführen	53
Abfrageergebnisse gruppieren	53
Abfrageergebnisse in Übersichten zusammenfassen.	54
Abfrageergebnisse speichern und Formatierungen ausführen	54

Abfrageergebnisse als Tabelle speichern	54
Abfrageergebnisse in Dateien speichern	54
Abfrageergebnisse drucken	55
Druckvorschau der Abfrageergebnisse vorab anzeigen	55
Abfrageergebnisse drucken	55

Kapitel 9. Mit statischen Abfragen arbeiten 57

Statische Abfragen	57
Statische Abfragen erstellen	57
Substitutionsvariablen in statischen Abfragen durch Host-Variablen ersetzen	58
Statische Abfrage ausführen.	59

Kapitel 10. Mit dem Tabelleneditor arbeiten 61

Tabelleneditor	61
Mit dem Tabelleneditor nach Zeilen suchen	61
Zeile hinzufügen	62
Zeile ändern	63
Zeile löschen.	63
Tabellen in der Sicht 'Abfrageergebnisse' bearbeiten.	63
Zeile in der Sicht 'Abfrageergebnisse' löschen.	63
Tabellen in der Sicht 'Abfrageergebnisse' aktualisieren	63
DB2 Forms	63

Kapitel 11. Daten verteilen. 65

Daten exportieren	65
Daten in Dateien exportieren	65
Daten importieren	67
Daten auf einem Datenbank-Server speichern	68
Mit dem Befehl 'Senden an' arbeiten	69
Mit dem Add-In für Microsoft Excel arbeiten	69
Beispielanwendungen verwenden.	70

Kapitel 12. Mit der QMF Berichtszentrale arbeiten 71

Erste Schritte in der QMF Berichtszentrale	71
Fenster 'QMF Berichtszentrale'.	71
Verbindung zum Server herstellen	73
Mit Berichten und Objekten arbeiten	74
Berichte ausführen	74
Mit Ordnern und Favoriten arbeiten	75
Berichte zu den Favoriten hinzufügen	76

Kapitel 13. Mit der API von QMF für Windows arbeiten 77

QMF für Windows über die API steuern	77
Blockierende Aufrufe	78
Verbindung zur Datenbank herstellen	78
API-Referenz.	79
AddDecimalHostVariable()	79
AddHostVariable()	80
BindDecimalHostVariable()	81
BindHostVariable()	82
BindSection().	83
CancelBind()	83
ChangePassword()	84
ClearList().	84
Close()	85
Commit()	85
CompleteQuery()	86
CopyToClipboard().	86
DeleteQMFObject().	88
EndBind().	88
Execute()	89
ExecuteEx()	89
ExecuteStoredProcedure()	90
ExecuteStoredProcedureEx().	91
Export()	93
ExportForm()	95
ExportReport()	96
FastSaveData()	98
FetchNextRow().	99
FetchNextRowEx()	100
FetchNextRows()	100
FetchNextRowsEx()	102
FlushQMFCache().	103
GetColumnCount()	103
GetColumnDataValue()	103
GetColumnHeader()	104
GetColumnHeaderEx()	104
GetColumnHeadings()	105
GetColumnValue()	106
GetColumnValueEx().	107
GetDefaultServerName()	107
GetGlobalVariable()	108
GetHostVariableNames()	108
GetHostVariableTypes()	109
GetLastErrorString()	109
GetLastErrorType()	110
GetLastSQLCode()	111
GetLastSQLError()	112
GetLastSQLState().	113
GetOption().	113
GetOptionEx()	115
GetProcText()	115

GetProcVariables()	116
GetQMFObjectInfo()	117
GetQMFObjectInfoEx()	119
GetQMFObjectList()	120
GetQMFObjectListEx()	121
GetQMFPProcText()	122
GetQMFPQueryText()	123
GetQueryText()	123
GetQueryVerb()	124
GetResourceLimit()	124
GetResourceLimitEx()	129
GetRowCount()	130
GetServerList()	130
GetServerListEx()	131
GetStoredProcedureResultSets()	132
GetVariables()	133
GetVariablesEx()	134
InitializeProc()	134
InitializeQuery()	135
InitializeServer()	136
InitializeStaticQuery()	138
IsStatic()	138
Open()	139
Prepare()	140
PrintReport()	140
ReinitializeServer()	140

Rollback()	141
RunProc()	141
SaveData()	142
SaveQMFPProc()	145
SaveQMFPQuery()	146
SetBindOption()	147
SetBindOwner()	149
SetBusyWindowButton()	149
SetBusyWindowMessage()	150
SetBusyWindowMode()	150
SetBusyWindowTitle()	151
SetGlobalVariable()	152
SetHostVariable()	152
SetOption()	153
SetParent()	154
SetProcVariable()	155
SetVariable()	156
ShowBusyWindow()	156
StartBind()	157







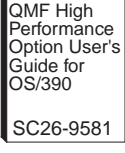
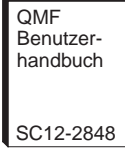
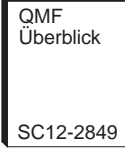





Anhang. Bemerkungen 159

Marken 162

Index 163

QMF-Bibliothek

Folgende Handbücher können Sie über Ihren IBM Ansprechpartner bestellen.

Einführung	 <p>QMF Überblick GC12-2858</p>			
Installation, Planung, Verwaltung und Diagnose	 <p>Installing and Managing QMF on OS/390 GC26-9575</p>	 <p>Installing and Managing QMF on VM/ESA GC26-9573</p>	 <p>Installing and Managing QMF on VSE/ESA GC26-9574</p>	 <p>QMF für Windows Installation und Verwaltung GC12-2859</p>
	 <p>QMF Messages and Codes GC26-9580</p>	 <p>QMF High Performance Option User's Guide for OS/390 SC26-9581</p>		
Verwendung	 <p>QMF Benutzer- handbuch SC12-2848</p>	 <p>QMF Überblick SC12-2849</p>	 <p>Erste Schritte mit QMF für Windows SC12-2847</p>	
	Anwendungs- programmierung	 <p>Developing QMF Applications SC26-9579</p>		
	Online- Bibliotheken	 <p>SK2T-0730 OS/390, VM und VSE</p>	 <p>SK2T-6700 (nur OS/390)</p>	 <p>SK2T-2067 (nur VM)</p>

Kapitel 1. Einführung

In diesem Kapitel wird ein Überblick über QMF für Windows gegeben. Darüber hinaus werden einige der grundlegenden Schritte für die Arbeit mit dem Produkt erläutert.

Datenbank-Server

Abfragen, Formate, Prozeduren und Tabellen werden auf einem Datenbank-Server ausgeführt und gespeichert.

DB2-Datenbank-Familie

QMF für Windows kann zu verschiedenen DB2-Datenbanken eine Verbindung herstellen.

- DB2 UDB für OS/390, DB2 für OS/390 und DB2 für MVS
- DB2 Server für VSE & VM und SQLDS
- DB2 Universal Database und DB2 Common Server
- DB2 Parallel Edition
- DataJoiner

Durch Ihre QMF für Windows-Lizenz wird bestimmt, unter welchen DB2-Produkten Sie eine Installation durchführen und zu welchen DB2-Datenbanken Sie mit Ihrer Kopie von QMF für Windows eine Verbindung herstellen können.

Benutzername oder technischer Name — Begriffserklärung

Die verschiedenen DB2-Versionen und -Typen bezeichnen Datenbanken mit entweder einem RDB-Namen (*Relational Database Name*, Name der relationalen Datenbank), einem Standortnamen oder einem anderen technischen Namen.

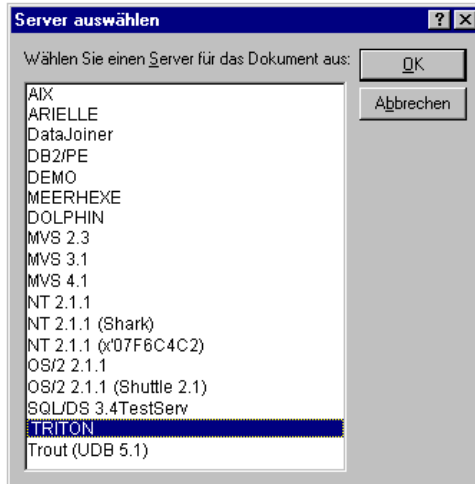
Unter QMF für Windows kann Ihr Administrator einem Datenbanknamen einen benutzerfreundlicheren und leichter zu merkenden Namen zuordnen, wie beispielsweise "Datenbank Einkauf" anstelle von "DB2P_01_EINKF".

QMF für Windows bezeichnet einen Datenbank-Server oder eine DB2-Datenbank als "Server".

Server-Namen festlegen

Bevor Sie eine Datenbank abfragen können, müssen Sie QMF für Windows mitteilen, wo die Datenbank gespeichert ist.

1. Wählen Sie hierzu im Menü **Datei** die Option **Neu** und anschließend **SQL-Abfrage** aus. Ein neues SQL-Abfragedokument wird anschließend geöffnet.
2. Wählen Sie dann im Menü **Abfrage** die Option **Server auswählen...** aus. Das Dialogfenster **Server auswählen** wird geöffnet.



3. Wählen Sie in der Liste der verfügbaren Server den Server aus, auf dem Sie eine Abfrage durchführen wollen, und klicken Sie **OK** an. Wenn Sie das nächste Mal eine QMF für Windows-Sitzung starten, stellt QMF für Windows die Verbindung zum selben Server automatisch wieder her.

Datenbanksicherheit

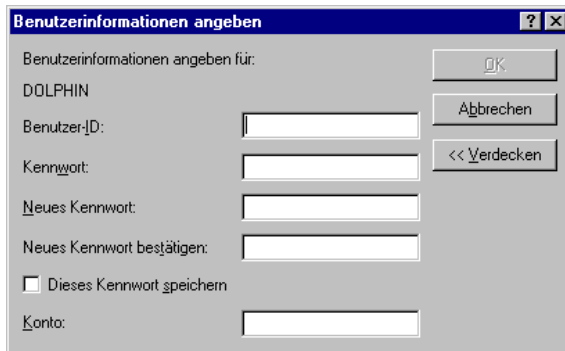
Bevor Sie eine Verbindung zu einem Server herstellen können, müssen Sie eine Benutzer-ID und ein Kennwort angeben.

Anmelden

Sie müssen eine gültige Benutzer-ID und ein gültiges Kennwort für den Datenbank-Server angeben, auf den Sie versuchen, zuzugreifen. Die Benutzer-ID und das Kennwort für den Datenbank-Server ist nicht notwendigerweise mit der Benutzer-ID und dem Kennwort Ihrer lokalen Maschine oder Ihres Netzcomputers identisch. Wenn Sie Windows verwenden, haben Sie die Möglichkeit, in QMF für Windows-Sitzungen benutzte Server-Kennwörter zu speichern. Wenn Sie momentan unter Windows angemeldet sind, wird im Dialogfenster **Benutzerinformationen angeben** ein zusätzliches Markierungsfeld mit dem Namen **Dieses Kennwort speichern** angezeigt. Wenn Sie dieses Feld markieren, wird das für den jeweiligen Server eingegebene Kennwort in der Kennwortliste von Windows gespeichert. Wenn Sie unter Windows angemeldet sind, kann QMF für Windows das Kennwort nun automatisch abrufen, so dass Sie es nicht mehr eingeben müssen. Wenn Sie nicht angemeldet sind, oder wenn Sie QMF für Windows ausführen, bzw. wenn Sie als ein anderer Benutzer angemeldet sind, fordert Sie QMF für Windows zur Eingabe einer Benutzer-ID und eines Kennworts auf.

Anmerkung: Wenn Sie angeben, dass ein Kennwort gespeichert werden soll, kann jeder, der sich unter Ihrem Windows-Konto anmelden kann, mit Ihrer (Server-)Benutzer-ID und Ihrem Kennwort auf Ihre Datenbank-Server zugreifen.

1. Wählen Sie im Menü **Abfrage** die Option **Benutzerinformationen angeben...** aus. Das Dialogfenster **Benutzerinformationen angeben** wird anschließend angezeigt.



2. Geben Sie Ihre Benutzer-ID und Ihr Kennwort in den entsprechenden Feldern ein.

Anmerkung: Bei der Eingabe der Benutzer-ID und des Kennworts müssen Sie auf Groß-/Kleinschreibung achten; dies bedeutet, dass eine Benutzer-ID oder ein Kennwort in Großbuchstaben auch in Großschreibung eingegeben werden muss. Bei einigen Datenbank-Servern muss bei den eingegebenen Benutzer-IDs und Kennwörtern auf Groß-/Kleinschreibung geachtet werden, bei anderen wiederum nicht.

3. Wählen Sie das Markierungsfeld **Dieses Kennwort speichern** aus, wenn Sie Ihre Benutzer-ID und Ihr Kennwort speichern wollen.
4. Klicken Sie **OK** an. QMF für Windows speichert diese Informationen für den Zugriff auf den Server.

Kennwörter korrigieren

Wenn Sie ein falsches Kennwort eingegeben haben, können Sie den Fehler korrigieren. Öffnen Sie dazu erneut das Dialogfenster **Benutzerinformationen angeben**.

1. Klicken Sie hierzu im Menü **Abfrage** die Option **Benutzerinformationen angeben...** an. Das Dialogfenster **Benutzerinformationen angeben** wird anschließend angezeigt.
2. Geben Sie das Kennwort erneut ein, und klicken Sie dann **OK** an. Das Kennwort wird entsprechend korrigiert.

Kennwörter ändern

Von QMF für Windows aus können Sie Ihr Kennwort auf dem Datenbank-Server ändern. Diese Funktion wird momentan lediglich von DB2 für OS/390 Version 5 und höher unterstützt.

1. Wählen Sie im Menü **Abfrage** die Option **Benutzerinformationen angeben...** aus. Das Dialogfenster **Benutzerinformationen angeben** wird anschließend angezeigt.
2. Klicken Sie **Ändern...** an. Die Felder **Neues Kennwort** und **Neues Kennwort bestätigen** werden anschließend angezeigt.
3. Geben Sie das neue Kennwort im Feld **Neues Kennwort** und anschließend zur Bestätigung erneut im Feld **Neues Kennwort bestätigen** ein, und klicken Sie dann **OK** an. Das Kennwort für den Datenbank-Server wird geändert.

Kontonamen angeben

Der Datenbank-Server verwendet Namen von Konten (*Accounts*) für die Überwachung der Systemverwendung und -auslastung. Wenden Sie sich an Ihren Datenbankadministrator, um zu erfahren, ob Ihr System mit dieser Funktionalität arbeitet.

1. Wählen Sie im Menü **Abfrage** die Option **Benutzerinformationen angeben...** aus. Das Dialogfenster **Benutzerinformationen angeben** wird anschließend angezeigt.
2. Geben Sie im Feld **Konto** den gewünschten Kontonamen an, und klicken Sie anschließend **OK** an. QMF für Windows speichert die Informationen für den Zugriff auf den Server.

Steuerung mit dem Governor

Der QMF für Windows-Governor läuft ständig im Hintergrund. Dieses Steuerungsprogramm überwacht die Verwendung von Datenbanken und Systemressourcen. Der Governor beschränkt ebenfalls die Art und die Größe der Abfragen, die Sie ausführen können.

Ressourcengrenzwerte anzeigen

Wählen Sie im Menü **Ansicht** die Option **Ressourcengrenzwerte...** aus. Das Dialogfenster **Ressourcengrenzwerte** wird angezeigt. Auf die Informationen in diesem Dialogfenster Ressourcengrenzwerte besteht lediglich Lesezugriff. Diese Grenzwerte werden von Ihrem Systemadministrator festgelegt.



Folgende Grenzwertarten und Steuerelemente können aktiv sein:

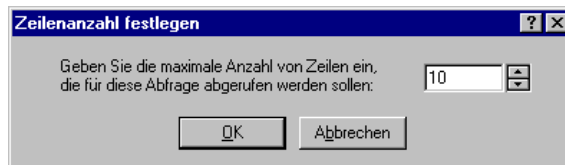
- Zeitlimits
- Grenzwerte
- SQL-Verben
- Optionen
- Daten speichern
- Binden
- Objektüberwachung

Eigenen Grenzwert für die Zeilenanzahl festlegen

Sie können die maximale Anzahl für die bei einer Abfrage abzurufenden Zeilen angeben. Wenn dieser Grenzwert erreicht wird, bricht QMF für Windows die Abfrage ab. Der maximal gültige Höchstwert, der in Ihrer Ressourcengrenzwertgruppe angegeben ist, hat Vorrang vor diesem Parameter. Geben Sie den Wert 0 in diesem Feld ein, um keinen Maximalwert anzugeben.

Zeilen, die QMF für Windows bereits in Überschreitung dieses Grenzwerts abgerufen hat, werden beibehalten und sind zur Anzeige verfügbar.

1. Klicken Sie im Menü **Abfrage** die Option **Zeilenanzahl festlegen...** an. Das Dialogfenster **Zeilenanzahl festlegen** wird anschließend angezeigt.



2. Geben Sie an, wieviele Zeilen von einer Abfrage maximal zurückgegeben werden sollen, und klicken Sie dann **OK** an. Dieser Zeilengrenzwert wird bei der nächsten Ausführung der Abfrage angewendet.

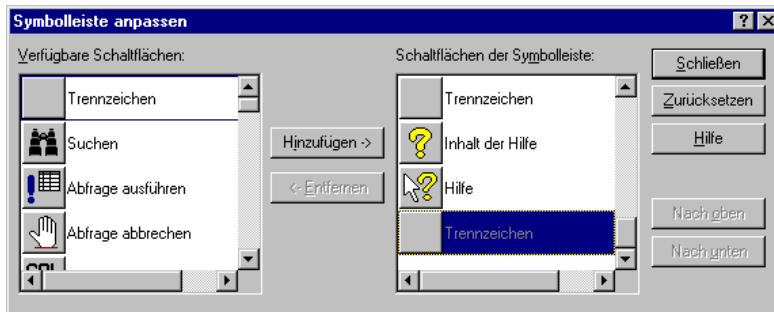
Symbolleiste anpassen

Sie können angeben, welche Knöpfe (Schaltflächen) in der Symbolleiste (Funktionsleiste) angezeigt werden sollen.

Knöpfe zur Symbolleiste hinzufügen

Sie haben die Möglichkeit, neue Knöpfe (Schaltflächen) der vorhandenen Symbolleiste von QMF für Windows hinzuzufügen. Diese zur Aufnahme in die Symbolleiste verfügbaren Schaltflächen stehen für Funktionen, die wahrscheinlich nicht alle Benutzer benötigen werden.

1. Klicken Sie den grauen Bereich um die Symbolleiste doppelt an. Das Dialogfenster **Symbolleiste anpassen** wird anschließend geöffnet.



2. Wählen Sie in der Spalte **Verfügbare Schaltflächen** die Schaltfläche aus, der hinzugefügt werden soll, und klicken Sie anschließend **Hinzufügen ->** an. Die Schaltfläche wird der Symbolleiste hinzugefügt.
3. Wenn Sie keine weiteren Schaltflächen mehr hinzufügen wollen, klicken Sie **Schließen** an. Das Dialogfenster wird geschlossen, und die neuen Schaltflächen werden der Symbolleiste hinzugefügt.

Knöpfe in der Symbolleiste verschieben

Sie haben die Möglichkeit, die Knöpfe (Schaltflächen) in der Symbolleiste von QMF für Windows neu anzuordnen.

1. Klicken Sie hierzu den grauen Bereich um die Symbolleiste doppelt an. Das Dialogfenster **Symbolleiste anpassen** wird anschließend geöffnet.
2. Wählen Sie in der Spalte **Verfügbare Schaltflächen** die Schaltfläche aus, die verschoben werden soll.
3. Verschieben Sie die Schaltflächen in der Symbolleiste mit Hilfe der Druckknöpfe **Nach oben** bzw. **Nach unten**.
4. Wenn Sie keine weiteren Schaltflächen mehr verschieben wollen, klicken Sie **Schließen** an. Das Dialogfenster wird geschlossen, und die Schaltflächen werden in ihren neuen Positionen angezeigt.

Knöpfe aus der Symbolleiste entfernen

Sie haben die Möglichkeit, Knöpfe (Schaltflächen) aus der Symbolleiste von QMF für Windows zu entfernen.

1. Klicken Sie hierzu den grauen Bereich um die Symbolleiste doppelt an. Das Dialogfenster **Symbolleiste anpassen** wird anschließend geöffnet.
2. Wählen Sie in der Spalte **Verfügbare Schaltflächen** die Schaltfläche aus, die entfernt werden soll, und klicken Sie anschließend **<- Entfernen** an. Die Schaltfläche wird aus der Symbolleiste entfernt.
3. Wenn Sie keine weiteren Schaltflächen mehr entfernen wollen, klicken Sie **Schließen** an. Das Dialogfenster wird geschlossen, und die Schaltflächen werden aus der Symbolleiste entfernt.

Kapitel 2. Mit SQL-Abfragen arbeiten

SQL (*Structured Query Language*) ist die grundlegendste Schnittstelle zwischen einem Benutzer und einer Datenbank. Abfragen werden in SQL geschrieben und von der Datenbank verarbeitet. Benutzer können QMF für Windows-Abfragen in SQL schreiben oder Abfragen mit Hilfe von Mausclicks erstellen.

SQL-Abfragen

Für SQL-Abfragen sind Kenntnisse der SQL-Befehle und der SQL-Syntax erforderlich. Benutzer, die über keine SQL-Kenntnisse verfügen, sollten menügesteuerte Abfragen erstellen.

Neue SQL-Abfragen erstellen

Klicken Sie in der Symbolleiste den Knopf **Neue SQL-Abfrage** an.



Ein neues Abfragedokument wird anschließend geöffnet.

SQL-Abfragen auf einem Datenbank-Server ausführen

1. Öffnen Sie ein neues Abfragedokument, und geben Sie eine Abfrage ein. Sie können auch eine vorhandene Abfragedatei oder eine Abfrage in der Datenbank öffnen.
2. Klicken Sie in der Symbolleiste den Knopf **Abfrage ausführen** an.



Die Abfrage wird ausgeführt, und die Ergebnisse werden angezeigt.

Zwischen der Ergebnissicht und der SQL-Sicht umschalten

Sie können entweder die Ergebnisse einer Abfrage oder die entsprechende SQL-Anweisung dafür anzeigen.

Klicken Sie in der **SQL-Sicht** einer ausgeführten Abfrage in der Symbolleiste den Knopf **Ergebnisse anzeigen** an.



Die Abfrageergebnisse werden anschließend angezeigt.

-oder-

Klicken Sie in der Ergebnissicht einer Abfrage den Knopf **SQL anzeigen** an.



Die SQL-Anweisung wird angezeigt.

Mit Schriftarten arbeiten

Sie können die Schriftart ändern, die für die Anzeige von Abfragen verwendet wird. Die jeweiligen Möglichkeiten bei der Auswahl der Schriftart hängen davon ab, welche Schriftarten auf Ihrem Computer installiert wurden. Informationen zum Hinzufügen neuer Schriftarten sind in der Hilfefunktion Ihres Betriebssystems enthalten.

Anmerkung: Wenn Sie die Abfrage nach der Auswahl einer neuen Schriftart zur Anzeige der Abfrage speichern, wird diese Abfrage zukünftig immer unter Verwendung der neuen Schriftarte angezeigt.

Schriftart zur Anzeige der Abfrage auswählen

1. Klicken Sie in der **SQL-Sicht** im Menü **Abfrage** die Option **Schriftart festlegen...** an. Das Dialogfenster **Schriftart** wird anschließend angezeigt.
2. Wählen Sie die Schriftart aus, in der der Text der Abfrage angezeigt werden soll, und klicken Sie dann **OK** an. Die Abfrage wird in der neuen Schriftart dargestellt.

Anmerkung: Wenn Sie diese Schriftart als Standardeinstellung für alle neuen Abfragen verwenden wollen, klicken Sie **Als Standard festlegen** an.

Mehrfachabfragen

Sie können gleichzeitig mehrere Abfragedokumente geöffnet haben. Sie können ebenfalls gleichzeitig mehr als eine Abfrage ausführen. Diese Funktion können sie verwenden, um mehrere Berichte zu generieren oder um SQL-Text in einer Abfrage auszuschneiden bzw. zu kopieren und dann in eine andere Abfrage einzufügen.

Mehrere Abfragen gleichzeitig anzeigen

1. Öffnen Sie mindestens zwei Abfragedokumente.

2. Wählen Sie im Menü **Fenster** einen der folgenden Befehle aus:

Befehl	Ergebnis
Überlappend	Abfragen werden gestaffelt angezeigt.
Horizontal anordnen	Abfragen werden nebeneinander in der gleichen Fenstergröße angezeigt.
Vertikal anordnen	Abfragen werden übereinander in der gleichen Fenstergröße angezeigt.

Die Abfragefenster werden entsprechend der ausgewählten Option angeordnet.

Abfragen zeigen (erstellen)

Verwenden Sie den Befehl **Abfrage zeigen...**, um neue SQL-Abfragedokumente zu erstellen. Sie müssen lediglich einen oder mehrere Tabellennamen und die Art der SQL-Anweisung angeben. QMF für Windows erstellt dann automatisch eine SQL-Anweisung, die auf die Namen und Datentypen in den Spalten der Tabelle verweist.

Neue SQL-Abfragen erstellen

1. Klicken Sie hierzu im Menü **Datei** die Option **Abfrage zeigen...** an. Das Dialogfenster **Abfrage zeigen** wird anschließend angezeigt.

The screenshot shows the 'Abfrage zeigen' dialog box. It features a title bar with a question mark and a close button. The main area includes a 'Server:' dropdown menu, 'OK' and 'Abbrechen' buttons, and a section for 'Abfrageart' with radio buttons for 'Auswählen (Select)', 'Aktualisieren (Update)', and 'Einfügen (Insert)'. Below this is the 'Abfragetabellen' section with input fields for 'Eigner', 'Name', and 'Kennung', and buttons for 'Einzufügen' and 'Von Liste hinzufügen...'. At the bottom is a table with columns 'Eigner', 'Name', and 'Kennung', and buttons for 'Nach oben bewegen', 'Nach unten bewegen', and 'Entfernen'.

2. Wählen Sie die Art der zu erstellenden Abfrage aus:

Abfrageart	Ergebnis
Auswählen (Select)	Zeilen aus einer oder mehreren Tabellen abrufen.
Aktualisieren (Update)	Die in einer Tabelle enthaltenen Informationen ändern.
Einfügen (Insert)	Einer Tabelle neue Zeilen hinzufügen.

3. Geben Sie den Eigner und den Namen der abzufragenden Tabelle an.

Anmerkung: Sie können mit Hilfe von Mustern Tabellennamen aus einer Liste mit entsprechenden Tabellen auswählen.

- Das Prozentzeichen wird als Platzhalter für eine Zeichenfolge beliebiger Länge verwendet, die wiederum aus beliebigen Zeichen besteht. Um beispielsweise alle Tabellen aufzulisten, deren Namen mit dem Buchstaben A beginnen, geben Sie A% ein.
- Das Unterstreichungszeichen (_) wird verwendet, um ein beliebiges einzelnes Zeichen darzustellen. Um beispielsweise alle Tabellen aufzulisten, die einen Eigner haben, der den Buchstaben A an der zweiten Stelle aufweist, geben Sie _A% ein.

Nachdem Sie ein Muster eingegeben haben, klicken Sie **Von Liste hinzufügen...** an, und wählen Sie eine Tabelle aus der nun angezeigten Liste aus.

4. Geben Sie für die Tabelle eine eindeutige Kennung ein.
5. Klicken Sie **Hinzufügen** an. Die Tabelle wird der Abfrage hinzugefügt.
6. Wenn Sie die Tabelle(n) hinzugefügt haben, die Sie abfragen wollen, klicken Sie **OK** an. Für die ausgewählten Tabellen wird eine SQL-Abfrage erstellt und angezeigt.

Substitutionsvariablen in SQL-Abfragen

Sie können mit ein und derselben Abfrage unterschiedliche Informationen abrufen, indem Sie Substitutionsvariablen verwenden und bei jeder Ausführung der Abfrage unterschiedliche Werte für die Variablen angeben. So müssen Sie zum Abrufen unterschiedlicher Informationen die Abfrage nicht erneut schreiben. Sie müssen lediglich andere Werte für die Substitutionsvariablen in der Abfrage angeben, wenn Sie sie ausführen.

Eine Substitutionsvariable besteht aus Text, den Sie in eine Abfrage integrieren können. Sie muss mit dem Et-Zeichen (&) beginnen und kann bis zu 18 Zeichen enthalten; dabei kann es sich um alphabetische oder numerische Zeichen oder um folgende Sonderzeichen handeln: ^ ! \$ % & ' { } ? @ # % \ oder _. Folgende Werte sind Beispiele für gültige Substitutionsvariablen:

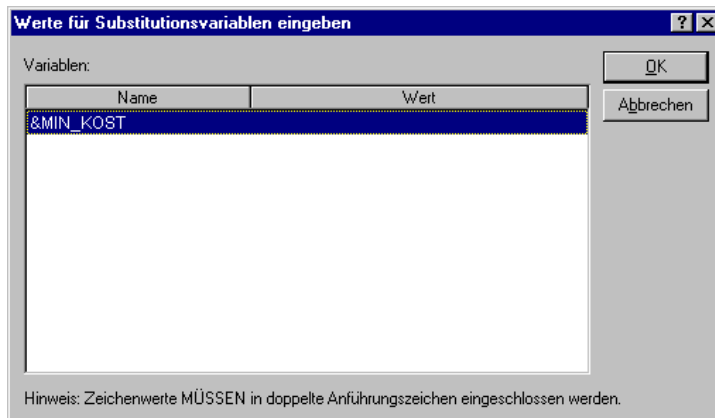
```
&VARIABLE1  
&KOSTENSTELLE_NUMMER
```

Eine Substitutionsvariable kann sich an einer beliebigen Stelle in einer Abfrage befinden und jeden Wert annehmen, der in einer Abfrage verwendet werden kann (mit Ausnahme von Kommentaren). So können Sie beispielsweise eine Substitutionsvariable anstelle eines Spaltennamens, einer Suchbedingung, einer Unterabfrage oder eines beliebigen Werts verwenden.

SQL-Abfragen mit Substitutionsvariablen ausführen

1. Öffnen Sie ein neues Abfragedokument, und geben Sie folgende SQL-Anweisung ein:

```
SELECT * FROM Q.PERSONAL WHERE KOST >= &MIN_KOST
```
2. Führen Sie die Abfrage aus. Das Dialogfenster **Werte für Substitutionsvariablen eingeben** wird anschließend geöffnet.



3. Geben Sie im Feld **Wert** den Wert 50 ein, und klicken Sie **OK** an. Die Abfrage wird ausgeführt, und die Abfrageergebnisse werden anschließend angezeigt.

Sie können in den Klauseln SELECT und FROM Werte ersetzen und auf diese Weise mit Substitutionsvariablen experimentieren. Führen Sie Ihre Abfragen einmal mit verschiedenen Eingaben aus, und betrachten Sie die unterschiedlichen Ergebnisse.

SQL-Abfragen speichern und öffnen

Sie können Abfragen in Dateien auf Ihrem PC, auf einem Datei-Server oder auf einem Datenbank-Server speichern.

SQL-Abfragen in Dateien speichern

1. Klicken Sie in einer geöffneten Abfrage den Knopf **Speichern** in der Symbolleiste an.



Wurde die Abfrage zuvor bereits gespeichert, wird sie jetzt erneut gespeichert. Wird die Abfrage jetzt zum ersten Mal gespeichert, wird das Dialogfenster **Speichern unter** geöffnet.

2. Geben Sie den Dateinamen ein, unter dem die Abfrage gespeichert werden soll, und klicken Sie **Speichern** an. Die Abfrage wird anschließend gespeichert.

Gespeicherte SQL-Abfragedateien öffnen

1. Klicken Sie den Knopf **Öffnen** in der Symbolleiste an.



Das Dialogfenster **Öffnen** wird anschließend angezeigt.

2. Wählen Sie die zu öffnende Datei aus, und klicken Sie dann **OK** an. Die ausgewählte Abfrage wird in einem neuen Abfragedokument geöffnet.

SQL-Abfragen auf dem Datenbank-Server speichern

Auf dem Server gespeicherte Abfragen, können für andere Benutzer verfügbar gemacht werden. Wenn Sie Ihre Abfragen zusammen mit anderen Benutzern gemeinsam verwenden wollen, speichern Sie sie auf dem Datenbank-Server.

1. Klicken Sie in einer geöffneten Abfrage den Knopf **Auf Server speichern** in der Symbolleiste an.



Das Dialogfenster **Abfrage auf <server_name> speichern** wird anschließend angezeigt.



2. Geben Sie einen Eigner sowie einen Namen ein, und wählen Sie aus, ob die gespeicherte Abfrage zusammen mit anderen Benutzern gemeinsam verwendet werden soll; klicken Sie anschließend **OK** an. Die Abfrage wird auf dem Server gespeichert.

Ist bereits eine Abfrage mit diesem Namen vorhanden, werden Sie aufgefordert, das Überschreiben der bereits vorhandenen Abfrage zu bestätigen.

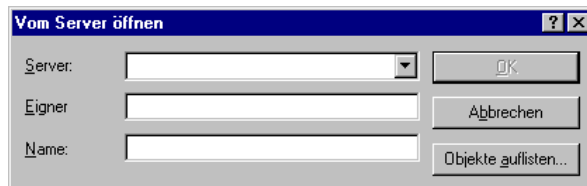
Gespeicherte SQL-Abfragen auf dem Datenbank-Server öffnen

Sie können Abfragen, die auf dem Datenbank-Server gespeichert wurden, öffnen.

1. Klicken Sie hierzu den Knopf **Vom Server öffnen** in der Symbolleiste an.



Das Dialogfenster **Vom Server öffnen** wird anschließend angezeigt.



2. Geben Sie einen Server, einen Eigner und einen Namen ein; klicken Sie anschließend **OK** an. Die SQL-Abfrage wird geöffnet.

SQL-Abfragen drucken

Sie können Ihre SQL-Abfragen vorab anzeigen und drucken.

Abfrage vorab anzeigen

1. Öffnen Sie eine Abfrage und wechseln Sie in die SQL-Sicht. Die SQL-Anweisung wird angezeigt.
2. Klicken Sie im Menü **Datei** die Option **Seite einrichten...** an. Das Dialogfenster **Seite einrichten** wird anschließend angezeigt.
3. Nehmen Sie am Layout der Seite alle gewünschten Änderungen vor, und klicken Sie dann **OK** an.
4. Klicken Sie in der Symbolleiste den Knopf **Druckvorschau...** an.



Es wird eine Voranzeige der zu druckenden Abfrage angezeigt.

SQL-Abfragen drucken

1. Öffnen Sie eine Abfrage und wechseln Sie in die SQL-Sicht. Die SQL-Anweisung wird angezeigt.
2. Klicken Sie im Menü **Datei** die Option **Seite einrichten...** an. Das Dialogfenster **Seite einrichten** wird anschließend angezeigt.
3. Nehmen Sie am Layout der Seite alle gewünschten Änderungen vor, und klicken Sie dann **OK** an.
4. Klicken Sie den Knopf **Drucken** in der Symbolleiste an.



Die Abfrage wird gedruckt.

Kapitel 3. Mit menügesteuerten Abfragen arbeiten

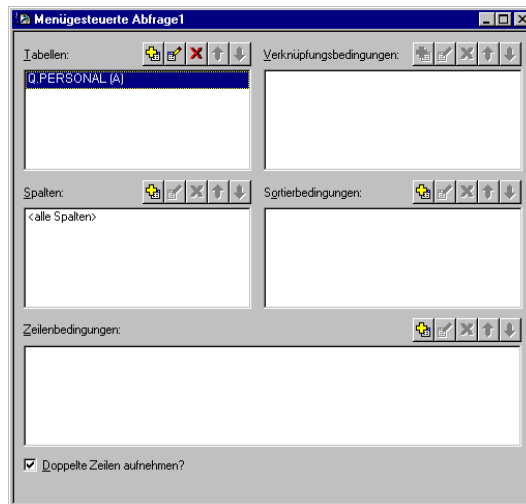
Mit menügesteuerten Abfragen können Sie auf einfache Weise eine Abfrage erstellen, indem Sie in Menüs und Listen entsprechende Optionen auswählen. Nachdem Sie eine menügesteuerte Abfrage erstellt haben, können Sie sie speichern oder sie in eine SQL-Abfrage umwandeln.

Einfache Abfragen erstellen

Sie können einfache Abfragen erstellen, indem Sie die Schnittstelle für menügesteuerte Abfragen verwenden.





Neue menügesteuerte Abfragen öffnen

- Klicken Sie hierzu im Menü **Datei** die Option **Neu** und anschließend **Menügesteuerte Abfrage** an. Ein neues Dokument für eine menügesteuerte Abfrage wird anschließend geöffnet.



Aktionsknöpfe für menügesteuerte Abfragen

Menügesteuerte Abfragen werden mit Hilfe der entsprechenden Aktionsknöpfe bearbeitet. Die Knöpfe werden jeweils über den einzelnen Abschnitten angezeigt, für deren Steuerung sie verwendet werden können.

Aktionsknopf für menügesteuerte Abfragen	Darstellung	Ergebnis
Hinzufügen		Klicken Sie diesen Knopf an, um der menügesteuerten Abfrage ein Element hinzuzufügen.
Bearbeiten		Klicken Sie diesen Knopf an, um das hervorgehobene Element in der Abfrage zu bearbeiten.
Löschen		Klicken Sie diesen Knopf an, um das ausgewählte Element zu löschen.
Nach oben bewegen + Nach unten bewegen		Klicken Sie diese Knöpfe an, um das ausgewählte Element in der Abfrage nach oben oder nach unten zu bewegen.

Menügesteuerten Abfragen Tabellen hinzufügen

1. Klicken Sie im Abschnitt **Tabellen** des Dokuments mit der menügesteuerte Abfrage den Knopf **Hinzufügen** an.



Das Dialogfenster **Tabellen** wird anschließend angezeigt.



2. Geben Sie den Eigner und den Namen der hinzuzufügenden Tabelle an, und klicken Sie anschließend den Knopf **Hinzufügen** an. Die Tabelle wird der Abfrage hinzugefügt.

Anmerkung: Sie können Objekte aus einer Liste mit entsprechenden Objekten mit Hilfe von Mustern auswählen.

- Das Prozentzeichen (%) wird als Platzhalter für eine Zeichenfolge beliebiger Länge verwendet, die wiederum aus beliebigen Zeichen besteht. Um beispielsweise alle Tabellen aufzulisten, deren Namen mit dem Buchstaben A beginnen, geben Sie A% ein.
- Das Unterstrichungszeichen (_) wird verwendet, um ein beliebiges einzelnes Zeichen darzustellen. Um beispielsweise alle Tabellen aufzulisten, die einen Ersteller haben, der den Buchstaben A an der zweiten Stelle aufweist, geben Sie _A% ein.

Nachdem Sie ein Muster eingegeben haben, klicken Sie **Von Liste hinzufügen...** an, und wählen Sie eine Tabelle aus der nun angezeigten Liste aus.

3. Fügen Sie der Abfrage gegebenenfalls zusätzliche Tabellenbedingungen hinzu, und klicken Sie **Schließen** an. Im Dokument mit der menügesteuerten Abfrage werden nun die neuen Tabellen aufgelistet.

Menügesteuerte Abfragen ausführen

Diese Abfragen werden auf die gleiche Weise ausgeführt, wie SQL-Abfragen. Klicken Sie hierzu den Knopf **Abfrage ausführen** in der Symbolleiste an.



Die menügesteuerte Abfrage wird ausgeführt.

Komplexe Abfragen erstellen

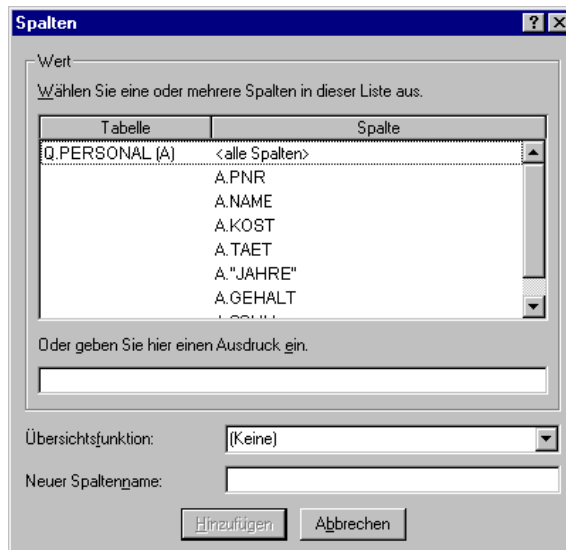
Sie können auch komplexere Abfragen erstellen, indem Sie die Schnittstelle für menügesteuerte Abfragen verwenden.

Menügesteuerten Abfragen Spalten hinzufügen

1. Klicken Sie im Abschnitt **Spalten** des Dokuments mit der menügesteuerte Abfrage den Knopf **Hinzufügen** an.



Das Dialogfenster **Spalten** wird angezeigt.



2. Wählen Sie die gewünschte Spalte aus, und klicken Sie **Hinzufügen** an. Die Spalte wird der menügesteuerten Abfrage hinzugefügt.
3. Fügen Sie der Abfrage gegebenenfalls zusätzliche Spalten hinzu, und klicken Sie **Schließen** an. Im Dokument mit der menügesteuerten Abfrage werden nun die neuen Spalten aufgelistet.

Anmerkung: Sie können eine Übersichtsfunktion der Spalte hinzufügen, indem Sie eine im Feld **Übersichtsfunktion** auswählen. Verfügbare Übersichtsfunktionen sind: MITTEL, ANZAHL, MAXIMUM, MINIMUM und SUMME.

Anmerkung: Sie können eine Spalte in der Abfrage umbenennen, indem Sie einen neuen Spaltennamen im Feld **Neuer Spaltenname** eingeben.

Mit Sortierbedingungen arbeiten

Mit Sortierbedingungen können Sie angeben, auf welche Weise die Zeilen in einer Abfrage sortiert werden sollen. Zeilen können in aufsteigender (A-Z) oder absteigender (Z-A) Reihenfolge sortiert werden.

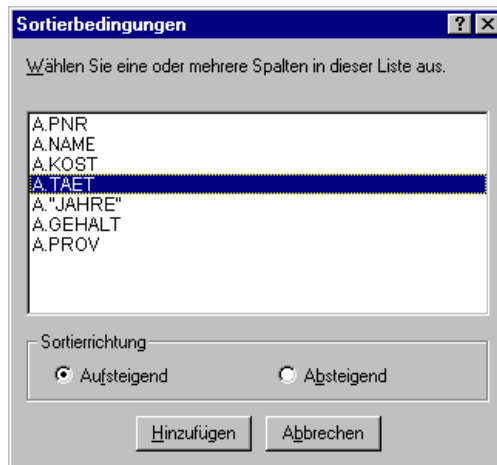
Wenn Sie Zeilen nach mehr als einer Spalte sortieren, wird die erste Spalte zuerst sortiert, die zweite Spalte wird innerhalb der Reihenfolge der ersten Spalte sortiert, usw.

Sortierbedingungen hinzufügen

1. Klicken Sie im Abschnitt **Sortierbedingungen** des Dokuments mit der menügesteuerte Abfrage den Knopf **Hinzufügen** an.



Das Dialogfenster **Sortierbedingungen** wird angezeigt.



2. Wählen Sie die Spalte, die sortiert werden soll, sowie die Sortierreihenfolge aus, und klicken Sie anschließend **Hinzufügen** an. Die Sortierbedingung wird der menügesteuerten Abfrage hinzugefügt.
3. Fügen Sie der Abfrage gegebenenfalls zusätzliche Sortierbedingungen hinzu, und klicken Sie **Schließen** an. Im Dokument mit der menügesteuerten Abfrage werden nun die neuen Sortierbedingungen aufgelistet.

Mit Zeilenbedingungen arbeiten

Oft wollen Sie lediglich nur bestimmte Zeilen in einer Tabelle anzeigen. Um bestimmte Zeilen zur Anzeige auszuwählen, fügen Sie der menügesteuerten Abfrage Zeilenbedingungen hinzu. Wenn Sie keine Zeilenbedingungen verwenden, werden alle in der Tabelle enthaltenen Zeilen angezeigt.

Die folgenden Zeilenbedingungen sind verfügbar:

- Gleich
- Kleiner als
- Kleiner als oder gleich
- Größer als
- Größer als oder gleich
- Zwischen
- Beginnend mit
- Endend mit
- Enthaltend
- NULL

Zeilenbedingungen werden von den folgenden Operatoren gesteuert:

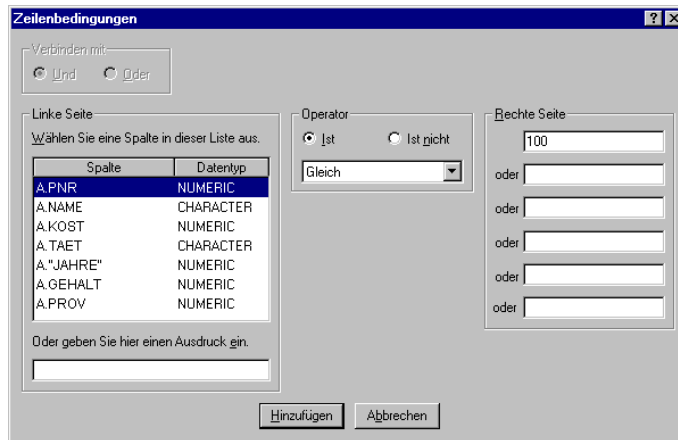
- Ist
- Ist nicht

Zeilenbedingungen hinzufügen

1. Klicken Sie im Abschnitt **Zeilenbedingungen** des Dokuments mit der menügesteuerte Abfrage den Knopf **Hinzufügen** an.



Das Dialogfenster **Zeilenbedingungen** wird angezeigt.



2. Wählen Sie die Abschnitte der bedingten Anweisung aus, und klicken Sie **Hinzufügen** an.

Teil der Zeilenbedingung	Funktion
Linke Seite	Wählen Sie die zu untersuchende Spalte aus.
Operator	Bestimmen Sie die Abhängigkeit zwischen der linken und der rechten Seite der Zeile.
Rechte Seite	Geben Sie die zu überprüfende Bedingung ein.

Die Zeilenbedingung wird der menügesteuerten Abfrage hinzugefügt.

3. Fügen Sie der Abfrage gegebenenfalls zusätzliche Zeilenbedingungen hinzu, und klicken Sie **Schließen** an. Im Dokument mit der menügesteuerten Abfrage werden nun die neuen Zeilenbedingungen aufgelistet.

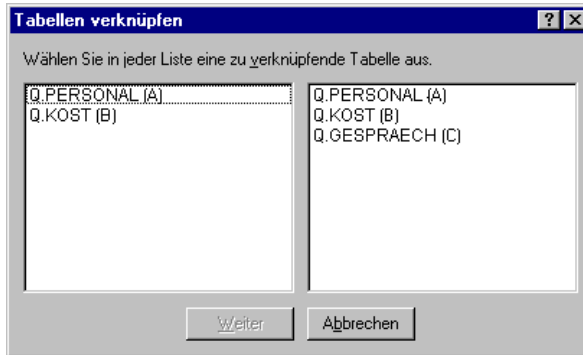
Mehrere Tabellen in menügesteuerten Abfragen verwenden

Sie können in einer menügesteuerten Abfrage Informationen aus mehreren Tabellen einschließen.

Sie müssen eine Verbindung zwischen den beiden Tabellen herstellen und in beiden jeweils eine oder mehrere "Verknüpfungsbedingungen" angeben. Bei den Ergebnissen werden nur Zeilen von den Tabellen berücksichtigt, in denen die Verknüpfungsspalten identisch sind. In einer Verknüpfungsbedingung muss der Datentyp aller Spalten identisch sein. Wenn Sie eine Beziehung zwischen zwei Spalten angegeben haben, "merkt sich" QMF für Windows diese Beziehung und schlägt sie in zukünftigen Abfragen vor. Auf diese Weise wird das Erstellen zukünftiger Abfragen einfacher und effizienter.

Verknüpfungsbedingungen für menügesteuerte Abfragen erstellen

1. Klicken Sie im Abschnitt **Tabellen** des Dokuments mit der menügesteuerte Abfrage den Knopf **Hinzufügen an**, und fügen Sie mindestens zwei Tabellen hinzu. Wenn die Tabellen zuvor noch nie verknüpft waren, wird das Dialogfenster **Tabellen verknüpfen** angezeigt. Wenn die Tabellen bereits zuvor verknüpft wurden, schlägt QMF für Windows die zuvor verwendete Verknüpfungsbedingung vor.



2. Wählen Sie aus jeder Tabelle eine Spalte mit demselben Datentyp und übereinstimmenden Daten aus, und klicken Sie **Weiter** an. In der menügesteuerten Abfrage werden die neuen Verknüpfungsbedingungen angezeigt.

Menügesteuerte Abfragen und SQL

Sie können die Schnittstelle für menügesteuerte Abfragen verwenden, um SQL zu erlernen.

SQL für menügesteuerte Abfragen anzeigen

Klicken Sie in der Sicht mit der menügesteuerte Abfrage den Knopf **SQL anzeigen** in der Symbolleiste an.



Die entsprechende SQL-Anweisung für die menügesteuerte Abfrage wird angezeigt. In dieser Sicht können Sie die SQL-Anweisung nicht ändern.

Menügesteuerte Abfragen in SQL konvertieren

Sie können eine menügesteuerte Abfrage in ein neues SQL-Abfragedokument konvertieren. Die neue SQL-Abfrage kann geändert, gespeichert, gedruckt und ausgeführt werden. Klicken Sie im Menü **Abfrage** die Option **In SQL konvertieren** an. Die Abfrage wird in ein neues SQL-Abfragedokument konvertiert.

Substitutionsvariablen in menügesteuerten Abfragen verwenden

Substitutionsvariablen können in menügesteuerten Abfragen auf die gleiche Weise verwendet werden, wie in SQL-Abfragen. Siehe hierzu den Abschnitt "Substitutionsvariablen in SQL-Abfragen".

Substitutionsvariablen können beispielsweise wie folgt verwendet werden:

- In einer Zeilenbedingung
KOST ist größer als oder gleich &MIN_KOST
- In einer Spaltenspezifikation
&EingabeNum

Menügesteuerte Abfragen speichern

Menügesteuerte Abfragen können in Dateien auf Ihrem PC bzw. auf einem Datei-Server oder auf einem Datenbank-Server gespeichert werden.

Menügesteuerte Abfragen in Dateien speichern

1. Klicken Sie in einer geöffneten menügesteuerten Abfrage den Knopf **Speichern** in der Symbolleiste an.



Anmerkung: Wurde die Abfrage zuvor bereits gespeichert, wird sie jetzt erneut gespeichert. Wird die Abfrage jetzt zum ersten Mal gespeichert, wird das Dialogfenster **Speichern unter** geöffnet.

2. Geben Sie den Dateinamen ein, unter dem die menügesteuerte Abfrage gespeichert werden soll, und klicken Sie **Speichern** an. Die Abfrage wird gespeichert.

Menügesteuerte Abfragen öffnen, die in Dateien gespeichert sind

1. Klicken Sie den Knopf **Öffnen** in der Symbolleiste an.



Das Dialogfenster **Öffnen** wird anschließend angezeigt.

2. Wählen Sie die zu öffnende Datei aus, und klicken Sie dann **OK** an. Die ausgewählte menügesteuerte Abfrage wird in einem neuen Abfragedokument geöffnet.

Menügesteuerte Abfragen auf dem Datenbank-Server speichern

1. Klicken Sie in einer geöffneten menügesteuerten Abfrage den Knopf **Auf Server speichern** in der Symbolleiste an.



Das Dialogfenster **Abfrage auf <server_name> speichern** wird anschließend angezeigt.

2. Geben Sie einen Eigner sowie einen Namen ein, und wählen Sie aus, ob die gespeicherte Abfrage zusammen mit anderen Benutzern gemeinsam verwendet werden soll; klicken Sie anschließend **OK** an. Die Abfrage wird auf dem Server gespeichert.

Ist bereits eine Abfrage mit diesem Namen vorhanden, werden Sie aufgefordert, das Überschreiben der bereits vorhandenen Abfrage zu bestätigen.

Gespeicherte menügesteuerte Abfragen auf dem Datenbank-Server öffnen

Sie können menügesteuerte Abfragen, die auf dem Datenbank-Server gespeichert wurden, öffnen.

1. Klicken Sie hierzu den Knopf **Vom Server öffnen** in der Symbolleiste an.



Das Dialogfenster **Vom Server öffnen** wird anschließend angezeigt.

2. Geben Sie einen Server, einen Eigner und einen Namen ein; klicken Sie anschließend **OK** an. Die menügesteuerte Abfrage wird geöffnet.

Menügesteuerte Abfragen drucken

Sie können Ihre menügesteuerte Abfrage drucken. Sie können auch den SQL-Text einer menügesteuerten Abfrage drucken. Siehe hierzu „SQL-Abfragen drucken“ auf Seite 16.

Menügesteuerte Abfragen vorab anzeigen

Sie können die Ergebnisse oder den Text einer menügesteuerten Abfrage vor dem Drucken anzeigen.

1. Öffnen Sie eine Abfrage und, wechseln Sie in die menügesteuerte Sicht. Die Abfrage wird angezeigt.
2. Klicken Sie im Menü **Datei** die Option **Seite einrichten...** an. Das Dialogfenster **Seite einrichten** wird anschließend angezeigt.
3. Nehmen Sie am Layout der Seite alle gewünschten Änderungen vor, und klicken Sie dann **OK** an.
4. Klicken Sie den Knopf **Druckvorschau...** in der Symbolleiste an.



Es wird eine Voranzeige der zu druckenden Abfrage angezeigt.

Kapitel 4. Mit Abfrageergebnissen arbeiten

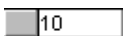



Auf die Abfrageergebnisse können Formatierungs-, Gruppierungs- und Berechnungsfunktionen direkt angewendet werden. Diese Formatierung kann zusammen mit der Abfrage gespeichert oder als ein Format exportiert werden.

Abfrageergebnisse sortieren und in der Größe ändern

Benutzer können die aus einer Abfrage resultierenden Ergebnisse auswählen, in der Größe ändern, neu anordnen und sortieren.

Spalten und Zeilen auswählen

Nachdem Sie eine Abfrage ausgeführt haben, können Sie die Steuerelemente in der **Ergebnissicht** verwenden, um Informationen zu bearbeiten und auszuwählen.

Spalten und Zeilenauswahlknöpfe	Darstellung	Funktion
Zeilenauswahlknopf		Klicken Sie diesen Knopf an, um alle Daten in einer Zeile auszuwählen.
Spaltenauswahlknopf		Klicken Sie diesen Knopf an, um alle Daten in einer Spalte auszuwählen.
Zelle		Klicken Sie eine Zelle direkt an, um sie auszuwählen.
Knöpfe für das Blättern zum Anfang und zum Ende		Klicken Sie diese Knöpfe an, um zum Anfang oder Ende von Abfrageergebnissen zu blättern.

Spalten und Zeilen in der Größe ändern

Sie können die Darstellung von Abfrageergebnissen ändern, indem Sie die entsprechenden Spalten und Zeilen in der Größe ändern.

1. Wählen Sie hierzu mit der Maus die schwarze Trennlinie zwischen zwei Spalten oder zwei Zeilen aus.
2. Ziehen Sie die Trennlinie nach links oder rechts bzw. nach oben oder unten, um die Spalte oder Zeile in der Größe zu ändern.

Anmerkung: Wenn Sie die Abfrage nach dem Ändern der Größe von Spalten oder Zeilen speichern, wird diese zukünftig immer unter Verwendung der neuen Formatierung angezeigt.

Automatisches Anpassen von Zeilen und Spalten

Sie können Spalten und Zeilen automatisch so in der Größe ändern, dass die darin enthaltenen Daten genau in den verfügbaren Platz passen.

Wählen Sie mit der Maus eine gesamte Spalte oder Zeile aus, und klicken Sie die schwarze Trennlinie zwischen dieser Spalte und dem benachbarten Objekt doppelt an. Die Spalte bzw. Zeile wird automatisch so in der Größe geändert, dass die darin enthaltenen Daten genau in den verfügbaren Platz passen.

Anmerkung: Wenn Sie die Abfrage nach dem Ändern der Größe von Spalten oder Zeilen speichern, wird diese zukünftig immer unter Verwendung der neuen Formatierung angezeigt.

Abfrageergebnisse sortieren

Nachdem Sie eine Abfrage ausgeführt haben, können Sie die Ergebnisse alphabetisch nach Spalte sortieren.

Wählen Sie in der Ergebnissicht einer Abfrage eine Spalte aus, und wählen Sie anschließend die Option **Aufsteigend sortieren** im Menü **Ergebnisse** aus.

Die Abfrageergebnisse werden in aufsteigender Reihenfolge sortiert.

-oder-

Wählen Sie in der Ergebnissicht einer Abfrage eine Spalte aus, und wählen Sie anschließend die Option **Absteigend sortieren** im Menü **Ergebnisse** aus.

Die Abfrageergebnisse werden in absteigender Reihenfolge sortiert.

Anmerkung: Um komplexere Sortieroperationen für die ausgewählte Spalte anzuwenden, wählen Sie die Option **Sortieren...** im Menü **Ergebnisse** aus.

Spalten neu anordnen

Sie können die Reihenfolge der Spalten in den Abfrageergebnissen ändern.

Wählen Sie in der Ergebnissicht einer Abfrage eine Spalte aus, und ziehen Sie sie anschließend mit der Maus an ihren neuen Platz.

Die Spalte wird nun in der neuen Reihenfolge angezeigt.

Abfrageergebnisse formatieren

Sie können die Schriftart ändern, die für die Anzeige von Abfragen und Abfrageergebnissen verwendet wird. Die jeweiligen Möglichkeiten bei der Auswahl der Schriftart hängen davon ab, welche Schriftarten auf Ihrem Computer installiert wurden. Informationen zum Hinzufügen neuer Schriftarten sind in der Hilfefunktion Ihres Betriebssystems enthalten.

Anmerkung: Wenn Sie die Abfrage nach der Auswahl einer neuen Schriftart zur Anzeige der Abfrageergebnisse speichern, werden diese Ergebnisse zukünftig immer unter Verwendung der neuen Schriftart angezeigt.

Schriftart zur Anzeige der Abfrageergebnisse auswählen

1. Wählen Sie in der **Ergebnissicht** im Menü **Ergebnisse** die Option **Schriftart festlegen...** aus. Das Dialogfenster **Schriftart** wird anschließend angezeigt.
2. Wählen Sie die Schriftart und die Schriftgröße aus, in der die Abfrageergebnisse angezeigt werden sollen, und klicken Sie dann **OK** an. Die Abfrageergebnisse werden anschließend im von Ihnen angegebenen Format angezeigt.

Anmerkung: Wenn Sie diese Schriftart als Standardeinstellung für alle Abfrageergebnisse verwenden wollen, klicken Sie **Als Standard festlegen** an.

Numerische Abfrageergebnisse formatieren

1. Wählen Sie in der Ergebnissicht eine Spalte aus, die numerische Werte enthält, und wählen Sie anschließend die Option **Format...** im Menü **Ergebnisse** aus. Das Dialogfenster **Format** wird angezeigt.
2. Geben Sie die Art der Formatierung an, die angewendet werden soll, und klicken Sie dann **OK** an. Die Werte werden entsprechend der von Ihnen getroffenen Auswahl formatiert.

Anmerkung: Wenn Sie diese Schriftart als Standardeinstellung für alle Abfrageergebnisse verwenden wollen, klicken Sie **Als Standard festlegen** an.

Formatierungen von Abfrageergebnissen in ein Format konvertieren

Sie können die Formatierungen von Abfrageergebnissen in ein Format konvertieren.

1. Wählen Sie im Menü **Ergebnisse** die Option **Bericht anzeigen...** aus.

Das Dialogfenster **Format auswählen** wird anschließend angezeigt.

2. Wählen Sie **Von Abfrage** aus, und klicken Sie **OK** an.

Die Formatierung der Abfrageergebnisse wird in ein Format konvertiert und in einem neuen Fenster **Format** geöffnet.

Gruppierungs- und Berechnungsfunktionen für Abfrageergebnisse ausführen

Sie können Formatierungsfunktionen für Gruppierungen, Berechnungen und Übersichten für Abfrageergebnisse anwenden.

Abfrageergebnisse gruppieren

Sie können Abfrageergebnisse mit oder ohne Übersichtsinformationen gruppieren.

1. Wählen Sie die zu gruppierende Spalte aus.

2. Wählen Sie im Menü **Ergebnisse** die Art der gewünschten Gruppierungsfunktion aus.

Die Spalte wird entsprechend der von Ihnen getroffenen Auswahl gruppiert.

Abfrageergebnisse in Übersichten zusammenfassen

Sie können Übersichten für Abfrageergebnisse nach Spalte erstellen.

1. Wählen Sie die zu gruppierende Spalte aus.

2. Wählen Sie im Menü **Ergebnisse** die Art der gewünschten Übersichtsfunktion aus.

Die Spalte wird entsprechend der von Ihnen getroffenen Auswahl in einer Übersicht zusammengefasst.

Abfrageergebnisse speichern und Formatierungen ausführen

Sie können Abfrageergebnisse speichern und die Formatierung als ein Format speichern.

Abfrageergebnisse als Tabelle speichern

Sie können Abfrageergebnisse als Tabelle auf einem Datenbank-Server speichern.

1. Wählen Sie im Menü **Ergebnisse** die Option **Daten In Datenbank speichern...** aus.

Das Dialogfenster **Daten speichern** wird anschließend geöffnet.

2. Geben Sie einen Eigner und einen Tabellennamen ein, und klicken Sie dann **OK** an.

Die Abfrageergebnisse werden als Tabelle auf dem Datenbank-Server gespeichert.

Abfrageergebnisse in Dateien speichern

Sie können Abfrageergebnisse in einer Datei auf Ihrem PC oder einem Datei-Server speichern.

1. Wählen Sie im Menü **Ergebnisse** die Option **Daten exportieren...** aus.

Das Dialogfenster **Daten exportieren** wird angezeigt.

2. Geben Sie Exportoptionen sowie die Lokation an, in der Sie die Datei speichern wollen, und klicken Sie anschließend **OK** an.

Die Abfrageergebnisse werden in einer Datei gespeichert.

Abfrageergebnisse drucken

Sie können Ihre Abfrageergebnisse vorab anzeigen und drucken.

Druckvorschau der Abfrageergebnisse vorab anzeigen

1. Öffnen Sie eine Abfrage, und führen Sie sie aus. Die Abfrageergebnisse werden anschließend angezeigt.
2. Wählen Sie im Menü **Datei** die Option **Seite einrichten...** aus. Das Dialogfenster **Seite einrichten** wird anschließend angezeigt.
3. Nehmen Sie am Layout der Seite alle gewünschten Änderungen vor, und klicken Sie dann **OK** an.
4. Klicken Sie den Knopf **Druckvorschau...** in der Symbolleiste an.



Es wird eine Voranzeige der zu druckenden Abfrageergebnisse angezeigt.

Abfrageergebnisse drucken

1. Öffnen Sie eine Abfrage, und wechseln Sie in die Ergebnissicht. Die Abfrageergebnisse werden anschließend angezeigt.
2. Wählen Sie im Menü **Datei** die Option **Seite einrichten...** aus. Das Dialogfenster **Seite einrichten** wird anschließend angezeigt.
3. Nehmen Sie am Layout der Seite alle gewünschten Änderungen vor, und klicken Sie dann **OK** an.
4. Klicken Sie den Knopf **Drucken** in der Symbolleiste an.



Die Abfrageergebnisse werden anschließend gedruckt.

Kapitel 5. Mit Berichten arbeiten

Berichte werden erstellt, indem Abfrageergebnisse mit den Formatierungen aus einem Format kombiniert werden.

Formate

Ein Format ist eine Gruppe an Formatierungsanweisungen, die zum Erstellen, Anzeigen und Drucken von Berichten verwendet werden.

Grundlegendes zu Formaten

Formate setzen sich aus verschiedenen Komponenten zusammen. Diese Komponenten können alle in einem Formatdokument bearbeitet werden.

Haupt Die primären Komponenten eines Formats umfassen Kopfzeilen, Fußzeilen und Gruppenwechsel.

Gruppenwechsel

Kenndaten, Inhalt und Position von bis zu sechs Zeilen mit Zwischensummen in einem Bericht.

Berechnungen

Berechnungsausdrücke für Berichte.

Anmerkung: Auf Ihrer Maschine muss IBM ObjectREXX installiert sein, um Formatberechnungen verwenden zu können.

Spalten

Darstellung und Formatierung von Spalten in einem Bericht. Zu den definierbaren Kenndaten gehören Spaltenreihenfolge, Formatierung, Verwendungscode, Einrückung und Breite.

Bedingungen

Bedingte Formatierungsvorgaben. Sie können beispielsweise angeben, dass im Format keine Zeilen angezeigt werden, die nicht bestimmte Bedingungen erfüllen.

Detailvariationen

Berichtsdetailüberschriften und Haupttextteil. Hier können Sie tabellarische Daten durch Daten ersetzen, die in freier Formatierung vorliegen, um Formatbriefe oder Adreßetiketten zu erstellen.

Schluss

Inhalt und Position des Schlusstexts des Berichts. Sie können beispielsweise auswählen, ob Sie am Ende des Berichts Schlusstext und Ergebnisdaten aufnehmen wollen.

HTML

Inhalt und Position der HTML-Befehle (*Hypertext Markup Language*) und Formatierung in HTML-Berichten.

Angaben

Verschiedene Darstellungsangaben für Ihren Bericht.

Seite Inhalt und Position der Seitenkopf- und Seitenfußzeile in Ihrem Bericht.

Bericht mit Hilfe eines Formats erstellen

Berichte werden erstellt, indem Abfrageergebnisse mit den in einem Format enthaltenen Formatierungsoptionen kombiniert werden. Sie können mehrere Berichte auf der Grundlage eines Abfrageergebnisses erstellen, indem Sie diesen Prozess wiederholen.

1. Klicken Sie in einer Abfrageergebnissicht den Knopf **Bericht anzeigen** in der Symbolleiste an.



Das Dialogfenster **Format auswählen** wird anschließend angezeigt.



2. Abhängig von der Art des Formats, das Sie im Dialogfenster **Format auswählen** anklicken, werden Sie dazu aufgefordert, zusätzliche Informationen anzugeben. Geben Sie die Lokation bzw. den Eigner oder Namen der Datei oder den Dokumenttitel an, und klicken Sie anschließend **OK** an. Der Bericht wird unter Verwendung des ausgewählten Formats und der aktuellen Abfrageergebnisse generiert.

Format bearbeiten

Im Fenster **Format** stehen zahlreiche Optionen zum Bearbeiten und Formatieren von Formaten zur Verfügung.

Rufen Sie von einem geöffneten Format aus das Menü **Format** auf. Im Menü **Format** werden sämtliche Optionen zum Bearbeiten und Formatieren von Formaten aufgelistet. Sämtliche dieser Komponenten können Sie auch durch Anklicken des entsprechenden Knopfs in der Symbolleiste bearbeiten.

Format erstellen

Diese Schritte enthalten Beispieldaten aus der Tabelle Q.PERSONAL. Experimentieren Sie mit den verschiedenen Einstellungen, um Ihre eigenen Formate entsprechend Ihren Vorstellungen und Anforderungen zu erstellen.

Schritt 1: Format erstellen

1. Führen Sie die folgende SQL-Abfrage aus, um die Daten abzurufen, die im Bericht angezeigt werden sollen:

```
SELECT * FROM Q.PERSONAL ORDER BY KOST, NAME
```

Die Abfrageergebnisse werden anschließend angezeigt.

2. Klicken Sie den Knopf **Bericht anzeigen** in der Symbolleiste an. Das Dialogfenster **Format auswählen** wird anschließend angezeigt.
3. Geben Sie an, dass Sie das **Standardformat** verwenden wollen und klicken Sie **OK** an. QMF für Windows zeigt das Standardformat an. Wenn Sie am Standardformat Änderungen vornehmen wollen, klicken Sie einen der Knöpfe für die einzelnen Formatkomponenten in der Symbolleiste an. Für jede Formatkomponente wird ein Knopf in der Formatsymbolleiste angezeigt.

Schritt 2: Spaltenreihenfolge ändern

Im Bericht soll NAME die erste Spalte sein und PNR (für *Personalnummer*) die zweite Spalte. Die Spaltenreihenfolge wird auf dem Format in der Komponente **Spalten** angegeben.

1. Klicken Sie im Menü **Format** die Option **Spalten...** an, um die Registerkarte **Spalten** des Dialogfensters **Format** aufzurufen.
2. Ändern Sie die Reihenfolge einer Spalte, indem Sie den vorhandenen Wert für die Reihenfolge überschreiben. Um NAME zur ersten Spalte des Berichts zu machen, ändern Sie deren Folgenummer (die Spalte in der Liste mit der Bezeichnung **Nr.**) in 1.
3. Um PNR zur zweiten Spalte des Berichts zu machen, ändern Sie deren Folgenummer in 2, und klicken Sie **OK** an. QMF für Windows zeigt den Bericht mit der neuen Spalte im Fenster **Format** an.

Schritt 3: Spaltenüberschriften ändern

Die erste Spaltenüberschrift soll MITARB (für *Mitarbeiter*) und die zweite Spaltenüberschrift soll PROVISION sein. Der Text für Spaltenüberschriften wird in der Komponente **Spalten** des Formats angegeben.

1. Klicken Sie im Menü **Format** die Option **Spalten...** an, um die Registerkarte **Spalten** des Dialogfensters **Format** aufzurufen.

2. Ändern Sie die Spaltenüberschrift, indem Sie den vorhandenen Text für die Spaltenüberschrift überschreiben. Ändern Sie die erste Spaltenüberschrift in MITARB und die letzte Spaltenüberschrift in PROVISION, und klicken Sie **OK** an. QMF für Windows zeigt den Bericht mit den neuen Spaltenüberschriften im Fenster **Format** an.

Schritt 4: Spaltenformatierung ändern

In der Spalte für das GEHALT soll das entsprechende Währungssymbol angezeigt werden. Die Formatierung einer Spalte wird durch ihren Editiercode bestimmt, der in der Komponente **Spalten** des Formats angegeben wird.

1. Klicken Sie im Menü **Format** die Option **Spalten...** an, um die Registerkarte **Spalten** des Dialogfensters **Format** aufzurufen.
2. Ändern Sie den Editiercode der Spalte GEHALT in D2, indem Sie den vorhandenen Editiercode überschreiben, und klicken Sie anschließend **OK** an. QMF für Windows zeigt im Fenster **Format** den Bericht mit dem entsprechenden Währungssymbol in der Spalte GEHALT an.

Schritt 5: Übersichtsinformationen hinzufügen

Der Bericht soll in einzelne Abschnitte für jede Kostenstelle (Abteilung) unterteilt werden. Zusätzlich soll für jede Kostenstelle am Ende jedes Abschnitts die Gesamtsumme für GEHALT und PROVISION angezeigt werden. Dazu muss angegeben werden, wie jede Spalte im Bericht verwendet werden soll. Die Verwendung einer Spalte wird durch ihren Verwendungscode bestimmt, der in der Komponente **Spalten** des Formats angegeben wird.

1. Klicken Sie im Menü **Format** die Option **Spalten...** an, um die Registerkarte **Spalten** des Dialogfensters **Format** aufzurufen.
2. Um den Bericht auf der Basis von KOST in Abschnitte zu unterteilen, ändern Sie den Verwendungscode für KOST in GRUWE1. Verwendungscode, die mit dem Wort GRUWE beginnen, erstellen für die angegebene Spalte einen Gruppenwechsel für einen Abschnitt. Die Zahl, die dem Wort GRUWE nachgestellt ist, bestimmt die Gruppenwechselebene. In einem Bericht werden bis zu sechs Ebenen für Gruppenwechsel unterstützt.
3. Um festzulegen, dass ein Gesamtbetrag für GEHALT und PROVISION für jede KOST aufgenommen werden soll, ändern Sie den Verwendungscode für GEHALT und PROVISION in SUMME.
4. Der Bericht wird leichter verständlich sein, wenn am Ende jedes Gruppenwechsels eines Abschnitts beschreibende Informationen angezeigt werden. Klicken Sie hierzu im Menü **Format** die Option **Gruppenwechsel...** an.
5. Geben Sie auf der Registerkarte **Gruppenwechsel** im Dialogfenster **Format** Fußzeilentext für den Gruppenwechsel an. Geben Sie für die erste Fußzeile für den Gruppenwechsel *Gesamtsumme - Kostenstelle* an, und klicken Sie anschließend **OK** an. QMF für Windows zeigt das Fenster **Format** an.

Schritt 6: Seitenkopfzeilen und -fußzeilen hinzufügen

Dem Bericht sollen Seitenkopfzeilen und -fußzeilen hinzugefügt werden. Seitenkopfzeilen und -fußzeilen werden auf dem Format in der Komponente **Seite** angegeben.

1. Klicken Sie im Menü **Format** die Option **Seite...** an, um die Registerkarte **Seite** des Dialogfensters **Format** aufzurufen.
2. Im oberen Bereich dieses Dialogfensters werden Angaben zur Seitenkopfzeile gemacht. Geben Sie für die erste Zeile der Seitenkopfzeile *Kostenstelle -Bericht* an und für die zweite Zeile *Gehalt und Provision - Gesamt*. Wählen Sie die Ausrichtung für die Seitenkopfzeile aus.
3. Im unteren Bereich dieses Dialogfensters wird die Fußzeile angegeben. Geben Sie für die erste Zeile der Seitenfußzeile *Seitenende* an. Wählen Sie die Ausrichtung für die Fußzeile aus, und klicken Sie **OK** an. QMF für Windows zeigt das Fenster **Format** an.

Formate speichern

Sie können Formate auf Ihrem PC, auf einem Datei-Server oder auf einem Datenbank-Server speichern.

Format in einer Datei speichern

1. Klicken Sie in einem geöffneten Format den Knopf **Speichern** in der Symbolleiste an.
2. Wenn das Format bereits zuvor gespeichert wurde, wählen Sie **Speichern** aus. Wird das Format jetzt zum ersten Mal gespeichert, wird das Dialogfenster **Speichern unter** geöffnet.
3. Geben Sie den Dateinamen ein, unter dem das Format gespeichert werden soll, und klicken Sie **Speichern** an. Das Format wird anschließend gespeichert.

Gespeicherte Formatdateien öffnen

1. Klicken Sie den Knopf **Öffnen** in der Symbolleiste an.



Das Dialogfenster **Öffnen** wird anschließend angezeigt.

2. Wählen Sie die zu öffnende Datei aus, und klicken Sie dann **OK** an. Das ausgewählte Format wird in einem neuen Formatdokument geöffnet.

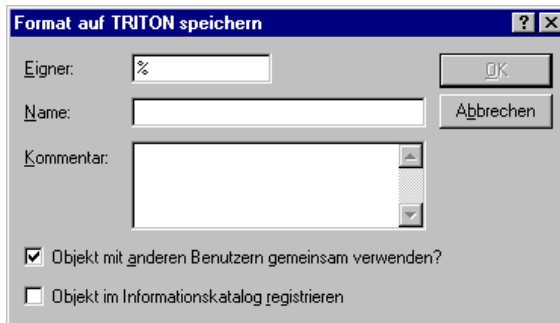
Formate auf dem Datenbank-Server speichern

Auf dem Server gespeicherte Formate, können für andere Benutzer verfügbar gemacht werden. Wenn Sie Ihre Formate zusammen mit anderen Benutzern gemeinsam verwenden wollen, speichern Sie sie auf dem Datenbank-Server.

1. Klicken Sie in einem geöffneten Format den Knopf **Auf Server speichern** in der Symbolleiste an.



Das Dialogfenster **Format auf <server_name> speichern** wird anschließend angezeigt.



2. Geben Sie einen Eigner sowie einen Namen ein, und wählen Sie aus, ob das gespeicherte Format zusammen mit anderen Benutzern gemeinsam verwendet werden soll; klicken Sie anschließend **OK** an. Das Format wird auf dem Server gespeichert.

Ist bereits ein Format mit diesem Namen vorhanden, werden Sie aufgefordert, das Überschreiben des bereits vorhandenen Formats zu bestätigen.

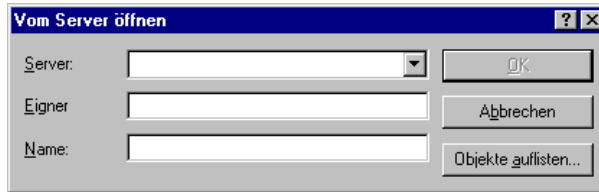
Formate auf dem Datenbank-Server öffnen

Sie können Formate, die auf dem Datenbank-Server gespeichert wurden, öffnen.

1. Klicken Sie hierzu den Knopf **Vom Server öffnen** in der Symbolleiste an.



Das Dialogfenster **Vom Server öffnen** wird anschließend angezeigt.



2. Geben Sie einen Server, einen Eigner und einen Namen ein; klicken Sie anschließend **OK** an. Das Format wird geöffnet.

Berichte drucken

Sie können Berichte drucken.

1. Öffnen Sie hierzu ein Format, und klicken Sie anschließend **Seite einrichten...** an.
2. Nehmen Sie am Layout der Seite alle gewünschten Änderungen vor, und klicken Sie dann **OK** an.
3. Klicken Sie im Menü **Datei** die Option **Bericht drucken...** an.
Der Bericht wird gedruckt.

Berichte exportieren

Sie können einen Bericht in eine Datei exportieren.

1. Öffnen Sie hierzu ein Format, und klicken Sie anschließend **Seite einrichten...** an.
2. Nehmen Sie am Layout der Seite alle gewünschten Änderungen vor, und klicken Sie dann **OK** an.
3. Klicken Sie im Menü **Datei** die Option **Bericht exportieren...** an. Das Dialogfenster **Bericht exportieren** wird anschließend angezeigt.



4. Geben Sie den Dateinamen ein, unter dem der Bericht gespeichert werden soll, und klicken Sie **OK** an. Der Bericht wird exportiert.

Kapitel 6. Mit Prozeduren arbeiten

Mit linearen Prozeduren können Sie mit einem einzigen Befehl Abfragen ausführen, Berichte generieren, Daten bearbeiten sowie weitere Funktionen ausführen. Eine vollständige Liste aller Prozedurbefehle, die von QMF für Windows unterstützt werden, ist in der Online-Hilfe enthalten.

Prozeduren mit Logik (REXX-Prozeduren) ähneln linearen Prozeduren, enthalten jedoch die IBM Programmiersprache Object REXX sowie Prozedurbefehle. Object REXX muss lokal installiert werden, damit Prozeduren mit Logik ausgeführt werden können.

Prozeduren ausführen

Prozeduren werden dazu verwendet, mehrere Funktionen mit einem Befehl auszuführen.

Neue lineare Prozedur erstellen

Wählen Sie im Menü **Datei** die Option **Neu** und anschließend **Prozedur** aus.

Ein neues Prozedurendokument wird geöffnet.

Neue Prozedur mit Logik erstellen

1. Wählen Sie im Menü **Datei** die Option **Neu** und anschließend **Prozedur** aus.
Ein neues Prozedurendokument wird geöffnet.
2. Geben Sie einen REXX-Kommentar als erste Zeile der Prozedur ein. REXX-Kommentarzeilen beginnen mit `/*` und enden mit `*/`.
3. Geben Sie alle QMF-Prozedurbefehle ein, die in der Prozedur erscheinen sollen. QMF-Befehle müssen in Großbuchstaben eingegeben und in doppelte Anführungszeichen eingeschlossen werden.
4. Geben Sie alle REXX-Befehle ein, die in der Prozedur erscheinen sollen.

Anmerkung: REXX-Befehle werden lokal und nicht auf dem Datenbank-Server ausgeführt. Object REXX muss dazu lokal installiert sein.

Prozedur auf einem Datenbank-Server ausführen

1. Öffnen Sie ein neues Prozedurendokument, und geben Sie eine Gruppe von Befehlen ein. Sie können auch eine vorhandene Prozedur aus einer Datei oder auf dem Datenbank-Server öffnen.

2. Klicken Sie den Knopf **Prozedur ausführen** in der Symbolleiste an.



Die Prozedur wird ausgeführt.

Prozeduren speichern

Sie können Prozeduren auf Ihrem PC, auf einem Datei-Server oder auf einem Datenbank-Server speichern.

Prozeduren in Dateien speichern

1. Klicken Sie in einer geöffneten Prozedur den Knopf **Speichern** in der Symbolleiste an.



Wurde die Prozedur zuvor bereits gespeichert, wird die vorhergehende Version überschrieben. Wird die Prozedur jetzt zum ersten Mal gespeichert, wird das Dialogfenster **Speichern unter** geöffnet.

2. Geben Sie den Dateinamen ein, unter dem die Prozedur gespeichert werden soll, und klicken Sie **Speichern** an. Die Prozedur wird gespeichert.

Gespeicherte Prozedurdatei öffnen

1. Klicken Sie den Knopf **Öffnen** in der Symbolleiste an.



Das Dialogfenster **Öffnen** wird angezeigt.

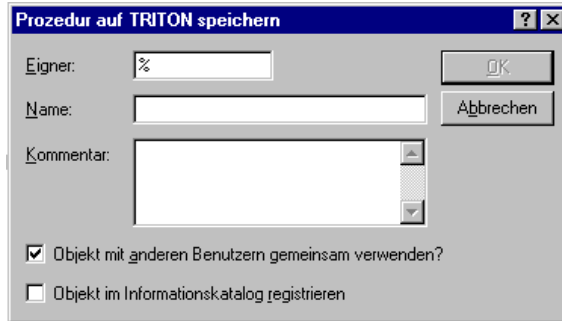
2. Wählen Sie die zu öffnende Datei aus, und klicken Sie dann **OK** an. Die ausgewählte Prozedur wird in einem neuen Prozedurdokument geöffnet.

Prozedur auf dem Datenbank-Server speichern

1. Klicken Sie in einer geöffneten Prozedur den Knopf **Auf Server speichern** in der Symbolleiste an.



Das Dialogfenster **Prozedur auf <server_name> speichern** wird anschließend angezeigt.



2. Geben Sie einen Eigner sowie einen Namen ein, und wählen Sie aus, ob die gespeicherte Prozedur zusammen mit anderen Benutzern gemeinsam verwendet werden soll; klicken Sie anschließend **OK** an. Die Prozedur wird auf dem Server gespeichert.

Ist bereits eine Prozedur mit diesem Namen vorhanden, werden Sie aufgefordert, das Überschreiben der bereits vorhandenen Prozedur zu bestätigen.

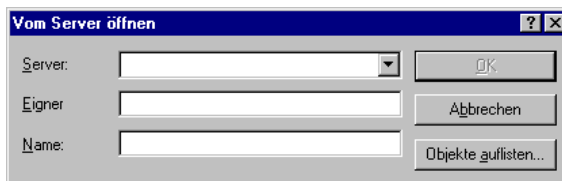
Gespeicherte Prozeduren auf dem Datenbank-Server öffnen

Sie können Prozeduren, die auf dem Datenbank-Server gespeichert wurden, öffnen.

1. Klicken Sie hierzu den Knopf **Vom Server öffnen** in der Symbolleiste an.



Das Dialogfenster **Vom Server öffnen** wird anschließend angezeigt.



2. Geben Sie einen Server, einen Eigner und einen Namen ein; klicken Sie anschließend **OK** an. Die Prozedur wird geöffnet.

Prozeduren drucken

Sie können den Text einer Prozedur drucken.

Prozedur vorab anzeigen

1. Öffnen Sie eine Prozedur. Die Prozedurbefehle werden im Fenster **Prozedur** angezeigt.
2. Klicken Sie im Menü **Datei** die Option **Seite einrichten...** an. Das Dialogfenster **Seite einrichten** wird anschließend angezeigt.
3. Nehmen Sie am Layout der Seite alle gewünschten Änderungen vor, und klicken Sie dann **OK** an.
4. Klicken Sie den Knopf **Druckvorschau...** in der Symbolleiste an.



Es wird eine Voranzeige der zu druckenden Prozedur angezeigt.

Prozedur drucken

1. Öffnen Sie eine Prozedur. Die Prozedurbefehle werden im Fenster **Prozedur** angezeigt.
2. Klicken Sie im Menü **Datei** die Option **Seite einrichten...** an. Das Dialogfenster **Seite einrichten** wird anschließend angezeigt.
3. Nehmen Sie am Layout der Seite alle gewünschten Änderungen vor, und klicken Sie dann **OK** an.
4. Klicken Sie den Knopf **Drucken** in der Symbolleiste an.



Die Prozedur wird gedruckt.

Kapitel 7. Mit Listen arbeiten

Listen stellen eine praktische Möglichkeit der Anzeige von Sammlungen (*Collections*) von QMF-Objekten dar.

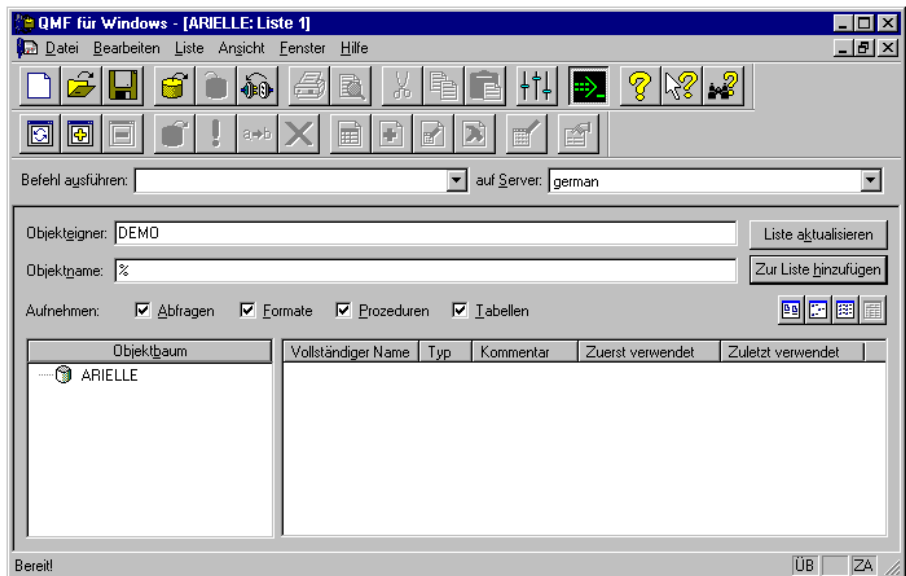
Objekte

QMF für Windows kann vier Objekttypen erkennen: Abfragen, Formate, Prozeduren und Tabellen. Sie können das Fenster **Liste** verwenden, um die Objekte auf der Basis des Objektname, des Objekteigners und des Objekttyps anzuzeigen.

Objekte auflisten

1. Wählen Sie hierzu im Menü **Datei** die Option **Neu** und anschließend **Liste** aus.

Das Fenster **Liste** wird anschließend geöffnet.



2. Geben Sie einen Objekteigner und einen Objektname an.

Anmerkung: Sie können Objekte aus einer Liste mit entsprechenden Objekten mit Hilfe von Mustern auswählen.

- Das Prozentzeichen (%) wird als Platzhalter für eine Zeichenfolge beliebiger Länge verwendet, die wiederum aus beliebigen Zeichen besteht. Um beispielsweise alle Tabellen aufzulisten, deren Namen mit dem Buchstaben A beginnen, geben Sie A% ein.
 - Das Unterstreichungszeichen (_) wird verwendet, um ein beliebiges einzelnes Zeichen darzustellen. Um beispielsweise alle Tabellen aufzulisten, die einen Ersteller haben, der den Buchstaben A an der zweiten Stelle aufweist, geben Sie _A% ein.
3. Wählen Sie den zu suchenden Objekttyp aus.
 4. Klicken Sie anschließend **Liste aktualisieren** an. Es wird eine Liste der auf dem Datenbank-Server gespeicherten Objekte angezeigt, die Ihren Anforderungen entsprechen.

Befehle im Fenster 'Liste'

Wenn Sie mit Maustaste 2 ein Objekt im Fenster **Liste** anklicken, wird eine Liste der Befehle angezeigt, die identisch mit den im Menü **Liste** enthaltenen Befehlen sind.

Objekt anzeigen

Öffnet das ausgewählte Objekt zur Anzeige. Verfügbar für Abfragen, Formate, Prozeduren und Tabellen.

Objekt ausführen

Führt das ausgewählte Objekt aus. Verfügbar für Abfragen und Prozeduren.

Objekt zeigen

Erstellt eine Abfrage auf der Basis der ausgewählten Tabelle. Sie können auswählen, dass eine SQL-Abfrage mit SELECT, mit UPDATE oder mit INSERT oder eine menügesteuerte Abfrage gezeigt werden soll. Verfügbar für Tabellen.

Objekt bearbeiten

Öffnet das ausgewählte Objekt zum Bearbeiten. Verfügbar für Tabellen.

Eigenschaften

Zeigt die Eigenschaften des ausgewählten Objekts an, u.a. Kommentare, Attribute und statistische Protokollinformationen zur Objektverwendung. Verfügbar für Abfragen, Formate, Prozeduren und Tabellen.

Listen erstellen

Sie können Listen erstellen, die als Sammlungen (*Collections*) von Objekten verwendet werden. Sie können beispielsweise eine Liste aller Abfragen, Formulare, Prozeduren, und Tabellen erstellen, die sich auf den Lagerbestand beziehen; damit wäre Ihre gesamte Arbeit in einem Objekt verfügbar. Ist die Liste erstellt, können Sie Objekte der Liste hinzufügen und aus ihr löschen sowie die Liste zur zukünftigen Verwendung speichern.

Objekte zu Listen hinzufügen

Sie können Objekte zu Listen hinzufügen.

Geben Sie dazu in einer geöffneten Liste den Eigner und den Namen der hinzuzufügenden Objekte an, und klicken Sie den Knopf **Zur Liste hinzufügen** in der Symbolleiste an.



Die Objekte, die dem Eigner und Namen entsprechen, werden der Liste hinzugefügt.

Objekte in Listen löschen

Sie können Objekte, die keinen Bezug zu einer Liste haben, in der Liste löschen.

Klicken Sie in einer geöffneten Liste den Knopf **Löschen** in der Symbolleiste an.



Das Objekt wird in der Liste entfernt, jedoch nicht gelöscht.

Listen in Dateien speichern

1. Klicken Sie in einer geöffneten Liste den Knopf **Speichern** in der Symbolleiste an.



Wurde die Liste zuvor bereits gespeichert, wird die vorhergehende Version überschrieben. Wird die Liste jetzt zum ersten Mal gespeichert, wird das Dialogfenster **Speichern unter** geöffnet.

2. Geben Sie den Dateinamen ein, unter dem die Liste gespeichert werden soll, und klicken Sie **Speichern** an. Die Liste wird gespeichert.

Gespeicherte SQL-Listendateien öffnen

1. Klicken Sie den Knopf **Öffnen** in der Symbolleiste an.



Das Dialogfenster **Öffnen** wird anschließend angezeigt.

2. Wählen Sie die zu öffnende Datei aus, und klicken Sie dann **OK** an. Die ausgewählte Liste wird in einem Listendokument geöffnet.

Kapitel 8. Mit Job-Dateien arbeiten

Sie können Prozeduren mit Job-Dateien zeitlich planen und ausführen. Job-Dateien verwenden den Windows-Scheduler ('Geplante Vorgänge') entsprechend der von Ihnen angegebenen Werte für Datum und Uhrzeit.

Job-Dateien

Sie können Jobdateien erstellen und diese lokal oder auf dem Datenbank-Server speichern.

Job-Dateien erstellen

1. Wählen Sie hierzu im Menü **Datei** die Option **Neu** und anschließend **Job** aus.

Ein neues Job-Dokument wird anschließend geöffnet.

Job-Dateien ausführen

Sie können lokal gespeicherte Job-Dateien ausführen.

1. Öffnen Sie eine Job-Datei.
2. Klicken Sie den Knopf **Job ausführen** in der Symbolleiste an.



3. Ziehen Sie die Trennlinie nach links oder rechts bzw. nach oben oder unten, um die Spalte oder Zeile in der Größe zu ändern.

Anmerkung: Wenn Sie die Abfrage nach dem Ändern der Größe von Spalten oder Zeilen speichern, wird diese zukünftig immer unter Verwendung der neuen Formatierung angezeigt.

Automatisches Anpassen von Zeilen und Spalten

Sie können Spalten und Zeilen automatisch so in der Größe ändern, dass die darin enthaltenen Daten genau in den verfügbaren Platz passen.

Wählen Sie mit der Maus eine gesamte Spalte oder Zeile aus, und klicken Sie die schwarze Trennlinie zwischen dieser Spalte und dem benachbarten Objekt doppelt an. Die Spalte bzw. Zeile wird automatisch so in der Größe geändert, dass die darin enthaltenen Daten genau in den verfügbaren Platz passen.

Anmerkung: Wenn Sie die Abfrage nach dem Ändern der Größe von Spalten oder Zeilen speichern, wird diese zukünftig immer unter Verwendung der neuen Formatierung angezeigt.

Abfrageergebnisse sortieren

Nachdem Sie eine Abfrage ausgeführt haben, können Sie die Ergebnisse alphabetisch nach Spalte sortieren.

Wählen Sie in der Ergebnissicht einer Abfrage eine Spalte aus, und wählen Sie anschließend die Option **Aufsteigend sortieren** im Menü **Ergebnisse** aus.

Die Abfrageergebnisse werden in aufsteigender Reihenfolge sortiert.

-oder-

Wählen Sie in der Ergebnissicht einer Abfrage eine Spalte aus, und wählen Sie anschließend die Option **Absteigend sortieren** im Menü **Ergebnisse** aus.

Die Abfrageergebnisse werden in absteigender Reihenfolge sortiert.

Anmerkung: Um komplexere Sortieroperationen für die ausgewählte Spalte anzuwenden, wählen Sie die Option **Sortieren...** im Menü **Ergebnisse** aus.

Spalten neu anordnen

Sie können die Reihenfolge der Spalten in den Abfrageergebnissen ändern.

Wählen Sie in der Ergebnissicht einer Abfrage eine Spalte aus, und ziehen Sie sie anschließend mit der Maus an ihren neuen Platz.

Die Spalte wird nun in der neuen Reihenfolge angezeigt.

Abfrageergebnisse formatieren

Sie können die Schriftart ändern, die für die Anzeige von Abfragen und Abfrageergebnissen verwendet wird. Die jeweiligen Möglichkeiten bei der Auswahl der Schriftart hängen davon ab, welche Schriftarten auf Ihrem Computer installiert wurden. Informationen zum Hinzufügen neuer Schriftarten sind in der Hilfefunktion Ihres Betriebssystems enthalten.

Anmerkung: Wenn Sie die Abfrage nach der Auswahl einer neuen Schriftart zur Anzeige der Abfrageergebnisse speichern, werden diese Ergebnisse zukünftig immer unter Verwendung der neuen Schriftart angezeigt.

Schriftart zur Anzeige der Abfrageergebnisse auswählen

1. Wählen Sie in der **Ergebnissicht** im Menü **Ergebnisse** die Option **Schriftart festlegen...** aus. Das Dialogfenster **Schriftart** wird anschließend angezeigt.

2. Wählen Sie die Schriftart und die Schriftgröße aus, in der die Abfrageergebnisse angezeigt werden sollen, und klicken Sie dann **OK** an. Die Abfrageergebnisse werden anschließend im von Ihnen angegebenen Format angezeigt.

Anmerkung: Wenn Sie diese Schriftart als Standardeinstellung für alle Abfrageergebnisse verwenden wollen, klicken Sie **Als Standard festlegen** an.

Numerische Abfrageergebnisse formatieren

1. Wählen Sie in der Ergebnissicht eine Spalte aus, die numerische Werte enthält, und wählen Sie anschließend die Option **Format...** im Menü **Ergebnisse** aus. Das Dialogfenster **Format** wird angezeigt.
2. Geben Sie die Art der Formatierung an, die angewendet werden soll, und klicken Sie dann **OK** an.

Anmerkung: Wenn Sie diese Schriftart als Standardeinstellung für alle Abfrageergebnisse verwenden wollen, klicken Sie **Als Standard festlegen** an.

Formatierungen von Abfrageergebnissen in ein Format konvertieren

Sie können die Formatierungen von Abfrageergebnissen in ein Format konvertieren.

1. Wählen Sie im Menü **Ergebnisse** die Option **Bericht anzeigen...** aus. Das Dialogfenster **Format auswählen** wird anschließend angezeigt.
2. Wählen Sie **Von Abfrage** aus, und klicken Sie **OK** an. Die Formatierung der Abfrageergebnisse wird in ein Format konvertiert und in einem neuen Fenster **Format** geöffnet.

Gruppierungs- und Berechnungsfunktionen für Abfrageergebnisse ausführen

Sie können Formatierungsfunktionen für Gruppierungen, Berechnungen und Übersichten für Abfrageergebnisse anwenden.

Abfrageergebnisse gruppieren

Sie können Abfrageergebnisse mit oder ohne Übersichtsinformationen gruppieren.

1. Wählen Sie die zu gruppierende Spalte aus.
2. Wählen Sie im Menü **Ergebnisse** die Art der gewünschten Gruppierungsfunktion aus.

Die Spalte wird entsprechend der von Ihnen getroffenen Auswahl gruppiert.

Abfrageergebnisse in Übersichten zusammenfassen

Sie können Übersichten für Abfrageergebnisse nach Spalte erstellen.

1. Wählen Sie die zu gruppierende Spalte aus.
2. Wählen Sie im Menü **Ergebnisse** die Art der gewünschten Übersichtsfunktion aus.

Die Spalte wird entsprechend der von Ihnen getroffenen Auswahl in einer Übersicht zusammengefasst.

Abfrageergebnisse speichern und Formatierungen ausführen

Sie können Abfrageergebnisse speichern und die Formatierung als ein Format speichern.

Abfrageergebnisse als Tabelle speichern

Sie können Abfrageergebnisse als Tabelle auf einem Datenbank-Server speichern.

1. Wählen Sie im Menü **Ergebnisse** die Option **Daten in Datenbank speichern...** aus.
Das Dialogfenster **Daten speichern** wird anschließend geöffnet.
2. Geben Sie einen Eigner und einen Tabellennamen ein und klicken Sie dann **OK** an.

Die Abfrageergebnisse werden als Tabelle auf dem Datenbank-Server gespeichert.

Abfrageergebnisse in Dateien speichern

Sie können Abfrageergebnisse in einer Datei auf Ihrem PC oder einem Datei-Server speichern.

1. Wählen Sie im Menü **Ergebnisse** die Option **Daten exportieren...** aus.
Das Dialogfenster **Daten exportieren** wird angezeigt.
2. Geben Sie Exportoptionen sowie die Lokation an, in der Sie die Datei speichern wollen, und klicken Sie anschließend **OK** an.

Die Abfrageergebnisse werden in einer Datei gespeichert.

Abfrageergebnisse drucken

Sie können Ihre Abfrageergebnisse vorab anzeigen und drucken.

Druckvorschau der Abfrageergebnisse vorab anzeigen

1. Öffnen Sie eine Abfrage, und führen Sie sie aus. Die Abfrageergebnisse werden anschließend angezeigt.
2. Wählen Sie im Menü **Datei** die Option **Seite einrichten...** aus. Das Dialogfenster **Seite einrichten** wird anschließend angezeigt.
3. Nehmen Sie am Layout der Seite alle gewünschten Änderungen vor, und klicken Sie dann **OK** an.
4. Klicken Sie den Knopf **Druckvorschau...** in der Symbolleiste an.



Es wird eine Voranzeige der zu druckenden Abfrageergebnisse angezeigt.

Abfrageergebnisse drucken

1. Öffnen Sie eine Abfrage, und wechseln Sie in die Ergebnissicht. Die Abfrageergebnisse werden anschließend angezeigt.
2. Wählen Sie im Menü **Datei** die Option **Seite einrichten...** aus. Das Dialogfenster **Seite einrichten** wird anschließend angezeigt.
3. Nehmen Sie am Layout der Seite alle gewünschten Änderungen vor, und klicken Sie dann **OK** an.
4. Klicken Sie den Knopf **Drucken** in der Symbolleiste an.



Die Abfrageergebnisse werden anschließend gedruckt.

Kapitel 9. Mit statischen Abfragen arbeiten

Eine statische Abfrage ist eine SQL-Abfrage, die bereits an einen Datenbank-Server übergeben und in ein Paket gebunden wurde. Wenn eine statische Abfrage ausgeführt wird, verwendet der Datenbank-Server den SQL-Text, der im Paket gebunden ist (und nicht den SQL-Text, der momentan im Fenster **Abfrage** angezeigt wird). Statische Abfragen sind im Hinblick auf Ressourcenbeanspruchung effizienter als dynamische Abfragen, sie können jedoch nicht bearbeitet werden.

Statische Abfragen

Statische Abfragen werden von SQL-Abfragen und menügesteuerten Abfragen erstellt, die bereits vorhanden sind.

Statische Abfragen erstellen

1. Wählen Sie im Menü **Abfrage** die Option **Paket binden...** aus. Das Dialogfenster **Statisches Paket binden** wird anschließend angezeigt.

Statisches Paket binden

Paket | Eingabevariablen

Collection-Name:

Paketname:

Eigner-ID:

Vorhandenes Paket (falls vorhanden) ersetzen?

Vorhandene Berechtigungen für Paket beibehalten?

Dezimaltrennzeichen:

Punkt (.)

Komma (,)

Zeichenfolgebegrenzer:

Apostroph (')

Anführungszeichen (")

Isolationsstufe:

Wiederholtes Lesen - Repeatable Read (RR)

Alle (RS)

Ändern (Change)

Kein Festschreiben - No Commit (UR)

2. Wählen Sie die Indexzunge **Paket** aus, geben Sie einen Collection-Namen sowie einen Paketnamen an, und ändern Sie gegebenenfalls weitere Optionen.
3. Wenn die Abfrage Substitutionsvariablen enthält, wählen Sie die Indexzunge **Eingabevariablen** aus. Ersetzen Sie alle Substitutionsvariablen durch Host-Variablen.
4. Klicken Sie **OK** an. Die statische Abfrage wird gebunden.

Anmerkung: Nachdem Sie eine Abfrage gebunden haben, müssen Sie diese Abfrage entweder in einer Datei oder auf dem Datenbank-Server speichern.

Substitutionsvariablen in statischen Abfragen durch Host-Variablen ersetzen

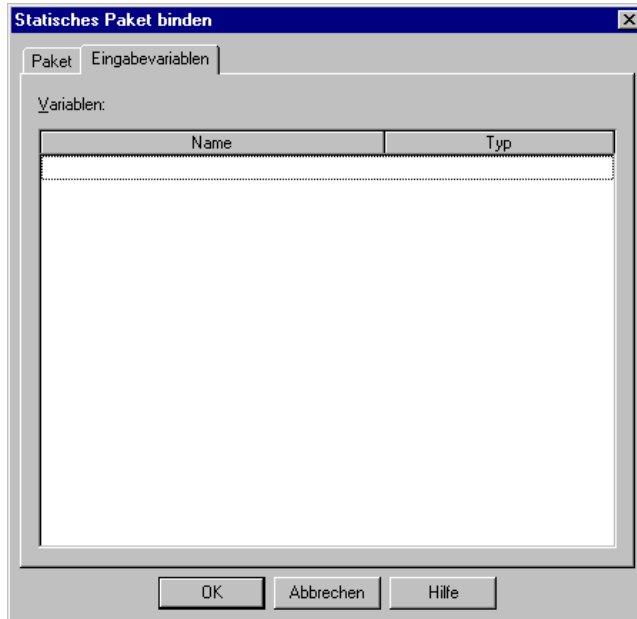
Wenn Sie ein Paket binden, müssen Sie für alle im SQL-Text vorkommenden Substitutionsvariablen jeweils eine Host-Variable angeben, die statt der Substitutionsvariable verwendet wird. Es können jedoch nicht alle Substitutionsvariablen direkt durch eine Host-Variable ersetzt werden. Substitutionsvariablen sorgen dafür, dass in der Abfrage Text direkt ersetzt wird, bevor er an den Datenbank-Server gesendet wird. Host-Variablen werden als ein Teil der Abfrage an den Datenbank-Server gesendet. Regeln dazu, wo und wie Host-Variablen in Abfragen verwendet werden können, finden Sie in der entsprechenden Dokumentation zu Ihrem Datenbank-Server.

Nachdem Sie eine Beziehung zwischen einer Substitutionsvariable und einer Host-Variable angegeben haben, "merkt sich" QMF für Windows diese Beziehung und schlägt sie in zukünftigen Abfragen vor. Auf diese Weise wird das Binden von Paketen einfacher und effizienter.

Folgende Datentypen sind für Host-Variablen zulässig:

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

1. Wählen Sie im Dialogfenster **Statisches Paket binden** die Indexzunge **Eingabevariablen** aus.



2. Geben Sie für jede Host-Variable einen Namen und die Art der Variable an, und klicken Sie **OK** an. Die Substitutionsvariablen werden in Host-Variablen umgesetzt.

Statische Abfrage ausführen

Sie können statische Abfragen wie jede andere Abfrage auch ausführen. Siehe hierzu "SQL-Abfragen" auf Seite „SQL-Abfragen“ auf Seite 9.

Kapitel 10. Mit dem Tabelleneditor arbeiten

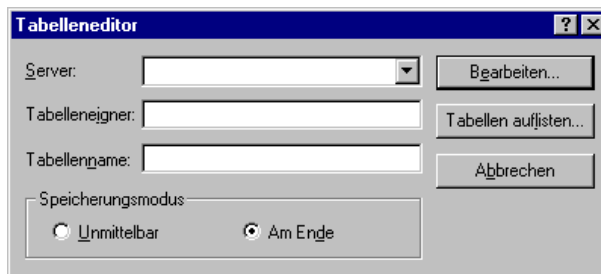
Verwenden Sie den Tabelleneditor, um in Tabellen gespeicherte Daten zu suchen, zu ergänzen, zu bearbeiten oder zu löschen, ohne dazu SQL-Anweisungen schreiben zu müssen.

Tabelleneditor

Mit dem Tabelleneditor sind Sie beim Bearbeiten Ihrer Daten und beim Suchen in diesen Daten sehr flexibel.

Mit dem Tabelleneditor nach Zeilen suchen

1. Wählen Sie hierzu im Menü **Datei** die Option **Tabelleneditor** aus. Das Dialogfenster **Tabelleneditor** wird anschließend angezeigt.



2. Geben Sie eine Tabelle an.

Anmerkung: Sie können mit Hilfe von Mustern Tabellennamen aus einer Liste mit entsprechenden Tabellen auswählen.

- Das Prozentzeichen (%) wird als Platzhalter für eine Zeichenfolge beliebiger Länge verwendet, die wiederum aus beliebigen Zeichen besteht. Um beispielsweise alle Tabellen aufzulisten, deren Namen mit dem Buchstaben A beginnen, geben Sie A% ein.
- Das Unterstrichungszeichen (_) wird verwendet, um ein beliebiges einzelnes Zeichen darzustellen. Um beispielsweise alle Tabellen aufzulisten, die einen Ersteller haben, der den Buchstaben A an der zweiten Stelle aufweist, geben Sie _A% ein.

Nachdem Sie ein Muster eingegeben haben, klicken Sie **Tabellen auflisten...** an, und wählen Sie eine Tabelle aus der nun angezeigten Liste aus.

3. Wählen Sie einen Speicherungsmodus aus.
 - **Unmittelbar** — Die Tabelle wird sofort nach jeder vorgenommenen Änderung auf dem Datenbank-Server aktualisiert.
 - **Am Ende** — Die Tabelle wird erst dann auf dem Datenbank-Server aktualisiert, wenn Sie alle Änderungen vorgenommen haben. Andere Benutzer sind nicht in der Lage, Änderungen an der Tabelle vorzunehmen, während Sie gerade dabei sind, Änderungen vorzunehmen.

Tabelle bearbeiten [?] [X]

Server: _____

Tabellenname: Q.STAFF

Spalte	Datentyp	Wert
PNR	SMALLINT NOT	
NAME	VARCHAR(9)	
KOST	SMALLINT	
TAET	CHAR(5)	
"JAHRE"	SMALLINT	
GEHALT	DECIMAL(7, 2)	
PROV	DECIMAL(7, 2)	

Zusätzliche Suchkriterien: _____

Aktionen

4. Klicken Sie **Bearbeiten...** an. Das Dialogfenster **Tabelle bearbeiten** wird anschließend angezeigt.
5. Geben Sie die Werte, nach denen Sie suchen wollen, in der Spalte **Wert** ein, oder geben Sie Suchkriterien im Feld **Zusätzliche Suchkriterien** ein, um komplexere Suchbedingungen zu verarbeiten. Sie können im Feld **Zusätzliche Suchkriterien** ein beliebiges gültiges SQL-Prädikat angeben.
6. Klicken Sie **Suche starten** an. In der Spalte **Wert** wird die erste gefundene Zeile angezeigt, die den Suchbedingungen entspricht.

Zeile hinzufügen

1. Geben Sie im Dialogfenster **Tabelle bearbeiten** die Informationen für den neuen Datensatz ein.
2. Klicken Sie **Zeile einfügen** an. Die neue Zeile wird der Tabelle hinzugefügt.
3. Klicken Sie **OK** an. Ihre Änderungen werden gespeichert.

Zeile ändern

1. Suchen Sie im Dialogfenster **Tabelle bearbeiten** die Zeile, die Sie ändern wollen.
2. Klicken Sie **Nächste Zeile** solange an, bis die zu ändernde Zeile angezeigt wird.
3. Bearbeiten Sie die Daten in der Spalte **Wert**, und klicken Sie **Zeile aktualisieren** an. Die Zeile wird aktualisiert.
4. Klicken Sie **OK** an. Ihre Änderungen werden gespeichert.

Zeile löschen

1. Suchen Sie im Dialogfenster **Tabelle bearbeiten** die Zeile, die Sie löschen wollen.
2. Klicken Sie **Nächste Zeile** solange an, bis die zu löschende Zeile angezeigt wird.
3. Klicken Sie **Zeile löschen** an. Die Zeile wird gelöscht.
4. Klicken Sie **OK** an. Ihre Änderungen werden gespeichert.

Tabellen in der Sicht 'Abfrageergebnisse' bearbeiten

Sie können Tabellen direkt in der Sicht **Abfrageergebnisse** bearbeiten.

Zeile in der Sicht 'Abfrageergebnisse' löschen

In der Sicht **Abfrageergebnisse** können Sie einzelne Zeilen in den Tabellen löschen.

Wählen Sie in der Sicht **Abfrageergebnisse** eine Zeile aus, und wählen Sie anschließend **Löschen** im Menü **Bearbeiten** aus. Die Zeile wird gelöscht.

Tabellen in der Sicht 'Abfrageergebnisse' aktualisieren

Sie können den Inhalt einzelner Spalten in der Sicht **Abfrageergebnisse** aktualisieren.

Klicken Sie hierzu in der Sicht **Abfrageergebnisse** eine Zeile doppelt an, geben Sie einen neuen Wert ein, und drücken Sie anschließend die Eingabetaste. Die Tabelle wird aktualisiert.

DB2 Forms

Wenn Sie die Komponente DB2 Forms auf Ihrer Maschine installiert haben, können Sie das Programm als Tabelleneditor für Tabellen verwenden, die keine LOB-Daten (*Large Objects*, große Objekte) enthalten. Weitere Informationen zu DB2 Forms finden Sie im 'Resource Center for DB2 Forms' unter der Web-Adresse www.rocketsoftware.com/db2forms.

Kapitel 11. Daten verteilen

Sie können Ihre Daten in andere Datenbanken und Anwendungen exportieren.

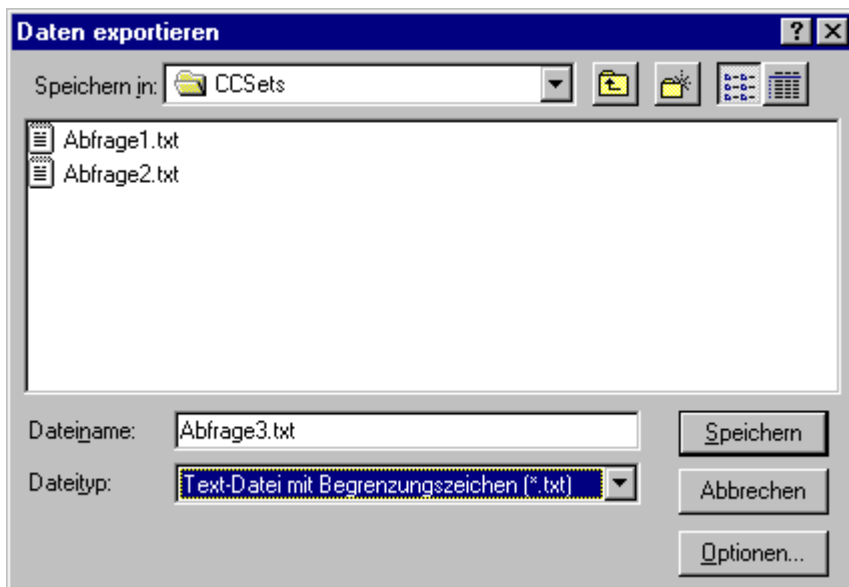
Daten exportieren

Sie können Daten von QMF für Windows in andere Anwendungen auf folgende Art und Weise exportieren:

- Export der Daten in eine Textdatei, in eine CSV-, IXF- oder HTML-Datei
- Speichern von Abfrageergebnissen in einer Tabelle
- Direktes Hinzufügen von Abfrageergebnissen in ein Arbeitsblatt der Tabellenkalkulation Microsoft Excel

Daten in Dateien exportieren

1. Wenn Sie Abfrageergebnisse anzeigen, wählen Sie im Menü **Ergebnisse** die Option **Daten exportieren...** aus. Das Dialogfenster **Daten exportieren** wird angezeigt.



2. Wählen Sie den gewünschten Typ der Ausgabedatei aus, und klicken Sie den Knopf **Optionen...** an. Abhängig vom ausgewählten Typ der Ausgabedatei wird nun das Dialogfenster **Als Text (mit Begrenzungszeichen) exportieren — Optionen**, das Dialogfenster **In HTML exportieren exportieren — Optionen**, das Dialogfenster **In IXF exportieren — Optionen** oder das Dialogfenster **In CSV exportieren — Optionen** angezeigt.
 - Sie können eine Textdatei mit der Erweiterung .TXT erstellen. Diese Datei ist eine Standard-ASCII-Datei mit entweder Zeichenfolgen- oder Spaltenbegrenzern (wie im Dialogfenster **Als Text (mit Begrenzungszeichen) exportieren — Optionen** angegeben).
 - Sie können eine HTML-Datei mit der Erweiterung .HTM (*Hypertext Markup Language*) erstellen. Diese HTML-Datei kann von jedem Web-Browser angezeigt werden. In der Datei werden automatisch alle HTML-Befehle generiert; sie kann direkt auf Ihrer Web-Site im Internet oder Intranet veröffentlicht werden. Die im Dialogfenster **In HTML exportieren — Optionen** ausgewählten Optionen bestimmen die Darstellung der exportierten Daten.
 - Sie können eine .IXF-Datei (*Integrated Data Exchange Format*, Integriertes Datenaustauschformat) erstellen. Ein IXF-Export behält alle Datenbankinformationen bei, einschließlich Spaltenüberschriften und Datentypen. Dieser Export wird in der Regel dazu verwendet, Informationen von einer Datenbank in eine andere zu übertragen.
 - Sie können eine .CSV-Datei (*Comma Separated Values*) erstellen. Ein CSV-Export ist einem Textexport sehr ähnlich; hier wird jedoch ein Komma als Spaltenbegrenzer verwendet. Dieses Format wird meist von Tabellenkalkulationsanwendungen verwendet.
3. Wählen Sie die Optionen für den ausgewählten Typ der Exportdatei aus, und klicken Sie **OK** an. Das Dialogfenster **Optionen** wird geschlossen.
4. Klicken Sie im Dialogfenster **Daten exportieren** den Knopf **Speichern** an. Die Daten werden exportiert.

Daten importieren

Sie können Daten importieren, die in einer IXF-Datei gespeichert wurden. Wenn Daten in einem Fenster **Abfrage** importiert werden, können sie auf einem Datenbank-Server gespeichert, in eine neue Datei exportiert oder für Berichte verwendet werden. PC/IXF-Dateien sowie IXF-Dateien im Zeichenmodus für System/370 werden unterstützt.

1. Wählen Sie im Menü **Datei** die Option **Daten importieren...** aus. Das Dialogfenster **Daten importieren** wird geöffnet.



2. Wählen Sie die zu importierende Datei aus, und klicken Sie dann **Öffnen** an. Die importierten Daten werden anschließend in einem neuen Fenster **Abfrage** angezeigt.

Daten auf einem Datenbank-Server speichern

Sie können importierte Abfrageergebnisse in einer Datenbanktabelle speichern.

1. Wenn Sie die Ergebnisse von importierten Abfragen anzeigen, wählen Sie im Menü **Ergebnisse** die Option **Daten in Datenbank speichern...** aus. Das Dialogfenster **Daten speichern** wird anschließend geöffnet.

Daten speichern [?] [X]

Server: DOLPHIN [v] [OK]

Tabelleneigner: Q [Abbrechen]

Tabellennamen: []

Kommentar: []

Tabellenbereich: []

Standardtabellenbereich:

Methode:

- Regulär (bereits abgerufene Daten speichern)
- Schnell (Daten nur auf dem Server abrufen und speichern)

Option für Quelldaten:

- Alle Daten speichern
- Nur die aktuell ausgewählten Daten speichern

Option für vorhandene Daten:

- Alle vorhandenen Daten ersetzen
- Vorhandenen Daten hinzufügen

Umfang zum Festschreiben: [] []

2. Wählen Sie einen Datenbank-Server aus, geben Sie einen Tabelleneigner und einen Tabellennamen ein, wählen Sie gegebenenfalls weitere Optionen aus, und klicken Sie **OK** an. Die Daten werden gespeichert.

Mit dem Befehl 'Senden an' arbeiten

QMF für Windows umfasst den Befehl **Senden an** sowie einen einfachen e-Mail-Client. Sie können den Befehl **Senden an** zusammen mit Job-Dateien verwenden, um Abfragen zeitlich zu planen und deren Ergebnisse zu verteilen.

1. Wählen Sie im Menü **Datei** die Option **Senden an** und dann **Internet-Mail-Empfänger** aus. Das Dialogfenster **Nachricht** wird angezeigt.
2. Geben Sie einen Empfänger für die Nachricht, das Thema sowie den Text der Nachricht an, und klicken Sie anschließend **Weiter** an. Das Dialogfenster **Anhänge** wird angezeigt.
3. Fügen Sie zur Nachricht ein oder mehrere Anhänge hinzu oder entfernen sie diese wieder von der Nachricht, und klicken Sie **Weiter** an. Das Dialogfenster **Nachricht senden** wird angezeigt.
4. Geben Sie den Namen Ihres Mail-Servers an, und klicken Sie **Fertig stellen** an. Die Nachricht wird gesendet.

Mit dem Add-In für Microsoft Excel arbeiten

Im Lieferumfang von QMF für Windows ist ein Add-In für Microsoft Excel 7.0 oder höher enthalten. Mit diesem Add-In können Sie QMF für Windows von Excel aus ausführen und Abfrageergebnisse direkt in ein Arbeitsblatt der Tabellenkalkulation zurückgeben. Das entsprechende Add-In wird automatisch installiert, wenn Sie die Installationsoption **Standard** auswählen, oder wenn Sie die Installationsoption **Benutzer** und dann im Fenster **Komponenten wählen** die Option **Add-In für Microsoft Excel** auswählen.

1. Klicken Sie in der Excel-Symbolleiste den Knopf **QMF für Windows** an.



QMF für Windows wird geöffnet.

2. Wählen Sie in QMF für Windows eine Abfrage aus, und führen Sie sie dann aus. Die Abfrageergebnisse werden anschließend angezeigt.
3. Wählen Sie die Daten aus, die Sie an Excel zurückgeben wollen.
4. Wählen Sie im Menü **Datei** die Option **Daten für Microsoft Excel zurückgeben...** aus. Excel wird gestartet, und das Dialogfenster **QMF für Windows — Add-In** wird angezeigt.
5. Geben Sie den Zielbereich für die Daten an, und klicken Sie anschließend **OK** an. Die Daten werden dem Arbeitsblatt hinzugefügt.

Beispielanwendungen verwenden

Für QMF für Windows sind verschiedene Beispielanwendungen und Integrationslösungen verfügbar. Näheres hierzu finden Sie auf der IBM Web-Site unter der Adresse <http://www.ibm.com/qmf/>.

Kapitel 12. Mit der QMF Berichtszentrale arbeiten

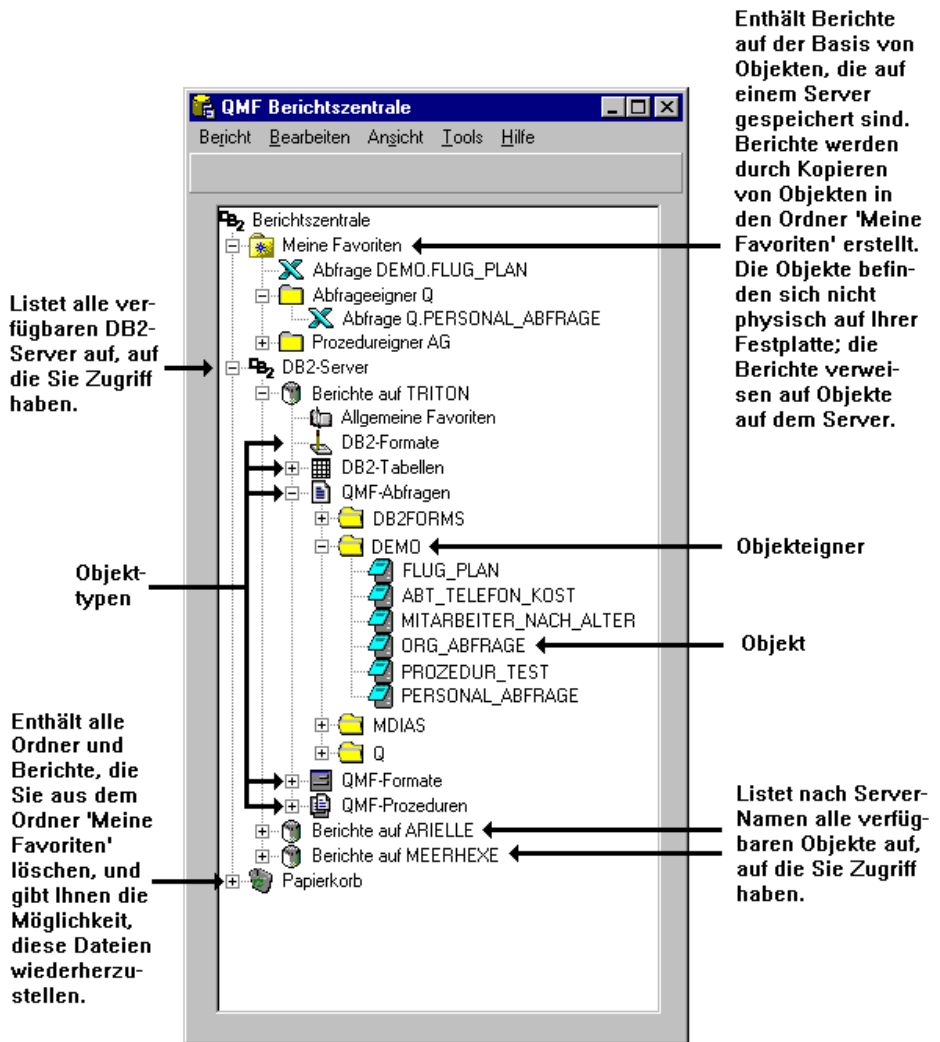
Mit Hilfe der QMF Berichtszentrale können Sie angepasste Berichte unter Verwendung von gemeinsam benutzten QMF-Abfragen, -Formaten, -Prozeduren und -Tabellen erstellen. Mit schnellem Zugriff auf diese Objekte können Sie Benutzervorgaben für das Datenformat angeben und angepasste Berichte erstellen, die in einer Vielzahl von Anwendungen angezeigt und bearbeitet werden können.

Erste Schritte in der QMF Berichtszentrale

- Klicken Sie mit der rechten Maustaste ein beliebiges Objekt oder einen beliebigen Ordner an, um dieselben Optionen zu aktivieren, die in den Menüs der Symbolleiste verfügbar sind.
- Klicken Sie das Plusymbol (+) neben einem beliebigen Ordner an, um die erste Inhaltsebene zu öffnen. Halten Sie die Umschalttaste gedrückt, und klicken Sie gleichzeitig das Plusymbol (+) an, um alle Ebenen unter dem Ordner zu öffnen.

Fenster 'QMF Berichtszentrale'

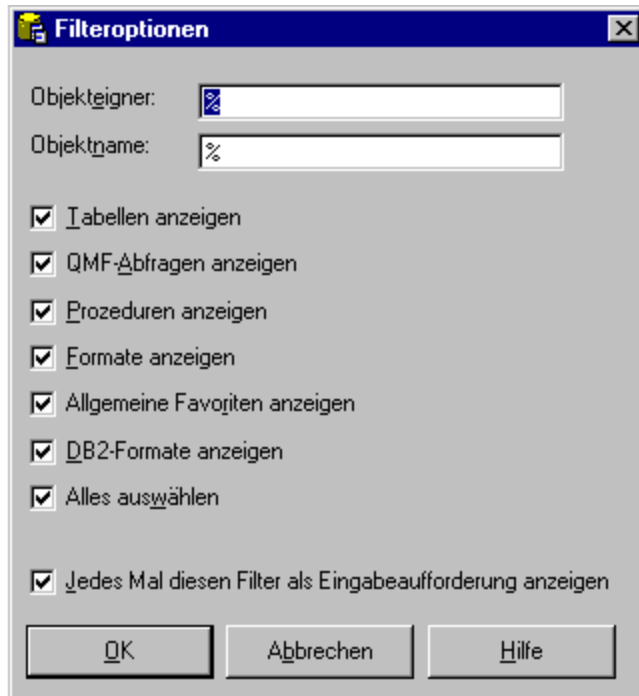
Das Fenster **QMF Berichtszentrale** enthält eine Struktur, die einer Baumstruktur ähnelt, in der verfügbare persönliche Favoriten ('Meine Favoriten'), DB2-Server, Allgemeine Favoriten, Objekte und ein Papierkorb angezeigt werden.



Objekte, die in diesem Fenster angezeigt werden, umfassen auch ein Symbol, das die Art der Anwendung repräsentiert, der die Objektausgabe zugeordnet ist.

Verbindung zum Server herstellen

1. Wenn unter "DB2-Server" keine Server-Namen angezeigt werden, klicken Sie das Plusymbol (+) an.
2. Klicken Sie das Plusymbol (+) neben einem Server an. Das Dialogfenster **Filteroptionen** wird angezeigt.



3. Wählen Sie die Typen von Objekten aus, die Sie anzeigen möchten, und klicken Sie anschließend **OK** an. Die auf dem Server verfügbaren Objekte werden nach Objekttyp getrennt angezeigt.

Mit Berichten und Objekten arbeiten

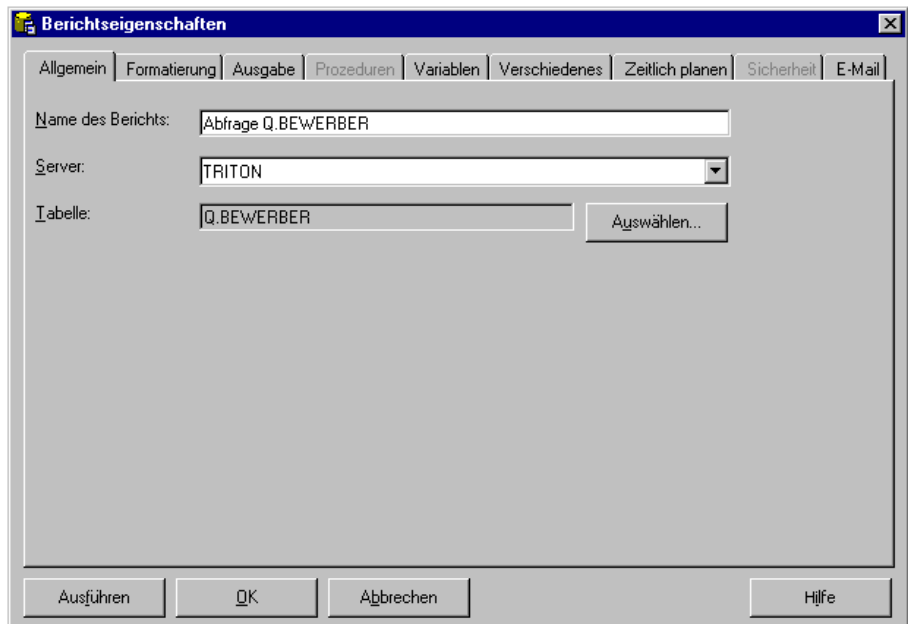
Berichte basieren auf QMF-Objekten. Alle Elemente in Ihren Ordnern mit persönlichen und allgemeinen Favoriten gelten als Berichte; Sie können die Formatierungs- und Anzeigeeoptionen für diese Elemente ändern. Die in den Ordnern für **Favoriten** enthaltenen Elemente stellen eine Verbindung zu den QMF-Objekten auf dem Server dar. QMF-Objekte werden nicht wirklich geändert. Geändert wird vielmehr die Verbindung zu dem Objekt, das als Bericht bezeichnet wird. Da Berichte auf Objekten basieren, gelten Eigenschaften von Objekten auch für Berichte.

Berichte können aus Objekten erstellt werden, die sich auf einem Server befinden; sie werden jedoch nicht auf dem Server gespeichert. Mit Hilfe dieser Funktionalität können Sie rasch Berichte zur einmaligen Ausführung erstellen. Nach dem Erstellen eines Berichts aus Objekten auf einem Server haben Sie jedoch die Option, diesen Bericht in Ihrem Ordner **Meine Favoriten** zu speichern.

Berichte ausführen

Sie können Berichte von Ihren Ordnern mit Favoriten aus oder von Objekten aus, die sich auf dem Server befinden, ausführen.

1. Klicken Sie bei einem ausgewähltem Bericht oder Objekt die Option **Eigenschaften...** im Menü **Bericht** an. Das Dialogfenster **Berichtseigenschaften** wird anschließend angezeigt.



2. Definieren Sie ggf. die Eigenschaften.
3. Klicken Sie den Knopf **Ausführen** an. Der Bericht wird verarbeitet; wenn Sie die Option **Bericht nach dem Publizieren anzeigen** auf der Registerkarte **Berichtseigenschaften** — **Ausgabe** ausgewählt haben, wird der Bericht in der von Ihnen angegebenen Anwendung angezeigt.

Sie können auch mit Hilfe einer der folgenden Methoden einen Bericht schnell ausführen:

- Wählen Sie den betreffenden Bericht und anschließend **Ausführen** im Menü **Bericht** aus.
- Klicken Sie den Bericht mit der rechten Maustaste an, und wählen Sie anschließend **Ausführen** aus.
- Klicken Sie den Berichtsnamen doppelt an.

Mit Ordnern und Favoriten arbeiten

Ordner werden dazu verwendet, Berichte und QMF-Objekte zu gruppieren; Ordner werden entsprechend den Objekteignernamen benannt. Sie können dieselben Operationen für Ordner ausführen, die Sie auch für Berichte ausführen können, wie beispielsweise Berichte ausführen und Berichtseigenschaften definieren. Wenn diese Operationen für einen Ordner ausgeführt werden, werden sie auch auf alle Berichte im betreffenden Ordner angewendet. Wenn Sie beispielsweise alle Berichte in einem Ordner nacheinander ausführen möchten, wählen Sie den betreffenden Ordner und anschließend **Ausführen** im Menü **Bericht** aus.

Die QMF Berichtszentrale enthält zwei Ordner der höchsten Ebene, in denen Sie Berichte speichern können. Diese Ordner enthalten Berichte, die auf Objekte auf Servern zeigen; die Objekte selbst sind nicht in den Ordnern mit Favoriten enthalten. Ihr Ordner **Meine Favoriten** befindet sich auf lokaler Ebene (auf Ihrem PC). Sie sind also der einzige Benutzer, der auf diesen Ordner und seinen Inhalt zugreifen kann. Der Ordner **Allgemeine Favoriten** befindet sich auf dem Server. Auf ihn können alle berechnigte Benutzer zugreifen. Sie können über Zugriff auf mehrere Ordner mit allgemeinen Favoriten verfügen. Dies hängt von den Ressourcengrenzwerten ab. Allerdings kann nie mehr als ein Ordner mit allgemeinen Favoriten auf einem Server sein.

Beim Kopieren von QMF-Objekten in Ordner mit Favoriten werden die Ordner automatisch umbenannt, so dass sie den Objekttyp und den Eigernamen enthalten. Beim Kopieren eines kompletten Objekttyps (also eines Ordners mit Objekten des gleichen Typs) von einem Server (z. B. alle Abfragen) wird der Server-Name ebenfalls in den neuen Ordnernamen aufgenommen.

Berichte zu den Favoriten hinzufügen

Sie können ein beliebiges Objekt oder einen beliebigen Bericht von einem Server zu Ihrem Ordner **Meine Favoriten** oder zum Ordner **Allgemeine Favoriten** auf einem Server hinzufügen (vorausgesetzt, der Systemadministrator hat Ihnen die entsprechende Berechtigung erteilt).

Gehen Sie wie folgt vor, um Berichte zu **Meine Favoriten** hinzuzufügen:

Klicken Sie bei ausgewähltem Bericht oder ausgewähltem Objekt die Option **Zu 'Meine Favoriten' hinzufügen** im Menü **Bericht** an. Ein Bericht wird am oberen Ende Ihres Ordners **Meine Favoriten** mit der folgenden Namenskonvention hinzugefügt: ObjekttypEIGNERNAME.OBJEKTNAME.

Gehen Sie wie folgt vor, um Berichte zu den **allgemeinen Favoriten** hinzuzufügen:

Ziehen Sie das QMF-Objekt oder den QMF-Bericht zum Ordner **Allgemeine Favoriten** auf einem Server. Sie können Berichte aus Ihrem Ordner **Meine Favoriten** oder von einem beliebigen Server hinzufügen.

Anmerkung: Wenn Sie Berichte den allgemeinen Favoriten hinzufügen oder Berichte in den allgemeinen Favoriten ändern, müssen Sie **Änderungen in 'Allgemeine Favoriten' speichern** im Menü **Bericht** auswählen, bevor Ihre Änderungen auf dem Server gesichert werden.

Weitere Informationen zum Arbeiten mit der QMF Berichtszentrale finden Sie in der Online-Hilfe.

Kapitel 13. Mit der API von QMF für Windows arbeiten

Sie können benutzerdefinierte Anwendungen über die API (*Application Programming Interface*, Anwendungsprogrammchnittstelle) von QMF für Windows erstellen.

QMF für Windows über die API steuern

Die folgenden Schritte stellen eine Übersicht dar, wie mit der API zum Steuern von QMF für Windows gearbeitet wird.

1. Erstellen Sie ein Exemplar (*Instance*) des QMF für Windows-API-Objekts. Wenn Sie Microsoft Visual Basic einsetzen, fügen Sie eine Referenz zur QMF für Windows-Typbibliothek qmfwin.tlb hinzu. Verwenden Sie anschließend die Anweisung Dim:

```
Dim QMFWin As New QMFWin
```

Oder die Anweisung CreateObject:

```
Dim QMFWin As Object
```

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

Anmerkung: Wenn Sie eine andere Entwicklungsumgebung verwenden, schlagen Sie in der jeweiligen Produktdokumentation nach. Dort sind Anweisungen zum Ausführen der jeweils erforderlichen Schritte enthalten.

2. Wählen Sie den DB2-Server aus, den Sie verwenden wollen, und rufen Sie InitializeServer() auf, um eine Verbindung zur Datenbank zu initialisieren.

Anmerkung: Ein Server wird erst dann erfolgreich initialisiert, wenn DB2 eine Benutzer-ID und ein Kennwort als gültig erkennt. Sie haben die Möglichkeit, die Benutzer-ID und das Kennwort von QMF für Windows abfragen zu lassen oder die Abfrage in Ihrer Anwendung durchzuführen und Benutzer-ID und Kennwort als Parameter des Funktionsaufrufs InitializeServer() zu übergeben.

3. Wählen Sie die Abfrage aus, die Sie ausführen wollen. Verwenden Sie hierzu InitializeQuery(). Falls die Abfrage Variablen enthält, verwenden Sie die Funktion SetVariable(), um die Variablenwerte zu definieren.
4. Öffnen Sie die Abfrage, oder führen Sie sie aus. Verwenden Sie die Funktion Open(), um den Cursor der Abfrage für SELECT-Anweisungen zu öffnen, und verwenden Sie die Funktion Execute(), um die SQL für Nicht-SELECT-Anweisungen auszuführen.

5. Ist die Abfrage eine SELECT-Anweisung, rufen Sie Zeilen mit Daten ab, indem Sie wiederholt `FetchNextRow()` aufrufen. Um mehrere Zeilen gleichzeitig abzurufen, verwenden Sie die Funktion `FetchNextRows()`, oder verwenden Sie die Funktion `CompleteQuery()`, um QMF für Windows anzuweisen, alle Zeilen abzurufen.
6. Ist die Abfrage eine SELECT-Anweisung, schließen Sie die Abfrage unter Verwendung der Funktion `Close()`.
7. Beenden Sie die Arbeitseinheit (*Unit-of-Work*) unter Verwendung der Funktion `Commit()` oder `Rollback()`.

Blockierende Aufrufe

Alle Funktionen der QMF für Windows-API sind synchrone Funktionen. Dies bedeutet, dass diese Funktionen blockieren (d. h. sie kehren nicht zurück), bis die angeforderte Datenbankaktion beendet ist. Diese Implementierung ist wünschenswert, da sie die Programmierung in der Client-Anwendung vereinfacht. Dieses Verfahren hat jedoch zur Folge, dass eine Client-Anwendung, die auf die Rückgabe einer Funktion der QMF für Windows-API wartet, nur dann auf Benutzereingaben reagieren oder Anzeigenaktualisierungen durchführen kann, wenn sie multi-thread-fähig ist.

Die QMF für Windows-API antwortet nur jeweils auf einen Funktionsaufruf von einem Client. Ist Ihre Client-Anwendung multi-threaded, müssen Sie

- auf die Beendigung eines Funktionsaufrufs warten, bevor Sie einen anderen starten, oder
- mehrere Exemplare (*Instances*) der QMF für Windows-API erstellen (ein Exemplar für jeden Thread, der die API verwendet).

Verbindung zur Datenbank herstellen

Jedes Exemplar (*Instance*) des QMF für Windows-API-Objekts stellt eine einzelne Verbindung zur Datenbank her und verwendet diese für alle Datenbankaktivitäten, für die Widerruf- (*Rollback*) oder Festschreibeoperationen (*Commit*) nachfolgend ausgeführt werden können; hierzu gehören Aktivitäten, wie das Öffnen einer Abfrage, das Abrufen von Daten und das Ausführen von SQL-Anweisungen.

Wenn Sie mehrere Abfragen in einem bestimmten Exemplar des QMF für Windows-API-Objekts erstellen, indem Sie die Funktion `InitializeQuery()` zwei oder mehrere Male aufrufen, benutzen alle Abfragen die gleiche Verbindung gemeinsam.

Die QMF für Windows-API antwortet nur jeweils auf einen Funktionsaufruf von einem Client. Ist Ihre Client-Anwendung multi-threaded, müssen Sie folgende Funktionen ausführen:

- `DeleteQMFObject()`
- `GetQMFObjectInfo()`

- GetQMFOBJECTInfoEx()
- GetQMFOBJECTList()
- GetQMFOBJECTListEx()
- GetQMFQueryText()
- SaveQMFQuery()

QMF für Windows erstellt eine zweite Verbindung zur Datenbank und verwendet dann diese, um administrative Datenbankaktivitäten abwickeln zu können (beispielsweise das Abrufen von QMF-Informationen). Diese zweite Verbindung wird benötigt, um einen einheitlichen Rollback- und Commit-Mechanismus für Client-Anwendungen zu unterstützen.

Das QMF für Windows-API-Objekt verwaltet automatisch diese Verbindungen zur Datenbank. Hat jedoch Ihr Administrator einen Grenzwert für die Anzahl der zulässigen Verbindungen festgelegt, sollten Sie bedenken, dass jedes Exemplar des QMF für Windows-API-Objekts zwei Verbindungen belegen kann.

API-Referenz

In dieser Referenz werden sämtliche Befehle aufgelistet, die zum Erstellen von Anwendungen über die API von QMF für Windows verfügbar sind.

AddDecimalHostVariable()

short AddDecimalHostVariable(long *AbfrageID*, short *Typ*, short *Genauigkeit*, short *AnzahlDerKommastellen*, const VARIANT& *Wert*)

Beschreibung

Diese Funktion wendet die Daten in *Wert* auf eine Variable in der statischen SQL-Anweisung an, die mit *AbfrageID* initialisiert wird. Sie rufen diese Funktion für jede Variable in der Anweisung auf. QMF für Windows unternimmt nichts, um Werte auf Variablen abzugleichen und zuzuordnen. Sie müssen also diese Funktion in der gleichen Reihenfolge aufrufen, wie die Variablen in der SQL-Anweisung.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeStaticQuery() zurückgegeben.
<i>Typ</i>	Der SQL-Datentyp des Werts, der an den Datenbank-Server weitergegeben werden soll. Dieser Wert beeinflusst die Konvertierung von <i>Wert</i> von einem Datentyp VARIANT in den tatsächlich übergebenen Wert. Der einzig gültige Wert für AddDecimalHostVariable() ist 484 (RSDT_DECIMAL).

<i>Genauigkeit</i>	Die Genauigkeit des Dezimalwerts.
<i>AnzahlDerKommastellen</i>	Die Anzahl der Kommastellen des Dezimalwerts.
<i>Wert</i>	Der in der Anweisung zu ersetzende Datenwert. Zur Angabe eines Nullwerts muss der Typ der Variante auf VT_EMPTY gesetzt werden.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()` oder `GetLastErrorType()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

AddHostVariable()

short AddHostVariable(long *AbfrageID*, short *Typ*, const VARIANT& *Wert*)

Beschreibung

Diese Funktion wendet die Daten in *Wert* auf eine Variable in der statischen SQL-Anweisung an, die mit *AbfrageID* initialisiert wird. Sie müssen diese Funktion für jede Variable in der Anweisung aufrufen. QMF für Windows unternimmt nichts, um Werte auf Variablen abzugleichen und zuzuordnen. Sie müssen also diese Funktion in der gleichen Reihenfolge aufrufen, wie die Variablen in der SQL-Anweisung.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeStaticQuery()</code> zurückgegeben.
<i>Typ</i>	Der SQL-Datentyp des Werts, der an den Datenbank-Server weitergegeben werden soll. Dieser Wert beeinflusst die Konvertierung von <i>Wert</i> von einem Datentyp VARIANT in den tatsächlich übergebenen Wert.
<i>Wert</i>	Der in der Anweisung zu ersetzende Datenwert. Zur Angabe eines Nullwerts muss der Typ der Variante auf VT_EMPTY gesetzt werden.

Gültige Werte für *Typ* umfassen:

Wert	Bedeutung
384 (RSDT_DATE)	Datum
388 (RSDT_TIME)	Zeit
392 (RSDT_TIMESTAMP)	Zeitmarke

448 (RSDT_VARCHAR)	Zeichenfolge mit variabler Länge
452 (RSDT_CHAR)	Zeichenfolge
464 (RSDT_VARGRAPHIC)	Grafik mit variabler Länge
468 (RSDT_GRAPHIC)	Grafik
480 (RSDT_FLOAT)	Gleitkommazahl
496 (RSDT_INTEGER)	4-Byte-Integer
500 (RSDT_SMALLINT)	2-Byte-Integer

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()` oder `GetLastErrorType()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

BindDecimalHostVariable()

short `BindDecimalHostVariable(BSTR CollectionName, BSTR Paketname, short Abschnittsnummer, short Nummer, BSTR Name, short Datentyp, short Genauigkeit, short AnzahlDerKommastellen)`

Beschreibung

Diese Funktion bindet eine Variable im angegebenen Abschnitt. Nehmen Sie den Text ":H" im SQL-Text als Platzhalter für eine Host-Variable auf. Für jede dezimale Host-Variable im SQL-Text müssen Sie `BindDecimalHostVariable()` aufrufen, um Informationen zur Variable anzugeben.

Parameter

Name	Beschreibung
<i>CollectionName</i>	Die Collection-ID des Pakets, das Sie binden wollen.
<i>Paketname</i>	Der Name des Pakets, das Sie binden wollen.
<i>Abschnittsnummer</i>	Die Abschnittsnummer der Anweisung innerhalb der zu bindenden Collection und des zu bindenden Pakets.
<i>Nummer</i>	Die Kennung für die Variable, die Sie binden wollen. Die erste Variable in der SQL-Anweisung ist Variable 0, usw.
<i>Name</i>	Vom Datenbank-Server für Diagnosezwecke verwendet. Dieser Wert wird von QMF für Windows nicht überprüft und ist auch nicht für QMF für Windows erforderlich.
<i>Datentyp</i>	Der SQL-Datentyp der Variable. Der einzig gültige Wert für <code>BindDecimalHostVariable()</code> ist 484 (RSDT_DECIMAL).
<i>Genauigkeit</i>	Die Genauigkeit des Dezimalwerts.
<i>AnzahlDerKommastellen</i>	Die Anzahl der Kommastellen des Dezimalwerts.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLException()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

BindHostVariable()

short BindHostVariable(BSTR *CollectionName*, BSTR *Paketname*, short *Abschnittsnummer*, short *Nummer*, BSTR *Name*, short *Datentyp*, short *Länge*)

Beschreibung

Diese Funktion bindet eine Variable im angegebenen Abschnitt. Nehmen Sie den Text ":H" im SQL-Text als Platzhalter für eine Host-Variable auf. Für jede Host-Variable im SQL-Text müssen Sie `BindHostVariable()` aufrufen, um Informationen zur Variable anzugeben.

Parameter

Name	Beschreibung
<i>CollectionName</i>	Die Collection-ID des Pakets, das Sie binden wollen.
<i>Paketname</i>	Der Name des Pakets, das Sie binden wollen.
<i>Abschnittsnummer</i>	Die Abschnittsnummer der Anweisung innerhalb der zu bindenden Collection und des zu bindenden Pakets.
<i>Nummer</i>	Die Kennung für die Variable, die Sie binden wollen. Die erste Variable in der SQL-Anweisung ist Variable 0, usw.
<i>Name</i>	Vom Datenbank-Server für Diagnosezwecke verwendet. Dieser Wert wird von QMF für Windows nicht überprüft und ist auch nicht für QMF für Windows erforderlich.
<i>Datentyp</i>	Der SQL-Datentyp der Variable.
<i>Länge</i>	Die Länge der Variable.

Gültige Werte für *Datentyp* umfassen:

Wert	Bedeutung
384 (RSDT_DATE)	Datum
388 (RSDT_TIME)	Zeit
392 (RSDT_TIMESTAMP)	Zeitmarke
448 (RSDT_VARCHAR)	Zeichenfolge mit variabler Länge
452 (RSDT_CHAR)	Zeichenfolge
464 (RSDT_VARGRAPHIC)	Grafik mit variabler Länge
468 (RSDT_GRAPHIC)	Grafik

480 (RSDT_FLOAT)	Gleitkommazahl
484 (RSDT_DECIMAL)	Dezimal
496 (RSDT_INTEGER)	4-Byte-Integer
500 (RSDT_SMALLINT)	2-Byte-Integer

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

BindSection()

short `BindSection(BSTR CollectionName, BSTR Paketname, short Abschnittsnummer, BSTR SQLText)`

Beschreibung

Diese Funktion legt den SQL-Text fest, der in der angegebenen Abschnittsnummer der Collection und des Pakets während des Bindens verwendet werden soll.

Parameter

Name	Beschreibung
<i>CollectionName</i>	Die Collection-ID des Pakets, das Sie binden wollen.
<i>Paketname</i>	Der Name des Pakets, das Sie binden wollen.
<i>Abschnittsnummer</i>	Die Abschnittsnummer der Anweisung innerhalb der zu bindenden Collection und des zu bindenden Pakets.
<i>SQLText</i>	Der SQL-Text der Anweisung, die Sie binden wollen.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

CancelBind()

short `CancelBind(BSTR CollectionName, BSTR Paketname)`

Beschreibung

Diese Funktion storniert eine zuvor initiierte Bindeoperation. Sämtliche Informationen zu dem benannten Paket werden freigegeben.

Parameter

Name	Beschreibung
<i>CollectionName</i>	Die Collection-ID des Pakets, das Sie binden wollen.
<i>Paketname</i>	Der Name des Pakets, das Sie binden wollen.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

ChangePassword()

short `ChangePassword(BSTR NeuesKennwort)`

Beschreibung

Diese Funktion ändert das Kennwort, für die zuvor im Aufruf `InitializeServer()` angegebene Benutzer-ID.

Anmerkung: Nicht alle Typen von Datenbank-Servern unterstützen das Ändern von Kennwörtern. Wenn der im Aufruf `InitializeServer()` angegebene Server nicht das Ändern von Kennwörtern unterstützt, wird ein Fehler zurückgegeben, und das Kennwort wird nicht geändert.

Parameter

Name	Beschreibung
<i>NeuesKennwort</i>	Das neue Kennwort.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

ClearList()

short `ClearList(short Typ)`

Beschreibung

Diese Funktion initialisiert die interne Liste, die im Parameter *Typ* angegeben ist, erneut.

Parameter

Name	Beschreibung
<i>Typ</i>	Der Wert RSL_SERVER oder der Wert RSL_QUERY.

Rückgabewert

Null, falls erfolgreich; RS_OUTOFRANGE, falls nicht erfolgreich.

Siehe auch

Open()

Close()

short Close(long *AbfrageID*)

Beschreibung

Diese Funktion schließt eine Abfrage und macht die *AbfrageID* ungültig. Ist für diese Abfrage ein Cursor geöffnet, wird der Cursor geschlossen und dadurch die Datenbank für andere Benutzer freigegeben. Diese Funktion beendet die Verbindung zum Datenbank-Server nicht. Da die Verbindung geöffnet bleibt, wird keine Rollback- oder Commit-Operation durchgeführt.

Anmerkung: Der Name dieser Funktion steht im Konflikt mit dem Schlüsselwort Close von Microsoft Access 2.0. Um diese Unverträglichkeit zu umgehen, müssen Sie — wenn Sie MS Access 2.0 verwenden — den Funktionsnamen in eckige Klammern einschließen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastSQLCode(), GetLastSQLError() oder GetLastSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

Execute()

Open()

Commit()

short Close(long *AbfrageID*)

Beschreibung

Diese Funktion schreibt alle von Ihnen in der aktuellen Arbeitseinheit vorgenommenen Änderungen fest, schließt alle offenen Cursor und macht alle noch offenen Abfrage-IDs ungültig.

Anmerkung: Der Name dieser Funktion steht im Konflikt mit dem Schlüsselwort Commit von Microsoft Access 2.0. Um diese Unverträglichkeit zu umgehen, müssen Sie — wenn Sie MS Access 2.0 verwenden — den Funktionsnamen in eckige Klammern einschließen.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() oder GetLastErrorSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

Rollback()

CompleteQuery()

short CompleteQuery(long *AbfrageID*)

Beschreibung

Diese Funktion liest alle Zeilen einer Ergebnisgruppe und speichert sie intern in QMF für Windows. Ist für diese Abfrage ein Cursor geöffnet, wird der Cursor geschlossen und dadurch die Datenbank für andere Benutzer freigegeben. Sie können die Funktion FetchNextRow() oder FetchNextRows() verwenden, um die Zeilen abzurufen. Rufen Sie Close() auf, wenn Sie mit der Abfrage fertig sind.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() oder GetLastErrorSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

CopyToClipboard()

short CopyToClipboard(long *AbfrageID*, long *ErsteZeile*, long *ErsteSpalte*, long *LetzteZeile*, long *LetzteSpalte*, BOOL *SpaltenüberschrEinschl*, [VARIANT *DatumZeitFormat*])

Beschreibung

Diese Funktion kopiert den angegebenen Bereich von Zeilen und Spalten in die Zwischenablage. Wenn Sie noch nicht für alle Zeilen, die in der Tabelle gespeichert werden sollen, Zeilendaten abgerufen haben, müssen Sie vor dem Aufrufen dieser Funktion die Funktion `CompleteQuery()` verwenden. Eine Fehlermeldung wird zurückgegeben, wenn Sie versuchen, Zeilen zu kopieren, die noch nicht aus der Datenbank abgerufen wurden.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>ErsteZeile</i>	Die erste Zeile, die in der Kopie aufgenommen werden soll.
<i>ErsteSpalte</i>	Die erste Spalte, die in der Kopie aufgenommen werden soll.
<i>LetzteZeile</i>	Die letzte Zeile, die in der Kopie aufgenommen werden soll, oder -1, wenn alle Zeilen aufgenommen werden sollen.
<i>LetzteSpalte</i>	Die letzte Spalte, die in der Kopie aufgenommen werden soll, oder -1, wenn alle Spalten aufgenommen werden sollen.
<i>Spaltenüberschr Einschl</i>	Verwenden Sie einen Wert ungleich Null, um die Spaltenüberschriften in der ersten Zeile aufzunehmen; verwenden Sie den Wert Null, um die Spaltenüberschriften nicht in der ersten Zeile aufzunehmen.
<i>DatumZeitFormat</i>	(Optional) Das Format, das für die Datums- und Zeitwerte verwendet werden soll. Folgende Werte sind zulässig: 0 (ISO-Format), 1 (USA-Format), 2 (EUR-Format), 3 (JIS-Format) oder 4 (Format der Windows-Systemsteuerung). Der Standardwert ist 4.

Anmerkung: Der Wert für die erste Zeile einer Ergebnisgruppe ist 0; der Wert für die letzte Zeile ist um eins niedriger als die Gesamtzahl der Zeilen. Der Wert für die erste Spalte einer Ergebnisgruppe ist 0; der Wert für die letzte Spalte ist um eins niedriger als die Gesamtzahl der Spalten.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()` oder `GetLastErrorType()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten. Ist die Ergebnisgruppe leer oder wurden keine Zeilen aus der Datenbank abgerufen, wird ein Wert ungleich Null zurückgegeben, es sei denn, der Parameter *ErsteZeile*=0 und der Parameter *LetzteZeile*=1. In diesem Fall wird Null zurückgegeben und eine leere Zeichenfolge wird in die Zwischenablage kopiert.

DeleteQMFObject()

short DeleteQMFObject(BSTR *EignerUndName*)

Beschreibung

Diese Funktion löscht ein QMF-Objekt (Abfrage, Format, Prozedur oder Tabelle).

Parameter

Name	Beschreibung
<i>EignerUndName</i>	Eine Zeichenfolge, die den Eigner und den Namen (getrennt durch einen Punkt) des Objekts umfasst, das Sie löschen wollen. Zum Beispiel: Udo.Abfrage2

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() oder GetLastErrorSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

EndBind()

short EndBind(BSTR *CollectionName*, BSTR *Paketname*)

Beschreibung

Diese Funktion schließt den Bindeprozess für ein statisches SQL-Paket ab. Der Aufruf dieser Funktion bewirkt, dass QMF für Windows die vollständigen Informationen für das aktuelle Paket an die Datenbank zur Verarbeitung sendet.

Parameter

Name	Beschreibung
<i>CollectionName</i>	Der im vorherigen Aufruf von StartBind() verwendete Collection-Name.
<i>Paketname</i>	Der im vorherigen Aufruf von StartBind() verwendete Paketname.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() oder GetLastErrorSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Execute()

short Execute(long *AbfrageID*)

Beschreibung

Diese Funktion führt eine SQL-Anweisung aus, die ein SQL-Verb verwendet (nicht SELECT). Verwenden Sie Execute(), wenn die Anweisung keine Ergebnisse zurückgibt. Bei Anweisungen, die Ergebnisse zurückgeben, verwenden Sie ExecuteEx(). Verwenden Sie bei Anweisungen, die das Verb SELECT verwenden, die Funktion Open() anstelle von Execute() oder ExecuteEx(). Verwenden Sie die Funktion GetQueryVerb(), um festzustellen, welches Verb von einer Abfrage verwendet wird.

Anmerkung: Der Name dieser Funktion steht im Konflikt mit dem Schlüsselwort Execute von Microsoft Access 2.0. Um diese Unverträglichkeit zu umgehen, müssen Sie — wenn Sie MS Access 2.0 verwenden — den Funktionsnamen in eckige Klammern einschließen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() oder GetLastErrorSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

Execute()
Open()

ExecuteEx()

short ExecuteEx(long *AbfrageID*, VARIANT* *Ergebnis*)

Beschreibung

Diese Funktion führt eine SQL-Anweisung aus, die ein SQL-Verb verwendet (nicht SELECT). Verwenden Sie ExecuteEx(), wenn die Anweisung Ergebnisse zurückgibt, beispielsweise bei einer SELECT INTO-Anweisung. Bei Anweisungen, die keine Ergebnisse zurückgeben, verwenden Sie Execute(). Verwenden Sie bei Anweisungen, die das Verb SELECT verwenden, die Funktion Open() anstelle von Execute() oder ExecuteEx(). Verwenden Sie die Funktion GetQueryVerb(), um festzustellen, welches Verb von einer Abfrage verwendet wird.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Ergebnis</i>	<p>Ein Zeiger auf eine VARIANT, in der die Ergebnisse gespeichert werden. Das Ergebnis ist eine Feldgruppe (Variant-Typ <code>VT_ARRAY</code> <code>VT_VARIANT</code>), die für jede Spalte im Ergebnis einen Wert enthält.</p> <p>Jeder Wert wird entweder in seinem Basisdatentyp oder dem am ehesten entsprechenden Variant-Datentyp angegeben. Folgende Rückgabetypen werden unterstützt: <code>string</code> (Variant-Typ <code>VT_BSTR</code>), <code>float</code> (Variant-Typ <code>VT_R4</code>), <code>double</code> (Variant-Typ <code>VT_R8</code>), <code>short</code> (Variant-Typ <code>VT_I2</code>), <code>long</code> (Variant-Typ <code>VT_I4</code>) und <code>binary</code> (Variant-Typ <code>VT_UI1</code> <code>VT_ARRAY</code>).</p> <p>Sie müssen die VARIANT korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen <code>VariantInit()</code> aufrufen.</p>

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

ExecuteStoredProcedure()

```
short ExecuteStoredProcedure(long AbfrageID,[VARIANT vaCommitOK], [VARIANT vaMaxErgebnisgruppen], [VARIANT vaSpaltennamen], [VARIANT vaSpaltenbezeichner], [VARIANT vaSpaltenkommentare])
```

Beschreibung

Diese Funktion führt eine SQL-Anweisung aus, die das SQL-Verb `CALL` verwendet, um eine gespeicherte Prozedur auf einem Datenbank-Server auszuführen. Verwenden Sie `ExecuteStoredProcedure()`, wenn die gespeicherte Prozedur keine Ergebnisse zurückgibt (anstelle von oder zusätzlich zu Ergebnisgruppen). Bei gespeicherten Prozeduren, die Ergebnisse zurückgeben, verwenden Sie `ExecuteStoredProcedureEx()`.

Zur Initialisierung einer gespeicherten Prozedur zur Ausführung mit `ExecuteStoredProcedure()`, rufen Sie zunächst `InitializeQuery()` auf, indem Sie eine SQL-Anweisung angeben, die die Anweisung `CALL` verwendet. Die gespeicherte Prozedur muss als ein Literal in der Anweisung `CALL` angegeben werden. Alle in der Anweisung `CALL` angegebenen Parameter (Konstanten oder sonstige) werden ignoriert. Verwenden Sie statt dessen `AddHostVariable()`, die Eingabe- und Ausgabe-Variablen anzugeben.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben. Im SQL-Text für die Abfrage muss eine CALL-Anweisung angegeben sein.
<i>vaCommitOK</i>	Ein optionaler Boolescher Wert, mit dem angegeben wird, ob die gespeicherte Prozedur die Arbeitseinheit (<i>Unit-of-Work</i>) festschreiben (<i>Commit</i>) kann, oder ob diese Operation eingeschränkt werden soll. Der Standardwert ist Wahr (<i>True</i>).
<i>vaMaxErgebnisgruppen</i>	Ein optionaler numerischer Wert, mit dem die maximale Anzahl der Ergebnisgruppen angegeben wird, die die gespeicherte Prozedur zurückgeben darf. Geben Sie null an, wenn die gespeicherte Prozedur keine Ergebnisgruppen zurückgeben soll, oder wenn der Datenbank-Server das Zurückgeben von Ergebnisgruppen von gespeicherten Prozeduren über DRDA nicht unterstützt.
<i>vaSpaltennamen</i>	Ein optionaler Boolescher Wert, mit dem angegeben wird, ob die Datenbank Spaltennamen für die in jeder zurückgegebenen Ergebnisgruppe enthaltenen Spalten zurückgeben soll.
<i>vaSpaltenbezeichner</i>	Ein optionaler Boolescher Wert, mit dem angegeben wird, ob die Datenbank Spaltenbezeichner für die in jeder zurückgegebenen Ergebnisgruppe enthaltenen Spalten zurückgeben soll.
<i>vaSpaltenkommentare</i>	Ein optionaler Boolescher Wert, mit dem angegeben wird, ob die Datenbank Spaltenkommentare für die in jeder zurückgegebenen Ergebnisgruppe enthaltenen Spalten zurückgeben soll.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastSQLCode(), GetLastSQLError() oder GetLastSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

ExecuteStoredProcedureEx()

short ExecuteStoredProcedureEx(long *AbfrageID*, VARIANT* *Ergebnis*, [VARIANT *vaCommitOK*], [VARIANT *vaMaxErgebnisgruppen*], [VARIANT *vaSpaltennamen*], [VARIANT *vaSpaltenbezeichner*], [VARIANT *vaSpaltenkommentare*])

Beschreibung

Diese Funktion führt eine SQL-Anweisung aus, die das SQL-Verb CALL verwendet, um eine gespeicherte Prozedur auf einem Datenbank-Server auszuführen. Verwenden Sie ExecuteStoredProcedureEx(), wenn die gespeicherte

Prozedur Ergebnisse zurückgibt (anstelle von oder zusätzlich zu Ergebnisgruppen). Bei gespeicherten Prozeduren, die Ergebnisse zurückgeben, verwenden Sie `ExecuteStoredProcedureEx()`.

Zur Initialisierung einer gespeicherten Prozedur zur Ausführung mit `ExecuteStoredProcedure()`, rufen Sie zunächst `InitializeQuery()` auf, indem Sie eine SQL-Anweisung angeben, die die Anweisung `CALL` verwendet. Die gespeicherte Prozedur muss als ein Literal in der Anweisung `CALL` angegeben werden. Alle in der Anweisung `CALL` angegebenen Parameter (Konstanten oder sonstige) werden ignoriert. Verwenden Sie statt dessen `AddHostVariable()`, die Eingabe- und Ausgabe-Variablen anzugeben.

Gibt die gespeicherte Prozedur Ergebnisgruppen zurück, rufen Sie `GetStoredProcedureResultSets()` auf, um die Abfrage-IDs für die Ergebnisgruppen abzurufen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben. Im SQL-Text für die Abfrage muss eine <code>CALL</code> -Anweisung angegeben sein.
<i>Ergebnis</i>	Ein Zeiger auf eine <code>VARIANT</code> , in der die Ergebnisse gespeichert werden. Das Ergebnis ist eine Feldgruppe (Variant-Typ <code>VT_ARRAY</code> <code>VT_VARIANT</code>), die für jede Spalte im Ergebnis einen Wert enthält. Jeder Wert wird entweder in seinem Basisdatentyp oder dem am ehesten entsprechenden Variant-Datentyp angegeben. Folgende Rückgabetypen werden unterstützt: <code>string</code> (Variant-Typ <code>VT_BSTR</code>), <code>float</code> (Variant-Typ <code>VT_R4</code>), <code>double</code> (Variant-Typ <code>VT_R8</code>), <code>short</code> (Variant-Typ <code>VT_I2</code>), <code>long</code> (Variant-Typ <code>VT_I4</code>) und <code>binary</code> (Variant-Typ <code>VT_UI1</code> <code>VT_ARRAY</code>). Sie müssen die <code>VARIANT</code> korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen <code>VariantInit()</code> aufrufen.
<i>vaCommitOK</i>	Ein optionaler Boolescher Wert, mit dem angegeben wird, ob die gespeicherte Prozedur die Arbeitseinheit (<i>Unit of Work</i>) festschreiben (<i>Commit</i>) kann, oder ob diese Operation eingeschränkt werden soll. Der Standardwert ist Wahr (<i>True</i>).

<i>vaMaxErgebnisgruppen</i>	Ein optionaler numerischer Wert, mit dem die maximale Anzahl der Ergebnisgruppen angegeben wird, die die gespeicherte Prozedur zurückgeben darf. Geben Sie null an, wenn die gespeicherte Prozedur keine Ergebnisgruppen zurückgeben soll, oder wenn der Datenbank-Server das Zurückgeben von Ergebnisgruppen von gespeicherten Prozeduren über DRDA nicht unterstützt.
<i>vaSpaltennamen</i>	Ein optionaler Boolescher Wert, mit dem angegeben wird, ob die Datenbank Spaltennamen für die in jeder zurückgegebenen Ergebnisgruppe enthaltenen Spalten zurückgeben soll.
<i>vaSpaltenbezeichner</i>	Ein optionaler Boolescher Wert, mit dem angegeben wird, ob die Datenbank Spaltenbezeichner für die in jeder zurückgegebenen Ergebnisgruppe enthaltenen Spalten zurückgeben soll.
<i>vaSpaltenkommentare</i>	Ein optionaler Boolescher Wert, mit dem angegeben wird, ob die Datenbank Spaltenkommentare für die in jeder zurückgegebenen Ergebnisgruppe enthaltenen Spalten zurückgeben soll.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Export()

short `Export(long AbfrageID, long ErsteZeile, long ErsteSpalte, long LetzteZeile, long LetzteSpalte, short Formatierung, short ZeichenfBegrenzer, short Spaltenbegrenzer, BOOL SpaltenüberschrEinschl, BSTR Dateiname, [VARIANT DatumZeitFomat])`

Beschreibung

Diese Funktion kopiert den angegebenen Bereich von Zeilen und Spalten in die Zwischenablage. Wenn Sie noch nicht für alle Zeilen, die in der Tabelle gespeichert werden sollen, Zeilendaten abgerufen haben, müssen Sie vor dem Aufrufen dieser Funktion die Funktion `CompleteQuery()` verwenden. Eine Fehlermeldung wird zurückgegeben, wenn Sie versuchen, Zeilen zu kopieren, die noch nicht aus der Datenbank abgerufen wurden.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>ErsteZeile</i>	Die erste Zeile, die im Export aufgenommen werden soll.
<i>ErsteSpalte</i>	Die erste Spalte, die im Export aufgenommen werden soll.

<i>LetzteZeile</i>	Die letzte Zeile, die in der Kopie aufgenommen werden soll, oder -1, wenn alle Zeilen aufgenommen werden sollen.
<i>LetzteSpalte</i>	Die letzte Spalte, die in der Kopie aufgenommen werden soll, oder -1, wenn alle Spalten aufgenommen werden sollen.
<i>Spaltenüberschr Einschl</i>	Verwenden Sie einen Wert ungleich Null, um die Spaltenüberschriften in der ersten Zeile aufzunehmen; verwenden Sie den Wert Null, um die Spaltenüberschriften nicht in der ersten Zeile aufzunehmen.
<i>Dateiname</i>	Eine Zeichenfolge, die den Namen der Datei enthält, in die Sie den Export schreiben wollen.
<i>DatumZeitFormat</i>	(Optional) Das Format, das für die Datums- und Zeitwerte verwendet werden soll. Folgende Werte sind zulässig: 0 (ISO-Format), 1 (USA-Format), 2 (EUR-Format), 3 (JIS-Format) oder 4 (Format der Windows-Systemsteuerung). Der Standardwert ist 4.

Anmerkung: Der Wert für die erste Zeile einer Ergebnisgruppe ist 0; der Wert für die letzte Zeile ist um eins niedriger als die Gesamtzahl der Zeilen. Der Wert für die erste Spalte einer Ergebnisgruppe ist 0; der Wert für die letzte Spalte ist um eins niedriger als die Gesamtzahl der Spalten.

Name	Beschreibung
<i>Formatierung</i>	Gibt die Ausgabeformatierung an.

Wert	Bedeutung
0 (RSEF_TEXT)	Die Ausgabedatei wird in einfacher Textformatierung geschrieben.
1 (RSEF_HTML)	Die Ausgabedatei wird in HTML-Formatierung (<i>Hypertext Markup Language</i>) geschrieben. Die Daten werden in einer HTML-Tabelle angeordnet.
2 (RSEF_CSV)	Die Ausgabedatei wird im CSV-Format (<i>Comma Separated Values</i> , Werte sind durch Kommas voneinander getrennt) geschrieben.
3 (RSEF_PCIXF)	Die Ausgabedatei wird im PC/IXF-Format (<i>Integrated Date Exchange Format</i> , Integriertes Datenaustauschformat) geschrieben.
4 (RSEF_S370IXF)	Die Ausgabedatei wird im System/370 IXF-Format (<i>Integrated Date Exchange Format</i> , Integriertes Datenaustauschformat) geschrieben.

Name	Beschreibung
------	--------------

<i>ZeichenfBegrenzer</i>	Gibt den Zeichenfolgebegrenzer an. Dieser Parameter wird ignoriert, wenn als <i>Formatierung</i> RSEF_HTML angegeben wurde.
--------------------------	---

Wert	Bedeutung
0 (RSSD_NONE)	Es wird kein Zeichenfolgebegrenzer verwendet.
1 (RSSD_SINGLEQUOTE)	Zeichenfolgen werden durch ein einfaches Anführungszeichen (') begrenzt.
2 (RSSD_DOUBLEQUOTE)	Zeichenfolgen werden durch ein doppeltes Anführungszeichen (") begrenzt.

Name	Beschreibung
<i>Spaltenbegrenzer</i>	Gibt den Spaltenbegrenzer an. Dieser Parameter wird ignoriert, wenn als <i>Formatierung</i> RSEF_HTML angegeben wurde.

Wert	Bedeutung
0 (RSCD_SPACE)	Spalten werden durch ein Leerzeichen begrenzt ().
1 (RSCD_TAB)	Spalten werden durch einen Tabulatorschritt begrenzt (\t).
2 (RSCD_COMMA)	Spalten werden durch ein Komma begrenzt (,).

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten. Ist die Ergebnisgruppe leer oder wurden keine Zeilen aus der Datenbank abgerufen, wird ein Wert ungleich Null zurückgegeben, es sei denn, der Parameter *ErsteZeile*=0 und der Parameter *LetzteZeile*=1. In diesem Fall wird Null zurückgegeben und eine leere Datei wird geschrieben.

Siehe auch

CopyToClipboard()

ExportForm()

short ExportForm(BSTR *EignerUndName*, BSTR *Dateiname*)

Beschreibung

Diese Funktion exportiert das angegebene QMF-Format in die angegebene Datei.

Parameter

Name	Beschreibung
<i>EignerUndName</i>	Eine Zeichenfolge, die den Eigner und den Namen (getrennt durch einen Punkt) des Formats umfasst, das Sie exportieren wollen. Zum Beispiel: Udo.Abfrage2
<i>Dateiname</i>	Eine Zeichenfolge, die den Namen der Datei enthält, in die Sie das exportierte Format schreiben wollen.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

`PrintReport()`

ExportReport()

`short ExportReport(long AbfrageID, short Quelle, BSTR Quelle, BSTR Ausgabedateiname, short Seitenlänge, short Seitenbreite, BOOL DatumZeitEinschl, BOOL SeitenzahlenEinschl, [VARIANT Formatierung], [VARIANT FormatseiteneinrichtungVerwenden])`

Beschreibung

Diese Funktion erstellt einen Bericht für die angegebene Abfrage und schreibt ihn in eine Datei. Die Formatierung und das Layout für den Bericht geben Sie in einem QMF-Format an. Die Ausgabedatei ist eine Textdatei in ASCII-Formatierung. In dieser Datei werden Zeilen durch je ein Zeichen für Zeilenschaltung (*Carriage Return*) und Zeilenvorschub (*Line Feed*) getrennt. Seiten werden durch ein Zeichen für Formularvorschub (*Form Feed*) getrennt. Es empfiehlt sich, die Ausgabedatei in einer Monospace-Schriftart anzuzeigen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Quelle</i>	Der Name (entweder ein <code>Dateiname</code> oder <code>Eigner.Name</code>) des Formats, das verwendet werden soll.
<i>Ausgabedateiname</i>	Der Name der Datei, in die der Bericht geschrieben werden soll.

<i>Seitenlänge</i>	Die Anzahl der Zeilen auf jeder Seite des Berichts. Der Wert -1 für den Parameter <i>Seitenlänge</i> gibt an, dass die Ausgabe fortlaufend, d. h. ohne Seitenumbrüche erfolgen soll (es sei denn, der Bericht ist breiter als die angegebene <i>Seitenbreite</i>).
<i>DatumZeitEinschl</i>	Ein Wert ungleich Null gibt an, dass Datum und Zeit am unteren Rand jeder Seite ausgegeben werden; der Wert Null gibt an, dass Datum und Zeit nicht ausgegeben werden.
<i>SeitenzahlenEinschl</i>	Ein Wert ungleich Null gibt an, dass die Seitenzahlen am unteren Rand jeder Seite ausgegeben werden; der Wert Null gibt an, dass Seitenzahlen nicht ausgegeben werden.
<i>Formatierung</i>	Ein optionaler Parameter, mit dem die Formatierung des exportierten Berichts angegeben wird. Falls Null, ist die Formatierung unformatierter Text. Hierdurch wird angegeben, dass die Ausgabe genau der entsprechen soll, die vom Format erzeugt wird (Text oder HTML, abhängig von Typ des Formats). Falls ungleich Null, ist das Format HTML. Hierdurch wird angegeben, dass die Ausgabe HTML sein soll. Bei Nicht-HTML-Formaten wird die Ausgabe in HTML konvertiert, indem HTML-Befehle am Anfang und am Ende der Ausgabe hinzugefügt werden. Der Standardwert ist Null.
<i>DatumZeitFormat</i>	(Optional) Das Format, das für die Datums- und Zeitwerte verwendet werden soll. Folgende Werte sind zulässig: 0 (ISO-Format), 1 (USA-Format), 2 (EUR-Format), 3 (JIS-Format) oder 4 (Format der Windows-Systemsteuerung). Der Standardwert ist 4.
<i>Formatierung</i>	Das Format der Ausgabedatei.
<i>Formatseiteneinrichtung Verwenden</i>	Ein optionaler Parameter, mit dem — wenn er ungleich Null ist — angegeben wird, dass die Parameter <i>Seitenlänge</i> , <i>Seitenbreite</i> , <i>DatumZeitEinschl</i> und <i>SeitenzahlenEinschl</i> ignoriert werden sollen und dass statt dessen dafür Werte verwendet werden sollen, die von den Werten "genommen" werden, die mit dem angegebenen Format gespeichert wurden. Der Standardwert ist Null.
Wert	Bedeutung
0 (RSF_DEFAULT)	Standardformat verwenden. Der Parameter <i>Formatname</i> muss eine leere Zeichenfolge sein.
1 (RSF_DATABASE)	Format aus der Datenbank verwenden. Geben Sie im Parameter <i>Formatname</i> den Eigner und Namen (Eigner.Name) des Formats an. Wenn ein Format verwendet werden soll, das sich auf einem anderen Datenbank-Server befindet, muss dieses Format zunächst mit der Funktion <code>ExportForm()</code> in eine Datei exportiert und anschließend im Parameter <i>Quellentyp</i> der Wert <code>RSF_FILE</code> angegeben werden.

2 (RSF_FILE)

Format aus einer Datei verwenden. Geben Sie im Parameter *Formatname* den Dateinamen an.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLException()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

`ExportForm()`

FastSaveData()

short `FastSaveData(long AbfrageID, BOOL Ersetzen, BSTR Tabellennamen, BSTR TabellenBerName, [VARIANT Kommentar])`

Beschreibung

Diese Funktion erstellt einen Bericht für die angegebene Abfrage und schreibt ihn in eine Datei. Die Formatierung und das Layout für den Bericht geben Sie in einem QMF-Format an. Die Ausgabedatei ist eine Textdatei in ASCII-Formatierung. In dieser Datei werden Zeilen durch je ein Zeichen für Zeilenschaltung (*Carriage Return*) und Zeilenvorschub (*Line Feed*) getrennt. Seiten werden durch ein Zeichen für Formularvorschub (*Form Feed*) getrennt. Es empfiehlt sich, die Ausgabedatei in einer Monospace-Schriftart anzuzeigen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Ersetzen</i>	Verwenden Sie einen Wert ungleich Null, wenn die angegebenen Daten alle in der Tabelle vorhandenen Daten ersetzen sollen. Verwenden Sie den Wert Null, wenn die angegebenen Daten an die in der Tabelle vorhandenen Daten angehängt werden sollen.
<i>Tabellennamen</i>	Der Name der Tabelle, in der die Daten gespeichert werden sollen. Ist die Tabelle nicht vorhanden, wird sie von QMF für Windows erstellt.
<i>TabellenBerName</i>	Der Name des Tabellenbereichs, in dem die Tabelle vorhanden ist oder erstellt werden wird. Ist der Parameter <i>TabellenBerName</i> NULL oder eine leere Zeichenfolge, verwendet QMF für Windows den standardmäßigen Tabellenbereich. Haben Sie QMF für Windows so konfiguriert, dass immer der Standardtabellenbereich verwendet wird, wird dieser Parameter ignoriert. Siehe <code>RSR_SDDIFFERENTTS</code> in der Beschreibung zu <code>GetResourceLimit()</code> .

<i>Kommentar</i>	Ein optionaler Parameter für eine Zeichenfolge, die einen Kommentar für die Tabelle angibt, in der die Daten gespeichert sind.
------------------	--

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

FetchNextRow()

short `FetchNextRow(long AbfrageID, VARIANT* Zeile)`

Beschreibung

Diese Funktion liest die nächste Datenzeile aus der Datenbank.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Zeile</i>	<p>Ein Zeiger auf eine <code>VARIANT</code>, in der die Ergebnisse gespeichert werden. Das Ergebnis ist eine Feldgruppe (Variant-Typ <code>VT_ARRAY</code> <code>VT_VARIANT</code>), die für jede Spalte im Ergebnis einen Wert enthält. Rufen Sie <code>GetColumnCount()</code> auf, um die Anzahl der Werte in der Feldgruppe (<i>Array</i>) festzustellen.</p> <p>Jeder Wert wird entweder in seinem Basisdatentyp oder dem am ehesten entsprechenden Variant-Datentyp angegeben. Folgende Rückgabetypen werden unterstützt: <code>string</code> (Variant-Typ <code>VT_BSTR</code>), <code>float</code> (Variant-Typ <code>VT_R4</code>), <code>double</code> (Variant-Typ <code>VT_R8</code>), <code>short</code> (Variant-Typ <code>VT_I2</code>), <code>long</code> (Variant-Typ <code>VT_I4</code>) und <code>binary</code> (Variant-Typ <code>VT_UI1</code> <code>VT_ARRAY</code>).</p> <p>Wird das Ende der Ergebnisgruppe erreicht (es sind keine Zeilen mehr vorhanden, die abgerufen werden können), oder wenn die Ergebnisgruppe leer ist, ist das Ergebnis leer (Variant-Typ <code>VT_EMPTY</code>) und nicht eine Feldgruppe (<i>Array</i>).</p> <p>Sie müssen die <code>VARIANT</code> korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen <code>VariantInit()</code> aufrufen.</p>

Anmerkung: Aufgrund eines Programmfehlers in Microsoft Excel 7.0 und Microsoft Access 7.0 — sowie möglicherweise anderen 32-Bit-Produkten von Microsoft, die die Visual Basic for Applications (VBA) verwenden -, werden Zeichenfolgedaten (*String Data*) in

Variant-Variablen, die aus QMF für Windows empfangen werden, nicht von Unicode (von OLE verwendet) in ANSI (von VBA verwendet) umgesetzt. Wenn dieser Fehler auftritt, wird nur das erste Zeichen der Zeichenfolge angezeigt. Um den Fehler zu beheben, müssen Sie die Variable gleich einer leeren Zeichenfolge definieren, bevor Sie die QMF für Windows-Funktion aufrufen, die die Variable verwendet.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wenn das Ende der Ergebnisgruppe erreicht wird, lautet der Rückgabewert -1. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

`FetchNextRows()`

FetchNextRowEx()

`short FetchNextRowEx(long AbfrageID)`

Beschreibung

Diese Funktion liest die nächste Datenzeile aus der Datenbank. Sie können diese Funktion in Umgebungen verwenden, die VARIANT-Feldgruppen (*Arrays*) nicht unterstützen, wie beispielsweise Microsoft Access 2.0. Verwenden Sie diese Funktion in Verbindung mit der Funktion `GetColumnValue()`, um für die aktuelle Zeile die Daten in jeder Spalte abzurufen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wenn das Ende der Ergebnisgruppe erreicht wird, lautet der Rückgabewert -1. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

`FetchNextRowsEx()`

FetchNextRows()

`short FetchNextRows(long AbfrageID, VARIANT* Zeilen, long* AnzZeilen)`

Beschreibung

Diese Funktion liest die nächsten im Parameter *AnzZeilen* angegebenen Zeilen mit Daten aus der Datenbank.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Zeile</i>	<p>Ein Zeiger auf eine VARIANT, in der die Ergebnisse gespeichert werden. Das Ergebnis ist eine zweidimensionale Feldgruppe (Variant-Typ <code>VT_ARRAY</code> <code>VT_VARIANT</code>), die für jede Spalte in jeder Zeile einen Wert enthält. Rufen Sie <code>GetColumnCount()</code> auf, um die Anzahl der Spalten in der Feldgruppe (<i>Array</i>) festzustellen. Die Dimensionen der Feldgruppe sind [<i>AnzZeilen</i>][<i>SpaltenZahl</i>] — auch wenn die Anzahl der ungelesenen Zeilen in der Ergebnisgruppe niedriger ist, als der Wert für <i>AnzZeilen</i> (in diesem Fall enthält die Feldgruppe zusätzliche, nicht verwendete Einträge).</p> <p>Jeder Wert wird entweder in seinem Basisdatentyp oder dem am ehesten entsprechenden Variant-Datentyp angegeben. Folgende Rückgabetyper werden unterstützt: <code>string</code> (Variant-Typ <code>VT_BSTR</code>), <code>float</code> (Variant-Typ <code>VT_R4</code>), <code>double</code> (Variant-Typ <code>VT_R8</code>), <code>short</code> (Variant-Typ <code>VT_I2</code>), <code>long</code> (Variant-Typ <code>VT_I4</code>) und <code>binary</code> (Variant-Typ <code>VT_UI1</code> <code>VT_ARRAY</code>).</p> <p>Wird das Ende der Ergebnisgruppe erreicht (es sind keine Zeilen mehr vorhanden, die abgerufen werden können), oder wenn die Ergebnisgruppe leer ist, ist das Ergebnis leer (Variant-Typ <code>VT_EMPTY</code>) und nicht eine Feldgruppe (<i>Array</i>).</p> <p>Sie müssen die VARIANT korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen <code>VariantInit()</code> aufrufen.</p>
<i>AnzZeilen</i>	Ein Zeiger auf einen Datentyp "long", der die Anzahl der zu lesenden Zeilen enthält. Ist die Anzahl der ungelesenen Zeilen in der Ergebnisgruppe niedriger, als der Wert für <i>AnzZeilen</i> , wird <i>AnzZeilen</i> auf die tatsächliche Anzahl von Zeilen, die in der Ergebnisgruppe enthalten sind, zurückgesetzt.

Anmerkung: Aufgrund eines Programmfehlers in Microsoft Excel 7.0 und Microsoft Access 7.0 — sowie möglicherweise anderen 32-Bit-Produkten von Microsoft, die die Visual Basic for Applications (VBA) verwenden -, werden Zeichenfolgedaten (*String Data*) in Variant-Variablen, die aus QMF für Windows empfangen werden, nicht von Unicode (von OLE verwendet) in ANSI (von VBA verwendet) umgesetzt. Wenn dieser Fehler auftritt, wird nur das erste Zeichen der Zeichenfolge angezeigt. Um den

Fehler zu beheben, müssen Sie die Variable gleich einer leeren Zeichenfolge definieren, bevor Sie die QMF für Windows-Funktion aufrufen, die die Variable verwendet.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wenn das Ende der Ergebnisgruppe erreicht wird, lautet der Rückgabewert -1. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

`FetchNextRow()`

FetchNextRowsEx()

`short FetchNextRowsEx(long AbfrageID, long* AnzZeilen)`

Beschreibung

Diese Funktion liest die nächsten im Parameter *AnzZeilen* angegebenen Zeilen mit Daten aus der Datenbank. Sie können diese Funktion in Umgebungen verwenden, die VARIANT-Feldgruppen (*Arrays*) nicht unterstützen, wie beispielsweise Microsoft Access 2.0. Verwenden Sie diese Funktion in Verbindung mit der Funktion `GetColumnValueEx()`, um für die aktuelle Zeile die Daten in jeder Spalte abzurufen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>AnzZeilen</i>	Ein Zeiger auf einen Datentyp "long", der die Anzahl der zu lesenden Zeilen enthält. Ist die Anzahl der ungelesenen Zeilen in der Ergebnisgruppe niedriger, als der Wert für <i>AnzZeilen</i> , wird <i>AnzZeilen</i> auf die tatsächliche Anzahl von Zeilen, die in der Ergebnisgruppe enthalten sind, zurückgesetzt.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wenn das Ende der Ergebnisgruppe erreicht wird, lautet der Rückgabewert -1. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

`FetchNextRowEx()`

FlushQMFCache()

void FlushQMFCache()

Beschreibung

Diese Funktion weist QMF für Windows an, die QMF-Informationen aus dem Zwischenspeicher zu entfernen und ihren Inhalt zu löschen. Wenn QMF für Windows das nächste Mal QMF-Informationen benötigt, werden diese aus der Datenbank abgerufen. Normalerweise stellt QMF für Windows die QMF-Informationen aus der Datenbank in den Zwischenspeicher, um den Datenverkehr zu verringern und den Durchsatz zu verbessern. Diese Funktion wird vor der Funktion GetQMFObjectInfo(), GetQMFFQueryText() oder GetQMFObjectList() aufgerufen, um sicherzustellen, dass die zurückgegebenen Informationen aktuell sind.

Rückgabewert

Keiner.

GetColumnCount()

long GetColumnCount(long *AbfrageID*)

Beschreibung

Diese Funktion gibt die Anzahl der Spalten in der Ergebnisgruppe zurück.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben.

Rückgabewert

Die Anzahl der Spalten in jeder Zeile, falls erfolgreich. 0 oder -1, falls nicht erfolgreich. Wird der Wert 0 oder -1 zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastSQLCode(), GetLastSQLError() oder GetLastSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetColumnDataValue()

short GetColumnDataValue(long *AbfrageID*, long *Index*)

Beschreibung

Diese Funktion gibt den Datenwert für die im Parameter *Index* angegebene Spalte in der aktuellen Datenzeile zurück. Nach dem Aufruf dieser Funktion kann die Eigenschaft *Wert* nach dem zurückgegebenen Wert abgefragt werden. Verwenden Sie diese Funktion zusammen mit der Funktion FetchNextRowEx(), um auf Daten in einer einzelnen Datenzeile zuzugreifen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Index</i>	Der Index (mit der Basis Null) der abzurufenden Zeile mit Daten.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLException()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetColumnHeader()

BSTR `GetColumnHeader(long AbfrageID, long Index, short* Ergebnis)`

Beschreibung

Diese Funktion gibt den Spaltenkennsatz (auch Spaltenname genannt) zurück, der dem Index *Index* zugeordnet ist.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Index</i>	Der Index (mit der Basis Null) der abzurufenden Zeile mit Daten.
<i>Ergebnis</i>	Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird für Ergebnis ein Wert ungleich Null zurückgegeben, können die Funktionen <code>GetLastErrorString()</code> , <code>GetLastErrorType()</code> , <code>GetLastSQLCode()</code> , <code>GetLastSQLException()</code> oder <code>GetLastSQLState()</code> aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Anmerkung: Spaltenüberschriften sind für statische SQL-Anweisungen nicht verfügbar. Bei Abfrage-IDs, die von `InitializeStaticQuery()` zurückgegeben werden, gibt `GetColumnHeader` eine Zeichenfolge in der Form "Spaltn" zurück; hierbei ist "n" die Nummer der Spalte.

Rückgabewert

Die zurückgegebene Zeichenfolge stellt den Spaltennamen dar, wie im Parameter *Index* angegeben.

GetColumnHeaderEx()

short `GetColumnHeaderEx(long AbfrageID, long Index)`

Beschreibung

Diese Funktion gibt den Spaltenkennsatz (auch Spaltenname genannt) zurück, der dem Index *Index* zugeordnet ist. Nach dem Aufruf dieser Funktion kann die Eigenschaft *Wert* nach dem zurückgegebenen Wert abgefragt werden.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Index</i>	Der Index (mit der Basis Null) der abzurufenden Zeile mit Daten.

Anmerkung: Spaltenüberschriften sind für statische SQL-Anweisungen nicht verfügbar. Bei Abfrage-IDs, die von `InitializeStaticQuery()` zurückgegeben werden, gibt `GetColumnHeader` eine Zeichenfolge in der Form "Spaltn" zurück; hierbei ist "n" die Nummer der Spalte.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Ist der Rückgabewert Null, sollten Sie die Eigenschaft *Wert* abfragen, um die Zeichenfolge, die für den Spaltennamen steht, zu ermitteln. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetColumnHeadings()

short `GetColumnHeadings(long AbfrageID, VARIANT* Überschriften)`

Beschreibung

Diese Funktion gibt die Spaltenüberschriften (auch Spaltennamen genannt) zurück.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Überschriften</i>	Ein Zeiger auf eine VARIANT, in der die Ergebnisse gespeichert werden. Das Ergebnis ist eine Feldgruppe mit Zeichenfolgen (Variant-Typ <code>VT_ARRAY VT_BSTR</code>), die für jede Spaltenüberschrift eine Zeichenfolge enthält.

Sie müssen die VARIANT korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen `VariantInit()` aufrufen.

Anmerkung: Aufgrund eines Programmfehlers in Microsoft Excel 7.0 und Microsoft Access 7.0 — sowie möglicherweise anderen 32-Bit-Produkten von Microsoft, die die Visual Basic for Applications (VBA) verwenden -, werden Zeichenfolgedaten (*String Data*) in Variant-Variablen, die aus QMF für Windows empfangen werden, nicht von Unicode (von OLE verwendet) in ANSI (von VBA verwendet) umgesetzt. Wenn dieser Fehler auftritt, wird nur das erste Zeichen der Zeichenfolge angezeigt. Um den Fehler zu beheben, müssen Sie die Variable gleich einer leeren Zeichenfolge definieren, bevor Sie die QMF für Windows-Funktion aufrufen, die die Variable verwendet.

Anmerkung: Spaltenüberschriften sind für statische SQL-Anweisungen nicht verfügbar. Bei Abfrage-IDs, die von `InitializeStaticQuery()` zurückgegeben werden, gibt `GetColumnHeadings()` die Zeichenfolgen "Col1", "Col2" usw. zurück.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLException()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetColumnValue()

short `GetColumnValue(long AbfrageID, long Index, VARIANT* Wert)`

Beschreibung

Diese Funktion gibt den Datenwert für die im Parameter *Index* angegebene Spalte in der aktuellen Datenzeile zurück. Verwenden Sie diese Funktion zusammen mit der Funktion `FetchNextRowEx()`, um auf Daten in einer einzelnen Datenzeile zuzugreifen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Index</i>	Der Index (mit der Basis Null) der abzurufenden Zeile mit Daten.
<i>Wert</i>	Ein Zeiger auf eine VARIANT, in der die Ergebnisse gespeichert werden sollen. Das Ergebnis ist ein Datenwert, der auf dem Variant-Typ basiert. Sie müssen die VARIANT korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen <code>VariantInit()</code> aufrufen.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetColumnValueEx()

short GetColumnValueEx(long *AbfrageID*, long *Zeilenindex*, long *Spaltenindex*, VARIANT* *Wert*)

Beschreibung

Diese Funktion gibt den Datenwert für die im Parameter *Spaltenindex* angegebene Spalte für die Datenspalte zurück, die in *Zeilenindex* angegeben ist. Verwenden Sie diese Funktion zusammen mit der Funktion `FetchNextRowsEx()`, um auf Daten in einer einzelnen Datenzeile zuzugreifen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Zeilenindex</i>	Der Index (mit der Basis Null) der abzurufenden Zeile.
<i>Spaltenindex</i>	Der Index (mit der Basis Null) der abzurufenden Spalte.
<i>Wert</i>	Ein Zeiger auf eine VARIANT, in der die Ergebnisse gespeichert werden sollen. Sie können die sich hieraus ergebende Variant abfragen, um den Datentyp für die weitere Verarbeitung zu ermitteln. Sie müssen die VARIANT korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen <code>VariantInit()</code> aufrufen.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetDefaultServerName()

BSTR GetDefaultServerName()

Beschreibung

Diese Funktion gibt eine Zeichenfolge zurück, die den Standard-Server-Namen enthält.

Rückgabewert

Eine Zeichenfolge, die den Standard-Server-Namen angibt.

GetGlobalVariable()

BSTR GetGlobalVariable(BSTR *Name*)

Beschreibung

Die Funktion ruft den Wert der angegebenen globalen Variable ab.

Parameter

Name	Beschreibung
<i>Name</i>	Eine Zeichenfolge, die den Namen der Variablen enthält, die definiert werden soll.

Rückgabewert

Eine Zeichenfolge, die den Wert der globalen Variablen oder NULL enthält, wenn die Variable keinen Wert hat oder wenn ein Fehler auftritt.

GetHostVariableNames()

short GetHostVariableNames(long *AbfrageID*, VARIANT* *Namen*)

Beschreibung

Diese Funktion gibt eine Feldgruppe (*Array*) der Namen aller Host-Variablen zurück, auf die in der angegebenen Abfrage verwiesen wird. Die Abfrage muss eine statische Abfrage sein, mit der auf Host-Variablen verwiesen wird (entweder in der QMF-Abfrage gespeichert oder durch AddHostVariable() erstellt).

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben.
<i>Namen</i>	Ein Zeiger auf eine VARIANT, in der die Ergebnisfeldgruppe gespeichert werden soll.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, kann die Funktion GetLastErrorString() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetHostVariableTypes()

short GetHostVariableTypes(long *AbfrageID*, VARIANT* *Typnamen*)

Beschreibung

Diese Funktion gibt eine Feldgruppe (*Array*) der Datentypen aller Host-Variablen zurück, auf die in der angegebenen Abfrage verwiesen wird. Die Abfrage muss eine statische Abfrage sein, mit der auf Host-Variablen verwiesen wird (entweder in der QMF-Abfrage gespeichert oder durch AddHostVariable() erstellt). Siehe hierzu AddHostVariable(). Dort finden Sie eine Liste aller Datentypen, die zurückgegeben werden können.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben.
<i>Typen</i>	Ein Zeiger auf eine VARIANT, in der die Ergebnisfeldgruppe gespeichert werden soll.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, kann die Funktion GetLastErrorString() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetLastErrorString()

BSTR GetLastErrorString()

Beschreibung

Diese Funktion gibt eine Zeichenfolge mit Informationen zum zuletzt aufgetretenen Fehler zurück. Wenn Sie diese Funktion nach einer Funktion aufrufen, die erfolgreich (ohne Fehler) ausgeführt wurde, gibt diese Funktion anschließend Informationen zu dem letzten Fehler zurück, der während eines vorhergehenden Funktionsaufrufs auftrat. Um Missverständnisse zu vermeiden, sollten Sie diese Funktion immer nur direkt nach dem Aufruf einer Funktion, die einen Fehler zurückgegeben hat, aufrufen.

Rückgabewert

Eine Zeichenfolge mit Fehlerinformationen. Wenn seit der Erstellung des QMF-API-Objekts keine Fehler aufgetreten sind, wird NULL zurückgegeben.

Siehe auch

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

GetLastErrorType()

short GetLastErrorType()

Beschreibung

Diese Funktion gibt den Typ des zuletzt aufgetretenen Fehlers zurück. Wenn Sie diese Funktion nach einer Funktion aufrufen, die erfolgreich (ohne Fehler) ausgeführt wurde, gibt diese Funktion anschließend Informationen zu dem letzten Fehler zurück, der während eines vorhergehenden Funktionsaufrufs auftrat. Um Missverständnisse zu vermeiden, sollten Sie diese Funktion immer nur direkt nach dem Aufruf einer Funktion, die einen Fehler zurückgegeben hat, aufrufen.

Rückgabewert

Der zurückgegebene Wert gibt den Typ des Fehlers an:

Wert	Bedeutung
0 (RS_ERROR_NONE)	Seit dem Erstellen des QMF für Windows-API-Objekts sind keine Fehler aufgetreten.
1 (RS_ERROR_SQL)	Ein SQL-Fehler ist aufgetreten. Ist der Fehler während des Aufrufs einer Funktion aufgetreten, die das Argument <i>AbfrageID</i> verwendet, rufen Sie die Funktion <i>Close()</i> auf, um diese Abfrage zu schließen. Es wird keine Rollback-Operation ausgeführt. Sie können das QMF für Windows-API-Objekt weiter verwenden, auch dann, wenn Sie auf zusätzliche Fehler stoßen.
2 (RS_ERROR_USER_CANCEL)	Ein Benutzer hat eine Operation abgebrochen, in der Regel durch Anklicken von Abbrechen im Vorgangsfenster . Hierdurch führt QMF für Windows eine implizite Rollback-Operation aus, wodurch alle ausstehenden Abfrage-IDs ungültig gemacht werden, und löscht die Verbindung zur Datenbank. Sie müssen <i>InitializeServer()</i> oder <i>ReinitializeServer()</i> aufrufen, wenn Sie fortfahren wollen.
3 (RS_ERROR_FATAL_GOV)	Ein schwerwiegender Governor-Fehler ist aufgetreten. Ein Grund hierfür ist, dass bei der QMF für Windows-API eine Zeitlimitüberschreitung (<i>Close()</i>) aufgetreten ist, da die maximal zulässige Inaktivitätszeitspanne überschritten wurde. Hierdurch führt QMF für Windows eine implizite Rollback-Operation aus, wodurch alle ausstehenden Abfrage-IDs ungültig gemacht werden, und löscht die Verbindung zur Datenbank. Sie müssen <i>InitializeServer()</i> oder <i>ReinitializeServer()</i> aufrufen, wenn Sie fortfahren wollen.

4 (RS_ERROR_NONFATAL_GOV)	Ein behebbarer Governor-Fehler ist aufgetreten. Es wurde entweder die maximale Anzahl abzurufender Zeilen überschritten oder das SQL-Verb ist nicht zulässig. Ist der Fehler während des Aufrufs einer Funktion aufgetreten, die das Argument <i>AbfrageID</i> verwendet, rufen Sie die Funktion <i>Close()</i> auf, um diese Abfrage zu schließen. Es wird keine Rollback-Operation ausgeführt und die Verbindung zur Datenbank bleibt unbeeinträchtigt. Sie können also weiter mit dem QMF für Windows-API-Objekt arbeiten.
5 (RS_ERROR_OTHER)	Ein allgemeiner Fehler ist aufgetreten. Es wird keine Rollback-Operation ausgeführt. Sie können das QMF für Windows-API-Objekt weiter verwenden, auch dann, wenn Sie auf zusätzliche Fehler stoßen.

Siehe auch

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

GetLastSQLCode()

long GetLastSQLCode()

Beschreibung

Diese Funktion gibt den SQL-Code des zuletzt aufgetretenen Fehlers zurück. Wenn Sie diese Funktion nach einer Funktion aufrufen, die erfolgreich (ohne Fehler) ausgeführt wurde, gibt diese Funktion anschließend Informationen zu dem letzten Fehler zurück, der während eines vorhergehenden Funktionsaufrufs auftrat. Um Missverständnisse zu vermeiden, sollten Sie diese Funktion immer nur direkt nach dem Aufruf einer Funktion, die einen Fehler zurückgegeben hat, aufrufen.

Rückgabewert

Der SQL-Code des zuletzt aufgetretenen Fehlers. Wenn seit der Erstellung des QMF für Windows-API-Objekts keine Fehler aufgetreten sind, oder wenn der letzte Fehler kein SQL-Fehler war, wird null zurückgegeben.

Siehe auch

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

GetLastSQLError()

VARIANT GetLastSQLError()

Beschreibung

Diese Funktion gibt detaillierte SQL-Fehlerinformationen zum zuletzt aufgetretenen Fehler zurück. Wenn Sie diese Funktion nach einer Funktion aufrufen, die erfolgreich (ohne Fehler) ausgeführt wurde, gibt diese Funktion anschließend Informationen zu dem letzten Fehler zurück, der während eines vorhergehenden Funktionsaufrufs auftrat. Um Missverständnisse zu vermeiden, sollten Sie diese Funktion immer nur direkt nach dem Aufruf einer Funktion, die einen Fehler zurückgegeben hat, aufrufen.

Rückgabewert

Eine Feldgruppe (Variant-Typ VT_ARRAY | VT_VARIANT), die Fehlerinformationen enthält. Wenn seit der Erstellung des QMF für Windows-API-Objekts keine Fehler aufgetreten sind, oder wenn der letzte Fehler kein SQL-Fehler war, wird leer (Variant-Typ VT_EMPTY) zurückgegeben. Die Feldgruppe hat die folgende Formatierung:

Element	Typ	Inhalt
0	long (VT_I4)	Code
1	string (VT_BSTR)	State
2	string (VT_BSTR)	ErrProc
3	string (VT_BSTR)	RDBName
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	string (VT_BSTR)	Warn0
11	string (VT_BSTR)	Warn1
12	string (VT_BSTR)	Warn2
13	string (VT_BSTR)	Warn3
14	string (VT_BSTR)	Warn4
15	string (VT_BSTR)	Warn5
16	string (VT_BSTR)	Warn6
17	string (VT_BSTR)	Warn7
18	string (VT_BSTR)	Warn8

19	string (VT_BSTR)	Warn9
20	string (VT_BSTR)	WarnA
21	string (VT_BSTR)	MessageTokens

Siehe auch

GetLastErrorString()
 GetLastErrorType()
 GetLastErrorSQLCode()
 GetLastErrorSQLState()

GetLastSQLState()

BSTR GetLastErrorSQLState()

Beschreibung

Diese Funktion gibt den SQL-Status des zuletzt aufgetretenen Fehlers zurück. Wenn Sie diese Funktion nach einer Funktion aufrufen, die erfolgreich (ohne Fehler) ausgeführt wurde, gibt diese Funktion anschließend Informationen zu dem letzten Fehler zurück, der während eines vorhergehenden Funktionsaufrufs auftrat. Um Missverständnisse zu vermeiden, sollten Sie diese Funktion immer nur direkt nach dem Aufruf einer Funktion, die einen Fehler zurückgegeben hat, aufrufen.

Rückgabewert

Eine Zeichenfolge mit dem SQL-Code des zuletzt aufgetretenen Fehlers. Wenn seit der Erstellung des QMF für Windows-API-Objekts keine Fehler aufgetreten sind, oder wenn der letzte Fehler kein SQL-Fehler war, wird Null zurückgegeben.

Siehe auch

GetLastErrorString()
 GetLastErrorType()
 GetLastErrorSQLCode()
 GetLastErrorSQLError()

GetOption()

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

Beschreibung

Ruft den Wert der angegebenen Option in QMF für Windows ab.

Parameter

Name	Beschreibung
<i>Option</i>	Gibt an, welche Option abgerufen werden soll.

Wert	Bedeutung
0 (RSO_SERVER_DEFINITION_FILE)	Name der Server-Definitionsdatei.
1 (RSO_CPIC_DLL)	Dateiname der CPI-C-Provider-DLL.
2 (RSO_CPIC_TIMEOUT_WARNING)	CPI-C-Zeitlimit für Warnung (in Sekunden). Dieses Limit wird nicht von der QMF für Windows-API verwendet.
3 (RSO_CPIC_TIMEOUT_CANCEL)	CPI-C-Zeitlimit für das Abbrechen (in Sekunden).
4 (RSO_TCP_TIMEOUT_WARNING)	TCP-Zeitlimit für Warnung (in Sekunden). Dieses Limit wird nicht von der QMF für Windows-API verwendet.
5 (RSO_TCP_TIMEOUT_CANCEL)	TCP-Zeitlimit für das Abbrechen (in Sekunden).
6 (RSO_DISPLAY_NULLS_STRING)	Die für die Anzeige von Nullwerten zu verwendende Zeichenfolge.
7 (RSO_ENTER_NULLS_STRING)	Die für die Eingabe von Nullwerten zu verwendende Zeichenfolge.
8 (RSO_ENTER_DEFAULTS_STRING)	Die für die Eingabe von Standardwerten zu verwendende Zeichenfolge.
9 (RSO_TRACE_FILE_1)	Name der Trace-Datei 1.
10 (RSO_TRACE_FILE_2)	Name der Trace-Datei 2.
11 (RSO_TCP_TRACE_LEVEL)	TCP-Trace-Stufe.
12 (RSO_CPIC_TRACE_LEVEL)	CPI-C-Trace-Stufe.
13 (RSO_DDM_TRACE_LEVEL)	DDM-Trace-Stufe.
<i>Wert</i>	Ein Zeiger auf eine VARIANT, in der die Ergebnisse gespeichert werden. Das Ergebnis ist eine Feldgruppe (Variant-Typ VT_ARRAY VT_VARIANT), die für jede Spalte im Ergebnis einen Wert enthält. Rufen Sie GetColumnCount() auf, um die Anzahl der Werte in der Feldgruppe (<i>Array</i>) festzustellen. Sie müssen die VARIANT korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen VariantInit() aufrufen.

Anmerkung: Aufgrund eines Programmfehlers in Microsoft Excel 7.0 und Microsoft Access 7.0 — sowie möglicherweise anderen 32-Bit-Produkten von Microsoft, die die Visual Basic for Applications (VBA) verwenden -, werden Zeichenfolgedaten (*String Data*) in Variant-Variablen, die aus QMF für Windows empfangen werden, nicht von Unicode (von OLE verwendet) in ANSI (von

VBA verwendet) umgesetzt. Wenn dieser Fehler auftritt, wird nur das erste Zeichen der Zeichenfolge angezeigt. Um den Fehler zu beheben, müssen Sie die Variable gleich einer leeren Zeichenfolge definieren, bevor Sie die QMF für Windows-Funktion aufrufen, die die Variable verwendet.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()` oder `GetLastErrorType()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

`SetOption()`

GetOptionEx()

short `GetOptionEx(short Option)`

Beschreibung

Ruft den Wert der angegebenen Option in QMF für Windows ab. Wenn der Wert der Option zurückgegeben wird, müssen Sie die das Merkmal *Option* für die Daten abfragen.

Parameter

Name	Beschreibung
<i>Option</i>	Die Optionswerte sind die gleichen wie die für den Aufruf <code>GetOption()</code> .

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()` oder `GetLastErrorType()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

`GetOption()`

`SetOption()`

GetProcText()

BSTR `GetProcText(long ProcID)`

Beschreibung

Diese Funktion gibt den Text zurück, der für die angegebene Prozedur nach der Variablenersetzung ausgeführt wird. Sie sollten `SetProcVariable()` verwenden, um den Wert von allen in der Prozedur verwendeten Variablen zu setzen, bevor Sie diese Funktion aufrufen.

Parameter

Name	Beschreibung
<i>ProzID</i>	Die ID der Prozedur, wie von InitializeProc() zurückgegeben.

Rückgabewert

Falls erfolgreich, wird eine Zeichenfolge zurückgegeben, die den Prozedurtext enthält. Falls nicht erfolgreich, wird NULL zurückgegeben. Wird der Wert NULL zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetProcVariables()

short GetProcVariables(long *ProzID*, VARIANT* *Variablen*)

Beschreibung

Ruft den Wert der angegebenen Option in QMF für Windows ab.

Parameter

Name	Beschreibung
<i>ProzID</i>	Die ID der Prozedur, wie von InitializeProc() zurückgegeben.
<i>Wert</i>	Ein Zeiger auf eine VARIANT, in der die Ergebnisse gespeichert werden. Das Ergebnis ist eine Feldgruppe mit Zeichenfolgen (Variant-Typ VT_ARRAY VT_BSTR), wobei jede Zeichenfolge den Namen einer Variablen enthält. Enthält die Prozedur keine Variablen, ist das Ergebnis ein leerer Wert (Variant-Typ VT_EMPTY). Sie müssen die VARIANT korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen VariantInit() aufrufen.

Anmerkung: Aufgrund eines Programmfehlers in Microsoft Excel 7.0 und Microsoft Access 7.0 — sowie möglicherweise anderen 32-Bit-Produkten von Microsoft, die die Visual Basic for Applications (VBA) verwenden -, werden Zeichenfolgedaten (*String Data*) in Variant-Variablen, die aus QMF für Windows empfangen werden, nicht von Unicode (von OLE verwendet) in ANSI (von VBA verwendet) umgesetzt. Wenn dieser Fehler auftritt, wird nur das erste Zeichen der Zeichenfolge angezeigt. Um den Fehler zu beheben, müssen Sie die Variable gleich einer leeren Zeichenfolge definieren, bevor Sie die QMF für Windows-Funktion aufrufen, die die Variable verwendet.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Enthält die Prozedur keine Variablen, ist der Rückgabewert RS_NO_ERROR_NO_DATA (-1). Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetQMFOBJECTINFO()

short GetQMFOBJECTINFO(BSTR *EignerUndName*, short *Typ*, short *Zeit*, VARIANT* *Wert*)

Beschreibung

Diese Funktion gibt Informationen zu einem QMF-Objekt (entweder ein Format oder eine Abfrage) zurück. Die zurückgegebenen Informationen werden mit den Parametern *Typ* und *Zeit* angegeben.

Parameter

Name	Beschreibung
<i>EignerUndName</i>	Eine Zeichenfolge, die den Eigner und den Namen (getrennt durch einen Punkt) des Objekts umfasst, für das Sie Informationen abrufen wollen. Zum Beispiel: Udo.Abfrage2
<i>Wert</i>	Ein Zeiger auf eine VARIANT, in der die Ergebnisse gespeichert werden. Für RSI_TIMESUSED, RSI_TIMESRUN, RSI_TIMESCANCELLED und RSI_LEVEL hat das Ergebnis den Datentyp "short" (Variant-Typ VT_I2). Für RSI_RESTRICTED hat das Ergebnis den Datentyp "Boolean" (Variant-Typ VT_BOOL). Für alle anderen Werte hat das Ergebnis den Datentyp "string" (Variant-Typ VT_BSTR). Sie müssen die VARIANT korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen VariantInit() aufrufen.

Anmerkung: Aufgrund eines Programmfehlers in Microsoft Excel 7.0 und Microsoft Access 7.0 — sowie möglicherweise anderen 32-Bit-Produkten von Microsoft, die die Visual Basic for Applications (VBA) verwenden -, werden Zeichenfolgedaten (*String Data*) in Variant-Variablen, die aus QMF für Windows empfangen werden, nicht von Unicode (von OLE verwendet) in ANSI (von VBA verwendet) umgesetzt. Wenn dieser Fehler auftritt, wird nur das erste Zeichen der Zeichenfolge angezeigt. Um den Fehler zu beheben, müssen Sie die Variable gleich einer leeren Zeichenfolge definieren, bevor Sie die QMF für Windows-Funktion aufrufen, die die Variable verwendet.

<i>Typ</i>	Gibt den Typ der abzurufenden Informationen an.
Wert	Bedeutung
0 (RSI_COMMENT)	Kommentar.
1 (RSI_LEVEL)	Stufe.
2 (RSI_TYPE)	Typ.
3 (RSI_SUBTYPE)	Subtyp.
4 (RSI_RESTRICTED)	Eingeschränkt.
5 (RSI_MODEL)	Modell.
6 (RSI_TIMESUSED)	Anzahl der Verwendungen.
7 (RSI_TIMESRUN)	Anzahl der Ausführungen.
8 (RSI_TIMESCANCELLED)	Anzahl der Abbrüche.
9 (RSI_DATE)	Datum der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
10 (RSI_TIME)	Zeit der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
11 (RSI_USERID)	Benutzer-ID bei der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
12 (RSI_SQLID)	SQL-ID bei der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
13 (RSI_ENVIRONMENT)	Umgebung bei der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
14 (RSI_MODE)	Modus bei der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
15 (RSI_COMMAND)	Befehl bei der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
<i>Zeit</i>	Zeitpunkt der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
Wert	Bedeutung
0 (RST_FIRSTUSED)	Erste Verwendung.
1 (RST_LASTUSED)	Letzte Verwendung.
2 (RST_LASTMODIFIED)	Letzte Änderung.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetQMFObjectInfoEx()

short GetQMFObjectInfoEx(BSTR *EignerUndName*, short *Typ*, short *Zeit*)

Beschreibung

Diese Funktion gibt Informationen zu einem QMF-Objekt zurück. Die zurückgegebenen Informationen werden mit den Parametern *Typ* und *Zeit* angegeben. Nach dem Aufruf dieser Funktion kann die Eigenschaft *QMFObjectInfo* nach dem zurückgegebenen Wert abgefragt werden.

Parameter

Name	Beschreibung
<i>EignerUndName</i>	Eine Zeichenfolge, die den Eigner und den Namen (getrennt durch einen Punkt) des Objekts umfasst, für das Sie Informationen abrufen wollen. Zum Beispiel: Udo.Abfrage2
<i>Typ</i>	Gibt den Typ der abzurufenden Informationen an.

Wert	Bedeutung
0 (RSI_COMMENT)	Kommentar.
1 (RSI_LEVEL)	Stufe.
2 (RSI_TYPE)	Typ.
3 (RSI_SUBTYPE)	Subtyp.
4 (RSI_RESTRICTED)	Eingeschränkt.
5 (RSI_MODEL)	Modell.
6 (RSI_TIMESUSED)	Anzahl der Verwendungen.
7 (RSI_TIMESRUN)	Anzahl der Ausführungen.
8 (RSI_TIMESCANCELLED)	Anzahl der Abbrüche.
9 (RSI_DATE)	Datum der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
10 (RSI_TIME)	Zeit der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
11 (RSI_USERID)	Benutzer-ID bei der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.

12 (RSI_SQLID)	SQL-ID bei der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
13 (RSI_ENVIRONMENT)	Umgebung bei der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
14 (RSI_MODE)	Modus bei der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
15 (RSI_COMMAND)	Befehl bei der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.
<i>Zeit</i>	Zeitpunkt der ersten Verwendung, der letzten Verwendung oder der letzten Änderung.

Wert	Bedeutung
0 (RST_FIRSTUSED)	Erste Verwendung.
1 (RST_LASTUSED)	Letzte Verwendung.
2 (RST_LASTMODIFIED)	Letzte Änderung.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetQMFOBJECTList()

short GetQMFOBJECTList(BSTR *Eigner*, BSTR *Name*, short *Typ*, VARIANT* *Liste*)

Beschreibung

Diese Funktion gibt eine Feldgruppe (*Array*) der Namen aller QMF-Objekte zurück, die mit in den Parametern *Eigner* und *Name* angegebenen Mustern übereinstimmen.

Parameter

Name	Beschreibung
<i>Eigner</i>	Eine Zeichenfolge, die den Eigner der Objekte enthält, die in der zurückgegebenen Liste aufgenommen werden sollen.
<i>Name</i>	Eine Zeichenfolge, die die Namen der Objekte enthält, die in der zurückgegebenen Liste aufgenommen werden sollen.

<i>Liste</i>	Ein Zeiger auf eine VARIANT, in der die Ergebnisse gespeichert werden. Das Ergebnis ist eine Feldgruppe mit Zeichenfolgen (Variant-Typ VT_ARRAY VT_BSTR), die alle das Format Eigner.Name aufweisen. Werden keine übereinstimmenden QMF für Windows-Abfragen gefunden, ist das Ergebnis ein leerer Wert (Variant-Typ VT_EMPTY). Sie müssen die VARIANT korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen VariantInit() aufrufen.
--------------	--

Anmerkung: Aufgrund eines Programmfehlers in Microsoft Excel 7.0 und Microsoft Access 7.0 — sowie möglicherweise anderen 32-Bit-Produkten von Microsoft, die die Visual Basic for Applications (VBA) verwenden -, werden Zeichenfolgedaten (*String Data*) in Variant-Variablen, die aus QMF für Windows empfangen werden, nicht von Unicode (von OLE verwendet) in ANSI (von VBA verwendet) umgesetzt. Wenn dieser Fehler auftritt, wird nur das erste Zeichen der Zeichenfolge angezeigt. Um den Fehler zu beheben, müssen Sie die Variable gleich einer leeren Zeichenfolge definieren, bevor Sie die QMF für Windows-Funktion aufrufen, die die Variable verwendet.

<i>Typ</i>	Gibt die Typen der QMF-Objekte an, die in die Liste aufgenommen werden sollen. Diese Werte können zusammengefügt werden, um mehrere Objekttypen anzugeben.
------------	--

Wert	Bedeutung
2048 (RSQ_MASK_QUERIES)	QMF-Abfragen in die Liste aufnehmen.
1024 (RSQ_MASK_FORMS)	QMF-Formate in die Liste aufnehmen.
512 (RSQ_MASK_PROCS)	QMF-Prozeduren in die Liste aufnehmen.
256 (RSQ_MASK_TABLES)	Tabellen in die Liste aufnehmen.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Werden keine übereinstimmenden QMF-Objekte gefunden, ist der Rückgabewert Null. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() oder GetLastErrorSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetQMFObjectListEx()

short GetQMFObjectListEx(BSTR *Eigner*, BSTR *Name*, short *Index*)

Beschreibung

Diese Funktion gibt den Namen des QMF-Objekts zurück, das mit in den Parametern *Eigner* und *Name* angegebenen Muster übereinstimmen, auf die durch den Parameter *Index* verwiesen wird. Nach dem Aufruf dieser Funktion kann die Eigenschaft *Wert* nach dem zurückgegebenen Wert abgefragt werden.

Parameter

Name	Beschreibung
<i>Eigner</i>	Eine Zeichenfolge, die den Eigner der Objekte enthält, die in der zurückgegebenen Liste aufgenommen werden sollen.
<i>Name</i>	Eine Zeichenfolge, die die Namen der Objekte enthält, die in der zurückgegebenen Liste aufgenommen werden sollen.
<i>Index</i>	Der Index der Liste der QMF-Objekte, die mit dem Muster übereinstimmen.
<i>Typ</i>	Gibt die Typen der QMF-Objekte an, die in die Liste aufgenommen werden sollen. Diese Werte können zusammengefügt werden, um mehrere Objekttypen anzugeben.
Wert	Bedeutung
2048 (RSQ_MASK_QUERIES)	QMF-Abfragen in die Liste aufnehmen.
1024 (RSQ_MASK_FORMS)	QMF-Formate in die Liste aufnehmen.
512 (RSQ_MASK_PROCS)	QMF-Prozeduren in die Liste aufnehmen.
256 (RSQ_MASK_TABLES)	Tabellen in die Liste aufnehmen.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Werden keine übereinstimmenden QMF-Objekte gefunden, ist der Rückgabewert RS_OUTOFRANGE. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetQMFProcText()

BSTR GetQMFProcText(BSTR *EignerUndName*)

Beschreibung

Diese Funktion gibt den Text zurück, der für die angegebene Prozedur nach der Variablenersetzung ausgeführt wird. Sie sollten `SetProcVariable()` verwenden, um den Wert von allen in der Prozedur verwendeten Variablen zu setzen, bevor Sie diese Funktion aufrufen.

Parameter

Name	Beschreibung
<i>EignerUndName</i>	Eine Zeichenfolge, die den Eigner und den Namen (getrennt durch einen Punkt) des Objekts umfasst, das Sie löschen wollen. Zum Beispiel: Udo.Proz2

Rückgabewert

Eine Zeichenfolge, die den Text der abgerufenen Prozedur enthält, oder NULL, wenn die Prozedur nicht abgerufen werden konnte. Wird der Wert NULL zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetQMFQueryText()

BSTR GetQMFQueryText(BSTR *EignerUndName*)

Beschreibung

Diese Funktion ruft den SQL-Text ab, der in der angegebenen Abfrage gespeichert ist.

Parameter

Name	Beschreibung
<i>EignerUndName</i>	Eine Zeichenfolge, die den Eigner und den Namen (getrennt durch einen Punkt) des Objekts umfasst, das Sie löschen wollen. Zum Beispiel: Udo.Abfrage2

Rückgabewert

Eine Zeichenfolge, die den Text der abgerufenen Abfrage enthält, oder NULL, wenn die Abfrage nicht abgerufen werden konnte. Wird der Wert NULL zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetQueryText()

BSTR GetQueryText(long *AbfrageID*)

Beschreibung

Diese Funktion gibt den SQL-Text zurück, der für die angegebene Abfrage nach der Variablensetzung ausgeführt wird. Sie sollten `SetVariable()` verwenden, um den Wert von allen in der Abfrage verwendeten Variablen zu setzen, bevor Sie diese Funktion aufrufen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.

Anmerkung: Der Abfragetext ist für statische SQL-Anweisungen nicht verfügbar. Für Abfrage-IDs, die von `InitializeStaticQuery()` zurückgegeben werden, gibt `GetQueryText()` eine leere Zeichenfolge zurück.

Rückgabewert

Falls erfolgreich, wird eine Zeichenfolge zurückgegeben, die den SQL-Text enthält. Falls nicht erfolgreich, wird NULL zurückgegeben. Wird der Wert NULL zurückgegeben, können die Funktionen `GetLastErrorString()` oder `GetLastErrorType()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetQueryVerb()

BSTR `GetQueryVerb(long AbfrageID)`

Beschreibung

Diese Funktion gibt eine Zeichenfolge mit dem SQL-Verb zurück, das Sie in der Abfrage verwendet haben.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.

Anmerkung: Das Abfrageverb ist für statische SQL-Anweisungen nicht verfügbar. Für Abfrage-IDs, die von `InitializeStaticQuery()` zurückgegeben werden, gibt `GetQueryVerb()` eine leere Zeichenfolge zurück.

Rückgabewert

Ist die Abfrage erfolgreich, wird eine Zeichenfolge zurückgegeben, die das SQL-Verb enthält. Falls nicht erfolgreich, wird NULL zurückgegeben. Wird der Wert NULL zurückgegeben, können die Funktionen `GetLastErrorString()` oder `GetLastErrorType()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetResourceLimit()

short `GetResourceLimit(short Ressource, long* Wert)`

Beschreibung

Diese Funktion ruft den angeforderten Ressourcengrenzwert ab. Sie müssen InitializeServer() vor dem Aufrufen dieser Funktion aufrufen, da Ressourcengrenzwerte jeweils für einen Server bearbeitet werden.

Parameter

Name	Beschreibung
<i>Ressource</i>	Die Ressourcenwerte umfassen:
Wert	Bedeutung
0 (RSR_IDLE_CONNECTION_TIMEOUT)	Zeitlimit für Verbindungen im Leerlauf (in Sekunden).
1 (RSR_IDLE_QUERY_TIMEOUT_CANCEL)	Zeitlimit für Abfragen im Leerlauf (in Sekunden).
2 (RSR_IDLE_QUERY_TIMEOUT_WARNING)	Zeitlimit für Abfragen im Leerlauf (in Sekunden). Dies ist das Zeitlimit für Warnungen; es wird für die QMF für Windows-API nicht aktiviert.
3 (RSR_SERVER_RESPONSE_TIMEOUT_CANCEL)	Zeitlimit für Server (in Sekunden).
4 (RSR_SERVER_RESPONSE_TIMEOUT_WARNING)	Zeitlimit für Server (in Sekunden). Dies ist das Zeitlimit für Warnungen; es wird für die QMF für Windows-API nicht aktiviert.
5 (RSR_MAX_ROWS_TO_FETCH_CANCEL)	Höchstzahl der abrufbaren Zeilen.
6 (RSR_MAX_ROWS_TO_FETCH_WARNING)	Höchstzahl der abrufbaren Zeilen. Dies ist das Zeitlimit für Warnungen; es wird für die QMF für Windows-API nicht aktiviert.
7 (RSR_MAX_BYTES_TO_FETCH_CANCEL)	Maximale Anzahl abzurufender Byte.

8 (RSR_MAX_BYTES_TO_FETCH_WARNING)	Maximale Anzahl abzurufender Byte. Dies ist das Zeitlimit für Warnungen; es wird für die QMF für Windows-API nicht aktiviert.
9 (RSR_MAX_CONNECTIONS)	Höchstzahl der zulässigen Verbindungen zu dem Datenbank-Server.
10 (RSR_ALLOW_SERVER_ACCESS_UI)	Ist der Zugriff auf den Datenbank-Server von der QMF für Windows-Schnittstelle aus zulässig?
11 (RSR_ALLOW_SERVER_ACCESS_API)	Ist der Zugriff auf den Datenbank-Server von der QMF für Windows-API aus zulässig?
12 (RSR_FETCH_ALL_ROWS)	Alle Zeilen automatisch abrufen?
13 (RSR_CONFIRM_UPDATES)	Datenbank-Server-Aktualisierungen bestätigen? Diese Option hat keine Auswirkungen auf die QMF für Windows-API; Datenbankaktualisierungen werden für die QMF für Windows-API nicht bestätigt.
14 (RSR_SUMMARY_TRACKING)	Ist die Übersichtsobjektüberwachung aktiviert?
15 (RSR_DETAILED_TRACKING)	Ist die detaillierte Objektüberwachung aktiviert?
16 (RSR_SQL_TRACKING)	Ist die SQL-Textüberwachung aktiviert?
17 (RSR_ADHOC_TRACKING)	Ist die Ad-Hoc-Objektüberwachung aktiviert?
18 (RSR_ALLOW_ACQUIRE)	Ist das SQL-Verb ACQUIRE zulässig?
19 (RSR_ALLOW_ALTER)	Ist das SQL-Verb ALTER zulässig?

20 (RSR_ALLOW_COMMENT)	Ist das SQL-Verb COMMENT zulässig?
21 (RSR_ALLOW_CREATE)	Ist das SQL-Verb CREATE zulässig?
22 (RSR_ALLOW_DELETE)	Ist das SQL-Verb DELETE zulässig?
23 (RSR_ALLOW_DROP)	Ist das SQL-Verb DROP zulässig?
24 (RSR_ALLOW_EXPLAIN)	Ist das SQL-Verb EXPLAIN zulässig?
25 (RSR_ALLOW_GRANT)	Ist das SQL-Verb GRANT zulässig?
26 (RSR_ALLOW_INSERT)	Ist das SQL-Verb INSERT zulässig?
27 (RSR_ALLOW_LABEL)	Ist das SQL-Verb LABEL zulässig?
28 (RSR_ALLOW_LOCK)	Ist das SQL-Verb LOCK zulässig?
29 (RSR_ALLOW_REVOKE)	Ist das SQL-Verb REVOKE zulässig?
30 (RSR_ALLOW_SELECT)	Ist das SQL-Verb SELECT zulässig?
31 (RSR_ALLOW_SET)	Ist das SQL-Verb SET zulässig?
32 (RSR_ALLOW_SIGNAL)	Ist das SQL-Verb SIGNAL zulässig?
33 (RSR_ALLOW_UPDATE)	Ist das SQL-Verb UPDATE zulässig?
34 (RSR_ALLOW_CALL)	Ist das SQL-Verb CALL zulässig?
35 (RSR_ALLOW_SAVE_DATA)	Ist der Befehl 'Daten speichern' zulässig?
36 (RSR_SAVE_DATA_TABLE_SPACE_NAME)	Der Standard-Collection-Name zum Binden von Paketen.
37 (RSR_SAVE_DATA_TABLE_SPACE_NAME_OVERRIDE)	Kann der der Standard-tabellenbereichsname für den Befehl 'Daten speichern' vom Benutzer überschrieben werden?

38 (RSR_ALLOW_BIND_PACKAGE)	Binden von Paketen zulassen?
39 (RSR_DEF_COLLECTION)	Der Standard-Collection-Name zum Binden von Paketen.
40 (RSR_DEF_COLLECTION_OVERRIDE)	Kann der Standard-Collection-Name zum Binden von Paketen vom Benutzer überschrieben werden?
41 (RSR_DEF_ISOLATION_LEVEL)	Die Standardisolationsstufe zum Binden von Paketen.
42 (RSR_DEF_ISOLATION_LEVEL_OVERRIDE)	Kann die Standardisolationsstufe zum Binden von Paketen vom Benutzer überschrieben werden?
43 (RSR_ALLOW_TABLE_EDIT)	Verwendung des Tabelleneditors zulässig?
44 (RSR_ALLOW_EXPORT)	Exportieren von Daten zulassen?
45 (RSR_ALLOW_SAVED_QUERIES_ONLY)	Darf der Benutzer lediglich gespeicherte Abfragen ausführen?
46 (RSR_ALLOW_DROP_PACKAGE)	Freigeben von Paketen zulassen?
47 (RSR_QUERY_ISOLATION_LEVEL)	Die beim Ausführen von Abfragen zu verwendende Isolationsstufe.
48 (RSR_ACCOUNT_STRING)	Die Zeichenfolge, mit der Kontoinformationen angegeben werden, die beim Herstellen der Verbindung zum Datenbank-Server übergeben werden.
49 (RSR_ACCOUNT_OVERRIDE)	Kann die Zeichenfolge, mit der Kontoinformationen angegeben werden (die beim Herstellen der Verbindung zum Datenbank-Server übergeben werden), durch den Benutzer überschrieben werden?

Wert	Ein Zeiger auf den Datentyp "long", in dem das Ergebnis gespeichert wird. Das Ergebnis ist der Wert des angeforderten Ressourcengrenzwerts. Für Boolesche Werte ist das Ergebnis ungleich Null bei wahr (true) und null bei falsch (false). Bei RSR_SAVE_DATA_TABLE_SPACE_NAME, RSR_DEF_COLLECTION und RSR_ACCOUNT_STRING wird -1 zurückgegeben und in der Eigenschaft <i>Ressourcengrenzwert</i> kann der zurückgegebene Zeichenfolgenwert abgerufen werden.
------	---

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()` oder `GetLastErrorType()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetResourceLimitEx()

short `GetResourceLimitEx(short Ressource)`

Beschreibung

Diese Funktion ruft den angeforderten Ressourcengrenzwert ab. Sie müssen `InitializeServer()` vor dem Aufrufen dieser Funktion aufrufen, da Ressourcengrenzwerte jeweils für einen Server bearbeitet werden. Nach dem Aufrufen dieser Funktion fragen Sie die Eigenschaft *Ressourcengrenzwert* ab, um das Ergebnis zu ermitteln.

Parameter

Name	Beschreibung
<i>Ressource</i>	Die Ressourcenwerte sind die gleichen wie die für den Aufruf <code>GetResourceLimit()</code> .

Anmerkung: Das Abfrageverb ist für statische SQL-Anweisungen nicht verfügbar. Für Abfrage-IDs, die von `InitializeStaticQuery()` zurückgegeben werden, gibt `GetQueryVerb()` eine leere Zeichenfolge zurück.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()` oder `GetLastErrorType()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetRowCount()

long GetRowCount(long *AbfrageID*)

Beschreibung

Diese Funktion gibt die Anzahl der Zeilen zurück, die sich momentan im internen Puffer von QMF für Windows befinden. Die Anzahl kann größer sein als die Zeilenanzahl, die mit der Funktion FetchNextRow() oder FetchNextRows() abgerufen wurde, da QMF für Windows Daten, die von der Datenbank empfangen wurden, in einen Puffer stellt.

Diese Funktion gibt die Anzahl der Zeilen zurück, die bereits von der Datenbank abgerufen wurden. Wenn Sie die Gesamtanzahl der Zeilen in der Ergebnisgruppe abrufen wollen, können Sie folgendes ausführen:

- Die Funktion CompleteQuery() aufrufen und alle Zeilen abrufen, die FetchNextRow() oder FetchNextRows() verwenden.
- FetchAllRows = TRUE angeben, wenn Sie die Funktion Open() aufrufen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben.

Rückgabewert

Die Anzahl der Zeilen, falls erfolgreich (0, wenn keine Zeilen abgerufen wurden), oder -1, falls nicht erfolgreich. Wird der Wert 1 zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetServerList()

short GetServerList(VARIANT* *Liste*)

Beschreibung

Diese Funktion gibt eine Feldgruppe (*Array*) zurück, die die Namen der Datenbank-Server enthält, die in der Server-Definitionsdatei (SDF, Server Definition File) von QMF für Windows definiert sind. Sie müssen einen Datenbank-Server in der SDF-Datei definieren, wenn Sie auf ihn über die QMF für Windows-API zugreifen wollen.

Parameter

Name	Beschreibung
<i>Liste</i>	Ein Zeiger auf eine VARIANT, in der die Ergebnisse gespeichert werden. Das Ergebnis ist eine Feldgruppe mit Zeichenfolgen (Variant-Typ VT_ARRAY VT_BSTR), wobei jede Zeichenfolge den Namen eines Datenbank-Servers enthält. Wenn Sie keine Datenbank-Server definiert haben, ist das Ergebnis ein leerer Wert (Variant-Typ VT_EMPTY). Sie müssen die VARIANT korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen VariantInit() aufrufen.

Anmerkung: Aufgrund eines Programmfehlers in Microsoft Excel 7.0 und Microsoft Access 7.0 — sowie möglicherweise anderen 32-Bit-Produkten von Microsoft, die die Visual Basic for Applications (VBA) verwenden -, werden Zeichenfolgedaten (*String Data*) in Variant-Variablen, die aus QMF für Windows empfangen werden, nicht von Unicode (von OLE verwendet) in ANSI (von VBA verwendet) umgesetzt. Wenn dieser Fehler auftritt, wird nur das erste Zeichen der Zeichenfolge angezeigt. Um den Fehler zu beheben, müssen Sie die Variable gleich einer leeren Zeichenfolge definieren, bevor Sie die QMF für Windows-Funktion aufrufen, die die Variable verwendet.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wenn Sie keine Datenbank-Server definiert haben, ist der Rückgabewert Null. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetServerListEx()

short GetServerListEx(short *Index*)

Beschreibung

Diese Funktion ruft den Namen des Servers ab, auf den durch den Parameter *Index* verwiesen wird. Nach dem Aufruf dieser Funktion kann die Eigenschaft *Wert* nach dem zurückgegebenen Wert abgefragt werden.

Parameter

Name	Beschreibung
<i>Index</i>	Ein Index der Liste der Server.

Rückgabewert

Null, falls erfolgreich; RS_OUTOFRANGE, wenn der Index größer als die Anzahl der verfügbaren Server ist; ungleich Null, falls nicht erfolgreich. Wenn Sie keine Datenbank-Server definiert haben, ist der Rückgabewert RS_OUTOFRANGE. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetStoredProcedureResultSets()

short GetStoredProcedureResultSets(long *AbfrageID*, VARIANT* *Ergebnisgruppen*)

Beschreibung

Diese Funktion ruft die Abfrage-IDs für die Ergebnisgruppen ab, die von der gespeicherten Prozedur zurückgegeben wurden, die mit der ursprünglichen *AbfrageID* ausgeführt wurden. Jede zurückgegebene Abfrage-ID kann mit FetchNextRow() oder FetchNextRows(), um die Zeilen der Ergebnisgruppe abzurufen, oder mit Close() verwendet werden, wenn das Ende jeder Ergebnisgruppe erreicht wird.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der ursprünglichen Abfrage, wie von InitializeQuery() zurückgegeben.
<i>Ergebnisgruppen</i>	Ein Zeiger auf eine VARIANT, in der die Abfrage-IDs auf die Ergebnisgruppen gespeichert werden. Das Ergebnis ist eine Feldgruppe (<i>Array</i>) mit long-Integern (Variant-Typ VT_ARRAY VT_I4), wobei jedes Integer die Abfrage-ID für die entsprechende Ergebnisgruppe darstellt. Hat die gespeicherte Prozedur keine Ergebnisgruppen zurückgegeben, ist das Ergebnis ein leerer Wert (Variant-Typ VT_EMPTY). Sie müssen die VARIANT korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen VariantInit() aufrufen.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetVariables()

short `GetVariables(long AbfrageID, VARIANT* Variablen)`

Beschreibung

Diese Funktion gibt eine Feldgruppe (*Array*) der Namen der Variablen zurück, die im SQL-Text der Abfrage enthalten sind. Vor dem Ausführen der Prozedur entweder mit `Open()` oder `Execute()` müssen Sie diesen Variablen Werte zuordnen, indem Sie `SetVariable()` aufrufen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.
<i>Variablen</i>	Ein Zeiger auf eine <code>VARIANT</code> , in der die Ergebnisse gespeichert werden. Das Ergebnis ist eine Feldgruppe mit Zeichenfolgen (Variant-Typ <code>VT_ARRAY VT_BSTR</code>), wobei jede Zeichenfolge den Namen einer Variablen enthält. Enthält die SQL-Anweisung keine Variablen, ist das Ergebnis ein leerer Wert (Variant-Typ <code>VT_EMPTY</code>). Sie müssen die <code>VARIANT</code> korrekt initialisieren, bevor Sie diese Funktion aufrufen. Visual Basic führt dies automatisch durch. Visual C++-Programmierer müssen <code>VariantInit()</code> aufrufen.

Anmerkung: Aufgrund eines Programmfehlers in Microsoft Excel 7.0 und Microsoft Access 7.0 — sowie möglicherweise anderen 32-Bit-Produkten von Microsoft, die die Visual Basic for Applications (VBA) verwenden -, werden Zeichenfolgedaten (*String Data*) in Variant-Variablen, die aus QMF für Windows empfangen werden, nicht von Unicode (von OLE verwendet) in ANSI (von VBA verwendet) umgesetzt. Wenn dieser Fehler auftritt, wird nur das erste Zeichen der Zeichenfolge angezeigt. Um den Fehler zu beheben, müssen Sie die Variable gleich einer leeren Zeichenfolge definieren, bevor Sie die QMF für Windows-Funktion aufrufen, die die Variable verwendet.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Enthält die SQL-Anweisung keine Variablen, ist der Rückgabewert RS_ERROR_NO_DATA (-1). Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

GetVariablesEx()

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

Beschreibung

Diese Funktion gibt den Namen der Variable im SQL-Text der Abfrage zurück, auf die durch den Parameter *Index* verwiesen wird. Nach dem Aufruf dieser Funktion kann die Eigenschaft *Wert* nach dem zurückgegebenen Wert abgefragt werden. Vor dem Ausführen der Prozedur entweder mit Open() oder Execute() müssen Sie dieser Variablen (und allen anderen im SQL-Text vorhandenen Variablen) Werte zuordnen, indem Sie SetVariable() aufrufen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben.
<i>Variablen</i>	Ein Index der internen Liste der Variablen. Fragen Sie die Eigenschaft <i>Wert</i> für die Zeichenfolge ab, die dem übergebenen Index entspricht. Enthält die SQL-Anweisung keine Variablen, gibt die Funktion den Wert RS_ERROR_NO_DATA zurück.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Enthält die SQL-Anweisung keine Variablen, ist der Rückgabewert RS_ERROR_NO_DATA (-1). Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

InitializeProc()

long InitializeProc(short *Quellentyp*, BSTR *Quelle*)

Beschreibung

Diese Funktion legt den Text fest, den Sie in einer Prozedur verwenden wollen. Sie können den Text als Parameter an diese Funktion übergeben, aus einer Textdatei lesen oder aus einer vorhandenen Prozedur übernehmen.

Parameter

Name	Beschreibung
<i>Quellentyp</i>	Gibt die Quelle für den Text der Prozedur an.

Wert	Bedeutung
0 (RSS_STRING)	Der Text ist im Parameter <i>Quelle</i> enthalten.
2 (RSS_FILE)	Der Text ist in der Textdatei enthalten, deren Name im Parameter <i>Quelle</i> angegeben ist.
3 (RSS_QMFPROC)	Der Text ist in der Prozedur enthalten, deren Eigner und Name im Parameter <i>Quelle</i> angegeben sind.

<i>Quelle</i>	Eine Zeichenfolge mit dem Text, dem Eigner und dem Namen (Eigner.Name) der Prozedur oder dem Namen einer Datei, die Text der Prozedur enthält.
---------------	--

Rückgabewert

Falls erfolgreich, die Kennung der Prozedur (ProzID). Falls nicht erfolgreich, -1. Sie müssen diesen Wert in allen Schnittstellenaufrufen verwenden, für die der Parameter *ProzID* erforderlich ist.

InitializeQuery()

long InitializeQuery(short *Quellentyp*, BSTR *Quelle*)

Beschreibung

Diese Funktion legt den Text fest, den Sie in einer Abfrage verwenden wollen. Sie können den SQL-Text als Parameter an diese Funktion übergeben, aus einer Textdatei lesen oder aus einer vorhandenen Abfrage übernehmen. Rufen Sie close() auf, wenn Sie mit der Abfrage fertig sind.

Parameter

Name	Beschreibung
<i>Quellentyp</i>	Gibt die Quelle für den Text der SQL-Anweisung an.

Wert	Bedeutung
0 (RSS_STRING)	Der SQL-Text ist im Parameter <i>Quelle</i> enthalten.
1 (RSS_QMFQUERY)	Der SQL-Text ist in der Abfrage enthalten, deren Eigner und Name im Parameter <i>Quelle</i> angegeben sind.

2 (RSS_FILE)

Der SQL-Text ist in der Textdatei enthalten, deren Name im Parameter *Quelle* angegeben ist.

Rückgabewert

Falls erfolgreich, die Kennung der Abfrage. Falls nicht erfolgreich, -1. Sie müssen diesen Wert in allen Schnittstellenaufrufen verwenden, für die der Parameter *Abfrage* erforderlich ist.

InitializeServer()

short InitializeServer(BSTR *ServerName*, BSTR *BenutzerID*, BSTR *Kennwort*, BOOL *DialogErzw*, [VARIANT *Konto*], [VARIANT *DialogUnterdrücken*])

Beschreibung

Diese Funktion initialisiert eine Verbindung zu einem Datenbank-Server. Sie müssen diese Funktion vor jeder anderen Funktion in der QMF für Windows-API aufrufen. Sie können diese Funktion beliebig oft aufrufen. Wenn Sie diese Funktion jedoch aufrufen und sie nicht durch Aufrufen der Funktion Commit() bzw. Rollback() beenden, hat dies eine implizite Rollback-Operation zur Folge.

Parameter

Name	Beschreibung
<i>ServerName</i>	Eine Zeichenfolge, die die Namen des zu verwendenden Datenbank-Servers enthält. Dieser Name muss mit einem der Namen übereinstimmen, die in der Server-Definitionsdatei (<i>Server Definition File</i> , SDF) von QMF für Windows definiert sind. Rufen Sie GetServerList() auf, um eine Liste der gültigen Server abzurufen.
<i>BenutzerID</i>	Eine Zeichenfolge, die die zu verwendende Benutzer-ID enthält. Ist die BenutzerID NULL oder eine leere Zeichenfolge, versucht QMF für Windows, die Benutzer-ID aus der letzten Abfrage zu verwenden, falls sie verfügbar ist. Andernfalls zeigt QMF für Windows das Dialogfenster Benutzerinformationen an, um die Eingabe einer Benutzer-ID und eines Kennworts zu ermöglichen.
<i>Kennwort</i>	Eine Zeichenfolge, die das Kennwort für die angegebene Benutzer-ID enthält. Ist das Kennwort NULL oder eine leere Zeichenfolge, versucht QMF für Windows, ein gespeichertes Kennwort zu verwenden falls ein solches Kennwort verfügbar ist. (Hierfür ist Windows für Workgroups erforderlich.) Ist kein Kennwort verfügbar, zeigt QMF für Windows das Dialogfenster Benutzerinformationen an, um die Eingabe eines Kennworts zu ermöglichen.

<i>DialogErzw</i>	Ein Wert ungleich Null gibt an, dass QMF für Windows das Dialogfenster Benutzerinformationen unabhängig davon anzeigen soll, ob eine Benutzer-ID und ein Kennwort angegeben wurden. Hierdurch kann der Benutzer die Informationen ändern, bevor sie verwendet werden. Ein Wert von Null gibt an, dass QMF für Windows das Dialogfenster Benutzerinformationen nur anzeigen soll, wenn es erforderlich ist.
<i>Konto</i>	Mit diesem optionalen Parameter wird eine Zeichenfolge angegeben, die Informationen zum Benutzereintrag enthält, die zum Zeitpunkt der Verbindungsherstellung an den Server weitergeleitet werden. Der Server kann diese Informationen in einem System zur Job-Abrechnung verwenden.
<i>DialogUnterdrücken</i>	Ein Wert ungleich Null gibt an, dass QMF für Windows das Dialogfenster Benutzerinformationen nicht anzeigen soll — auch dann nicht, wenn keine Benutzer-ID und kein Kennwort angegeben wurden. Die Option ist dann sehr hilfreich, wenn die Ausführung in einer Umgebung stattfindet, in der kein Benutzer vorhanden ist, um auf das Dialogfenster Benutzerinformationen zu reagieren — beispielsweise auf einem Web-Server.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch
`SetParent()`

InitializeStaticQuery()

long InitializeStaticQuery(BSTR *CollectionName*, BSTR *Paketname*, BSTR *KonsistenzToken*, short *Abschnittsnummer*)

Beschreibung

Mit dieser Funktion wird der Abschnitt eines Pakets angegeben, den Sie als eine statische Abfrage ausführen wollen.

Parameter

Name	Beschreibung
<i>CollectionName</i>	Der Name einer zuvor gebundenen Collection.
<i>Paketname</i>	Der Name eines zuvor gebundenen Pakets.
<i>KonsistenzToken</i>	Das Token, das von der benannten Collection und dem benannten Paket, die oben angegeben sind, verwendet wird.
<i>Abschnittsnummer</i>	Die Abschnittsnummer der Anweisung innerhalb der auszuführenden Collection und des auszuführenden Pakets.

Rückgabewert

Falls erfolgreich, die Kennung der Abfrage. Falls nicht erfolgreich, -1. Sie müssen diesen Wert in allen Schnittstellenaufrufen verwenden, für die der Parameter *AbfrageID* erforderlich ist.

IsStatic()

BOOL IsStatic(long *AbfrageID*)

Beschreibung

Mit dieser Funktion wird bestimmt, ob sich die angegebene Abfrage-ID auf eine statische Abfrage oder auf eine dynamische Abfrage bezieht.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() oder InitializeStaticQuery() zurückgegeben.

Rückgabewert

Gibt einen Wert ungleich Null zurück, falls erfolgreich, und *AbfrageID* bezieht sich auf eine statische Abfrage; andernfalls wird Null zurückgegeben.

Open()

short Open(long *AbfrageID*, long *Zeilengrenzwert*, BOOL *AlleZeilenAbrufen*)

Beschreibung

Verwenden Sie diese Funktion, um eine Abfrage auszuführen, die das SQL-Verb SELECT verwendet, indem Sie einen Cursor in der Datenbank für die Abfrage öffnen. Verwenden Sie FetchNextRow() oder FetchNextRows(), um die Daten für die Abfrage abzurufen, und rufen Sie Close() auf, wenn Sie damit fertig sind. Wenn QMF für Windows so konfiguriert ist, dass automatisch alle Zeilen abgerufen werden (siehe RSR_FETCHALLROWS in der Beschreibung der Funktion GetResourceLimit()) oder wenn der Parameter *AlleZeilenAbrufen* ungleich Null ist, ruft QMF für Windows vor der Rückgabe von diesem Aufruf alle Zeilen der Ergebnisgruppe in seinen internen Puffer ab.

Anmerkung: Der Name dieser Funktion steht im Konflikt mit dem Schlüsselwort Open von Microsoft Access 2.0. Um diese Unverträglichkeit zu umgehen, müssen Sie — wenn Sie MS Access 2.0 verwenden — den Funktionsnamen in eckige Klammern einschließen.

Anmerkung: Verwenden Sie diese Funktion nur in Anweisungen, die das SQL-Verb SELECT enthalten. Rufen Sie für Anweisungen, die ein anderes Verb (z. B. SET) enthalten, rufen Sie statt dessen die Funktion Execute() auf. Verwenden Sie die Funktion GetQueryVerb(), um festzustellen, welches Verb von einer Abfrage verwendet wird.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben.
<i>Zeilengrenzwert</i>	Eine Zahl, die die maximale Anzahl der aus der Datenbank abzurufenden Zeilen angibt. Null gibt an, dass keine Begrenzung (mit Ausnahme des Zeilengrenzwerts, der mit "QMF für Windows Administrator" eingerichtet wurde) aktiviert werden soll.
<i>AlleZeilenAbrufen</i>	Ein Boolescher Wert, der angibt, ob alle Zeilen der Ergebnisgruppe automatisch in den internen Puffer von QMF für Windows abgerufen werden sollen. Ist der Wert ungleich Null, werden alle Zeilen automatisch abgerufen, wobei der Cursor geschlossen und die Datenbank vorübergehend für die Benutzung durch Andere gesperrt wird. Dieses Vorgehen entspricht dem Aufruf der Funktion CompleteQuery().

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Prepare()

short `Prepare(long AbfrageID)`

Beschreibung

Die Funktion bereitet die durch `AbfrageID` angegebene *Abfrage* vor. Die Anweisung wird vom Datenbank-Server untersucht — das Vorhandensein des Objekts und die erforderlichen Berechtigungen werden überprüft. Wenn es sich bei der Abfrage um eine `SELECT`-Anweisung handelt, sind Informationen zu den Spalten, die durch die Anweisung zurückgegeben werden, nach der Beendigung von `Prepare()` verfügbar.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeQuery()</code> zurückgegeben.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

`Execute()`
`Open()`

PrintReport()

short `PrintReport(long AbfrageID, short Quellentyp, BSTR Quelle, BSTR Ausgabedateiname, short Seitenlänge, short Seitenbreite, BOOL DatumZeitEinschl, BOOL SeitenzahlenEinschl, [VARIANT Formatierung], [VARIANT Formatseiten-einrichtungVerwenden])`

Beschreibung

`PrintReport()` ist ein Synonym für die Funktion `ExportReport()`.

ReinitializeServer()

short `ReinitializeServer()`

Beschreibung

Diese Funktion initialisiert die Verbindung zum Datenbank-Server erneut. Im allgemeinen muss diese Funktion nur aufgerufen werden, wenn eine andere

Funktion der QMF für Windows-API einen Fehler zurückgibt. Durch den Aufruf dieser Funktion werden alle geöffneten Cursor geschlossen und alle ausstehenden Abfrage-IDs ungültig gemacht.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Rollback()

short Rollback()

Beschreibung

Diese Funktion storniert alle von Ihnen in der aktuellen Arbeitseinheit (*Unit-of-Work*) vorgenommenen Änderungen, schließt alle offenen Cursor und macht alle noch offenen Abfrage-IDs ungültig.

Anmerkung: Der Name dieser Funktion steht im Konflikt mit dem Schlüsselwort `Rollback` von Microsoft Access 2.0. Um diese Unverträglichkeit zu umgehen, müssen Sie — wenn Sie MS Access 2.0 verwenden — den Funktionsnamen in eckige Klammern einschließen.

Anmerkung: Diese Rollback-Operation wirkt sich nur auf SQL-Änderungen aus, die durch den Aufruf der Funktion `Open()` oder `Execute()` ausgeführt wurden. Die Rollback-Operation wirkt sich nicht auf Änderungen aus, die von anderen Funktionen der QMF für Windows-API, wie beispielsweise `FastSaveData()`, `SaveData()`, oder `DeleteQMFObject()`, gemacht wurden.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

`Commit()`

RunProc()

short RunProc(long *ProzID*)

Beschreibung

Diese Funktion führt die angegebene Prozedur aus. Die Prozedur wird bis zum vollständigen Abschluss ausgeführt bzw. solange ausgeführt, bis ein Fehler auftritt. Sie können über diese Programmierschnittstelle auf keine

Ergebnisse der Prozedur zugreifen (beispielsweise nicht auf Daten von einer ausgeführten Abfrage). Alle Dateien, die exportiert wurden, bzw. alle Daten, die durch die Prozedur gespeichert wurden, stehen jedoch nach der Ausführung zur Verfügung.

Parameter

Name	Beschreibung
<i>ProzID</i>	Die ID der Prozedur, wie von InitializeProc() zurückgegeben.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() oder GetLastErrorSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

SaveData()

short SaveData(long *AbfrageID*, long *ErsteZeile*, long *ErsteSpalte*, long *LetzteZeile*, long *LetzteSpalte*, BOOL *Ersetzen*, BSTR *Tabellenname*, BSTR *TabellenBerName*, BSTR *ServerName*, BSTR *BenutzerID*, BSTR *Kennwort*, BOOL *DialogErzw*, [VARIANT *Konto*], [VARIANT *Kommentar*], [VARIANT *CommitUmfang*])

Beschreibung

Diese Funktion speichert den angegebenen Bereich von Zeilen und Spalten in der angegebenen Tabelle im angegebenen Tabellenbereich. Wenn Sie noch nicht für alle Zeilen, die in der Tabelle gespeichert werden sollen, Zeilendaten abgerufen haben, müssen Sie vor dem Aufrufen dieser Funktion die Funktion CompleteQuery() verwenden. Wenn Sie versuchen, Zeilen zu speichern, die noch nicht aus der Datenbank abgerufen wurden, schlägt die Speicherungsoperation fehl. Ist die Tabelle bereits vorhanden, muss die neue Tabelle die gleiche Spaltenanzahl und die gleichen Spaltentypen wie die vorhandene Tabelle aufweisen.

Diese Funktion wird in einer anderen Arbeitseinheit ausgeführt, als die übrigen API-Funktionen; die Ergebnisse werden in einer COMMIT-Operation automatisch festgeschrieben. Der Aufruf von Commit() oder Rollback() hat keine Auswirkungen auf Änderungen, die Sie unter Verwendung dieser Funktion vornehmen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben.

<i>ErsteZeile</i>	Die erste Zeile, die in der Speicherungsoperation aufgenommen werden soll. Der Wert für die erste Zeile einer Ergebnisgruppe ist 0.
<i>ErsteSpalte</i>	Die erste Spalte, die in der Speicherungsoperation aufgenommen werden soll. Der Wert für die erste Spalte einer Ergebnisgruppe ist Null.
<i>LetzteZeile</i>	Die letzte Zeile, die in der Speicherung aufgenommen werden soll, oder -1, wenn alle Zeilen aufgenommen werden sollen. Der Wert für die letzte Zeile in einer Ergebnisgruppe ist um eins niedriger als die Gesamtzahl der Zeilen.
<i>LetzteSpalte</i>	Die letzte Spalte, die in der Speicherungsoperation aufgenommen werden soll, oder -1, wenn alle Spalten aufgenommen werden sollen. Der Wert für die letzte Spalte in einer Ergebnisgruppe ist um eins niedriger als die Gesamtzahl der Spalten.
<i>Ersetzen</i>	Ein Wert ungleich Null gibt an, dass bereits in der Tabelle bestehende Daten (falls vorhanden) durch die angegebenen Daten ersetzt werden. Der Wert Null gibt an, dass die angegebenen Daten an bereits bestehende Daten in der Tabelle (falls vorhanden) angefügt werden.
<i>Tabellenname</i>	Der Name der Tabelle, in der die Daten gespeichert werden. Ist die Tabelle nicht vorhanden, wird sie erstellt.
<i>TabellenBerName</i>	Der Name des Tabellenbereichs, in dem die Tabelle vorhanden ist oder erstellt werden wird. Ist der Parameter <i>TabellenBerName</i> NULL oder eine leere Zeichenfolge, wird der Standardtabellenbereich verwendet. Wenn Sie QMF für Windows so konfiguriert haben, dass immer der Standardtabellenbereich verwendet wird (siehe RSR_SDDIFFERENTTS in der Beschreibung der Funktion GetResourceLimit()), wird dieser Parameter ignoriert.
<i>ServerName</i>	Der Name des Datenbank-Servers auf dem die Tabelle gespeichert ist. Ist <i>ServerName</i> NULL oder eine leere Zeichenfolge, wird der Server-Name verwendet, der beim Aufruf der Funktion InitializeServer() angegeben wurde, und die Parameter <i>BenutzerID</i> , <i>Kennwort</i> , <i>DialogErzw</i> und <i>Konto</i> werden ignoriert.
<i>BenutzerID</i>	Wurde im Parameter <i>ServerName</i> ein anderer Server angegeben, ist <i>BenutzerID</i> die für diesen Server verwendete Benutzer-ID. Wenn Sie keine Benutzer-ID angeben, verwendet QMF für Windows die zuletzt für diesen Server angegebene Benutzer-ID (falls vorhanden), oder QMF für Windows zeigt ein Dialogfenster an, wenn keine Benutzer-ID zuvor vorhanden war. Dieser Parameter wird ignoriert, wenn im Parameter <i>ServerName</i> NULL oder eine leere Zeichenfolge angegeben wurde.

<i>Kennwort</i>	Wurde im Parameter <i>ServerName</i> ein anderer Server angegeben, ist <i>Kennwort</i> das für diesen Server verwendete Kennwort. Wenn Sie kein Kennwort angeben, verwendet QMF für Windows das zuletzt für diesen Server angegebene Kennwort (falls vorhanden), oder QMF für Windows zeigt ein Dialogfenster an, wenn kein Kennwort zuvor vorhanden war. Dieser Parameter wird ignoriert, wenn im Parameter <i>ServerName</i> NULL oder eine leere Zeichenfolge angegeben wurde.
<i>DialogErzw</i>	Wenn Sie im Parameter <i>ServerName</i> einen anderen Server angegeben haben, wird durch einen Wert ungleich Null erzwungen, dass QMF für Windows ein Dialogfenster zur Eingabe von Anmeldeinformationen anzeigt, auch dann, wenn eine Benutzer-ID und ein Kennwort angegeben wurden oder anderweitig verfügbar sind. Ein Wert von Null gibt an, dass QMF für Windows dieses Dialogfenster nur dann anzeigen soll, wenn es erforderlich ist. Dieser Parameter wird ignoriert, wenn im Parameter <i>ServerName</i> NULL oder eine leere Zeichenfolge angegeben wurde.
<i>Konto</i>	Wenn Sie in <i>ServerName</i> einen anderen Server angegeben haben, wird mit diesem optionalen Parameter eine Zeichenfolge angegeben, die Informationen zum Benutzereintrag enthält, die zum Zeitpunkt der Verbindungsherstellung an diesen Server weitergeleitet werden. Der Server kann diese Informationen in einem System zur Job-Abrechnung verwenden. Dieser Parameter wird ignoriert, wenn im Parameter <i>ServerName</i> NULL oder eine leere Zeichenfolge angegeben wurde.
<i>Kommentar</i>	Ein optionaler Parameter für eine Zeichenfolge, die einen Kommentar für die Tabelle angibt, in der die Daten gespeichert sind.
<i>CommitUmfang</i>	Mit diesem optionalen Parameter wird angegeben, wieviele Zeilen gleichzeitig in die Tabelle eingefügt werden sollen, bevor die die Arbeitseinheit (<i>Unit-of-Work</i>) festgeschrieben (<i>Commit</i>) wird. Durch die Angabe von Null wird angezeigt, dass sämtliche Zeilen vor dem Festschreiben eingefügt werden sollen. Durch Angabe von 10 wird beispielsweise angezeigt, dass ein Festschreiben dann erfolgen soll, nachdem jeweils zehn Zeilen eingefügt wurden.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten. Ist die Ergebnisgruppe leer oder wurden keine Zeilen aus der Datenbank abgerufen, wird ein Wert ungleich Null zurückgegeben, es sei denn, es gilt folgendes für folgende Parameter: `ErsteZeile = 0` und `LetzteZeile = -1`. In diesem Fall wird Null zurückgegeben und eine leere Tabelle erstellt.

SaveQMFProc()

short SaveQMFProc(BSTR *EignerUndName*, BSTR *Text*, BSTR *Kommentar*, BOOL *Ersetzen*, BOOL *GemeinsBen*)

Beschreibung

Mit dieser Funktion wird eine Prozedur auf einem Datenbank-Server gespeichert.

Parameter

Name	Beschreibung
<i>EignerUndName</i>	Eine Zeichenfolge, die den Eigner und den Namen (getrennt durch einen Punkt) der Prozedur umfasst, die Sie speichern wollen. Zum Beispiel: Udo.Proz2
<i>Text</i>	Eine Zeichenfolge, die den Text enthält, der in der Prozedur gespeichert werden soll.
<i>Kommentar</i>	Eine Zeichenfolge, die einen beliebigen Kommentar enthält, der zusammen mit der Prozedur gespeichert werden soll. Soll kein Kommentar gespeichert werden, muss dieser Parameter entweder als NULL oder als leere Zeichenfolge übergeben werden.
<i>Ersetzen</i>	Ein Wert ungleich Null gibt an, dass eine bereits bestehende Prozedur mit dem gleichen Namen ersetzt werden soll; der Wert Null gibt an, dass die Operation abgebrochen werden soll, falls bereits eine Prozedur mit dem gleichen Namen existiert.
<i>GemeinsBen</i>	Ein Wert ungleich Null gibt an, dass diese Prozedur gemeinsam mit anderen Benutzern verwendet werden soll; der Wert Null gibt an, dass sie nicht gemeinsam mit anderen Benutzern verwendet werden soll.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() oder GetLastErrorSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

SaveQMFQuery()

short SaveQMFQuery(BSTR *EignerUndName*, BSTR *Text*, BSTR *Kommentar*, BOOL *Ersetzen*, BOOL *GemeinsBen*)

Beschreibung

Mit dieser Funktion wird eine Abfrage auf einem Datenbank-Server gespeichert.

Parameter

Name	Beschreibung
<i>EignerUndName</i>	Eine Zeichenfolge, die den Eigner und den Namen (getrennt durch einen Punkt) der Abfrage umfasst, die gespeichert werden soll. Zum Beispiel: Udo.Abfrage2
<i>Text</i>	Eine Zeichenfolge, die den Text enthält, der in der Abfrage gespeichert werden soll.
<i>Kommentar</i>	Eine Zeichenfolge, die einen beliebigen Kommentar enthält, der zusammen mit der Abfrage gespeichert werden soll. Soll kein Kommentar gespeichert werden, muss dieser Parameter entweder als NULL oder als leere Zeichenfolge übergeben werden.
<i>Ersetzen</i>	Ein Wert ungleich Null gibt an, dass eine bereits bestehende Abfrage mit dem gleichen Namen ersetzt werden soll; der Wert Null gibt an, dass die Operation abgebrochen werden soll, falls bereits eine Abfrage mit dem gleichen Namen existiert.
<i>GemeinsBen</i>	Ein Wert ungleich Null gibt an, dass diese Abfrage gemeinsam mit anderen Benutzern verwendet werden soll; der Wert Null gibt an, dass sie nicht gemeinsam mit anderen Benutzern verwendet werden soll.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() oder GetLastErrorSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

SetBindOption()

short SetBindOption(BSTR *CollectionName*, BSTR *Paketname*, short *Option*, short *Wert*)

Beschreibung

Diese Funktion legt Optionen für die Collection und das Paket vor dem Aufruf von EndBind() fest.

Parameter

Name	Beschreibung
<i>CollectionName</i>	Die Collection-ID des Pakets, für das Sie die Option festlegen wollen.
<i>Paketname</i>	Der Name des Pakets, für das Sie die Option festlegen wollen.
<i>Option</i>	Eine der nachfolgend aufgelisteten Optionen.
<i>Wert</i>	Ein Wert ungleich Null gibt an, dass eine bereits bestehende Abfrage mit dem gleichen Namen ersetzt werden soll; der Wert Null gibt an, dass die Operation abgebrochen werden soll, falls bereits eine Abfrage mit dem gleichen Namen existiert.
<i>GemeinsBen</i>	Einer der nachfolgend für die angegebene Option aufgelisteten Werte.

Die einzelnen Optionen haben folgende Bedeutung und nehmen folgende Werte an:

Option	Bedeutung	Beschreibung
DDM_PKGRPLOPT(0x211C)	Kann ein vorhandenes Paket mit der gleichen Collection-ID und -Namen ersetzt werden?	DDM_PKGRPLALW (0x241F) Ja DDM_PKGRPLNA (0x2420) Nein
DDM_STTDECDEL(0x2121)	Der Begrenzer, der für das Dezimalzeichen in SQL-Anweisungen im Paket verwendet wird.	DDM_DECDELPRD (0x243C) Punkt DDM_DECDELCMA (0x243D) Komma
DDM_STTSTRDEL(0x2120)	Der Begrenzer, der für Zeichenfolgewerte in SQL-Anweisungen im Paket verwendet wird.	DDM_STRDELAP (0x2426) Apostroph DDM_STRDELQ (0x2427) Doppeltes Anführungszeichen

DDM_PKGISOLVL(0x2124)	Die Isolationsstufe für das Paket.	DDM_ISOLVLALL (0x2443) Alle DDM_ISOLVLCHG (0x2441) Ändern (Change) DDM_ISOLVLCS (0x2442) Cursorstabilität DDM_ISOLVLNC (0x2445) Kein Festschreiben (No Commit) DDM_ISOLVLR (0x2444) Wiederholbares Lesen (Repeatable Read)
DDM_PKGATHOPT(0x211E)	Können vorhandene Berechtigungen für das Paket beibehalten werden?	DDM_PKGATHKP (0x2425) Beibehalten DDM_PKGATHRVK (0x2424) Widerrufen
DDM_QRYBLKCTL(0x2132)	Die Methode, die beim Abrufen von Datenzeilen für Abfragen im Paket verwendet werden soll.	DDM_FIXROWPRC (0x2418) Jeweils eine Zeile DDM_LMTBLKPRC (0x2417) Jeweils ein Zeilenblock
DDM_RDBRLSOPT(0x2129)	Zeitpunkt des Freigebens von Datenbankressourcen, die beim Ausführen des Pakets übernommen wurden.	DDM_RDBRLSCMM (0x2438) Festschreiben (Commit) DDM_RDBRLSCNV (0x2439) Dialogfreigabe
DDM_STTDATFMT(0x2122)	Formatierung für abgerufene Datumswerte.	DDM_ISODATFMT (0x2429) ISO DDM_USADATFMT (0x242A) US DDM_EURDATFMT (0x242B) Europa DDM_JISDATFMT (0x242C) Japan (Industriestandard)
DDM_STTTIMFMT(0x2123)	Formatierung für abgerufene Zeitwerte.	DDM_ISOTIMFMT (0x242E) ISO DDM_USATIMFMT (0x242F) US DDM_EURTIMFMT (0x2430) Europa DDM_JISTIMFMT (0x2431) Japan (Industriestandard)

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

SetBindOwner()

short SetBindOwner(BSTR *CollectionName*, BSTR *Paketname*, BSTR *EignerID*)

Beschreibung

Mit dieser Funktion können Sie einen Eigner, der sich von Ihrer Benutzer-ID unterscheidet, für das Paket angeben, das Sie momentan binden. Dies ist möglicherweise dann erforderlich, wenn Ihre Benutzer-ID (im Gegensatz zur anderen Benutzer-ID) nicht über die erforderlichen Berechtigungen zum Binden des Pakets verfügt.

Parameter

Name	Beschreibung
<i>CollectionName</i>	Die Collection-ID des Pakets, für das Sie den Eigner angeben wollen.
<i>Paketname</i>	Der Name des Pakets, für das Sie den Eigner angeben wollen.
<i>Kommentar</i>	Eine Zeichenfolge, die einen beliebigen Kommentar enthält, der zusammen mit der Abfrage gespeichert werden soll. Soll kein Kommentar gespeichert werden, muss dieser Parameter entweder als NULL oder als leere Zeichenfolge übergeben werden.
<i>EignerID</i>	Die gewünschte Eigner-ID für das Paket, das Sie binden.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` oder `GetLastSQLState()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

SetBusyWindowButton()

void SetBusyWindowButton(BSTR *Text*)

Beschreibung

Diese Funktion gibt den Text an, der auf dem Knopf **Abbrechen** im **Vorgangsfenster** angezeigt wird.

Parameter

Name	Beschreibung
<i>Text</i>	Eine Zeichenfolge, mit der der Text angegeben wird, der auf dem Knopf Abbrechen im Vorgangsfenster angezeigt wird. Der Standardwert ist "Abbrechen". Wenn Sie eine leere Zeichenfolge angeben, wird der Knopf verdeckt. Unabhängig von dem Text, den Sie angeben, hat der Knopf immer die Funktion zum Abbrechen bzw. zum Schließen des Fensters.

Rückgabewert

Keiner.

Siehe auch

SetBusyWindowMessage()

SetBusyWindowMode()

SetBusyWindowTitle()

ShowBusyWindow()

SetBusyWindowMessage()

void SetBusyWindowMessage(BSTR *Nachricht*)

Beschreibung

Diese Funktion gibt den Text an, der im Nachrichtenbereich im **Vorgangsfenster** angezeigt wird.

Parameter

Name	Beschreibung
<i>Nachricht</i>	Eine Zeichenfolge, mit der der Text angegeben wird, der im Nachrichtenbereich im Vorgangsfenster angezeigt wird.

Rückgabewert

Keiner.

Siehe auch

SetBusyWindowButton()

SetBusyWindowMode()

SetBusyWindowTitle()

ShowBusyWindow()

SetBusyWindowMode()

void SetBusyWindowMode(short *Modus*)

Beschreibung

Diese Funktion legt fest, ob QMF für Windows das **Vorgangsfenster** anzeigt. Das **Vorgangsfenster** ist hilfreich, wenn Sie dem Benutzer eine Rückmeldung über ausgeführte Aktionen geben wollen oder wenn Sie ihm die Möglichkeit geben wollen, eine anstehende Datenbankaktion abubrechen. Diese Änderungen werden nicht sofort wirksam; sie treten in Kraft, wenn QMF für Windows das nächste Mal eine Operation ausführt, durch die das **Vorgangsfenster** angezeigt oder verdeckt würde.

Parameter

Name	Beschreibung
<i>Modus</i>	Gibt an, wann QMF für Windows das Vorgangsfenster anzeigt.

Wert	Bedeutung
0 (RSM_NEVER)	Das Fenster wird nicht angezeigt. Dies ist die Standardeinstellung.
1 (RSM_WHENBUSY)	Das Fenster wird dann angezeigt, wenn QMF für Windows mit der Datenbank kommuniziert. QMF für Windows zeigt dieses Fenster automatisch bei Bedarf an.
2 (RSM_CLIENTCONTROLLED)	Das Fenster wird angezeigt, nachdem Sie die Funktion ShowBusyWindow(TRUE) aufrufen und nachdem Sie die Funktion ShowBusyWindow(FALSE) aufrufen. Der Client legt fest, wann das Fenster angezeigt werden soll.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

SetBusyWindowButton()
SetBusyWindowMessage()
SetBusyWindowTitle()
SetParent()
ShowBusyWindow()

SetBusyWindowTitle()

void SetBusyWindowTitle(BSTR *Titel*)

Beschreibung

Diese Funktion gibt den Text an, der in der Titelleiste des **Vorgangsfensters** angezeigt wird.

Parameter

Name	Beschreibung
<i>Titel</i>	Eine Zeichenfolge, mit der der Text angegeben wird, der in der Titelleiste des Vorgangsfensters angezeigt wird.

Rückgabewert

Keiner.

Siehe auch

SetBusyWindowButton()
SetBusyWindowMode()
SetBusyWindowMessage()
ShowBusyWindow()

SetGlobalVariable()

short SetGlobalVariable(BSTR *Name*, BSTR *Wert*)

Beschreibung

Diese Funktion ordnet der angegebenen globalen Variable einen Wert zu. Dieser Wert steht für die Verwendung in Abfragen, Formaten und Prozeduren zur Verfügung.

Parameter

Name	Beschreibung
<i>Name</i>	Eine Zeichenfolge, die den Namen der Variablen enthält, die definiert werden soll.
<i>Wert</i>	Eine Zeichenfolge, die den Wert enthält, der der angegebenen Variablen zugeordnet werden soll.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

SetHostVariable()

short SetHostVariable(long *AbfrageID*, VARIANT *Index*, VARIANT *Wert*)

Beschreibung

Diese Funktion ordnet der angegebenen Host-Variablen, auf die durch die Abfrage verwiesen wird, einen Wert zu. Die Abfrage muss eine statische Abfrage sein, mit der auf Host-Variablen verwiesen wird (entweder in der QMF-Abfrage gespeichert oder durch AddHostVariable() erstellt). Mit *Index* kann entweder der numerische Index der Host-Variable oder der Name der Host-Variable angegeben werden.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von <code>InitializeStaticQuery()</code> zurückgegeben.
<i>Index</i>	Entweder eine Nummer (Variant-Typ <code>VT_I2</code>), mit der der Index der Host-Variable in der Abfrage angegeben wird, oder eine Zeichenfolge (Variant-Typ <code>VT_BSTR</code>), mit der der Name der Host-Variable angegeben wird.
<i>Wert</i>	Der Wert der Host-Variable. Zur Angabe eines Nullwerts muss der Typ der Variante auf <code>VT_EMPTY</code> gesetzt werden.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()` oder `GetLastErrorType()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

SetOption()

short `SetOption(short Modus, VARIANT Wert)`

Beschreibung

Diese Funktion setzt den angegebenen Optionswert in QMF für Windows. Bei einigen Optionen werden die Änderungen erst nach einem Neustart von QMF für Windows wirksam. Normalerweise starten Sie QMF für Windows erst dann neu, wenn Sie alle Exemplare (*Instances*) des QMF für Windows-API-Objekts zerstört haben.

Parameter

Name	Beschreibung
<i>Modus</i>	Gibt an, welcher Modus gesetzt werden soll:

Wert	Bedeutung
0 (<code>RSO_SERVER_DEFINITION_FILE</code>)	Name der Server-Definitionsdatei.
1 (<code>RSO_CPIC_DLL</code>)	Dateiname der CPI-C-Provider-DLL.
2 (<code>RSO_CPIC_TIMEOUT_WARNING</code>)	CPI-C-Zeitlimit für Warnung (in Sekunden). Dieses Limit wird nicht von der QMF für Windows-API verwendet.
3 (<code>RSO_CPIC_TIMEOUT_CANCEL</code>)	CPI-C-Zeitlimit für das Abbrechen (in Sekunden).

4 (RSO_TCP_TIMEOUT_WARNING)	TCP-Zeitlimit für Warnung (in Sekunden). Dieses Limit wird nicht von der QMF für Windows-API verwendet.
5 (RSO_TCP_TIMEOUT_CANCEL)	TCP-Zeitlimit für das Abbrechen (in Sekunden).
6 (RSO_DISPLAY_NULLS_STRING)	Die zur Anzeige von Nullwerten zu verwendende Zeichenfolge.
7 (RSO_ENTER_NULLS_STRING)	Die für die Eingabe von Nullwerten zu verwendende Zeichenfolge.
8 (RSO_ENTER_DEFAULTS_STRING)	Die für die Eingabe von Standardwerten zu verwendende Zeichenfolge.
9 (RSO_TRACE_FILE_1)	Name der Trace-Datei 1.
10 (RSO_TRACE_FILE_2)	Name der Trace-Datei 2.
11 (RSO_TCP_TRACE_LEVEL)	TCP-Trace-Stufe.
12 (RSO_CPIC_TRACE_LEVEL)	CPI-C-Trace-Stufe.
13 (RSO_DDM_TRACE_LEVEL)	DDM-Trace-Stufe.
<hr/>	
Name	Beschreibung
<i>Wert</i>	Der Wert, auf den diese Option gesetzt werden soll.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen `GetLastErrorString()` oder `GetLastErrorType()` aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

`GetOption()`

SetParent()

`short SetParent(long ElterWnd)`

Beschreibung

Diese Funktion definiert das Elternfenster für Dialogfenster. Wenn QMF für Windows ein Dialogfenster (im **Vorgangsfenster** oder das Dialogfenster **Benutzerinformationen**) anzeigt, erscheint das Dialogfenster normalerweise in Bezug auf das Hauptfenster von QMF für Windows zentriert und modal zu diesem Fenster. Mit dieser Funktion können Sie erzwingen, dass die Dialogfenster in QMF für Windows in Bezug auf das Fenster der Client-Anwendung zentriert und modal zu diesem Fenster erscheinen.

Parameter

Name	Beschreibung
<i>ElterWnd</i>	Die HWND des neuen Elternfensters. Geben Sie NULL an, um das Hauptfenster von QMF für Windows als Elternfenster zu verwenden.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

ShowBusyWindow()

SetProcVariable()

short SetProcVariable(long *ProzID*, BSTR *Name*, BSTR *Wert*)

Beschreibung

Diese Funktion ordnet der angegebenen Variablen einen Wert zu. Die Variable wird vor der Ausführung der Prozedur durch diesen Wert ersetzt. Enthält Ihre Prozedur eine oder mehrere Variablen, müssen Sie die Werte für die Variablen definieren, indem Sie diese Funktion aufrufen, bevor Sie die Funktion RunProc() aufrufen.

Parameter

Name	Beschreibung
<i>ProzID</i>	Die ID der Prozedur, wie von InitializeProc() zurückgegeben.
<i>Name</i>	Eine Zeichenfolge, die den Namen der Variablen enthält, die definiert werden soll.
<i>Wert</i>	Eine Zeichenfolge, die den Wert enthält, der der angegebenen Variablen zugeordnet werden soll.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

SetVariable()

short SetVariable(long *AbfrageID*, BSTR *Name*, BSTR *Wert*)

Beschreibung

Diese Funktion ordnet der angegebenen Variablen einen Wert zu. Die Variable wird vor der Ausführung der SQL-Anweisung durch diesen Wert ersetzt. Enthält die SQL-Anweisung eine oder mehrere Variablen, müssen Sie die Werte für die Variablen definieren, indem Sie diese Funktion aufrufen, bevor Sie die Funktion Open() oder Execute() aufrufen.

Parameter

Name	Beschreibung
<i>AbfrageID</i>	Die ID der Abfrage, wie von InitializeQuery() zurückgegeben.
<i>Name</i>	Eine Zeichenfolge, die den Namen der Variablen enthält, die definiert werden soll.
<i>Wert</i>	Eine Zeichenfolge, die den Wert enthält, der der angegebenen Variablen zugeordnet werden soll.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString() oder GetLastErrorType() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

ShowBusyWindow()

void ShowBusyWindow(BOOL *Anzeigen*)

Beschreibung

Diese Funktion weist QMF für Windows an, das **Vorgangsfenster** anzuzeigen oder zu verdecken. Das **Vorgangsfenster** ist hilfreich, wenn Sie dem Benutzer eine Rückmeldung über ausgeführte Aktionen geben wollen oder wenn Sie ihm die Möglichkeit geben wollen, eine anstehende Datenbankaktion abzubrechen. Diese Funktion kann nur verwendet werden, wenn Sie die Funktion SetBusyWindowMode() mit dem Modus RSM_CLIENTCONTROLLED aufrufen. Wenn Sie mit Hilfe der Funktion SetParent() ein Elternfenster definieren, ist das **Vorgangsfenster** modal zu dem angegebenen Fenster.

Parameter

Name	Beschreibung
<i>Anzeigen</i>	Ein Wert ungleich Null gibt an, dass QMF für Windows das Vorgangsfenster anzeigen soll; der Wert Null gibt an, dass das Vorgangsfenster verdeckt werden soll. Ist der Wert ungleich Null, wird das Vorgangsfenster angezeigt, bis die Funktion ShowBusyWindow() mit dem Wert Null für den Parameter <i>Anzeigen</i> aufgerufen wird.

Rückgabewert

Keiner.

StartBind()

short StartBind(BSTR *CollectionName*, BSTR *Paketname*, BSTR *KonsistenzToken*)

Beschreibung

Diese Funktion startet den Prozess des Bindens eines Pakets in der Datenbank.

Parameter

Name	Beschreibung
<i>CollectionName</i>	Die gewünschte Collection-ID für das Paket.
<i>Paketname</i>	Der gewünschte Name für das Paket.
<i>KonsistenzToken</i>	Eine 16 Zeichen lange Zeichenfolge, die die hexadezimale Darstellung eines aus acht Byte bestehenden Tokens enthält, das verwendet wird, um die Konsistenz zwischen dem in der Datenbank gebundenen Paket und der Anwendung, die dieses Paket verwendet, sicherzustellen. Wird ein Abschnitt innerhalb des Pakets ausgeführt, müssen Sie exakt diesen Wert angeben.

Rückgabewert

Null, falls erfolgreich; ungleich Null, falls nicht erfolgreich. Wird ein Wert ungleich Null zurückgegeben, können die Funktionen GetLastErrorString(), GetLastErrorType(), GetLastSQLCode(), GetLastSQLError() oder GetLastSQLState() aufgerufen werden, um zusätzliche Fehlerinformationen zu erhalten.

Siehe auch

EndBind()
CancelBind()

Anhang. Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanfragen sind schriftlich an folgende Adresse zu richten:

IBM Europe,
Director of Licensing,
92066 Paris La Defense Cedex,
France

Anfragen an obige Adresse müssen auf englisch formuliert werden.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekanntgegeben. IBM kann jederzeit ohne vorherige Ankündigung Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in dieser Veröffentlichung auf Web-Sites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Web-Sites dar. Das über diese Web-Sites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Web-Sites geschieht auf eigene Verantwortung.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation
J74/G4
555 Bailey Avenue
P.O. Box 49023
San Jose, CA 95161-9023
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen — in einigen Fällen auch von der Zahlung einer Gebühr — abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programme oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer gesteuerten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Garantie, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen. Diese Daten stellen deshalb keine Leistungsgarantie dar.

Informationen über Produkte anderer Hersteller als IBM wurden von den Herstellern dieser Produkte zur Verfügung gestellt, bzw. aus von ihnen veröffentlichten Ankündigungen oder anderen öffentlich zugänglichen Quellen entnommen. IBM hat diese Produkte nicht getestet und übernimmt im Hinblick auf Produkte anderer Hersteller keine Verantwortung für einwandfreie Funktion, Kompatibilität oder andere Ansprüche. Fragen hinsichtlich des Leistungsspektrums von Produkten anderer Hersteller als IBM sind an den jeweiligen Hersteller des Produkts zu richten.

Die oben genannten Erklärungen bezüglich der Produktstrategien und Absichtserklärungen von IBM stellen die gegenwärtige Absicht der IBM dar, unterliegen Änderungen oder können zurückgenommen werden, und repräsentieren nur die Ziele der IBM.

Alle von IBM angegebenen Preisen sind empfohlene Richtpreise und können jederzeit ohne weitere Mitteilung geändert werden. Händlerpreise können u. U. von den hier genannten Preisen abweichen.

Diese Veröffentlichung dient nur zu Planungszwecken. Die in dieser Veröffentlichung enthaltenen Informationen können geändert werden, bevor die beschriebenen Produkte verfügbar sind.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des täglichen Geschäftsablaufes. Diese Beispiele enthalten Namen von Personen, Firmen, Marken oder Produkten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHT-LIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispielprogramme wurden nicht unter allen denkbaren Bedingungen getestet.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

Marken

Die folgenden in diesem Buch verwendeten Namen sind Marken der IBM Corporation in den USA und/oder anderen Ländern:

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	RACF
DB2	S/390
DB2 Universal Database	SQL/DS
Distributed Relational Database Architecture	Virtual Machine/Enterprise Systems Architecture
DRDA	Visual Basic
DXT	VM/XA
GDDM	VM/ESA
IBM	VSE/ESA
	VTAM

Java und alle auf Java basierenden Marken und Logos sowie Solaris sind Marken oder eingetragene Marken von Sun Microsystems, Inc. in den USA und/oder anderen Ländern.

Lotus und 1-2-3 sind Marken von Lotus Development Corporation in den USA und/oder anderen Ländern.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken oder eingetragene Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Mit ** gekennzeichnete Namen können Marken oder Dienstleistungsmarken anderer Unternehmen sein.

Index

A

Abfrage

- Aktionsknöpfe für menügesteuerte Abfragen 18
- auf dem Datenbank-Server speichern 26
- ausführen 19
- drucken 16
- erstellen 11
- gespeicherte Abfragedateien öffnen 25
- gespeicherte SQL-Abfragedateien öffnen 14
- mehrere gleichzeitig anzeigen 10
- mehrere Tabellen in menügesteuerten Abfragen verwenden 23
- menügesteuerte Abfrage auf dem Datenbank-Server öffnen 26
- menügesteuerte Abfrage drucken 27
- menügesteuerte Abfrage erstellen 17
- menügesteuerte Abfrage in SQL konvertieren 24
- menügesteuerte Abfrage vorab anzeigen 27
- menügesteuerte Abfragen als Dateien speichern 25
- Schriftart auswählen 10
- Spalten hinzufügen 20
- SQL anzeigen 24
- SQL in menügesteuerten Abfragen verwenden 24
- statische Abfragen erstellen 57
- Substitutionsvariablen verwenden 25
- Tabellen hinzufügen 19
- Verknüpfungsbedingungen für menügesteuerte Abfragen erstellen 24
- vorab anzeigen 16, 34, 55

Abfragedokumente, mehrere 10

Abfrageergebnisse

- Formatierungen in ein Format konvertieren 32, 53
- Schriftart auswählen 31, 52

Abfrageergebnisse drucken 34, 55

- Abfrageergebnisse formatieren 31, 53
- Abfrageergebnisse gruppieren 32, 53
- Abfrageergebnisse in Dateien speichern 33, 54
- Abfrageergebnisse in Übersichten zusammenfassen 32, 54
- Abfrageergebnisse sortieren 30, 52
- Abfrageergebnisse speichern 33, 54
- Abfragen erstellen 20
- Add-In
 - Microsoft Excel 69
- AddDecimalHostVariable() 79
- AddHostVariable() 80
- Aktionsknöpfe
 - für menügesteuerte Abfrage 18
- Ändern der Größe von Spalten und Zeilen 29
- Ändern einer Zeile
 - im Tabelleneditor 63
- Ändern von Kennwörtern 4
- Angaben
 - Formate 36
- Angeben von Kontonamen 5
- Anmelden 3
- Anordnen von Spalten 30, 52
- Anpassen
 - Funktionsleiste 7
 - Symbolleiste 7
 - Zeilen und Spalten automatisch 30, 51
- Anzeige der Abfrage
 - Schriftart auswählen 10
- Anzeige der Abfrageergebnisse
 - Schriftart auswählen 31, 52
- Anzeigen
 - Abfragen 11
 - Ergebnisse 9
 - mehrere gleichzeitig 10
 - menügesteuerte Abfrage vorab anzeigen 27
 - Objekte 48
 - SQL in menügesteuerten Abfragen 24
 - zu druckende Abfrage 16, 34, 55
 - zu druckende Prozedur 46
- API-Referenz 79

Arbeiten mit

- Add-In für Microsoft Excel 69
- Schriftarten 10
- Sortierbedingungen 21
- Zeilenbedingungen 22

Arbeiten mit Beispielanwendungen 70

Auflisten

- Objekte 47

Aufrufen

- gespeicherte SQL-Abfragedateien 14

Ausdrucken

- Druckvorschau für Abfragen anzeigen 34, 55
- SQL-Abfrage 16

Ausführen

- menügesteuerte Abfragen 19
- SQL-Abfrage auf einem Datenbank-Server 9
- statische Abfragen 59

Ausführen auf dem Datenbank-Server

- SQL-Abfragen 9

Ausführen von SQL-Abfragen mit Substitutionsvariablen 13

Auswählen

- Schriftart zur Anzeige der Abfrage 10
- Schriftart zur Anzeige der Abfrageergebnisse 31, 52
- Zeilen und Spalten 29

Automatisches Anpassen von Zeilen und Spalten 30, 51

B

Bedingungen

- beim Sortieren 21
- Formate 35
- Sortierbedingungen hinzufügen 21
- Verknüpfungsbedingungen für menügesteuerte Abfragen erstellen 24
- verwenden 22
- Zeilenbedingungen hinzufügen 22

Bedingungen für Zeilen

- Beginnend mit 22
- Endend mit 22

- Bedingungen für Zeilen (*Forts.*)
 - Enthaltend 22
 - Gleich 22
 - Größer als 22
 - Größer als oder gleich 22
 - Kleiner als 22
 - Kleiner als oder gleich 22
 - NULL 22
 - Zwischen 22
 - Bedingungen für Zeilen (Operatoren)
 - Ist 22
 - Ist nicht 22
 - Beginnend mit (Zeilenbedingung) 22
 - Beispielanwendungen 70
 - Bemerkungen 159
 - Berechnungen
 - Formate 35
 - Berichte
 - drucken 41
 - exportieren 42
 - mit Hilfe von Formaten erstellen 36
 - vorab anzeigen 36
 - BindDecimalHostVariable() 81
 - BindHostVariable() 82
 - BindSection() 83
 - Blockierende Aufrufe 78
- C**
- CancelBind() 83
 - ChangePassword() 84
 - ClearList() 84
 - Close() 85
 - Commit() 85
 - CompleteQuery() 86
 - CopyToClipboard() 86
- D**
- Datei
 - SQL-Abfrage in Dateien speichern 14
 - Dateien
 - Daten in Dateien exportieren 65
 - Daten exportieren
 - auf einem Datenbank-Server 68
 - in Dateien 65
 - Datenbank
 - Sicherheit 3
 - Verbindung herstellen 78
 - Datenbank-Server
 - Daten exportieren 68
 - Formate speichern 40
 - gespeicherte Formate öffnen 41
 - gespeicherte Prozeduren öffnen 45
 - Datenbank-Server (*Forts.*)
 - gespeicherte SQL-Abfragen öffnen 15
 - menügesteuerte Abfrage öffnen 26
 - menügesteuerte Abfragen speichern 26
 - Prozeduren speichern 44
 - SQL-Abfrage auf einem Datenbank-Server 9
 - SQL-Abfragen speichern 14
 - DB2 Forms 63
 - DeleteQMFOBJECT() 88
 - Detailvariationen
 - Formate 35
 - Drucken
 - Berichte 41
 - Druckvorschau für Abfragen anzeigen 34, 55
 - menügesteuerte Abfrage 27
 - Prozedur 46
 - SQL-Abfrage 16
 - Drucken der Abfrageergebnisse 34, 55
 - Druckvorschau
 - für Abfragen 34, 55
 - für Prozeduren 46
 - menügesteuerte Abfrage 27
- E**
- EndBind() 88
 - Endend mit (Zeilenbedingung) 22
 - Entfernen von Knöpfen aus der Symbolleiste 8
 - Entfernen von Objekten aus Listen 49
 - Enthaltend (Zeilenbedingung) 22
 - Ergebnisse
 - drucken 34
 - formatieren 31, 53
 - Ergebnisse der Abfrage sortieren 30
 - Ergebnisse von Abfragen sichern 54
 - Ergebnissicht 9
 - Erstellen
 - Abfragen 11
 - komplexe Abfragen 20
 - menügesteuerte Abfrage 17
 - neue SQL-Abfrage 9
 - Verknüpfungsbedingungen für menügesteuerte Abfragen 24
 - Erstellen einer linearen Prozedur 43
 - Erstellen von Job-Dateien 51
 - Excel
 - Add-In 69
 - Execute() 89
 - ExecuteEx() 89
 - ExecuteStoredProcedure() 90
 - ExecuteStoredProcedureEx() 91
 - Export() 93
 - ExportForm() 95
 - Exportieren
 - Berichte 42
 - Exportieren von Daten
 - in andere Tabellen 68
 - ExportReport() 96
- F**
- FastSaveData() 98
 - Festlegen
 - Schriftart zur Anzeige der Abfrage 10
 - Server 2
 - FetchNextRow() 99
 - FetchNextRowEx() 100
 - FetchNextRows() 100
 - FetchNextRowsEx() 102
 - FlushQMFCache() 103
 - Formate
 - Angaben 36
 - auf dem Datenbank-Server speichern 40
 - Bedingungen 35
 - Berechnungen 35
 - Detailvariationen 35
 - einen Bericht erstellen 36
 - gespeicherte Datei öffnen 39
 - gespeicherte Formate auf dem Datenbank-Server öffnen 41
 - Gruppenwechsel 35
 - Haupt 35
 - HTML 36
 - in Dateien speichern 39
 - Schluss 35
 - Seite 36
 - Spalten 35
 - Formatieren von numerischen Abfrageergebnissen 31, 53
 - Formatierungen von Abfrageergebnissen in ein Format konvertieren 32, 53
 - Funktionsleiste
 - anpassen 7
 - Knöpfe entfernen 8
 - Knöpfe hinzufügen 7
 - Knöpfe verschieben 8
- G**
- Gespeicherte Prozeduren
 - auf dem Datenbank-Server öffnen 45

- Gespeicherte SQL-Abfragen
 - auf dem Datenbank-Server öffnen 15
- Gespeicherte SQL-Listendateien öffnen 50
- GetColumnCount() 103
- GetColumnDataValue() 103
- GetColumnHeader() 104
- GetColumnHeaderEx() 104
- GetColumnHeadings() 105
- GetColumnValue() 106
- GetColumnValueEx() 107
- GetDefaultServerName() 107
- GetGlobalVariable() 108
- GetHostVariableNames() 108
- GetHostVariableTypes() 109
- GetLastErrorString() 109
- GetLastErrorType() 110
- GetLastSQLCode() 111
- GetLastSQLError() 112
- GetLastSQLState() 113
- GetOption() 113
- GetOptionEx() 115
- GetProcText() 115
- GetProcVariables() 116
- GetQMFObjectInfo() 117
- GetQMFObjectInfoEx() 119
- GetQMFObjectList() 120
- GetQMFObjectListEx() 121
- GetQMFProcText() 122
- GetQMFQueryText() 123
- GetQueryText() 123
- GetQueryVerb() 124
- GetResourceLimit() 124
- GetResourceLimitEx() 129
- GetRowCount() 130
- GetServerList() 130
- GetServerListEx() 131
- GetStoredProcedureResultSets() 132
- GetVariables() 133
- GetVariablesEx() 134
- Gleich (Zeilenbedingung) 22
- Gleichzeitiges Anzeigen mehrerer Abfragen 10
- Governor, Steuerung 5
- Größer als (Zeilenbedingung) 22
- Größer als oder gleich (Zeilenbedingung) 22
- Gruppenwechsel
 - Formate 35
- Gruppieren von Abfrageergebnissen 32, 53

H

- Haupt
 - Formate 35

- Herstellen der Verbindung zur Datenbank 78
- Hinzufügen
 - Sortierbedingungen 21
 - Spalten zu menügesteuerten Abfragen 20
 - Tabellen zu menügesteuerten Abfragen 19
 - Zeile im Tabelleneditor 62
 - Zeilenbedingungen 22
- Hinzufügen von Knöpfen zur Symbolleiste 7
- Hinzufügen von Objekte zu Listen 49
- Host-Variablen
 - in statischen Abfragen verwenden 57
- HTML
 - Formate 36

I

- InitializeProc() 134
- InitializeQuery() 135
- InitializeServer() 136
- InitializeStaticQuery() 138
- Internet-Mail 69
- IsStatic() 138
- Ist (Operator für Zeilenbedingungen) 22
- Ist nicht (Operator für Zeilenbedingungen) 22

J

- Job-Dateien erstellen 51

K

- Kennwörter
 - korrigieren 4
- Kennwörter ändern 4
- Kleiner als (Zeilenbedingung) 22
- Kleiner als oder gleich (Zeilenbedingung) 22
- Knöpfe
 - Aktionsknöpfe für menügesteuerte Abfragen 18
 - aus der Symbolleiste entfernen 8
 - in der Symbolleiste hinzufügen 7
 - in der Symbolleiste verschieben 8
- Komplexe Abfragen
 - erstellen 20
- Kontonamen angeben 5

- Konvertieren
 - Formatierungen von Abfrageergebnissen in ein Format 32, 53
- Korrigieren
 - Kennwörter 4

L

- Lineare Prozedur erstellen 43
- Listen
 - gespeicherte Dateien öffnen 50
- Listen, Objekte entfernen 49
- Listen, Objekte hinzufügen 49
- Logon 3
- Löschen einer Zeile
 - im Tabelleneditor 63
- Löschen von Objekten aus Listen 49

M

- Mehrfachabfragen
 - gleichzeitig anzeigen 10
- Mehrfachabfragen, Dokumente 10
- Menügesteuerte Abfrage
 - Aktionsknöpfe 18
 - als Datei speichern 25
 - auf dem Datenbank-Server öffnen 26
 - auf dem Datenbank-Server speichern 26
 - ausführen 19
 - drucken 27
 - erstellen 17
 - gespeicherte Abfragedateien öffnen 25
 - in SQL konvertieren 24
 - mehrere Tabellen verwenden 23
 - Spalten hinzufügen 20
 - SQL anzeigen 24
 - SQL verwenden 24
 - Substitutionsvariablen verwenden 25
 - Tabellen hinzufügen 19
 - Verknüpfungsbedingungen erstellen 24
 - vorab anzeigen 27
- Menügesteuerte Abfragen
 - erstellen 20
- Microsoft Excel
 - Add-In 69

N

- Neu
 - menügesteuerte Abfrage 17
 - SQL-Abfrage 9
- Neu Anordnen von Spalten 30, 52

- NULL (Zeilenbedingung) 22
- Numerische Abfrageergebnisse formatieren 31, 53
- O**
- Objekt anzeigen 48
- Objekt ausführen 48
- Objekt bearbeiten 48
- Objekt zeigen 48
- Objekte
 - auflisten 47
 - aus Listen löschen 49
- Öffnen
 - gespeicherte Datei 39
 - gespeicherte Formate auf dem Datenbank-Server 41
 - gespeicherte Prozedurdatei 44
 - gespeicherte SQL-Abfragedateien 14
 - gespeicherte SQL-Listendateien 50
 - menügesteuerte Abfrage 17
 - menügesteuerte Abfragen, die in Dateien gespeichert sind 25
 - menügesteuerte Abfragen auf dem Datenbank-Server öffnen 26
 - neues SQL-Abfragedokument 9
 - Prozeduren auf dem Datenbank-Server 45
 - SQL-Abfragen auf dem Datenbank-Server 15
- Open() 139
- P**
- Prepare() 140
- PrintReport() 140
- Prozedur
 - drucken 46
- Prozedur erstellen 43
- Prozedur mit Logik 43
- Prozeduren
 - auf dem Datenbank-Server speichern 44
 - gespeicherte Prozedurdatei öffnen 44
 - in Dateien speichern 44
- R**
- ReinitializeServer() 140
- REXX-Prozedur 43
- Rollback() 141
- RunProc() 141
- S**
- SaveData() 142
- SaveQMFProc() 145
- SaveQMFQuery() 146
- Schaltflächen der Symbolleiste hinzufügen 7
- Schaltflächen entfernen 8
- Schaltflächen in der Symbolleiste verschieben 8
- Schluss
 - Formate 35
- Schriftart
 - zur Anzeige der Abfrage 10
 - zur Anzeige der Abfrageergebnisse auswählen 31, 52
- Schriftarten 10
- Seite
 - Formate 36
- Senden an 69
- Server
 - festlegen 2
 - Formate auf dem Datenbank-Server speichern 40
 - gespeicherte Formate auf dem Datenbank-Server öffnen 41
 - gespeicherte Prozeduren auf dem Datenbank-Server öffnen 45
 - menügesteuerte Abfrage auf dem Datenbank-Server öffnen 26
 - Prozeduren auf dem Datenbank-Server speichern 44
- SetBindOption() 147
- SetBindOwner() 149
- SetBusyWindowButton() 149
- SetBusyWindowMessage() 150
- SetBusyWindowMode() 150
- SetBusyWindowTitle() 151
- SetGlobalVariable() 152
- SetHostVariable() 152
- SetOption() 153
- SetParent() 154
- SetProcVariable() 155
- SetVariable() 156
- ShowBusyWindow() 156
- Sicherheit
 - Datenbank 3
- Sichern
 - Formate auf dem Datenbank-Server 40
 - Prozeduren auf dem Datenbank-Server 44
 - Prozeduren in Dateien 44
 - SQL-Abfrage in Dateien 14
 - SQL-Abfragen auf dem Datenbank-Server 14
- Sichern von Abfrageergebnissen 54
- Sichten
 - SQL-Sicht und Ergebnissicht 9
- Sortierbedingungen
 - hinzufügen 21
 - verwenden 21
- Sortieren der Abfrageergebnisse 30, 52
- Spalten
 - Formate 35
 - zu menügesteuerten Abfragen hinzufügen 20
- Spalten auswählen 29
- Spalten automatisch anpassen 30, 51
- Spalten neu anordnen 30, 52
- Spalten und Zeilen auswählen 29
- Spalten und Zeilen in der Größe ändern 29
- Speichern
 - Formate auf dem Datenbank-Server 40
 - Formate in Dateien 39
 - menügesteuerte Abfragen als Dateien 25
 - menügesteuerte Abfragen auf dem Datenbank-Server 26
 - Prozeduren auf dem Datenbank-Server 44
 - Prozeduren in Dateien 44
 - SQL-Abfragen auf dem Datenbank-Server 14
- Speichern in Dateien
 - SQL-Abfrage 14
- Speichern von Abfrageergebnissen 33, 54
- Speichern von Abfrageergebnissen in Dateien 33, 54
- Spreadsheet
 - Microsoft Excel 69
- SQL
 - in menügesteuerten Abfragen verwenden 24
 - menügesteuerte Abfrage in SQL konvertieren 24
- SQL-Abfrage
 - auf dem Datenbank-Server ausführen 9
 - auf dem Datenbank-Server öffnen 15
 - auf dem Datenbank-Server speichern 14
 - drucken 16
 - gespeicherte Datei öffnen 14
 - in Dateien speichern 14
 - mit Substitutionsvariablen ausführen 13
 - neue 9

- SQL-Abfrage (*Forts.*)
 - neues SQL-Abfragedokument öffnen 9
 - Substitutionsvariablen 12
 - vorab anzeigen 16, 34, 55
- SQL-Listendateien öffnen 50
- SQL-Sicht
 - anzeigen 9
- StartBind() 157
- Statische Abfragen
 - ausführen 59
 - erstellen 57
- Statische Abfragen erstellen
 - Substitutionsvariablen verwenden 57
- Steuerung mit dem Governor 5
- Substitutionsvariablen
 - durch Host-Variablen ersetzen 57
 - in menügesteuerten Abfragen verwenden 25
 - in SQL-Abfragen 12
 - in statischen Abfragen verwenden 57
 - SQL-Abfragen ausführen 13
- Suchen
 - mit dem Tabelleneditor nach Zeilen 61
- Symbolleiste
 - anpassen 7
 - Schaltflächen entfernen 8
 - Schaltflächen hinzufügen 7
 - Schaltflächen verschieben 8

T

- Tabellen
 - Daten exportieren 68
 - mehrere Tabellen in menügesteuerten Abfragen verwenden 23
 - zu menügesteuerten Abfragen hinzufügen 19
- Tabelleneditor 61
 - eine Zeile hinzufügen 62
 - nach Zeilen suchen 61
 - Zeile ändern 63
 - Zeile löschen 63
- Tabellenkalkulation
 - Microsoft Excel 69

U

- Übersichten für Abfrageergebnisse 32
- Umordnen von Spalten 30, 52

- Umschalten
 - zwischen der Ergebnissicht und der SQL-Sicht 9
- Umsetzen
 - menügesteuerte Abfrage in SQL 24
- Umwandeln
 - menügesteuerte Abfrage in SQL 24

V

- Variablen
 - Substitutionsvariablen in menügesteuerten Abfragen verwenden 25
- Verbindung zur Datenbank herstellen 78
- Verknüpfungsbedingungen
 - für menügesteuerte Abfragen erstellen 24
- Verschieben von Knöpfen in der Symbolleiste 8
- Verwenden
 - Add-In für Microsoft Excel 69
- Verwenden von Schriftarten 10
- Vorab anzeigen 55
 - Berichte 36
 - menügesteuerte Abfragen 27
 - zu druckende Abfrage 16, 34
 - zu druckende Prozedur 46

W

- Wechseln
 - zwischen der Ergebnissicht und der SQL-Sicht 9

Z

- Zeigen
 - Abfragen 11
- Zeile ändern
 - im Tabelleneditor 63
- Zeile hinzufügen
 - im Tabelleneditor 62
- Zeile löschen
 - im Tabelleneditor 63
- Zeilen auswählen 29
- Zeilen automatisch anpassen 30, 51
- Zeilen in der Größe ändern 29
- Zeilenbedingungen
 - Beginnend mit 22
 - Endend mit 22
 - Enthaltend 22
 - Gleich 22
 - Größer als 22
 - Größer als oder gleich 22
 - hinzufügen 22
- Zeilenbedingungen (*Forts.*)
 - Kleiner als 22
 - Kleiner als oder gleich 22
 - NULL 22
 - verwenden 22
 - Zwischen 22
- Zeilenbedingungen (Operatoren)
 - Ist 22
 - Ist nicht 22
- Zusammenfassen von Abfrageergebnissen 32, 54
- Zwischen (Zeilenbedingung) 22

Antwort

Query Management Facility
Erste Schritte mit QMF für Windows
Version 7

IBM Form SC12-2847-00

Anregungen zur Verbesserung und Ergänzung dieser Veröffentlichung nehmen wir gerne entgegen. Bitte informieren Sie uns über Fehler, ungenaue Darstellungen oder andere Mängel.

Zur Klärung technischer Fragen sowie zu Liefermöglichkeiten und Preisen wenden Sie sich bitte entweder an Ihre IBM Geschäftsstelle, Ihren IBM Geschäftspartner oder Ihren Händler.

Unsere Telefonauskunft "HALLO IBM" (Telefonnr.: 01803/31 32 33) steht Ihnen ebenfalls zur Klärung allgemeiner Fragen zur Verfügung.

Kommentare:

Danke für Ihre Bemühungen.

Sie können ihre Kommentare betr. dieser Veröffentlichung wie folgt senden:

- Als Brief an die Postanschrift auf der Rückseite dieses Formulars
- Als E-Mail an die folgende Adresse: comment@tcv.vnet.ibm.com

Name

Adresse

Firma oder Organisation

Rufnummer

E-Mail-Adresse

Antwort
SC12-2847-00



IBM Deutschland Informationssysteme GmbH
SW NLS Center

70548 Stuttgart



Programmnummer: 5675-DB2
5697-F42
5697-G24
5697-G22
5648-D35
5697-G23

Printed in Denmark

SC12-2847-00

