DB2 Universal Database

Image, Audio und Video Extender Verwaltung und Programmierung

Version 8

SH12-3037-00

DB2 Universal Database

Image, Audio und Video Extender Verwaltung und Programmierung

Version 8

SH12-3037-00

Anmerkung:
Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter *Bemerkungen* gelesen werden.

- Die IBM Homepage finden Sie im Internet unter: **ibm.com**
- IBM und das IBM Logo sind eingetragene Marken der International Business Machines Corporation.
- Das e-business-Symbol ist eine Marke der International Business Machines Corporation.
- Infoprint ist eine eingetragene Marke der IBM.
- ActionMedia, LANDesk, MMX, Pentium und ProShare sind Marken der Intel Corporation in den USA und/oder anderen Ländern.
- C-bus ist eine Marke der Corollary, Inc. in den USA und/oder anderen Ländern.
- Java und alle auf Java basierenden Marken und Logos sind Marken der Sun Microsystems, Inc. in den USA und/oder anderen Ländern.
- Microsoft Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.
- PC Direct ist eine Marke der Ziff Communications Company in den USA und/oder anderen Ländern.
- SET und das SET-Logo sind Marken der SET Secure Electronic Transaction LLC.
- UNIX ist eine eingetragene Marke der Open Group in den USA und/oder anderen Ländern.
- Marken anderer Unternehmen/Hersteller werden anerkannt.

Erste Ausgabe, Juni 2003

Diese Veröffentlichung ist eine Übersetzung des Handbuchs

DB2 Universal Database Image, Audio, and Video Extenders Administration and Programming Version 8,
IBM Form SH12-6747-00,
herausgegeben von International Business Machines Corporation, USA

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 TSC Germany
Kst. 2877
Juni 2003

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellen

Zu diesem Handbuch

In diesem Handbuch wird beschrieben, wie die DB2 Extender zur Vorbereitung und Verwaltung einer DB2-Datenbank für Abbild-, Audio- oder Videodaten verwendet werden. Außerdem wird beschrieben, wie Sie die benutzerdefinierten Funktionen (UDFs) und die Anwendungsprogrammierschnittstellen (APIs) verwenden können, die von den DB2-Extendern für den Zugriff und die Bearbeitung dieser Datentypen zur Verfügung gestellt werden. Durch die Integration von UDFs in die SQL-Anweisungen Ihres Programms und die Integration von APIs können Sie auf nicht traditionelle Daten, wie z. B. Abbilder und Videoclips, sowie auf traditionelle numerische Daten und Zeichendaten zugreifen.

Verweise auf ″DB2″ in diesem Buch beziehen sich auf DB2 UDB.

Zielgruppe

Dieses Handbuch ist für DB2-Datenbankadministratoren konzipiert, die mit den Konzepten, Tools und Methoden der DB2-Verwaltung vertraut sind.

Dieses Handbuch ist außerdem für DB2-Anwendungsprogrammierer konzipiert, die mit SQL und einer oder mehreren Programmiersprachen vertraut sind, die für DB2-Anwendungsprogramme verwendet werden können.

Dieses Handbuch richtet sich an Personen, die mit dem DB2 Image Extender, Audio Extender und Video Extender arbeiten werden. Personen, die mit dem Text Extender arbeiten wollen, sollten das Handbuch *DB2 Universal Database Text Extender Verwaltung und Programmierung* lesen.

Benutzung des Handbuchs

Dieses Handbuch ist wie folgt strukturiert:

„Teil 1. Einführung”

Dieser Teil gibt einen Überblick über die DB2 Extender. Lesen Sie diesen Teil, wenn Sie noch keine Verwaltungs- oder Programmieraufgaben mit den DB2 Extendern durchgeführt haben.

„Teil 2. Abbild-, Audio- und Videodaten verwalten”

In diesem Teil wird beschrieben, wie eine DB2-Datenbank für Abbild-, Audio- und Videodaten vorbereitet und verwaltet werden kann. Lesen Sie diesen Teil, wenn Sie eine DB2-Datenbank verwalten wollen, die Abbild-, Audio- oder Videodaten enthält.

„Teil 3. Programmierung für Abbild-, Audio- und Videodaten”

In diesem Teil wird beschrieben, wie die benutzerdefinierten Funktionen (UDFs) und Anwendungsprogrammierschnittstellen (APIs) der DB2 Extender verwendet werden können, um bestimmte Operationen für Abbild-, Audio- oder Videodaten anzufordern. Lesen Sie diesen Teil, wenn Sie auf Abbild-, Audio- oder Videodaten in einem DB2-Anwendungsprogramm zugreifen und diese bearbeiten müssen.

„Teil 4. Referenz"

In diesem Teil werden Referenzinformationen für die UDFs, APIs, Verwaltungsbefehle und Diagnoseinformationen der DB2 Extender (z. B. Nachrichten und Codes) erläutert. Lesen Sie diesen Teil, wenn Sie mit den Konzepten und Tasks der DB2 Extender vertraut sind, aber Informationen zu einer bestimmten UDF, API oder Nachricht bzw. zu einem bestimmten Befehl oder Code der DB2 Extender benötigen.

„Anhänge"

In den Anhängen werden folgende Themen beschrieben:

- Definieren der Umgebungsvariablen, die von den DB2 Extendern verwendet werden, um Dateien zu suchen und Anzeige- oder Wiedergabeprogramme für Abbild-, Audio- und Videoobjekte zu identifizieren.
- Installieren und Verwenden der Beispielprogramme und Multimediadateien, die mit den Extendern geliefert werden.

Plattformspezifische Informationen

Die DB2 Extender können in Verbindung mit der Einzelpartitionsdatenbankumgebung von DB2 Universal Database oder mit der Mehrpartitionsdatenbankumgebung der DB2 Universal Database Enterprise Extended Edition verwendet werden.

Dieses Handbuch enthält Informationen zur Verwendung der DB2 Extender in beiden Umgebungen. Informationen, die sich nur auf die Verwendung der Extender in der Mehrpartitionsumgebung der DB2 Universal Database Enterprise Extended Edition beziehen, sind mit "**Nur EEE**" markiert. Informationen, die sich nur auf die Verwendung der Extender in der Einzelpartitionsumgebung der DB2 Universal Database beziehen, sind mit "**Nur Nicht-EEE**" markiert. Informationen, die nicht für eine bestimmte Umgebung markiert sind, gelten für beide Umgebungen.

Hervorhebungskonventionen

Im vorliegenden Handbuch werden folgende Hervorhebungskonventionen verwendet:

Fett Fettdruck kennzeichnet die Definition eines neues Begriffs.

Kursiv Kursivdruck kennzeichnet Variablenparameter, die durch einen Wert ersetzt werden, oder hebt Wörter im allgemeinen Text hervor.

GROSSBUCHSTABEN
Großbuchstaben kennzeichnen Folgendes:
- Datentypen
- Verzeichnisnamen
- Feldnamen
- API-Aufrufe
- Befehle
- Schlüsselwörter
- Variablennamen

`Monospace`
Monospace-Schrift kennzeichnet eine Systemnachricht oder einen Wert, den Sie eingeben. Monospace-Schrift wird außerdem für Codierungsbeispiele verwendet.

Beschreibung der Syntaxdiagramme

Im vorliegenden Handbuch wird die Syntax von Befehlen und SQL-Anweisungen mit Hilfe von Syntaxdiagrammen beschrieben. Die Syntaxdiagramme sind wie folgt strukturiert:

- Syntaxdiagramme werden von links nach rechts und von oben nach unten gelesen, wobei die Linie die Richtung angibt.

  Das Symbol ►►— kennzeichnet den Anfang einer Anweisung.

  Das Symbol —► gibt an, dass die Anweisungssyntax in der nächsten Zeile fortgesetzt wird.

  Das Symbol ►— gibt an, dass es sich um die Fortsetzung einer Anweisung aus der vorherigen Zeile handelt.

  Das Symbol —►◄ kennzeichnet das Ende einer Anweisung.
- Erforderliche Elemente sind auf der horizontalen Linie (dem Hauptpfad) dargestellt.

- Wahlfreie Elemente werden unterhalb des Hauptpfads dargestellt.

- Besteht die Möglichkeit, aus mehreren Elementen auszuwählen, sind diese untereinander angeordnet.

  Wenn Sie eines dieser Elemente auswählen *müssen*, ist eines der untereinander angeordneten Elemente auf dem Hauptpfad dargestellt.

  Ist es nicht unbedingt erforderlich, dass Sie eines der Elemente auswählen, werden alle auswählbaren Elemente untereinander unterhalb des Hauptpfads dargestellt.

  Ein Wiederholungspfeil oberhalb der aufgelisteten Elemente gibt an, dass Sie mehrere der aufgeführten Elemente auswählen können.

- Schlüsselwörter sind in Großbuchstaben dargestellt (z. B. `/DB2IMAGE:`). Diese Schreibweise muss exakt eingehalten werden. Variablen sind in Kleinbuchstaben dargestellt (z. B. `srcpath`). Mit den Variablen werden in der Syntax Namen oder Werte dargestellt, die vom Benutzer eingegeben werden.
- Sind Interpunktionszeichen, runde Klammern, arithmetische Operatoren oder andere, ähnliche Symbole dargestellt, müssen diese als Teil der Syntax eingegeben werden.

Zugehörige Informationen

DB2 Universal Database Version 8

DB2 Universal Database Version 8 für DB2-Server Einstieg (GC12–3047), für DB2-Clients Einstieg (GC12–3052), Connect Personal Edition Einstieg (GC12–3049), Version 8 Personal Edition Einstieg (GC12–3045) und Data Links Manager Einstieg (GC12–3056). In diesen Büchern wird die Planung, Installation, Konfiguration und Migration von DB2 Universal Database auf den entsprechenden Plattformen beschrieben.

IBM DB2 Universal Database Version 8 Systemverwaltung: Konzept (SC12–3057), Optimierung (SC12–3058) und Implementierung (SC12–3059). In diesen Handbüchern wird der Entwurf und die Implementierung einer DB2-Datenbank beschrieben.

IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1, Version 8 (SC09–4849). In diesem Buch wird beschrieben, wie Anwendungen entwickelt werden, die unter Verwendung der DB2 Call Level Interface, einer aufrufbaren SQL-Schnittstelle, die mit der Microsoft-ODBC-Spezifikation kompatibel ist, auf DB2-Datenbanken zugreifen.

IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2, Version 8 (SC09–4850). In diesem Buch wird beschrieben, wie Anwendungen entwickelt werden, die unter Verwendung der DB2 Call Level Interface, einer aufrufbaren SQL-Schnittstelle, die mit der Microsoft-ODBC-Spezifikation kompatibel ist, auf DB2-Datenbanken zugreifen.

IBM DB2 Universal Database Command Reference Version 8 (SC09–4829). In diesem Buch wird beschrieben, wie der DB2-Befehlszeilenprozessor verwendet wird, und werden Referenzinformationen zu DB2-Befehlen gegeben.

DB2 Universal Database Text Extender

DB2 Universal Database Text Extender Verwaltung und Programmierung, Version 7, SC12-2893. In diesem Buch wird beschrieben, wie eine DB2-Datenbank für Textdaten verwaltet wird. Außerdem wird beschrieben, wie die Anwendungsprogrammierschnittstellen verwendet werden, die vom DB2 Text Extender für den Zugriff und die Bearbeitung von Textdaten zur Verfügung gestellt werden.

DB2 Universal Database XML Extender

DB2 Universal Database XML Extender Verwaltung und Programmierung. In diesem Buch wird beschrieben, wie eine DB2-Datenbank für XML-Dokumente verwaltet wird. Außerdem wird beschrieben, wie die Anwendungsprogrammierschnittstellen verwendet werden, die vom DB2 XML Extender für den Zugriff und die Bearbeitung von XML-Dokumenten und -Daten zur Verfügung gestellt werden.

DB2 Universal Database Spatial Extender

Spatial Extender Benutzer- und Referenzhandbuch, IBM Form SC12-2894. Dieses Handbuch enthält Informationen zur Installation, Konfiguration, Verwaltung, Programmierung und Fehlerbehebung des Spatial Extender. Außerdem enthält es wichtige Beschreibungen der Konzepte für räumliche Daten sowie Referenzinformationen (Nachrichten und SQL), die speziell für den Spatial Extender gelten.

DB2 Universal Database für z/OS Image, Audio und Video Extender

DB2 Universal Database for z/OS Version 6 Image, Audio, and Video Extenders Administration and Programming, SC26-9650. In diesem Buch wird beschrieben, wie ein DB2 für z/OS-Datenbankserver für Abbild-, Audio- und Videodaten verwaltet wird. Außerdem wird die Verwendung von benutzerdefinierten Funktionen und Anwendungsprogrammierschnittstellen beschrieben, die von den DB2 für z/OS Image, Audio und Video Extendern bereitgestellt werden, um auf Abbild-, Audio- und Videodaten zuzugreifen und diese Daten zu bearbeiten.

DB2 Universal Database für z/OS Text Extender

DB2 Universal Database for z/OS Version 6 Text Extender Administration and Programming, SC26-9651. In diesem Buch wird beschrieben, wie ein DB2 für z/OS-Datenbankserver für Textdaten verwaltet wird. Außerdem wird beschrieben, wie die benutzerdefinierten Funktionen und Anwendungsprogrammierschnittstellen, die vom DB2 für z/OS Text Extender bereitgestellt werden, verwendet werden können, um auf Textdaten zuzugreifen und diese zu bearbeiten.

Kommentare zum vorliegenden Buch

Ihr Feedback unterstützt IBM bei der Bereitstellung qualitativ hochwertiger Informationsmaterialen. Bitte senden Sie uns Ihre Kommentare zum vorliegenden Handbuch oder zu anderen DB2 Extender-Dokumentationen. Zur Abgabe von Kommentaren können Sie folgendermaßen vorgehen:

- Senden Sie Ihre Kommentare in einer E-Mail an die Adresse swsdid@de.ibm.com.

Bitte geben Sie unbedingt die Version des Produkts sowie den Titel und die IBM Formnummer (sofern vorhanden) der Veröffentlichung an, auf die sich Ihr Kommentar bezieht. Geben Sie bei Kommentaren zu einer spezifischen Textstelle bitte auch die Position dieser Textstelle (z. B. Abbildungs- oder Seitennummer) innerhalb der Veröffentlichung an.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Weitere Informationen dazu, wie Sie IBM kontaktieren können, finden Sie unter .

Teil 1. Einführung

Kapitel 1. Einführung

Extender-Server verwalten

Die DB2 Extender werden in der DB2-Client/Server-Umgebung ausgeführt. Diese Umgebung besteht aus einem Datenbankserver und einem oder mehreren fernen Datenbankclients. Die DB2 Extender-Services werden auf dem Server ausgeführt. Bevor Sie auf sie zugreifen können, müssen Sie sie starten.

Ist die Umgebung definiert, können Sie die Extender-Services vom Client aus stoppen und erneut starten. Sowohl vom Client als auch vom Server aus können Sie den Status der Extender abrufen.

Nur EEE: In einer Mehrpartitionsumgebung können Sie außerdem Datenbankpartitionen hinzufügen und löschen.

Extender-Umgebungen einrichten

Auf dem Server in der Befehlszeile des Betriebssystems den Befehl DMBSTART eingeben, um die Extender-Services zu starten:

```
dmbstart
```

Mit dem Befehl DMBSTART werden die Extender-Services für alle Datenbanken gestartet, die zur Speicherung von Extender-Daten aktiviert sind. Der Befehl startet außerdem DB2, wenn es noch nicht aktiv ist. Sie benötigen die Berechtigung SYSADM, SYSCTRL oder SYSMAINT, um den Befehl ausführen zu können. Unter AIX müssen Sie als Extender-Exemplareigner angemeldet sein.

Jetzt kann Ihre C-Anwendung über die APIs auf Extender-Services zugreifen, wenn die Anwendung eine Verbindung zur Datenbank herstellt. Ebenso müssen Sie, wenn Sie die db2ext-Befehlszeile verwenden wollen, eine Verbindung zur Datenbank herstellen, mit der Sie arbeiten wollen. Für die db2ext-Befehlszeile ist eine unabhängige Verbindung erforderlich, separat von derjenigen, die von der DB2-Befehlszeile verwendet wird.

Öffnen Sie den db2ext-Befehlszeilenprozessor auf dem Client, und führen Sie den DB2 Extender-Befehl CONNECT aus. Im folgenden Beispiel stellt der Befehl eine Verbindung zur Datenbank PERSONNL her. Er greift unter Verwendung des Kennworts ANPASS auf Tabellen mit dem Qualifikationsmerkmal ANITAS zu:

```
connect to personnl user anitas using anpass
```

Nur EEE: Wenn Sie die DB2 Extender in einer Umgebung für partitionierte Datenbanken verwenden, startet der Befehl DMBSTART die Extender-Services auf allen Datenbankpartitionsservern, die für das Exemplar definiert sind. Wenn Sie die Extender-Services nur auf einem Datenbankpartitionsserver starten wollen, geben Sie im Befehl den Knoten an, der gestartet werden soll. Das folgende Beispiel zeigt den Befehl zum Starten der Extender-Services am Knoten mit der Nummer 2.

```
dmbstart nodenum 2
```

Nur EEE: Bevor Sie einen Einzel-Partitionsdatenbankserver starten, müssen Sie auf dem gewünschten Knoten DB2 starten.

Jetzt können Sie die übrigen DB2 Extender-Befehle ausführen, die in Kapitel 15, „Verwaltungsbefehle für den Client", auf Seite 449, aufgelistet sind.

Datenbankpartitionen hinzufügen und löschen (nur EEE)

Damit Sie die Extender in einer Umgebung für partitionierte Datenbanken verwenden können, müssen die für die Extender definierten Partitionen mit denen übereinstimmen, die für DB2 definiert sind. Mit dem Befehl DMBSTART werden die Extender-Server auf allen Knoten gestartet, die für das aktuelle Exemplar definiert sind. Der Server stellt automatisch fest, ob der Knoten, auf dem er aktiv ist, kürzlich erstellt wurde, und führt die notwendigen Initialisierungen aus. Wenn ein Knoten von DB2 gelöscht wurde, müssen die Extender-Dateien, die diesem Knoten zugeordnet sind, manuell gelöscht werden.

Weitere Informationen zu DB2-Befehlen zum Hinzufügen und Löschen von Partitionen finden Sie im Buch *IBM DB2 Universal Database Enterprise Extended-Edition Einstieg*.

Folgende Schritte sind erforderlich, um eine Datenbankpartition hinzuzufügen:

1. Erstellen Sie unter Verwendung des Befehls DB2NCRT oder des Befehls DB2START ADDNODE eine Partition für DB2.
2. Erstellen Sie unter Verwendung des Extender-Befehls DMBSTART NODENUM eine Partition für die Extender.
3. Verteilen Sie die DB2-Daten neu, um die Vorteile der neuen Knotenkonfiguration zu nutzen. Verwenden Sie dazu den DB2-Befehl REDISTRIBUTE NODEGROUP.
4. Verteilen Sie die Extender-Daten neu, um die Vorteile der neuen Knotenkonfiguration zu nutzen. Verwenden Sie dazu den Extender-Befehl REDISTRIBUTE NODEGROUP.

Folgende Schritte sind erforderlich, um eine Datenbankpartition zu löschen:

1. Verteilen Sie die DB2-Daten neu, um sie von der Partition zu entfernen, die Sie löschen wollen. Verwenden Sie dazu den DB2-Befehl REDISTRIBUTE NODEGROUP.
2. Verteilen Sie die Extender-Daten neu, um sie von der Partition zu entfernen, die Sie löschen wollen. Verwenden Sie dazu den Extender-Befehl REDISTRIBUTE NODEGROUP.
3. Löschen Sie eine Partition für DB2, indem Sie den DB2-Befehl DB2NDROP bzw. den Befehl DB2STOP DROP verwenden.
4. Löschen Sie eine Partition für die Extender, indem Sie den Extender-Befehl DMBSTART NODENUM verwenden.
5. Löschen Sie manuell die Extender-Dateien, die der gelöschten Partition zugeordnet sind.

Die Extender-Daten für eine Datenbankpartition befinden sich in dem Unterverzeichnis node*num*, wobei *num* die Knotennummer ist, die der Datenbankpartition entspricht. Das Unterverzeichnis befindet sich in einem Verzeichnis, das durch den Wert der Umgebungsvariablen DB2MMDATAPATH angegeben ist. Löschen Sie, um die Extender-Daten für eine gelöschte Datenbankpartition zu löschen, das entsprechende Unterverzeichnis node*num* und alle Unterverzeichnisse darunter. (Weitere Informationen zu DB2MMDATAPATH finden Sie im Abschnitt „Umgebungsvariable DB2MMDATAPATH verwenden (nur EEE)" auf Seite 523.)

Extender-Server stoppen und starten

Wenn Sie Anwendungen stoppen, die die Extender-Services verwenden, bleibt der Server aktiv, bis Sie ihn explizit stoppen oder bis die Servermaschine erneut gestartet wird. Sie können alle Extender-Server stoppen, indem Sie von der Servermaschine aus in der Befehlszeile für das Betriebssystem den Befehl DMBSTOP eingeben.

Um die Extender-Services vom Client aus zu stoppen und zu starten, führen Sie die Befehle STOP SERVER und START SERVER von der db2ext-Befehlszeile aus aus. Mit diesen Befehlen werden die Extender-Services für die aktuelle Datenbank gestoppt und gestartet.

Nur EEE: In einer Umgebung für partitionierte Datenbanken kann der Befehl DMBSTART verwendet werden, um entweder alle Datenbankpartitionsserver, die für das Exemplar definiert sind, oder nur einen Einzel-Datenbankpartitionsserver zu starten. Der Befehl DMBSTART ohne Parameter startet alle Datenbankpartitionsserver. Wenn Sie nur einen Datenbankpartitionsserver starten wollen, geben Sie im Befehl wie folgt den Knoten an, der gestartet werden soll:

```
dmbstart nodenum 2
```

Wenn Sie den Server auf einem bestimmten Knoten gestartet haben, müssen Sie die Verbindung von diesem Server zur Datenbank wiederherstellen. Verwenden Sie dazu den folgenden Extender-Befehl RECONNECT SERVER:

```
reconnect server at nodenum 2
```

Nur EEE: Wenn Sie die DB2 Extender in einer Umgebung für partitionierte Datenbanken verwenden, können mit dem Befehl DMBSTOP ohne Parameter alle Datenbankpartitionsserver gestoppt werden, die für das Exemplar definiert sind. Wenn Sie nur einen Datenbankpartitionsserver stoppen wollen, müssen Sie zunächst die Verbindung zwischen diesem Server und der Datenbank unterbrechen. Verwenden Sie dazu den folgenden Extender-Befehl DISCONNECT SERVER:

```
disconnect server at nodenum 2
```

Anschließend können Sie den Befehl DMBSTOP ausführen und dabei den Knoten angeben, der gestoppt werden soll. Das folgende Beispiel zeigt den Befehl zum Eingeben in der Befehlszeile des Servers, mit dem die Extender-Services am Knoten mit der Nummer 2 gestoppt werden.

```
dmbstop nodenum 2
```

Nur EEE: Führen Sie den Befehl DMBSTOP nicht für einen bestimmten Knoten aus, es sei denn, Ihre Datenbank läuft im Wartungsmodus. Darüber hinaus müssen Sie sicherstellen, dass auf dem Knoten während des Ausschaltens keine Extender-Aktivitäten ausgelöst werden. Andernfalls kann es zu unerwarteten Folgen kommen.

Serverstatus anzeigen

Vom Server aus können Sie mit dem Befehl DMBSTAT den Status des Extender-Servers anzeigen. Beispielsweise werden mit dem folgenden Befehl die aktivierten Datenbanken aufgelistet und wird angeben, ob die Extender aktiv sind. Stellen Sie eine Verbindung zum Server her, bevor Sie diesen Befehl ausführen.

```
dmbstat
```

Vom Client aus können Sie unter Verwendung des Befehls GET SERVER STATUS den Status des Extender-Servers für eine Datenbank abrufen. Beispielsweise wird mit dem folgenden Befehl der Status der Datenbank PERSONNL abgerufen:

```
get server status personnl
```

Mehrere Serverexemplare erstellen und verwalten

Sie können mehrere Exemplare des DB2 Extender-Servers erstellen und verwenden. Sie sollten mehrere Exemplare erstellen, wenn Sie mehrere Exemplare des DB2-Servers erstellt haben. Jedes Exemplar des DB2 Extender-Servers ist einem Exemplar des DB2-Servers zugeordnet und hat den gleichen Namen. Sie können auch die Exemplare des DB2 Extender-Servers, die auf dem System verfügbar sind, auflisten, mehrere Exemplare gleichzeitig ausführen und Exemplare löschen.

Mehrere DB2 Extender-Serverexemplare erstellen

Ein DB2 Extender-Anfangs- oder -Standardexemplar wird erstellt, wenn Sie die DB2 Extender installieren. Diesem Exemplar wird der gleiche Name zugeordnet wie dem DB2-Standardexemplar. Unter Windows hat das DB2 Extender-Standardexemplar den Namen 'DB2'. Unter UNIX ist diesem der gleiche Name zugeordnet wie dem zu Anfang definierten DB2-Standardexemplar. Um zusätzliche Exemplare des DB2 Extender-Servers zu erstellen, müssen Sie über die Berechtigung SYSADMIN und unter UNIX über Root-Berechtigung verfügen.

Verwenden Sie den Befehl DMBICRT, um ein zusätzliches Exemplar des DB2 Image, Audio und Video Extender-Servers zu erstellen. Wenn Sie ein DB2 Extender-Serverexemplar für das DB2-Exemplar DEVINST erstellen wollen, geben Sie Folgendes in einer Befehlszeile des Betriebssystems ein:

```
dmbicrt  devinst
```

Wenn Sie den Befehl DMBICRT ausführen, wird ein Unterverzeichnis für das Exemplar erstellt und das Exemplar wird zur Liste der Exemplare hinzugefügt, die von den DB2 Extendern verwaltet wird.

Nur EEE:

- Unter Windows hat das Standardexemplar der DB2 Extender-Server den Namen 'DB2MPP'.
- Bei Verwendung des Befehls DMBICRT zum Erstellen von zusätzlichen Exemplaren des DB2 Image, Audio und Video Extender-Servers müssen Sie das Verzeichnis angeben, das die Extender für die verschiedenen Operationen in einer Umgebung für partitionierte Datenbanken verwenden. Dies ist das Verzeichnis, das unter UNIX in der Umgebungsvariablen DB2MMDATAPATH und unter Windows im Registrierungsdatenbankeintrag angegeben ist. Es muss ein gemeinsam benutztes Verzeichnis sein und es muss auf allen Knoten für das Exemplar existieren.
- Sie müssen außerdem einen Bereich von TCP/IP-Anschlüssen angeben, der unter Windows verwendet werden soll. Unter UNIX muss der Anschlussbereich zur Datei `/etc/services` hinzugefügt werden (siehe „DMBICRT" auf Seite 10).

Exemplare auflisten

Verwenden Sie den Befehl DMBILIST, um alle Exemplare des DB2 Extender-Servers, die auf dem System verfügbar sind, aufzulisten. Um festzustellen, welches Exemplar aktiv ist, geben Sie folgenden Befehl ein:

```
echo  %DB2INSTANCE%    (unter Windows)

echo $DB2INSTANCE      (unter UNIX)
```

Mehrere Exemplare gleichzeitig ausführen

Um mehrere Exemplare des DB2 Extender-Servers gleichzeitig auszuführen, führen Sie folgende Schritte aus:

Unter Windows

Führen Sie von einer Befehlszeile aus Folgendes aus:

1. Setzen Sie die Variable DB2INSTANCE auf den Namen des Exemplars, das Sie starten wollen, indem Sie Folgendes eingeben:

   ```
   set db2instance=exemplarname
   ```

2. Starten Sie die Extender-Services.

Unter UNIX

1. Melden Sie sich als Exemplareigner oder als Benutzer mit Systemverwaltungsberechtigung für das Exemplar an.
2. Richten Sie die Umgebung ein.
3. Starten Sie den Datenbankmanager.

Das aktuelle Exemplar einrichten

Wenn Sie Befehle ausführen, um Services für ein Exemplar zu starten oder zu stoppen, gelten die Befehle für das aktuelle Exemplar. Sie geben an, welches Exemplar des DB2 Extender-Servers verwendet werden soll, indem Sie die Variable DB2INSTANCE auf den Exemplarnamen setzen.

Exemplare löschen

Um ein Exemplar der DB2 Extender zu löschen, führen Sie folgende Schritte aus:

1. Stoppen Sie alle Anwendungen, die das Exemplar momentan verwenden.
2. Stoppen Sie die Extender-Services und alle Sitzungen des db2ext-Befehlszeilenprozessors, indem Sie die Befehle DMBSTOP und db2ext TERMINATE verwenden.
3. Erstellen Sie eine Sicherungskopie der Dateien im Verzeichnis des Extender-Exemplars, die Sie sichern wollen, z. B. die QBIC-Katalogdateien. Die Dateien in diesem Verzeichnis werden gelöscht, wenn das Exemplar gelöscht wird.
4. Geben Sie den Befehl DMBIDROP für das Exemplar ein, das gelöscht werden soll. Um beispielsweise das Exemplar DEVINST zu löschen, geben Sie Folgendes ein:

   ```
   dmbidrop devinst
   ```

Durch das Löschen eines Exemplars der DB2 Extender mit Hilfe des Befehls DMBIDROP wird das zugehörige DB2-Exemplar nicht gelöscht. Sie müssen das zugehörige DB2-Exemplar separat löschen. Wenn Sie das DB2-Exemplar löschen, das einem Exemplar der DB2 Extender zugeordnet ist, wird das DB2 Extender-Exemplar nicht gelöscht. Sie können es jedoch nicht verwenden.

Exemplare migrieren

Auf UNIX-Systemen sollten Sie nach der Installation einer neuen Version von DB2 UDB und der DB2 Extender Ihre DB2 Extender-Exemplare migrieren.

Um bestehende DB2 Extender-Exemplare zu migrieren, die mit einer früheren Version erstellt wurden, führen Sie folgende Schritte aus:

1. Migrieren Sie das DB2 UDB-Exemplar, das dem DB2 Extender-Exemplar zugeordnet ist.
2. Geben Sie den Befehl DMBIMIGR ein, um das Exemplar zu migrieren. Um beispielsweise das Exemplar OLDINST zu migrieren, geben Sie Folgendes ein:

   ```
   dmbimigr  oldinst
   ```

Tabellen zur Verwaltungsunterstützung bereinigen

Bei der Arbeit mit den DB2 Extendern können sich veraltete Einträge in den Tabellen zur Verwaltungsunterstützung sammeln. Beispielsweise löscht ein Benutzer eine Multimediadatei, aber nicht den entsprechenden Verweis in der Datenbank. Durch das Löschen von veralteten Metadaten kann die Leistung verbessert und Speicherbereich zurückgefordert werden.

Verwendung der API: Der Beispielcode in Abb. 1 bereinigt die Abbildmetadaten für alle Benutzertabellen, deren Eigner ANITAS ist. Er enthält unter anderem Fehlerprüfcode. Das vollständige Beispielprogramm befindet sich in der Datei API.C im Unterverzeichnis SAMPLES.

```
/*---- query database using DBiAdminReorgMetadata API ----*/
step="DBiAdminReorgMetadata API";
rc = DBiAdminReorgMetadata("anitas");
if (rc < 0) {
        printf("%s: %s FAILED!\n", argv[0], step);
   printMsg(rc);
        DBiGetError(&sqlcode, errorMsgText);
        printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
        fail = TRUE;
} else if (rc > 0) {
        printf("%s: %s, warning detected.\n", argv[0], step);
   printMsg(rc);
        DBiGetError(&sqlcode, errorMsgText);
        printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
     } else
        printf("%s: %s PASSED\n\n", argv[0], step);

/*---- end of query using DBiAdminReorgMetadata API ----*/
```

Abbildung 1. Beispielcode, der Tabellen zur Verwaltungsunterstützung bereinigt

Verwendung der db2ext-Befehlszeile:

```
reorg database user anitas for db2image
```

Wenn Sie nicht über Datenbankadministratorberechtigung, jedoch über die Berechtigung CONTROL verfügen, können Sie die APIs DBxReorgMetadata oder den Befehl REORG verwenden, um Metadaten für Ihre eigenen Tabellen zu bereinigen.

Verwaltungsbefehle für den Server

Die Befehle in diesem Kapitel werden von der Befehlszeile des Betriebssystems auf dem Server aus ausgeführt. Führen Sie sie nicht von der DB2-Befehlszeile oder der db2ext-Befehlszeile aus aus. Führen Sie den Befehl DMBSTART immer aus, wenn Sie Ihr Serversystem herunterfahren und erneut starten.

Nur EEE: Sie können außerdem die Serverbefehle DMBSTART und DMBSTOP in einer Mehrpartitionsdatenbankumgebung ausgeben. Wenn Sie einen Serverbefehl in einer Mehrpartitionsdatenbankumgebung ausgeben, gilt der Befehl für alle Knoten, es sei denn, Sie geben eine Knotennummer an. In diesem Fall gilt der Befehl nur für den angegebenen Knoten.

Nur EEE: Der Befehl DMBSTAT kann nicht in einer Mehrpartitionsumgebung ausgeführt werden. Der Serverstatus kann in einer Mehrpartitionsumgebung geprüft werden, indem der Clientbefehl GET SERVER STATUS ALL ausgeführt wird.

DMBICRT

Image	Audio	Video
X	X	X

Erstellt ein DB2 Extender-Exemplar. Sie sollten mehrere Exemplare des DB2 Extender-Servers erstellen, wenn Sie mit mehreren DB2-Exemplaren arbeiten. Unter UNIX erstellen Sie ein Clientexemplar, wenn Sie den DB2 Extender-Client installieren; beim Erstellen des Clientexemplars wird Ihre Umgebung für die Verwendung der DB2 Extender eingerichtet.

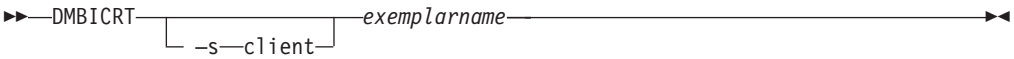
Autorisierung

SYSADM

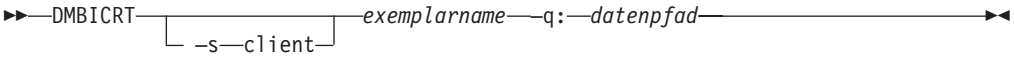
Unter UNIX müssen Sie über Root-Berechtigung verfügen.

Befehlssyntax

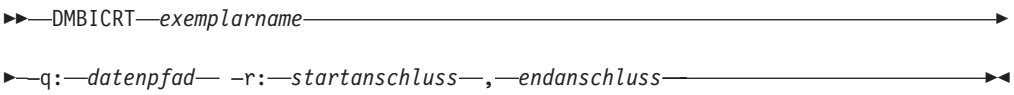
In einer Umgebung für nicht partitionierte Datenbanken:



In einer Umgebung für partitionierte Datenbanken unter UNIX:



In einer Umgebung für partitionierte Datenbanken unter Windows:



Befehlsparameter

exemplarname
Der Name eines bestehenden DB2-Exemplars. Wenn ein DB2-Exemplar mit diesem Namen nicht existiert, werden Sie aufgefordert, es zu erstellen.

–s client
Angabe zum Erstellen eines Exemplars nur für den Client. Bei Verwendung dieses Parameters ist *exemplarname* die Benutzer-ID des Clients. Bei der Erstellung eines Clientexemplars wird die Umgebung für den Client eingerichtet. **(Nur UNIX)**

datenpfad
Der Name eines gemeinsam benutzten Verzeichnisses oder Dateisystems; das Verzeichnis muss auf allen Knoten existieren. Dieses Verzeichnis ist unter UNIX in der Umgebungsvariablen DB2MMDATAPATH und unter Windows in der Registrierungsdatenbank gesetzt. **(Nur EEE)**

startanschluss, endanschluss

Bereich, den die TCP/IP-Anschlüsse verwenden sollen. Der Anschlussbereich muss gleich oder größer als die Anzahl an Knoten sein, mit denen Sie arbeiten. Die Anschlussnummern werden in die Windows-Registrierungsdatenbank geschrieben. **(Nur Windows EEE)**

Beispiele

Erstellen Sie ein Exemplar des DB2 Extender-Servers für das DB2-Exemplar DEVINST in einer Umgebung für nicht partitionierte Datenbanken:

```
dmbicrt  devinst
```

Hinweise

Der Befehl DMBICRT erstellt ein DB2 Extender-Exemplarverzeichnis für die Dateien, die vom Exemplar verwendet werden. Der Verzeichnisname lautet:

- *install_verzeichnis*\INSTANCE*exemplarname*, wobei *install_verzeichnis* das Verzeichnis ist, in dem Sie die DB2 Extender installiert haben **(Windows, OS/2)**
- *INSTHOME*/dmb , wobei *INSTHOME* das Ausgangsverzeichnis des Exemplareigners ist **(UNIX)**

Wenn ein DB2-Exemplar mit dem angegebenen Namen nicht existiert, wenn Sie den Befehl DMBICRT verwenden, werden Sie aufgefordert, es zu erstellen.

Nur EEE:

Obwohl Sie den Befehl DMBICRT unter der Root-Benutzer-ID von jedem teilnehmendem Knoten aus ausführen können, sollten Sie alle DB2 Extender-Serverexemplare unter dem selben Knoten erstellen. Der Knoten sollte derselbe sein, den Sie zur Erstellung des DB2-Exemplars verwendet haben und auf dem sich das DB2-Exemplarverzeichnis befindet. Wenn Sie zum Erstellen des DB2 Extender-Serverexemplars einen anderen Knoten verwenden, ist die Liste der Exemplare auf den jeweiligen Knoten möglicherweise nicht vollständig.

Das gemeinsam benutzte Verzeichnis oder Dateisystem, das als *datenpfad* angegeben ist, wird als Wert der Umgebungsvariablen DB2MMDATAPATH in *$INSTHOME*/dmb/dmbprofile unter UNIX und im Registrierungsschlüssel unter Windows gespeichert:

```
\\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2 Extenders\PROFILES
      \exemplarname\DB2MMDATAPATH
```

Unter UNIX müssen Sie den Anschlussbereich zur Datei /etc/services hinzugefügt haben, bevor Sie das Exemplar erstellen. Fügen Sie zwei Einträge unter Verwendung der folgenden Syntax zur Datei hinzu:

– DMB_*exemplarname startanschluss*

– DMB_*exemplarname*_END *endanschluss*

Der Bereich muss groß genug sein für alle Knoten in einer Umgebung für partitionierte Datenbanken.

Sie müssen das DB2-Exemplar erstellen, bevor Sie ein Exemplar des DB2 Extender-Servers erstellen.

Um ein Text Extender-Exemplar in einer Umgebung für partitionierte Datenbanken zu erstellen, verwenden Sie den Befehl TXTICRT, wie im Buch *DB2 Universal Database Text Extender Verwaltung und Programmierung* beschrieben.

DMBIDROP

Image	Audio	Video
X	X	X

Löscht ein DB2 Extender-Exemplar.

Autorisierung
SYSADM

Unter UNIX müssen Sie über Root-Berechtigung verfügen.

Befehlssyntax

Befehlsparameter

exemplarname
Der Name des DB2 Extender-Exemplars, das gelöscht werden soll.

Beispiele
Löschen des DB2 Extender-Serverexemplars mit dem Namen DEVINST:

```
dmbidrop devinst
```

Hinweise
Führen Sie folgende Schritte aus, bevor Sie diesen Befehl ausführen:

- Stoppen Sie alle Anwendungen, die das Exemplar verwenden, und alle db2ext-Befehlszeilenprozessoren.
- Stoppen Sie die Extender-Services.

Der Befehl DMBIDROP löscht das DB2 Extender-Exemplarverzeichnis, löscht den Exemplareintrag aus der Liste der Exemplare und löscht weitere Informationen zum Exemplar.

Der Befehl DMBIDROP löscht nur das DB2 Extender-Exemplar; er löscht nicht das zugehörige DB2-Exemplar. Sie müssen das DB2-Exemplar explizit löschen.

Wenn Sie das DB2-Exemplar löschen, das einem DB2 Extender-Serverexemplar zugeordnet ist, wird das DB2 Extender-Serverexemplar nicht gelöscht. Sie können es jedoch nicht verwenden.

(Nur EEE) Wenn Sie ein DB2 Extender-Exemplar löschen, müssen Sie Einträge für den Start- und Endanschluss aus der Datei `/etc/services` unter UNIX und aus der Datei `\WINNT\system32\drivers\etc\Services` unter Windows entfernt werden. Diese Einträge sind `DMB_`*`exemplarname`*`mpp` und `DMB_`*`exemplarname`*`mpp_END`.

DMBILIST

Image	Audio	Video
X	X	X

Listet alle Exemplare der DB2 Extender auf.

Autorisierung
Keine

Befehlssyntax

Befehlsparameter
Keine

Beispiele
Auflisten der DB2 Extender-Exemplare:

```
dmbilist
```

DMBIMIGR

Image	Audio	Video
X	X	X

(Nur UNIX) Migriert ein DB2 Extender-Exemplar von einem früheren Release auf das aktuelle Release.

Autorisierung

Sie müssen über Root-Berechtigung verfügen.

Befehlssyntax

Befehlsparameter

exemplarname
Der Name des DB2 Extender-Exemplars, das migriert werden soll.

Beispiele

Migrieren des DB2 Extender-Exemplars mit dem Namen OLDINST:

```
dmbimigr  oldinst
```

Hinweise

Führen Sie folgende Schritte aus, bevor Sie diesen Befehl ausführen:

- Sie müssen das aktuelle Release der DB2 Extender installiert haben.
- Sie müssen das zugehörige DB2-Exemplar migrieren.

Führen Sie den Befehl DMBIMIGR einmal für jedes DB2 Extender-Exemplar aus. Verwenden Sie den Befehl DMBILIST, um die Exemplare aufzulisten.

DMBSTART

Image	Audio	Video
X	X	X

Startet alle Extender-Services für das Extender-Exemplar.

Nur EEE: Wenn ein Knoten angegeben ist, startet der Befehl die Extender-Services nur auf diesem Knoten. Der Befehl DMBSTART leitet außerdem auf jedem Knoten die Funktion zum Erstellen/Löschen von Knoten ein. Die Funktion zum Erstellen/Löschen von Knoten stellt sicher, dass die Knoten, die für DB2 definiert sind, mit den Knoten übereinstimmen, die für die Extender definiert sind. Stimmen sie nicht überein, fügt die Funktion zum Erstellen/Löschen von Knoten je nach Bedarf Knoten hinzu oder löscht Knoten.

Autorisierung
SYSADM

Befehlssyntax

Befehlsparameter

knotennummer
Der Knoten, auf dem Sie die Extender-Services starten wollen. **(Nur EEE)**

Beispiele
Starten der Extender-Services:

```
dmbstart
```

Starten der Extender-Services am Knoten mit der Nummer 2:

```
dmbstart nodenum 2
```

Hinweise
Führen Sie diesen Befehl in folgenden Situationen aus:
- Während Sie als Exemplareigner angemeldet sind (unter AIX, HP-UX oder Solaris).
- Von einem Fenster aus, wobei die Umgebungsvariable DB2INSTANCE mit dem Exemplar übereinstimmt, das Sie starten wollen (unter Windows).
- Immer beim Herunterfahren und anschließenden Neustart des Serversystems.

In einer Einzelpartitionsumgebung wird mit dem Befehl DMBSTART außerdem das DB2-Exemplar gestartet, wenn es noch nicht aktiv ist.

Nur EEE:

In einer Mehrpartitionsumgebung startet der Befehl DMBSTART das DB2-Exemplar nicht. Sie müssen DB2 starten, bevor Sie DMBSTART in einer partitionierten Umgebung ausführen.

Wenn der Befehl DMBSTART fehlschlägt, führen Sie folgende Prüfungen aus:

– Stellen Sie sicher, dass der Wert der Variablen DBD2MMDATAPATH korrekt ist.

– Stellen Sie sicher, dass das gemeinsam benutzte Verzeichnis oder Dateisystem in der Variablen existiert und auf allen Knoten im Zugriff ist.

DMBSTAT

Image	Audio	Video
X	X	X

Zeigt an, welche Datenbanken aktiviert sind und ob die Extender-Services für diese Datenbanken aktiv sind.

Autorisierung
Keine

Befehlssyntax

Befehlsparameter
Keine

Beispiele
Anzeigen des Status der Extender-Services:

```
dmbstat
```

DMBSTOP

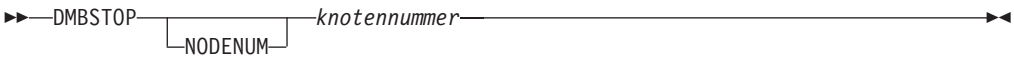
Image	Audio	Video
X	X	X

Stoppt die Extender-Services für das Extender-Exemplar.

Nur EEE: Wenn ein Knoten angegeben ist, stoppt der Befehl DMBSTOP die Extender-Services nur auf diesem Knoten.

Autorisierung
SYSADM

Befehlssyntax



Befehlsparameter

knotennummer
Der Knoten, auf dem Sie die Extender-Services stoppen wollen. **(Nur EEE)**

Beispiele
Stoppen der Extender-Services:

```
dmbstop
```

Stoppen der Extender-Services am Knoten mit der Nummer 2:

```
dmbstop nodenum 2
```

Hinweise
Führen Sie diesen Befehl in folgenden Situationen aus:

- Während Sie als Exemplareigner angemeldet sind (unter AIX, HP-UX oder Solaris).
- Von einem Fenster aus, wobei die Umgebungsvariable DB2INSTANCE mit dem Exemplar übereinstimmt, das Sie stoppen wollen (unter Windows).

Mit dem Befehl DMBSTOP wird das DB2-Exemplar nicht gestoppt.

Nur EEE: Führen Sie den Befehl DMBSTOP nicht für einen bestimmten Knoten aus, es sei denn, Sie arbeiten im Wartungsmodus. Darüber hinaus müssen Sie sicherstellen, dass auf dem Knoten während des Ausschaltens keine Extender-Aktivitäten ausgelöst werden. Andernfalls kann es zu unerwarteten Folgen kommen.

Extender-Daten in einem partitionierten Datenbanksystem neu verteilen (nur EEE)

Mit DB2 Extended Enterprise Edition können Sie Datenbankpartitionsserver (auch Knoten genannt) in einer Umgebung für partitionierte Datenbanken hinzufügen und löschen. Nachdem Knoten hinzugefügt wurden (oder bevor sie gelöscht werden), können bestehende Daten neu verteilt werden, um die Vorteile der neuen Konfiguration zu nutzen.

Zwei Schritte sind erforderlich, um Extender-Daten neu zu verteilen. Zunächst müssen Sie DB2-Daten neu verteilen. Anschließend können Sie DB2 Extender-Daten neu verteilen.

DB2-Daten neu verteilen

Bevor Sie DB2 Extender-Daten neu verteilen, müssen Sie unter Verwendung des DB2-Befehls REDISTRIBUTE NODEGROUP DB2-Daten neu verteilen.

Weitere Informationen zum Neuverteilen von DB2-Daten finden Sie im Handbuch *DB2 Systemverwaltung*.

Extender-Daten neu verteilen

Nachdem Sie DB2-Daten neu verteilt haben, können Sie Extender-Daten neu verteilen. Geben Sie den Extender-Befehl REDISTRIBUTE NODEGROUP ein, um die Neuverteilung von Extender-Daten zu starten.

```
redistribute nodegroup
```

Mit dem Befehl `REDISTRIBUTE NODEGROUP` werden Daten des Audio, Image und Video Extenders sowie QBIC-Merkmaldaten neu verteilt, wobei sie auf denselben Knoten gestellt werden wie die entsprechenden Benutzerdaten.

Wenn der Neuverteilungsprozess einen Fehler zurückgibt, können Sie den Befehl erneut ausführen. Sie können den Befehl mit oder ohne Parameter CONTINUE erneut ausführen, je nachdem, welche Anweisungen in der Antwort auf den Befehl angegeben werden. Mit dieser Option wird das System angewiesen, an der Stelle fortzufahren, an der es gestoppt wurde, und nicht am Anfang erneut zu starten. Der Parameter CONTINUE kann nicht verwendet werden, wenn der Befehl REDISTRIBUTE NODEGROUP das erste Mal nach Ausführung des DB2-Befehls REDISTRIBUTE NODEGROUP verwendet wird.

Um die Datenintegrität zu gewähren, müssen Sie die einzelnen Knotengruppen nacheinander neu verteilen. Warten Sie, bis die Neuverteilung für eine Knotengruppe beendet ist, bevor Sie die nächste starten.

Sie müssen zunächst eine Verbindung zur Datenbank herstellen, bevor Sie diesen Befehl verwenden.

Sie benötigen die Berechtigung SYSADM oder DBADM, um diesen Befehl ausführen zu können.

Szenenwechsel bei Videoobjekten ermitteln

In diesem Kapitel wird beschrieben, wie Szenenwechsel in einem Videoclip mit Hilfe der DB2 Video Extender-APIs ermittelt werden können. Diese APIs sind auf allen DB2 Video Extender-Plattformen verfügbar mit Ausnahme von Windows 3.1. Die Ermittlung von Videoszenenwechseln wird nur für Videoclips im MPEG-1-Format unterstützt.

Was ist ein Szenenwechsel in einem Video?

Stellen Sie sich ein Fernsehstudio vor, das Programme auf Videoband aufzeichnet, um sie später auszustrahlen. Kürzlich hat das Studio begonnen, die Clips der Videobänder unter Verwendung des Video Extenders in einer DB2-Datenbank zu speichern. Dadurch kann das Personal des Studios die herkömmlichen Informationen zu ihren Programmen abfragen und auch die Clips der Programme ansehen.

Das Studio möchte die Möglichkeit haben, ein Videoclip vorab anzuzeigen. Sie möchten eine visuelle Zusammenfassung, ein so genanntes Storyboard, anzeigen. Ein Beispiel für ein Storyboard wird in Abb. 2 gezeigt. Das Anzeigen eines Storyboards kann die Studiomitarbeiter dabei unterstützen, das Wesentliche eines Videos abzurufen, ohne das gesamte Video anzeigen zu müssen. Es kann die Mitarbeiter außerdem bei der Entscheidung unterstützen, ob ein Video für ihre Anforderungen geeignet ist (z. B., ob es sich lohnt, es herunterzuladen und anzuzeigen). Diese Voraussetzung ist für das Studio sehr wichtig. Durch das Anzeigen eines Storyboards an Stelle des gesamten Videos kann der Zeitaufwand für das Herunterladen und Anzeigen erheblich verringert werden. Weitere Informationen zur Verwendung der Funktionen zur Szenenwechselermittlung bei Videos auf diese Weise befinden sich im Abschnitt „Informationen zu allen Aufnahmen in einem Video speichern” auf Seite 34.

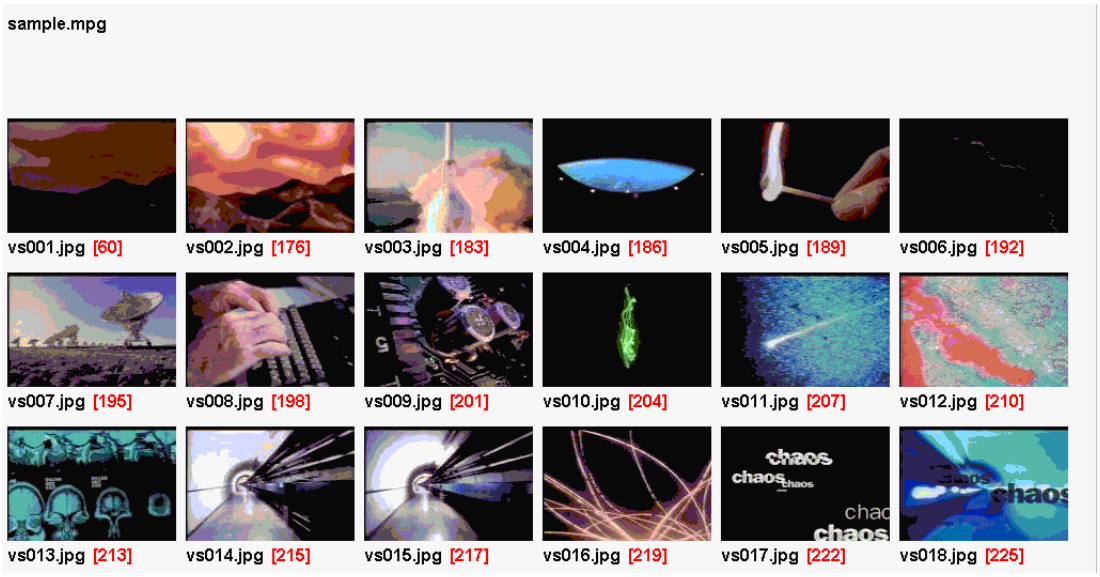


Abbildung 2. Ein Storyboard eines Videos. Repräsentative Vollbilder fassen den Inhalt und den Ablauf eines Videos zusammen.

Das Studio plant, die Ermittlungsfunktionen für Videoszenenwechsel des Video Extenders zu verwenden, um repräsentative Vollbilder für ihre Storyboards zu erfassen.

Ein **Szenenwechsel** ist der Punkt in einem Videoclip, an dem ein deutlicher Unterschied zwischen zwei aufeinander folgenden Vollbildern besteht. Ein Szenenwechsel findet beispielsweise statt, wenn eine Kamera während der Aufzeichnung eines Videos ihren Standpunkt ändert. Die Vollbilder zwischen zwei Szenenwechseln bilden eine **Aufnahme**.

Wenn der Video Extender einen Szenenwechsel[1] feststellt, zeichnet er die Daten für die zugehörige Aufnahme auf. Zu diesen Daten gehört die Nummer des Vollbilds, mit dem die Aufnahme beginnt, die Nummer des Vollbilds, mit dem die Aufnahme endet und die Nummer eines repräsentativen Vollbilds innerhalb der Aufnahme. Die Aufnahmedaten können außerdem den Pixelinhalt des repräsentativen Vollbilds enthalten.

Szenenwechsel suchen und verwenden

Der Video Extender stellt eine Gruppe von Anwendungsprogrammierschnittstellen zur Verfügung, mit denen Sie Aufnahmen oder Vollbilder in einem Videoclip suchen können. Wenn Sie eine Aufnahme oder ein Vollbild gefunden haben, können Sie auf die zugehörigen Daten zugreifen, wie z. B. die Anfangs- und Endvollbildnummern oder den Pixelinhalt eines Vollbilds. Sie können dann diese Informationen zur weiteren Verarbeitung an ein Programm übergeben. Beispielsweise können Sie den Inhalt eines Vollbilds an ein Programm übergeben, das diesen Inhalt anzeigen kann.

Der Video Extender stellt außerdem APIs zur Verfügung, mit denen Aufnahmedaten in einem **Aufnahmekatalog** gespeichert werden können. Der Aufnahmekatalog kann in einer Datenbank oder einer Datei sein. Sie können auf den Aufnahmekatalog in einer Datei oder auf eine schreibgeschützte Sicht des Aufnahmekatalogs in einer Datenbank zugreifen.

Eine Aufnahmekatalogdatei enthält Felder für die folgenden Daten:

- Name des Aufnahmekatalogs
- Werte zur Steuerung, wie der Video Extender eine Aufnahme ermittelt, z. B. die Mindestanzahl von Vollbildern in einer Aufnahme
- Werte zur Steuerung, wie viele und welche Vollbilder als repräsentative Vollbilder für eine Aufnahme gespeichert werden
- Aufnahmenummer
- Nummer des ersten Vollbilds
- Nummer des letzten Vollbilds
- Repräsentative Vollbildnummer
- Name der Datei, die den Inhalt des repräsentativen Vollbilds enthält

Die Sicht des Aufnahmekatalogs in einer Datenbank enthält Spalten für die folgenden Daten:

- Aufnahmekennung
- Videotabellenname
- Videospaltenname
- Videokennung
- Videodateiname
- Nummer des ersten Vollbilds

1. Der Ermittlungscode für den Videoszenenwechsel umfasst den MPEG-Decoder der University of California in Berkeley mit den Änderungen des Boston University Multimedia Communication Laboratory.

- Nummer des letzten Vollbilds
- Repräsentative Vollbildnummer
- Repräsentative Vollbilddaten
- Kommentar

Sie können auf die Daten in einer Aufnahmekatalogdatei zugreifen oder die Daten abfragen, wenn sich der Aufnahmekatalog in einer Datenbank befindet. Die Informationen zu repräsentativen Vollbildern sind besonders beim Anzeigen von Storyboards sinnvoll. Darüber hinaus können Sie, wenn sich der Aufnahmekatalog in einer Datenbank befindet, die Aufnahmedaten mit zugehörigen Daten in anderen Tabellen verknüpfen. Beispielsweise kann das Personal in einem Fernsehstudio einen Aufnahmekatalog in einer Datenbank erstellen. Es kann dann die Katalogdaten mit einer Tabelle verknüpfen, die die Videoclips sowie Informationen zu den Clips enthält. Auf diese Weise kann das Personal eine einzige Abfrage verwenden, um einen Clip und geschäftsinterne Informationen zum Clip abzufragen sowie Aufnahmen innerhalb des Clips zu identifizieren.

Datenstrukturen bei der Aufnahmeermittlung

Daten, die mit der Aufnahmeermittlung zusammenhängen, werden in Strukturen gespeichert, die in der Kopfdatei für die Aufnahmeermittlung, dmbshot.h, enthalten sind. Für viele der APIs für die Aufnahmeermittlung ist erforderlich, dass Sie auf eine oder mehrere dieser Strukturen zeigen. Einige dieser Strukturen werden verwendet, um Daten zu speichern, die der Video Extender als Eingabe verwendet. Beispielsweise enthält die Steuerstruktur für Aufnahmen Informationen, die die Aufnahmeermittlung steuern. Die meisten der Strukturen werden vom Video Extender verwendet, um Daten zu speichern, die von einem Video abgerufen werden. Beispielsweise enthält die Datenstruktur für Videovollbilder den Pixelinhalt eines Vollbilds.

Die Strukturen, die für die Aufnahmeermittlung verwendet werden, sind DBvIOType, DBvShotControl, DBvShotType, DBvFrameData und DBvStoryboardCtrl.

DBvIOType: Die Datenstruktur DBvIOType enthält Daten zu einem Video, wie z. B. Format, Dimensionen und Anzahl der Vollbilder. Die Datenstruktur ist wie folgt definiert:

```
typedef struct {

   FILE *hFile;                    /* file handle for the video */
   char vhandle[255];              /* video handle (if from database)
   char vtable[255];               /* video table name (if from database) */
   char vcolumn[255];              /* video column name (if from database) */
   char vFile[255];                /* name of video file */
   char idxFile[255];              /* name of index file */
   char isIdx;                     /* 1 if the index exists, 0 otherwise */
   char isInDb;                    /* 1 if from DB, 0 if from file */
   int format;                     /* Format of the video */
   unsigned long dx, dy;           /* Dimensions of the video */
   unsigned long totalFrames;      /* TotalFrames in the video */
   unsigned long markFrame;        /* used by shot detection */
   unsigned long currentFrame;     /* The current video frame */
   DBvFrameData fd;                /* Frame data for current frame */
   DBvDCFrameData fdDc;            /* Frame data for DC images */
   unsigned char BGRValid;         /* reserved */
   unsigned short usDeviceID;      /* reserved */
   unsigned long hwnd;             /* reserved */
   int videoReset;                 /* Flag if video is opened or seeked */
```

```
   int firstshot;                /* Used internally to indicate the first call */
   void *reserved                /* reserved */

} DBvIOType;
```

DBvShotControl: Die Datenstruktur DBvShotControl enthält Informationen, die zur Steuerung der Aufnahmeermittlung verwendet werden, wie z. B. die Ermittlungsmethode. Die Datenstruktur ist wie folgt definiert:

```
typedef struct {

  unsigned long reserved;
  unsigned long method;          /* detection method */

    #define DETECT_CORRELATION  0x00000001
    #define DETECT_HISTOGRAM    0x00000002
    #define DETECT_CORRHIST     0x00000003
    #define DETECT_CORRHISTDISS 0x00000004


  int normalCorrValue;           /* Correlation threshold */
  int sceneCutSkipXY;            /* reserved */
  int CorrHistThresh;            /* Histogram threshold */
  int DissThresh;                /* Dissolve threshold */
  int DissCacheSize;             /* Dissolve cache size */
  int DissNumCaches;             /* Dissolve cache number */
  int minShotSize;               /* Minimum frames in a shot */

} DBvShotControl;
```

In Tabelle 1 werden die einzelnen Felder in DBvShotControl und deren zulässige Einstellungen und Standardeinstellungen beschrieben. Um diese Felder mit deren Standardwerten zu initialisieren, verwenden Sie die API DBvInitShotControl, wie im Abschnitt „Werte in Datenstrukturen bei der Aufnahmeermittlung initialisieren” auf Seite 25 beschrieben.

DBvShotControl-Einstellungen hängen vom Videotyp ab: Szenenwechsel in digitalisierten Videos variieren je nach Inhalt und Format des Videos sehr. Außerdem variiert die Genauigkeit des Algorithmus für Szenenwechsel je nach Video. Klar definierte Szenenwechsel mit offensichtlichen Unterschieden in der allgemeinen Vollbilddarstellung werden akkurater ermittelt als weniger klar definierte Arten von Szenenwechseln oder Änderungen, bei denen der allgemeine Farbinhalt der gleiche bleibt. Obwohl die Standardeinstellungen für das DBvShotControl-Feld für die meisten Anwendungen gut funktionieren, müssen Sie diese Einstellungen möglicherweise verkleinern, um die Ermittlung falscher oder fehlender Aufnahmen zu verhindern.

Tabelle 1. DBvShotControl-Felder

Feld	Bedeutung
method	Gibt die Methode an, die der Video Extender zur Ermittlung von Szenenwechseln verwendet. Sie können eine der folgenden Methoden auswählen: DETECT_CORRELATION. Vergleicht die Pixel in zwei aufeinander folgenden Vollbildern. Überschreitet die Differenz die Korrelationsschwelle, wird ein Szenenwechsel ermittelt. DETECT_HISTOGRAM. Vergleicht die Histogrammwerte von zwei aufeinander folgenden Vollbildern. Der Histogrammwert misst die Verteilung von Farben im Vollbild. Überschreitet die Differenz die Histogrammschwelle, wird ein Szenenwechsel ermittelt. DETECT_CORRHIST. Verwendet die Korrelationsmethode, um mögliche Szenenwechsel zu identifizieren und verwendet dann die Histogrammmethode für die Vollbilder, die als mögliche Szenenwechsel markiert sind. Wird die Histogrammschwelle überschritten, wird ein Szenenwechsel ermittelt. DETECT_CORRHISTDISS. Arbeitet wie DETECT_CORRHIST, prüft jedoch zusätzliche Vollbilder auf Überblendung. Die Standardmethode ist DETECT_CORRHIST.
normalCorrValue	Ein ganzzahliger Wert von 0 bis 100, der die Korrelationsschwelle angibt. Hierdurch wird der Mindestwert des Korrelationskoeffizienten zwischen den Pixeln in zwei Vollbildern angegeben. Der Wert 0 bedeutet, dass immer ein Szenenwechsel für das nächste Vollbild ermittelt wird. Der Wert 100 bedeutet, dass ein Szenenwechsel nur ermittelt wird, wenn sich alle Pixel von einem Vollbild zum nächsten Vollbild ändern. Der Standardwert ist 60.
sceneCutSkipXY	Reserviert.
CorrHistThresh	Ein ganzzahliger Wert von 0 bis 100, der die Histogrammschwelle angibt. Hierdurch wird die Differenz zwischen Histogrammwerten von nachfolgenden Vollbildern gemessen. Der Wert 0 bedeutet, dass ein Szenenwechsel nur ermittelt wird, wenn die Histogrammwerte von einem Vollbild zum nächsten gänzlich unterschiedlich sind. Der Wert 100 bedeutet, dass immer ein Szenenwechsel für das nächste Vollbild ermittelt wird. Der Standardwert ist 10.
DissThresh	Ein ganzzahliger Wert von 0 bis 100, der die Schwelle für den Überblendungstest angibt. Hierdurch wird der Prozentsatz an Pixel in einem Vollbild gemessen, die einen Überblendungstest bestehen müssen, bevor eine Überblendung ermittelt wird. Der Wert 0 bedeutet, dass immer eine Überblendung für das Vollbild ermittelt wird. Der Wert 100 bedeutet, dass eine Überblendung nur ermittelt wird, wenn alle Pixel in dem Vollbild den Überblendungstest bestehen. Der Standardwert ist 15.
DissCacheSize	Ein ganzzahliger Wert, der die Anzahl von Vollbildern angibt, die im Abweichungsanteil des Überblendungstests verwendet werden. Der Standardwert ist 4.
DissNumCaches	Ein ganzzahliger Wert, der die Anzahl von Vollbildern angibt, die im Konsistenzanteil des Überblendungstests verwendet wird. Der Standardwert ist 7.
minShotSize	Ein ganzzahliger Wert, der die Mindestanzahl von Vollbildern für eine Aufnahme angibt. Damit eine Aufnahme ermittelt wird, muss sie zumindest so viele Vollbilder haben wie die Mindestanzahl. Der Standardwert ist 5.

DBvShotType: Die Datenstruktur DBvShotType enthält Informationen zu einer Aufnahme, wie z. B. Anfang- und Endvollbildnummer, repräsentative Vollbildnummer sowie einen Zeiger auf den Pixelinhalt des repräsentativen Vollbilds. Die Datenstruktur ist wie folgt definiert:

```
typedef struct {

  unsigned long startFrame;       /* starting frame number */
  unsigned long endFrame;         /* ending frame number */
  unsigned long repFrame;         /* representative frame number */
  DBvFrameData fd;                /* data for representative shot */
  unsigned long dx;               /* frame data width in pixels */
  unsigned long dy;               /* frame data height in pixels */
  char *comment;                  /* shot remark */

} DBvShotType;
```

DBvFrameData: Die Datenstruktur DBvFrameData enthält den Pixelinhalt eines Vollbilds. Die Datenstruktur ist wie folgt definiert:

```
typedef struct                     /* video frame data */
{
  /* MPEG 1 pixels */
  unsigned char *luminance;        /* Luminance pixel plane (black and white) */
  unsigned char *Cr;               /* Cr pixel plane */
  unsigned char *Cb;               /* Cb pixel plane */
  unsigned char *reserved;

  } DBvFrameData;
```

DBvStoryboardCtrl: Die Datenstruktur DBvStoryboardCtrl enthält Werte, mit denen gesteuert wird, wie viele und welche repräsentative Vollbilder für eine Aufnahme in einem Videokatalog gespeichert werden. Eine Beschreibung zur Verwendung dieser Werte befindet sich im Abschnitt „Storyboard erstellen" auf Seite 35. Die Datenstruktur ist wie folgt definiert:

```
typedef struct {

  int thresh1;                    /* threshold for small to medium scenes */
  int thresh2;                    /* threshold for medium to large scenes */
  int delta;                      /* offset used for representative frames */

} DBvStoryboardCtrl;
```

In Tabelle 2 werden die einzelnen Felder in DBvStoryboardCtrl und deren Standardeinstellungen beschrieben. Um diese Felder mit ihren Standardwerten zu initialisieren, verwenden Sie die API DBvInitStoryboardCtrl, wie im Abschnitt „Werte in Datenstrukturen bei der Aufnahmeermittlung initialisieren" auf Seite 25 beschrieben.

Die Einstellungen für DBvStoryboardCtrl hängen vom Typ des Videos ab: Welche und wie viele repräsentative Vollbilder für ein Storyboard optimal sind, hängt möglicherweise von den unterschiedlichen Typen von Videos ab. Obwohl die Standardeinstellungen für das Feld DBvStoryboardCtrl für die meisten Typen von Video gut funktionieren, möchten Sie möglicherweise diese Einstellungen für eine Testuntergruppe von Videos verwenden. Sie können dann die Einstellung entsprechend verbessern, bevor Sie Storyboards für eine größere Gruppe von Videos erstellen.

Tabelle 2. DBvStoryboardCtrl-Felder

Feld	Bedeutung
thresh1	Gibt die Schwelle für kurze Aufnahmen an. Aufnahmen, die weniger Vollbilder enthalten, als der Wert von thresh1 angibt, sind kurze Aufnahmen. Beim Katalogisieren enthalten die Informationen für eine kurze Aufnahme ein repräsentatives Vollbild (das mittlere Vollbild). Der Standardwert ist 90. Wenn der Wert von thresh1 auf -1 gesetzt ist, wird eine Aufnahme (unabhängig von der tatsächlichen Länge) als kurze Aufnahme betrachtet.
thresh2	Gibt die Schwelle für mittellange bis lange Aufnahmen an. Aufnahmen, die maximal so viele Vollbilder enthalten, wie der Wert thresh2 angibt, aber mindestens so viele Vollbilder, wie der Wert thresh1 angibt, werden als mittellange Aufnahmen betrachtet. Beim Katalogisieren enthalten die Informationen für eine mittellange Aufnahme zwei repräsentative Vollbilder. Die Position der repräsentativen Vollbilder wird durch den Wert des Deltafeldes gesteuert. Aufnahmen, die mehr Vollbilder enthalten, als der Wert von thresh2 angibt, sind lange Aufnahmen. Beim Katalogisieren enthalten die Informationen für eine lange Aufnahme drei repräsentative Vollbilder. Die Position des ersten und des letzten repräsentativen Vollbilds wird durch den Wert des Deltafeldes gesteuert. Das zweite repräsentative Vollbild ist das mittlere Vollbild. Der Standardwert ist 150. Wenn der Wert von thresh2 auf -1 gesetzt ist, wird eine Aufnahme (unabhängig von der tatsächlichen Länge) als kurze Aufnahme betrachtet.
delta	Gibt die relative Position an, die für repräsentative Vollbilder verwendet wird. Bei mittellangen und langen Aufnahmen wird das erste repräsentative Vollbild vom Anfang der Aufnahme die Deltazahl an Vollbildern eingerückt. Das letzte repräsentative Vollbild wird vom Ende der Aufnahme die Deltazahl an Vollbildern eingerückt. Der Standardwert ist 5.

Werte in Datenstrukturen bei der Aufnahmeermittlung initialisieren: Die Werte in der Datenstruktur DBvShotControl steuern die Aufnahmeermittlung. Die Werte in der Datenstruktur DBvStoryboardCtrl steuern das Erstellen eines Storyboards. Für die Felder in diesen Datenstrukturen können Sie explizit Werte angeben. Darüber hinaus können Sie die Werte in diesen Strukturen mit ihren Standardwerten initialisieren. Tabelle 1 auf Seite 23 enthält die Standardwerte der Datenstruktur DBvShotControl. Tabelle 2 enthält die Standardwerte der Datenstruktur DBvStoryboardCtrl.

Verwenden Sie die API DBvInitShotControl, um die Werte in der Datenstruktur DBvShotControl zu initialisieren. Wenn Sie die API verwenden, müssen Sie die Steuerstruktur für Aufnahmen angeben. Beispielsweise initialisiert die folgende Anweisung die Felder in der Struktur DBvShotControl mit den Standardwerten:

```
DBvShotControl    shotCtrl;

rc=DBvInitShotControl(
              shotCtrl);          /* pointer to shot control structure */
```

Verwenden Sie die API DBvInitStoryboardCtrl, um die Werte in der Datenstruktur DBvStoryboardCtrl zu initialisieren. Wenn Sie die API verwenden, müssen Sie die Steuerstruktur für das Storyboard angeben. Beispielsweise initialisiert die folgende Anweisung die Felder in der Struktur DBvStoryboardCtrl mit den Standardwerten:

```
DBvStoryboardCtrl    sbCtrl;

rc=DBvInitStoryboardCtrl(
             sbCtrl);          /* pointer to storyboard control structure */
```

Aufnahme oder Vollbild abrufen

Sie können den Video Extender verwenden, um eine Aufnahme oder ein Vollbild eines Videos abzurufen. Bevor Sie eine Aufnahme oder ein Vollbild abrufen können, müssen Sie das Video für die Aufnahmeermittlung öffnen. Der Video Extender verwendet einen Index, um auf Vollbilder und Aufnahmen zuzugreifen. Bevor Sie eine Aufnahme oder ein Vollbild abrufen können, müssen Sie einen Index für das Video erstellen.

Nachdem das Video geöffnet und ein Index erstellt wurde, können Sie die nächste Aufnahme oder das nächste Vollbild in einem Video abrufen oder ein bestimmtes Vollbild nach Vollbildnummer abrufen. Der Video Extender kann Videoclips im MPEG-1-Format verarbeiten. Wenn Sie planen, ein abgerufenes Vollbild mit einem Programm zu verwenden, für das das RGB-Format erforderlich ist, können Sie das Vollbild durch Verwendung einer Video Extender-API in das gewünschte Format umsetzen.

Video für die Aufnahmeermittlung öffnen: Verwenden Sie die API DBvOpenFile, um ein Video zu öffnen, das in einer Datei gespeichert ist. Auf die Datei muss vom Client aus zugegriffen werden können. Verwenden Sie die API DBvOpenHandle, um ein Video zu öffnen, das in einer Datenbank gespeichert ist. Die Anwendung muss zunächst mit der Datenbank verbunden sein. Ist das Video in einer Datenbanktabelle gespeichert, kopiert der Video Extender das Video in eine temporäre Datei. Die temporäre Datei befindet sich in einem Verzeichnis, das in der Umgebungsvariablen DB2VIDEOTEMP angegeben ist. Durch das Öffnen wird ein Video für die Aufnahmeermittlung initialisiert. Der Video Extender setzt einen Zeiger an den Anfang des Videos, das heißt auf das Vollbild 0.

Wenn Sie eine der APIs verwenden, müssen Sie auf einen Bereich zeigen, der für den Zeiger auf die Videodatenstruktur (DBvIOType) verwendet wird. Der Video Extender ordnet diese Struktur als Antwort auf den API-Aufruf zu und verwendet die Struktur, um Informationen zum Video zu speichern. Die Struktur zeigt außerdem auf die Vollbilddatenstruktur (DBvFrameData), die den Pixelinhalt des aktuellen Vollbilds enthält. Eine Beschreibung dieser Strukturen befindet sich im Abschnitt „Datenstrukturen bei der Aufnahmeermittlung" auf Seite 21. Für die API DBvOpenFile müssen Sie außerdem den Namen der Videodatei angeben. Für die API DBvOpenHandle müssen Sie die Videokennung angeben.

Beispielsweise öffnet die folgende Anweisung ein Video, das in einer Datei gespeichert ist, für die Aufnahmeermittlung:

```
DBvIOType    *videoptr;

rc=DBvOpenFile (
       &videoptr,                        /* pointer to video structure pointer */
       "/employee/video/rsmith.mpg");    /* video file */
```

Die folgende Anweisung öffnet ein Video, das in einer Datenbanktabelle gespeichert ist, für die Aufnahmeermittlung:

```
EXEC SQL BEGIN DECLARE SECTION;
char Vid_hdl[251];
  EXEC SQL END DECLARE SECTION;

DBvIOType    *videoptr;

EXEC SQL SELECT VIDEO INTO :Vid_hdl
  FROM EMPLOYEE
WHERE NAME="Anita Jones";

rc=DBvOpenHandle(
       &videoptr,                        /* pointer to video structure pointer */
       vid_hdl);                         /* video handle*/
```

Video indexieren: Der Video Extender verwendet einen Index, um auf Vollbilder und Aufnahmen in einem Video zuzugreifen. Sie müssen einen Index für ein Video erstellen, bevor Sie eine Aufnahme oder ein Vollbild des Videos abrufen können (das MPEG-Format liefert keinen Index für Vollbilder und Aufnahmen). Der Index ordnet Vollbildnummern den Bitströmen zu, die ein MPEG-1-Video bilden.

Sie können durch die Verwendung der API DBvCreateIndexFromVideo oder der API DBvCreateIndex einen Index für ein Video erstellen. Wenn Sie jedoch ein Video unter Verwendung der API DBvOpenFile oder der API DBvOpenHandle für die Aufnahmeermittlung geöffnet haben, brauchen Sie nicht explizit einen Index zu erstellen; der Video Extender wird automatisch einen Index für Sie erstellen. (Informationen zum Öffnen des Videos befinden sich im Abschnitt „Video für die Aufnahmeermittlung öffnen" auf Seite 26.)

Wenn ein Index (entweder explizit oder automatisch) erstellt wurde, versucht der DB2 Video Extender, den Index im selben Pfad zu speichern wie die Videodatei. Er versucht zunächst, die Indexdatei als 'fname.ext.idx' zu speichern, wobei 'fname' der Name der Videodatei und 'ext' die Erweiterung der Videodatei ist. Wenn dieser Versuch fehlschlägt, versucht der Video Extender, die Datei als 'fname.idx' in demselben Verzeichnis zu speichern wie die Videodatei. Wenn auch dieser Versuch fehlschlägt, versucht er, die Indexdatei im lokalen Verzeichnis zu speichern, zunächst als 'fname.ext.idx' und anschließend als 'fname.idx'.

Wenn die Datei geöffnet wird, sucht der Video Extender in der folgenden Reihenfolge nach der Indexdatei:

1. Eine Version der Indexdatei mit Schreibzugriff, vor einer Version mit Lesezugriff.
2. Eine Indexdatei im selben Pfad wie die Videodatei, vor einer Indexdatei im aktuellen Verzeichnis.
3. Ein Index mit dem Namen 'fname.ext.idx', vor einem Index mit dem Namen 'fname.idx', wobei 'fname' der Name der Videodatei und 'ext' die Erweiterung der Videodatei ist.

Beispielsweise sucht der Video Extender, wenn ein Index für die Videodatei 'myvideo.mpg' erstellt ist, zunächst nach einem Index mit Schreibzugriff mit dem Namen 'myvideo.mpg.idx' in dem Pfad, in dem sich die Videodatei befindet.

Wenn Sie die API DBvCreateIndexFromVideo verwenden, geben Sie die Datenstruktur DBvIOType an. Der Video Extender speichert den Namen der Indexdatei in der Struktur. Eine Beschreibung dieser Struktur befindet sich im Abschnitt

„Datenstrukturen bei der Aufnahmeermittlung" auf Seite 21. Beispielsweise erstellt die folgende Anweisung einen Index für ein Video, das zuvor für die Aufnahmeermittlung geöffnet wurde:

```
DBvIOType      *video;

rc=DBvCreateIndexFromVideo(
       video);                 /* pointer to video structure */
```

Wenn Sie die API DBvCreateIndex verwenden, geben Sie den Namen der Videodatei an. Der Video Extender speichert den Index in einer Datei (im gleichen Verzeichnis, in dem sich das Video befindet). Beispielsweise erstellt die folgende Anweisung einen Index für eine Videodatei (die Datei wurde zuvor nicht für die Aufnahmeermittlung geöffnet):

```
rc=DBvCreateIndex(
       "/employee/video/rsmith.mpg");    /* video file */
```

Sie können auch ermitteln, ob ein Index für ein Video existiert. Verwenden Sie die API DBvIsIndex, um nach einem Index zu suchen. Die API setzt die Statusvariable auf 0, wenn kein Index existiert, oder auf 1, wenn ein Index für die Datei existiert. Beispielsweise prüft die folgende Anweisung die Existenz eines Indexes für eine Videodatei:

```
short  *status

rc=DBvIsIndex(
       "/employee/video/rsmith.mpg",     /* video file */
       &status);                         /* status indicator */
```

Videoindex sichern: Sichern Sie die Datei mit dem Videoindex für den Fall, dass Sie sie wiederherstellen müssen. Die Datei befindet sich in dem Verzeichnis, in dem der Video Extender installiert ist.

Vollbild abrufen: Sie können das aktuelle Vollbild in einem Video abrufen. Sie können außerdem das aktuelle Vollbild auf eine bestimmte Vollbildnummer setzen. Verwenden Sie die API DBvGetFrame, um das aktuelle Vollbild in einem Video abzurufen. Verwenden Sie die API DBvSetFrameNumber, um das aktuelle Vollbild auf eine bestimmte Vollbildnummer zu setzen.

Wenn Sie die API DBvGetFrame verwenden, geben Sie die Videostruktur an. Beispielsweise ruft die folgende Anweisung das aktuelle Vollbild in einem Video ab:

```
DBvIOType      *video;

rc=DBvGetFrame(
       video);                 /* pointer to video structure */
```

Wenn Sie die API DBvSetFrameNumber verwenden, geben Sie die Videostruktur und die Nummer des Vollbilds an, das Sie als aktuelles Vollbild festlegen wollen. Beispielsweise setzen die folgenden Anweisungen das aktuelle Vollbild auf die Vollbildnummer 85 und rufen das Vollbild anschließend ab:

```
DBvIOType      *video;

rc=DBvSetFrameNumber(
       video,                  /* pointer to video structure */
       85);                    /* frame number */

rc=DBvGetFrame(
       video);                 /* pointer to video structure */
```

Bei der Ausgabe bringt die API DBvSetFrameNumber das Feld currentFrame in der Struktur DBvIOType in Grundstellung. Die API DBvGetFrame stellt den Pixelinhalt des Vollbilds in die Struktur DBvFrameData. Eine Beschreibung dieser Strukturen befindet sich im Abschnitt „Datenstrukturen bei der Aufnahmeermittlung" auf Seite 21.

Aufnahme abrufen: Verwenden Sie die API DBvDetectShot, um die nächste Aufnahme in einem Video abzurufen. Wenn Sie die API DBvDetectShot verwenden, müssen Sie auf die folgenden Datenstrukturen zeigen:
- Video (DBvIOType)
- Aufnahmesteuerung (DBvShotControl)
- Aufnahmetyp (DBvShotType)

Sie müssen außerdem auf ein Anfangsvollbild für die Suche zeigen. Der Video Extender beginnt an dieser Position des Videos seine Suche nach der nächsten Aufnahme.

Als Ergebnis der API setzt der Video Extender eine Markierung shotDetected und zeigt auf das Anfangsvollbild der nächsten Aufnahme und dessen Vollbilddaten. Ist die Markierung shotDetected auf 1 gesetzt, wurde eine Aufnahme ermittelt. In diesem Fall führt der Video Extender folgende Aktionen aus:
- Er setzt das Feld currentFrame in der Struktur DBvIOType auf das Anfangsvollbild der nächsten Aufnahme.
- Er stellt die Daten für das Anfangsvollbild der nächsten Aufnahme in das Feld fd in der Struktur DBvIOType.
- Er setzt die Struktur DBvShotType so, dass sie die Anfangs- und Endvollbildnummer, die repräsentative Vollbildnummer, die repräsentativen Vollbilddaten und den Kommentar für die nächste Aufnahme enthält.

Ist die Markierung shotDetected auf 0 gesetzt, wurde keine Aufnahme ermittelt. In diesem Fall gibt der Video Extender eine Code zurück, der angibt, dass das Ende des Videos erreicht wurde. Eine Beschreibung dieser Strukturen befindet sich im Abschnitt „Datenstrukturen bei der Aufnahmeermittlung" auf Seite 21.

Beispielsweise fordern die folgenden Anweisungen die nächste Aufnahme in einem Video an:

```
DBvIOType      *video;
long start_frame = 1;
char shotDetected = 0;
DBvShotControl    shotCtrl;
DBvShotType    shot;
shotCtrl→method=DETECT_CORRHIST
shotCtrl→normalCorrValue=60;
shotCtrl→sceneCutSkipXY=1;
shotCtrl→CorrHistThresh=10;
shotCtrl→DissThresh=10;
shotCtrl→DissCacheSize=4;
shotCtrl→DissNumCaches=7;
shotCtrl→minShotSize=0;
rc=DBvDetectShot(
       video,              /* pointer to video structure */
       start_frame,        /* starting frame for search */
       &shotDetected,      /* shot detected flag */
                           /* 1=detected, 0=not detected */
       shotCtrl,           /* pointer to shot control structure */
       &shot);             /* pointer to shot type structure */
```

Format eines abgerufenen Vollbilds umsetzen: Der Inhalt eines MPEG-1-Vollbilds ist im YUV-Format, ein Format, das Informationen zur Luminanzpixelebene, Cr-Pixelebene und Cb-Pixelebene eines Vollbilds enthält. Wenn Sie ein Videovollbild editieren wollen, ist es möglicherweise sinnvoll, das Vollbild vom YUV-Format in das RGB-Format umzusetzen. Der Video Extender stellt die API DBvFrameDatato24BitRGB zur Verfügung, um ein abgerufenes MPEG-1-Vollbild vom YUV-Format in das 24-Bit-RGB-Format umzusetzen. Um die API zu verwenden, müssen Sie zunächst einen Zielpuffer anlegen.

Wenn Sie die API verwenden, müssen Sie auf den Zielpuffer und die Vollbilddaten, die Sie umsetzen wollen, zeigen. Außerdem müssen Sie die Höhe und die Breite des Vollbilds angeben. (Sie können die Daten, die Höhe und die Breite des Vollbilds aus der Struktur DBvIOType für das Vollbild abrufen.) Beispielsweise setzen die folgenden Beispiele ein MPEG-1-Vollbild in ein 24-Bit-RGB-Format um:

```
char RGB[18000];
DBvIOType      *video;
DBvFrameData   fd;

rc=DBvGetNextFrame(
       video);                  /* pointer to video structure */

fd=video.fd
dx=video.dx
dy=video.dy

rc=DBvFrameDataTo24BitRGB (
       RGB,                      /* pointer to target buffer */
       &fd,                      /* pointer to frame data */
       dx,                       /* frame width */
       dy);                      /* frame height */
```

Videodatei schließen: Verwenden Sie die API DBvClose, um eine Videodatei zu schließen, die für die Aufnahmeermittlung geöffnet wurde. Wenn Sie die API verwenden, geben Sie einen Zeiger auf die Videostruktur für die Datei an.

Beispielsweise schließt die folgende Anweisung eine Videodatei, die für die Aufnahmeermittlung geöffnet wurde:

```
DBvIOType      *video;

rc=DBvClose (video);
```

Abgerufenes Vollbild anzeigen: Der Inhalt eines abgerufenen MPEG-1-Vollbild ist im YUV-Format. Dabei handelt es um ein Format, das von den meisten Programmen zur Abbildanzeige nicht angezeigt werden kann. Um ein abgerufenes Videovollbild anzeigen zu können, müssen Sie es in ein Format umsetzen, das von einem Programm zur Abbildanzeige verstanden wird, z. B. in das BMP-Format. Führen Sie beispielsweise folgende Schritte aus, um ein MPEG-1-Vollbild anzuzeigen:

1. Verwenden Sie die API DBvFrameDatato24BitRGB, um das Format eines abgerufenen MPEG-1-Vollbilds vom YUV-Format in das 24-Bit-RGB-Format umzusetzen. Informationen zur Verwendung der API DBvFrameDatato24BitRGB befinden sich im Abschnitt „Format eines abgerufenen Vollbilds umsetzen".
2. Hängen Sie die entsprechende Kopfzeile an das umgesetzte Vollbild an. Beispielsweise ist für das BMP-Format eine Kopfzeile erforderlich, die Informationen z. B. zur Höhe und Breite des Abbilds enthält.
3. Kopieren Sie den Vollbildinhalt (mit Kopfzeile) in eine Datei.

4. Verwenden Sie die API DBiBrowse, um die Datei anzuzeigen. Informationen zur Verwendung der API DBiBrowse befinden sich im Abschnitt „Anzeige- oder Wiedergabe-APIs verwenden" auf Seite 147.

Aufnahmen katalogisieren

Sie können die Informationen zu einer Aufnahme in einem Aufnahmekatalog speichern. Der Video Extender stellt APIs für folgende Funktionen zur Verfügung:

- Erstellen und Verwalten eines Aufnahmekatalogs in einer Datenbank. Sie können die APIs zu folgenden Zwecken verwenden:
  - Einen Aufnahmekatalog in einer Datenbank erstellen
  - Informationen zu einer einzelnen Aufnahme in einem Aufnahmekatalog speichern
  - Informationen zu allen Aufnahmen in einem Video in einem Aufnahmekatalog speichern
  - Informationen, die für eine Aufnahme in einem Aufnahmekatalog gespeichert sind, ändern
  - Aufnahmeinformationen in einem Aufnahmekatalog mischen
  - Aufnahmeinformationen aus einem Aufnahmekatalog löschen
  - Einen Aufnahmekatalog aus einer Datenbank löschen
- Erstellen einer Aufnahmekatalogdatei und Speichern der Informationen zu allen Aufnahmen eines Videos. Eine API wird zur Verfügung gestellt, die die Katalogdatei erstellt und sie mit den Aufnahmedaten füllt. Sie können auf die Daten in der Aufnahmekatalogdatei zugreifen und sie bearbeiten, aber dazu stehen keine APIs zur Verfügung.

Katalogisierte Aufnahmen liefern die Eingabe für Storyboards: Nachdem Sie Aufnahmeinformationen in einem Aufnahmekatalog (in einer Datenbank oder Datei) gespeichert haben, können Sie diese Informationen in einer aufnahmebezogenen Anwendung verwenden. Beispielsweise können Sie eine Anwendung erstellen, die die repräsentativen Vollbilder für alle Aufnahmen in einem Video abruft und sie in einem Storyboard anzeigt.

Sie brauchen nur einen Aufnahmekatalog für eine Datenbank zu erstellen: Sie brauchen nur einen Aufnahmekatalog zu erstellen, wenn Sie wollen, dass der Katalog in einer Datenbank gespeichert wird. Der Video Extender erstellt automatisch eine Aufnahmekatalogdatei, wenn Sie Daten für die Aufnahmen in einem Video speichern und angeben, dass die Ausgabe in einer Datei erfolgen soll.

Vor dem Erstellen eines Katalogs (nur Datenbank): Bevor Sie einen Katalog in einer Datenbank erstellen und verwenden, müssen Sie

- einen SQLConnect-Aufruf eingeben. Ein Aufnahmekatalog in einer DB2-Datenbank besteht aus einer Sammlung von Tabellen. Bevor Sie einen Aufnahmekatalog in einer Datenbank erstellen oder Operationen damit ausführen können, müssen Sie mit einem SQLConnect-Aufruf eine Verbindung zur Datenbank herstellen. (SQLConnect ist ein Aufruf der DB2 Call Level Interface.) Der Aufruf gibt eine Verbindungskennung zurück, die Sie benötigen, um die APIs anzugeben, die den Aufnahmekatalog verwalten.
- die Datenbank für Abbilddaten aktivieren. Sie müssen eine Datenbank für den Datentyp DB2Image aktivieren, bevor Sie einen Aufnahmekatalog in der Datenbank erstellen. Zusätzlich zu den anderen Informationen im Aufnahmekatalog speichert der Video Extender repräsentative Vollbilddaten für jede katalogisierte Aufnahme. Der Datentyp von repräsentativen Vollbilddaten ist DB2Image.

Aufnahmekatalog erstellen (nur Datenbank): Verwenden Sie die API DBvCreateShotCatalog, um einen Aufnahmekatalog in einer Datenbank zu erstellen. (Der Video Extender erstellt automatisch eine Aufnahmekatalogdatei, wenn Sie Daten für die Aufnahmen speichern und angeben, dass die Ausgabe in einer Datei erfolgen soll.). Der Katalog besteht aus Tabellen, die die aufnahmebezogenen Informationen speichern. Sie können eine Sicht der Tabellen durch Verwendung von SQL abfragen. In Tabelle 3 werden die Spalten in der Sicht gezeigt.

Tabelle 3. Spalten in der Sicht eines Aufnahmekatalogs

Spaltenname	Datentyp	Beschreibung
SHOTHANDLE	CHAR(36)	Aufnahmekennung
VIDEOHANDLE	VARCHAR(254)	Videokennung. Die Spalte enthält nur einen Wert, wenn das Video mit der API DBvOpenHandle geöffnet wurde.
VIDEOTABLE	VARCHAR(254)	Die Tabelle, die das Video enthält. Die Spalte enthält nur einen Wert, wenn das Video mit der API DBvOpenHandle geöffnet wurde.
VIDEOCOLUMN	VARCHAR(254)	Die Tabellenspalte, die das Video enthält. Die Spalte enthält nur einen Wert, wenn das Video mit der API DBvOpenHandle geöffnet wurde.
VIDEOFILE	VARCHAR(254)	Videodateiname. Die Spalte enthält nur einen Wert, wenn das Video mit der API DBvOpenFile geöffnet wurde.
STARTFRAME	INTEGER	Nummer des ersten Vollbilds
ENDFRAME	INTEGER	Nummer des letzten Vollbilds
REPFRAME	INTEGER	Repräsentative Vollbildnummer
REPFRAMEDATA	DB2IMAGE	Repräsentative Vollbilddaten
COMMENTS	LONG VARCHAR	Kommentar

Sie sind flexibel, wie viele Aufnahmekataloge Sie in einer Datenbank erstellen wollen und für welche Aufnahmen Sie Informationen in den einzelnen Aufnahmekatalogen speichern wollen. Sie können einen Katalog erstellen, um Aufnahmeinformationen für viele Videos zu speichern, können Aufnahmeinformationen für jedes Video in einem separaten Katalog speichern oder Informationen für mehrere Aufnahmen innerhalb eines Videos in mehreren Katalogen speichern.

Wenn Sie die API verwenden, müssen Sie einen Namen für den Katalog angeben. Namen, die länger als 16 Zeichen sind, werden abgeschnitten. Außerdem müssen Sie die Kennung für die Datenbankverbindung angeben, die durch den SQLConnect-Aufruf zur Datenbank zurückgegeben wird.

Beispielsweise erstellen die folgenden Anweisungen einen Aufnahmekatalog mit dem Namen 'hotshots':

```
SQLHDBC      hdbc;

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);

rc=DBvCreateShotCatalog(
               "hotshots",          /* shot catalog name */
               hdbc);               /* database connection handle */
```

Sichten von Aufnahmekatalogen werden MMDBSYS.SV`katname` genannt, wobei `katname` der Name des Aufnahmekatalogs ist. Beispielsweise hat eine Sicht des Katalogs 'hotshots' den Namen MMDBSYS.SVHOTSHOTS.

Informationen zu einer einzelnen Aufnahme speichern (nur Datenbank): Verwenden Sie die API DBvInsertShot, um Informationen zu einer einzelnen Aufnahme in einem Aufnahmekatalog zu speichern. Sie können Informationen zu einer einzelnen Aufnahme in einem Video nur speichern, wenn sich der Aufnahmekatalog in einer Datenbank befindet. Zu den Informationen, die im Katalog gespeichert werden, gehören:

- Aufnahmekennung
- Videotabellenname (für Videoclips, die in einer Tabelle gespeichert sind)
- Videospaltenname (für Videoclips, die in einer Tabelle gespeichert sind)
- Videokennung (für Videoclips, die in einer Tabelle gespeichert sind)
- Videodateiname (für Videoclips, die in einer Datei gespeichert sind)
- Nummer des ersten Vollbilds
- Nummer des letzten Vollbilds
- Repräsentative Vollbildnummer
- Repräsentative Vollbilddaten

Ein Kommentar für die Aufnahme wird jedoch nicht gespeichert. Eine Beschreibung, wie Sie einen Kommentar zu den für eine Aufnahme gespeicherten Informationen hinzufügen können, befinden sich im Abschnitt „Kommentar für eine Aufnahme angeben (nur Datenbank)" auf Seite 37.

Wenn Sie die API DBvInsertShot verwenden, müssen Sie den Namen des Aufnahmekatalogs und einen Zeiger auf die Aufnahme angeben. Eine Möglichkeit, den Zeiger auf die Aufnahme zu setzen, besteht darin, die nächste Aufnahme abzurufen, wie im Abschnitt „Aufnahme abrufen" auf Seite 29 beschrieben. Außerdem müssen Sie die Kennung für die Datenbankverbindung angeben, die durch den SQLConnect-Aufruf zur Datenbank zurückgegeben wird. Beispielsweise rufen die folgenden Anweisungen die nächste Aufnahme nach dem Vollbild 1 auf und speichern anschließend die Informationen zur Aufnahme im Aufnahmekatalog 'hotshots':

```
SQLHDBC      hdbc;
SQLHENV      henv;
DBvIOType      *video;
long start_frame = 1;
char shotDetected = 0;
DBvShotControl    shotCtrl;
DBvShotType    shot;

shotCtrl→method=DETECT_CORRHIST
shotCtrl→normalCorrValue=60;
shotCtrl→sceneCutSkipXY=1;
```

```
shotCtrl→CorrHistThresh=10;
shotCtrl→DissThresh=10;
shotCtrl→DissCacheSize=4;
shotCtrl→DissNumCaches=7;
shotCtrl→minShotSize=0;

SQLAllocConnect(henv,&hdbc)

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);

rc=DBvDetectShot(
       video,
       start_frame,
       &shotDetected,
       &shotCtrl,
       &shot)

rc=DBvInsertShot (
        "hotshots",             /*shot catalog name*/
       shot,                    /*pointer to shot*/
        hdbc);                  /*database connection handle*/
```

Informationen zu allen Aufnahmen in einem Video speichern: Verwenden Sie die API DBvBuildStoryboardTable oder DBvBuildStoryboardFile, um Informationen zu allen Aufnahmen eines Videos in einem Aufnahmekatalog zu speichern. Die API DBvBuildStoryboardTable speichert die Informationen in einem Aufnahmekatalog, der sich in einer Datenbank befindet. Die API DBvBuildStoryboardFile erstellt eine Aufnahmekatalogdatei und speichert die Informationen in dieser Datei.

Für jede der APIs kann das Quellenvideo in einer Datenbank oder einer Datei sein.

Wenn Sie eine der APIs verwenden, müssen Sie

- den Namen des Aufnahmekatalogs angeben.
- auf die Videostruktur zeigen.
- auf die Datenstruktur DBvShotControl zeigen.
- auf die Datenstruktur DBvStoryboardCtrl zeigen. Die Werte in dieser Datenstruktur steuern, wie viele und welche Videovollbilder als repräsentative Vollbilder im Aufnahmekatalog gespeichert werden. Weitere Informationen zum Festlegen dieser Werte befinden sich im Abschnitt „Storyboard erstellen” auf Seite 35.

Nur bei der API DBvBuildStoryboardTable müssen Sie außerdem die Kennung für die Datenbankverbindung angeben, die durch den SQLConnect-Aufruf zur Datenbank zurückgegeben wird.

Beispielsweise speichern die folgenden Anweisungen Informationen zu allen Aufnahmen eines Videos in einem Aufnahmekatalog. Der Aufnahmekatalog befindet sich in einer Datenbank.

```
SQLHDBC      hdbc;
SQLHENV      henv;
DBvIOType      *video;
DBvShotControl    shotCtrl;
DBvStoryBoardCtrl    sbCtrl;

sbCtrl→thresh1=50
sbCtrl→thresh2=500;
sbCtrl→delta=20;

SQLAllocConnect(henv,&hdbc)

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);
```

```
rc=DBvBuildStoryboardTable (
        "hotshots",              /*shot catalog name*/
       video,                    /*pointer to video structure*/
       shotCtrl,                 /*pointer to shot control structure*/
       sbctrl,                   /*pointer to storyboard control structure*/
        hdbc);                   /*database connection handle*/
```

Die folgenden Anweisungen erstellen eine Aufnahmekatalogdatei und speichern in dieser Datei Informationen zu allen Aufnahmen in einem Video.

```
DBvIOType           *video;
DBvShotControl      shotCtrl;
DBvStoryBoardCtrl   sbCtrl;

sbCtrl→thresh1=50
sbCtrl→thresh2=500;
sbCtrl→delta=20;

rc=DBvBuildStoryboardFile (
       "hotshots",               /*shot catalog file name*/
       video,                    /*pointer to video structure*/
       shotCtrl,                 /*pointer to shot control structure*/
       sbctrl);                  /*pointer to storyboard control structure*/
```

Storyboard erstellen: Wie ihre Namen implizieren, sind die APIs DBvBuildStoryboardTable und DBvBuildStoryboardFile besonders geeignet, Informationen zu speichern, die in einem Storyboard verwendet werden sollen. Ein **Storyboard** ist eine visuelle Zusammenfassung eines Videos. Sie können ein Storyboard erstellen, indem Sie die repräsentativen Vollbilder anzeigen, die für ein Video in einem Aufnahmekatalog gespeichert wurden.

Die APIs DBvBuildStoryboardTable und DBvBuildStoryboardFile speichern ein oder mehrere repräsentative Vollbilder für eine Aufnahme. Die Werte, die Sie in der Struktur DBvStoryboardCtrl angeben, steuern die Anzahl der repräsentativen Vollbilder, die für eine Aufnahme speichert werden, und steuern, welche Vollbilder verwendet werden. Die Definition der Struktur DBvStoryboardCtrl befindet sich im Abschnitt „Datenstrukturen bei der Aufnahmeermittlung" auf Seite 21. Abb. 3 zeigt, wie die Werte in den DBvStoryboardCtrl-Feldern verwendet werden.

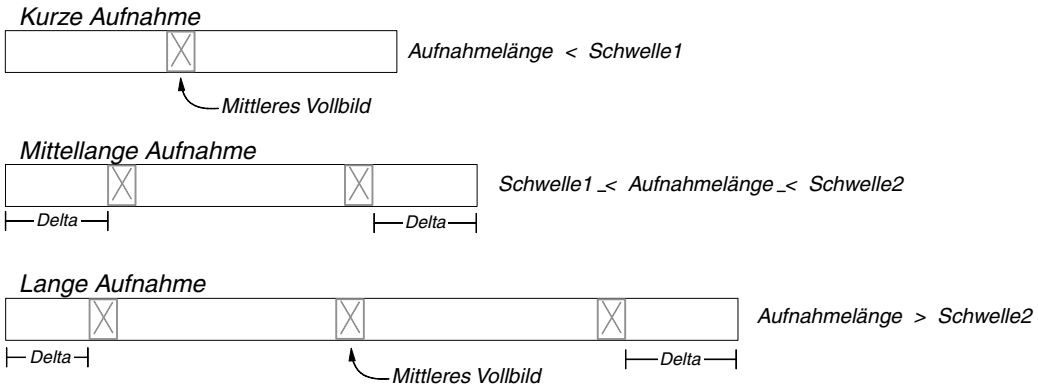


Abbildung 3. Verwendungsweise der Werte in der Struktur DBvStoryboardCtrl

Die Abb. 3 zeigt Folgendes:

- Nur ein repräsentatives Vollbild wird für eine kurze Aufnahme gespeichert. Die Anzahl an Vollbildern in einer kurzen Aufnahme ist kleiner als der Wert Schwelle1 in der Datenstruktur DBvStoryboardCtrl. Das repräsentative Vollbild ist das mittlere Vollbild der Aufnahme.

- Zwei repräsentative Vollbilder werden für eine mittellange Aufnahme gespeichert. Die Anzahl an Vollbildern in einer mittellangen Aufnahme ist größergleich dem Wert Schwelle1 und kleiner-gleich dem Wert Schwelle2 in der Datenstruktur DBvStoryboardCtrl. Durch den Deltawert in der Datenstruktur DBvStoryboardCtrl ist die Anzahl an Vollbildern vom Start des Videos bis zum ersten repräsentativen Vollbild festgelegt. Der Deltawert legt außerdem die Anzahl an Vollbildern vom zweiten repräsentativen Vollbild zum Ende des Videos fest.
- Drei repräsentative Vollbilder werden für eine lange Aufnahme gespeichert. Die Anzahl an Vollbildern in einer langen Aufnahme ist größer als der Wert Schwelle2 in der Datenstruktur DBvStoryboardCtrl. Durch den Deltawert in der Datenstruktur DBvStoryboardCtrl ist die Anzahl an Vollbildern vom Start des Videos bis zum ersten repräsentativen Vollbild festgelegt. Das zweite repräsentative Vollbild ist das mittlere Vollbild des Videos. Der Abstand zwischen dem dritten repräsentativen Vollbild und dem Ende des Videos ist durch den Deltawert festgelegt.

Jede Aufnahme kann als kurze Aufnahme verarbeitet werden, wenn der Wert Schwelle1 oder Schwelle2 auf -1 gesetzt wird. In diesem Fall wird nur ein repräsentatives Vollbild, das mittlere Vollbild, für die Aufnahme im Aufnahmekatalog gespeichert.

Neben den Werten in der Datenstruktur DBvStoryboardCtrl hat eine Reihe von Feldern in der Datenstruktur DBvShotControl Auswirkungen darauf, welche repräsentativen Vollbilder für eine nachfolgende Anzeige in einem Storyboard gespeichert werden. Beispielsweise geben die Felder CorrHistThresh, normalcorrValue und minShotSize in der Datenstruktur DBvShotControl Schwellen für die Aufnahmeermittlung an und beeinflussen somit, welche Vollbilder in einem Storyboard eines Videos angezeigt werden. Wenn Sie die API DBvBuildStoryBoardTable und die API DBvBuildStoryBoardFile verwenden, um Aufnahmeinformationen zur Verwendung in einem Storyboard zu speichern, müssen Sie zunächst einen Testlauf mit den Anfangswerten für die Datenstrukturen DBvStoryBoardCtrl und DBvShotControl ausführen. Sie können dann Ihre Ergebnisse optimieren, indem Sie die Werte in den verschiedenen Feldern dieser Datenstrukturen ändern.

Storyboard anzeigen: Sie können ein Programm erstellen, mit dem ein Storyboard angezeigt wird. Sie greifen dazu auf die repräsentativen Vollbilder zu, die in einem Aufnahmekatalog für ein Video gespeichert sind. Wenn die API DBvBuildStoryboardFile zum Speichern der Aufnahmen für das Video verwendet wurde, zeigt die Aufnahmekatalogdatei auf GIF-Dateien für die repräsentativen Vollbilder. Sie können diese GIF-Dateien mit Hilfe eines geeigneten Browsers oder Anzeigeprogramms anzeigen.

Wenn die API DBvBuildStorybBoardTable zum Speichern der Aufnahmen für das Video verwendet wurde, enthält der Aufnahmekatalog (der in einer Datenbank gespeichert ist) die Daten für die repräsentativen Vollbilder. Sie können auf die Daten von repräsentativen Vollbildern in der Aufnahmekatalogsicht zugreifen (eine Beschreibung der Sicht befindet sich in Tabelle 3 auf Seite 32). Die Daten von repräsentativen Vollbildern sind im YUV-Format. Dabei handelt es um ein Format, das von den meisten Programmen zur Abbildanzeige nicht angezeigt werden kann. Um die repräsentativen Vollbilder anzuzeigen, können Sie die Vollbilddaten mit Hilfe der API DBvFrameDatato24BitRGB (wie in „Abgerufenes Vollbild anzeigen" auf Seite 30 beschrieben) umsetzen. Sie können anschließend die repräsentativen Vollbilder mit Hilfe eines geeigneten Browsers oder Anzeigeprogramms anzeigen.

Storyboard-Beispielprogramme: Das Unterverzeichnis SAMPLES enthält zwei Beispielprogramme, die das Erstellen und Anzeigen eines Storyboards für ein Video demonstrieren. Ein Beispielprogramm, in der Datei `makesf.exe`, verwendet die API DBvBuildStoryBoardFile, um eine Aufnahmekatalogdatei zu erstellen und Aufnahmedaten in der Datei zu speichern. Das andere Beispielprogramm, `makehtml.exe`, greift auf die Aufnahmekatalogdatei zu und erstellt HTML-Seiten für die Anzeige durch einen Webbrowser.

Kommentar für eine Aufnahme angeben (nur Datenbank): Sie können einen Kommentar angeben, der mit den anderen Informationen für eine Aufnahme in einem Aufnahmekatalog gespeichert werden soll. Verwenden Sie die API DBvSetShotComment, um den Kommentar anzugeben.

Wenn Sie die API verwenden, müssen Sie den Namen des Aufnahmekatalog angeben, in dem der Kommentar gespeichert werden soll, die Kennung der Aufnahme, zu der der Kommentar hinzugefügt werden soll, sowie den Kommentar. Außerdem müssen Sie die Kennung für die Datenbankverbindung angeben, die durch den SQLConnect-Aufruf zur Datenbank zurückgegeben wird. Beispielsweise fügen die folgenden Anweisungen einen Kommentar für eine Aufnahme (die mit der Vollbildnummer 85 beginnt) zum Aufnahmekatalog 'hotshots' hinzu:

```
SQLHDBC      hdbc;
SQLHENV      henv;
char shothandle[37];

SQLAllocConnect(henv,&hdbc)

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);

EXEC SQL SELECT SHOTHANDLE INTO :shothandle
  FROM MMDBSYS.SVHOTSHOTS
  WHERE STARTFRAME=85;

rc=DBvSetShotComment (
        "hotshots",                         /*shot catalog name*/
        shothandle,                    /*shot handle*/
        "shot of beach at sunset",     /*comment*/
        hdbc);                         /*database connection handle*/
```

Informationen, die für eine Aufnahme gespeichert sind, ändern (nur Datenbank): Sie können die Information, die für eine Aufnahme in einem Aufnahmekatalog gespeichert sind, ändern. Verwenden Sie dazu die API DBvUpdateShot. Stellen Sie die Ersetzungsinformationen in eine Struktur DBvShotType. Sie müssen außerdem Informationen für die übrigen Felder angeben, auch wenn diese unverändert bleiben. Wenn Sie die API DBvUpdateShot verwenden, geben Sie den Namen des Katalogs und einen Zeiger auf die Struktur DBvShotType an. Außerdem müssen Sie die Kennung für die Datenbankverbindung angeben, die durch den SQLConnect-Aufruf zur Datenbank zurückgegeben wird.

Wenn Sie die Informationen für die Aufnahme ändern, haben Sie die Möglichkeit, den Kommentar, der (falls vorhanden) mit den Informationen gespeichert ist, zu ändern. Wenn Sie den Kommentar ändern wollen, geben Sie ihn in der Struktur DBvShotType an. Wenn Sie den alten Kommentar beibehalten wollen, geben Sie einen Nullwert in der Struktur DBvShotType an.

Beispielsweise ändern die folgenden Anweisungen die Informationen, die für eine Aufnahme im Katalog 'hotshots' gespeichert sind; die Aufnahme beginnt bei Vollbildnummer 85:

```
SQLHDBC         hdbc;
SQLHENV         henv;
char shothandle [37];
DBvShotType     shot;
DBvFrameData    fd110;

/* get shot handle */

EXEC SQL SELECT SHOTHANDLE INTO :shothandle
  FROM MMDBSYS.SVHOTSHOTS
  WHERE STARTFRAME=85;

/* change shot attribute */

shot.startFrame=110;
shot.endFrame=200;
shot.repframe=110;
shot.fd=fd110;
shot.comment=NULL;

/* update shot information */

SQLAllocConnect(henv,&hdbc)

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);

rc=DBvUpdateShot (
        "hotshots",             /*shot catalog name*/
        shot,                   /*shot information*/
        hdbc);                  /*database connection handle*/
```

Mischen von Aufnahmeinformationen in einem Aufnahmekatalog (nur Datenbank): Sie können die Informationen, die für zwei Aufnahmen in einem Aufnahmekatalog gespeichert sind, mischen. Wenn Sie Aufnahmeinformationen mischen, geben Sie eine Reihenfolge an, indem Sie eine erste und eine zweite Aufnahme kennzeichnen. Die Anfangsvollbildnummer der ersten Aufnahme wird als Anfangsvollbildnummer der gemischten Aufnahme gespeichert. Die Nummer des größten Vollbilds zwischen der ersten und zweiten Aufnahme wird als Endvollbildnummer der gemischten Aufnahme gespeichert. Beim Mischen werden die Informationen, die für die erste Aufnahme gespeichert sind, durch die Informationen für die gemischte Aufnahme ersetzt. Die Informationen, die für die zweite Aufnahme gespeichert sind, werden aus dem Aufnahmekatalog gelöscht.

Verwenden Sie die API DBvMergeShots, um die Informationen für zwei Aufnahmen in einem Aufnahmekatalog zu mischen. Wenn Sie die API verwenden, geben Sie den Namen des Aufnahmekatalogs, gefolgt von den Kennungen der ersten und zweiten Aufnahme an, die gemischt werden sollen. Außerdem müssen Sie die Kennung für die Datenbankverbindung angeben, die durch den SQLConnect-Aufruf zur Datenbank zurückgegeben wird. Beispielsweise mischen die folgenden Anweisungen die Informationen, die für zwei Aufnahmen im Katalog 'hotshots' gespeichert sind; die erste Aufnahme beginnt bei Vollbildnummer 85, und die zweite Aufnahme beginnt bei Vollbildnummer 210:

```
SQLHDBC      hdbc;
SQLHENV      henv;
char shothandle1[37];
char shothandle2[37];
```

```
EXEC SQL SELECT SHOTHANDLE INTO :shothandle1
  FROM MMDBSYS.SVHOTSHOTS1
  WHERE STARTFRAME=85;

EXEC SQL SELECT SHOTHANDLE INTO :shothandle2
  FROM MMDBSYS.SVHOTSHOTS2
  WHERE STARTFRAME=210;

SQLAllocConnect(henv,&hdbc)

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);

rc=DBvMergeShots (
        "hotshots",             /*shot catalog name*/
        shothandle1,            /*shot handle for first shot*/
        shothandle2,            /*shot handle for second shot*/
        hdbc);                  /*database connection handle*/
```

Aufnahmeinformationen aus einem Aufnahmekatalog löschen (nur Datenbank): Verwenden Sie die API DBvDeleteShot, um Informationen zu einer Aufnahme aus dem Aufnahmekatalog zu löschen. Wenn Sie die API verwenden, geben Sie den Namen des Aufnahmekatalogs, gefolgt von der Aufnahmekennung an. Außerdem müssen Sie die Kennung für die Datenbankverbindung angeben, die durch den SQLConnect-Aufruf zur Datenbank zurückgegeben wird. Beispielsweise löschen die folgenden Anweisungen Informationen zu einer Aufnahme (die mit der Vollbildnummer 85 beginnt) aus dem Aufnahmekatalog 'hotshots':

```
SQLHDBC          hdbc;
SQLHENV          henv;
char shothandle  [37];

EXEC SQL SELECT shothandle INTO :shothandle
  FROM mmdbsys.svhotshots
  WHERE startframe=85;

rc=DBvDeleteShot (
        "hotshots",             /*shot catalog name*/
        shothandle,             /*shot handle*/
        hdbc);                  /*database connection handle*/
```

Aufnahmekatalog löschen (nur Datenbank): Verwenden Sie die API DBvDeleteShotCatalog, um einen Aufnahmekatalog zu löschen. Wenn Sie die API verwenden, geben Sie den Namen des zu löschenden Aufnahmekatalogs und die Kennung für die Datenbankverbindung an, die durch den SQLConnect-Aufruf zur Datenbank zurückgegeben wird. Beispielsweise löscht die folgende Anweisung den Aufnahmekatalog 'hotshots':

```
SQLHDBC      hdbc;
SQLHENV      henv;

rc=DBvDeleteShotCatalog (
        "hotshots",             /*shot catalog name*/
        hdbc);                  /*database connection handle*/
```

Szenenwechsel verwenden

Kapitel 2. Überblick

DB2 (DB2) Universal Database (UDB) ist ein leistungsstarker, objektrelationaler Datenbankmanager. Er speichert und schützt herkömmliche numerische Daten und Zeichendaten sowie große, komplexe Objekte (LOBs). Die DB2 Extender helfen Ihnen, die Funktionen für die objektrelationale Verarbeitung von DB2 zu nutzen. Die Extender definieren eindeutige Datentypen und Sonderfunktionen für Abbild-, Audio-, Video- und Textobjekte. Dadurch sparen die Extender für Sie die Zeit und den Aufwand, die/der für die Definition dieser Datentypen und Funktionen in Ihren Anwendungen erforderlich wäre. Die Datentypen und Funktionen sind über SQL verfügbar. Aufgrund dessen stellen die Extender Ihren Anwendungen einen einzelnen Zugriffspunkt auf einen oder alle diese Datentypen, zusammen mit den numerischen Daten und Zeichendaten, zur Verfügung. Darüber hinaus stellen die Extender Ihren Anwendungen neue Wege für die Informationssuche zur Verfügung. Beispielsweise können Ihre Anwendungen nach Abbildern anhand ihrer visuellen Merkmale suchen, wobei sie visuelle Beispiele für Farbe oder Textur verwenden.

DB2 nutzen

Die Extender nutzen die objektorientierten Funktionen von DB2 aus. Insbesondere können Sie mit DB2 folgende Funktionen ausführen:

- LOBs (große Objekte) von bis zu 2 Gigabyte in einer DB2-Datenbank speichern.
- Eindeutige Datentypen für diese großen, komplexen Objekte definieren. Sie können diese benutzerdefinierten Typen (UDTs) verwenden, um den Datentyp zu identifizieren, der durch ein Objekt dargestellt wird, z. B. ein Abbild oder ein Ton.
- Spezifische Funktionen definieren, die für einen benutzerdefinierten Datentyp angefordert werden können. Beispielsweise können Sie eine Funktion definieren, die die Anzahl von Farben in einem Abbild zählt oder die Abtastrate eines Tons abruft. Sie können die benutzerdefinierten Funktionen (UDFs) in einer SQL-Anweisung auf die gleiche Weise anfordern wie andere SQL-Funktionen.

Die DB2 Extender erstellen UDFs für Abbild-, Audio-, Video- und Textobjekte. Die UDTs und UDFs können eine wertvolle Hilfe bei der Ausführung der folgenden Operationen darstellen:

- Entwicklung von Anwendungen. Da die Extender die Datentypen und Funktionen definieren, brauchen Sie sie nicht in Ihren Anwendungen zu definieren.
- Gewährleistung der Konsistenz. Die gleiche Gruppe von Extender-UDTs und -UDFs steht in allen auf Ihrem System implementierten Anwendungen zur Verfügung. Dadurch wird eine fertige Konsistenzstufe geboten, die andernfalls zwischen Anwendungen, die große Objekte verarbeiten, möglicherweise schwer zu erzielen ist.
- Erstellung leistungsstarker Abfragen. Da die UDFs auf die gleiche Weise angefordert werden wie andere SQL-Funktionen, können Ihre Anwendungen Abfragen für mehrere Datentypen enthalten. Eine SQL-Anweisung kann auf Abbild-, Audio-, Video- und Textobjekte sowie auf herkömmliche numerische Daten und Zeichendaten zugreifen. Sie können UDFs und UDTs in eingebetteten SQL-Anweisungen und auch in Aufrufen der DB2 Call Level Interface (DB2 CLI) angeben.

Da die Objekte, die die Extender verarbeiten, in einer DB2-Datenbank gespeichert werden können, gilt für diese Objekte die gleiche Sicherheit, Integrität und der gleiche Wiederherstellungsschutz wie für herkömmliche Datentypen, die in der Datenbank gespeichert werden.

Darüber hinaus nutzen die DB2 Extender die Umgebung für partitionierte Datenbanken von DB2 Universal Database Enterprise Extended Edition aus. Durch Partitionierung können Anwendungen eine Datenbank verwenden, die zu groß für einen einzelnen Computer ist. Durch Partitionierung können außerdem SQL-Operationen parallel ausgeführt werden, wodurch SQL-Abfragen oder -Dienstprogramme beschleunigt werden.

Neue leistungsstarke Wege für die Informationssuche

Die DB2 Extender geben Ihren Anwendungen viel Flexibilität bei der Suche nach Informationen. Die Anwendungen können nach Objekten suchen, die den herkömmlichen Datentypen, die in einer Datenbank gespeichert sind, zugeordnet sind. Beispielsweise können Sie nach einem Audioclip anhand dessen Beschreibung oder anhand des Datums, an dem er aufgezeichnet wurde, suchen. Die Anwendungen können außerdem nach Objekten anhand ihrer eindeutigen Merkmale, wie z. B. die Spieldauer eines Videoclips, suchen. Die Extender bestimmen diese Merkmale automatisch und speichern sie für die Verwendung bei Suchvorgängen.

Die Anwendungen können sogar nach Abbildern anhand des Inhalts suchen. Stellen Sie sich eine Anwendung vor, die visuelle Beispiele für die Suche nach Abbildern verwendet. Mit einer solchen Anwendungen könnten Benutzer ein Beispielabbild auswählen und die Anwendung andere Abbilder suchen lassen, die ähnliche Farben oder Texturen haben wie das Beispiel. Mit der DB2 Extender-Funktion QBIC (Query by Image Content = Abfrage anhand des Inhalts) können Sie Anwendungen erstellen, die auf diese visuelle Art und Weise nach Abbildern suchen.

Die DB2 Extender

Die DB2 Extender bestehen aus dem Image Extender, dem Audio Extender, dem Video Extender und dem Text Extender.

In diesem Handbuch werden der Image Extender, der Audio Extender und der Video Extender erläutert. Alle nachfolgenden Verweise auf „Extender" oder „DB2 Extender" in diesem Handbuch beziehen sich, sofern nichts anderes angegeben wurde, auf den Image, Audio und Video Extender. Informationen zum Text Extender befinden sich im Handbuch *Text Extender Verwaltung und Programmierung*. Informationen zum XML Extender befinden sich im Handbuch *XML Extender Verwaltung und Programmierung*.

Das SDK und die Laufzeitumgebungen

Das Installationspaket der DB2 Extender umfasst ein Software Developers Kit (SDK) sowie Client- und Serverlaufzeitumgebungen. Sie können DB2 Extender-Anwendungen auf einer Client- oder Servermaschine, auf der Sie das DB2 Extender SDK installiert haben, entwickeln.

Sie können DB2 Extender-Anwendungen auf einer Servermaschine ausführen, die den DB2 Extender-Clientlaufzeitcode und -Serverlaufzeitcode enthält. (Der Clientlaufzeitcode wird automatisch installiert, wenn Sie den Serverlaufzeitcode installieren.) Sie können außerdem DB2 Extender-Anwendungen auf einer Clientmaschine ausführen, auf der der DB2 Extender-Clientlaufzeitcode installiert ist. Wenn Sie eine Extender-Anwendung von einer Clientmaschine aus ausführen, müssen Sie sicherstellen, dass eine Verbindung zum Server hergestellt werden kann.

Verwendung der Extender

Sie können die benutzerdefinierten Funktionen (UDFs) der Extender in einem DB2-Anwendungsprogramm anfordern, oder Sie können sie interaktiv, unter Verwendung des DB2-Befehlszeilenprozessors, anfordern.

Die Extender stellen außerdem folgende APIs (Anwendungsprogrammierschnittstellen) zur Verfügung:

- Verwaltungs-APIs, um eine Datenbank für Abbild-, Audio- und Videodaten vorzubereiten und zu verwalten.
- Anzeige- und Wiedergabe-APIs, um Abbilder anzuzeigen und Video- und Audioclips wiederzugeben.
- QBIC-APIs, um Abbilder für die Suche anhand des Inhalts vorzubereiten und um die Suche anhand des Inhalts anzufordern. (Eine Inhaltssuche kann auch durch UDFs angefordert werden.)
- APIs zur Ermittlung von Videoaufnahmen, um Vollbildfolgen auf der Basis von Szenenwechseln in einem Video zu identifizieren.

Die DB2 Extender stellen außerdem einen Befehlszeilenprozessor zur Verfügung, mit dem Sie Verwaltungsbefehle absetzen können. Um zwischen dem von den Extendern bereitgestellten Befehlszeilenprozessor und dem durch DB2 zur Verfügung gestellten Befehlszeilenprozessor zu unterscheiden, wird die DB2 Extender-Komponente im Folgenden als „db2ext-Befehlszeilenprozessor" und die DB2-Komponente als „DB2-Befehlszeilenprozessor" bezeichnet.

Beispiele

Eine Werbeagentur verwaltet eine DB2-Datenbank mit Ihren Werbekampagnen. In der Vergangenheit speicherte die Agentur numerische Daten und Zeichendaten zu jeder Werbekampagne, wie z. B. Name des Kunden und das Datum der Fertigstellung. Mit der Installation von DB2 UDB und den DB2 Extendern speichert die Agentur jetzt auch den Inhalt der Objekte für die Werbung in der Datenbank.Dazu gehören Abbilder von Werbeanzeigen, Videos von Fernsehwerbespots und Aufzeichnungen von Radiowerbespots. Wie Abb. 4 auf Seite 44 zeigt, befinden sich alle zugehörigen Informationen zur Werbung in einer Datenbanktabelle mit dem Namen ADS.

Abbildung 4. Eine Multimediadatenbanktabelle. Die Tabelle enthält Abbild-, Audio- und Videodaten sowie herkömmliche Datentypen. Ein Video, Audiodaten und ein Abbild werden angezeigt. Diese Abbildung wurde dem amerikanischen Buch als Beispiel entnommen und daher nicht übersetzt.

Beispiel 1: Ein Video anhand seiner Merkmale abrufen

Ein Kundenbetreuer in der Werbeagentur möchte die Videowerbespots anzeigen, die für den Kunden IBM im Jahre 1997 erstellt wurden, wobei er nur die Spots sehen möchte, die höchstens 30 Sekunden lang sind.

Abb. 5 zeigt eine Abfrage, die auf die Videos zugreift. Beachten Sie die Video Extender-UDFs `Filename` und `Duration` in der Abfrage.

```
SELECT FILENAME(ADS_VIDEO)
       FROM ADS
WHERE CLIENT='IBM' AND
SHIP_DATE>='01/01/1997' AND
DURATION(ADS_VIDEO) <=30
```

Abbildung 5. Eine Abfrage für den Zugriff auf Videos

Die Abfrage gibt die Dateinamen der gewünschten Videos zurück. Der Kundenbetreuer kann danach seine bevorzugte Videowiedergabeeinheit starten und den Inhalt der einzelnen Videodateien wiedergeben.

Abb. 5 auf Seite 44 ist ein Beispiel für eine Abfrage, die der Kundenbetreuer interaktiv eingeben kann. Normalerweise würde der Kundenbetreuer ein Anwendungsprogramm verwenden, um die Videos zu suchen und wiederzugeben. Beispielsweise zeigt Abb. 6 einige Schlüsselelemente einer solchen Anwendung, die in C codiert ist. Die Anwendung ruft die Videodateinamen in eine DB2-Hostvariable mit dem Namen `hvVid_fname` ab. Beachten Sie außerdem, dass die Anwendung die Wiedergabe-API, `DBvPlay`, verwendet, um die Videos wiederzugeben.

```
#include <dmbvideo.h>


int count = 0;

EXEC SQL BEGIN DECLARE SECTION;
char hvClient[30];                      /*client name*/
char hvCampaign[30];                    /*campaign name*/
char hvSdate[8];                        /*ship date*/
char hvVid_fname [251]                  /*video file name*/
EXEC SQL END DECLARE SECTION;


EXEC SQL DECLARE c1 CURSOR FOR
     SELECT CLIENT, CAMPAIGN, SHIP_DATE, FILENAME(ADS_VIDEO)
       FROM ADS
       WHERE CLIENT='IBM' AND
         SHIP_DATE≥'01/01/1997' AND
         DURATION(ADS_VIDEO)≤30
FOR FETCH ONLY;
```

Abbildung 6. Eine Anwendung für den Zugriff auf und die Wiedergabe von Videos (Teil 1 von 2)

```
EXEC SQL OPEN c1;
for (;;) {
        EXEC SQL FETCH c1 INTO :hvClient, :hvCampaign,
                         :hvSdate, :hvVid_fname;

         if (SQLCODE != 0)
                   break;

    printf("\nRecord %d:\n", ++count);
    printf("Client = '%s'\n", hvClient);
    printf("Campaign = '%s'\n", hvCampaign);
    printf("Sdate = '%s'\n", hvSdate);

    rc=DBvPlay(NULL,MMDB_PLAY_FILE,hvVid_fname,MMDB_PLAY_WAIT);
}
EXEC SQL CLOSE c1;
```

Abbildung 6. Eine Anwendung für den Zugriff auf und die Wiedergabe von Videos (Teil 2 von 2)

Beispiel 2: Abbilder anhand des Inhalts suchen

Ein Grafiker in der Werbeagentur entwickelt eine neue Anzeige für einen Kunden. Der Grafiker möchte einen besonderen Blauton als Hintergrundfarbe in der Anzeige verwenden und möchte wissen, ob diese Farbe bereits zuvor in einer Anzeige der Agentur verwendet wurde. Dazu führt er eine Anwendung aus, die nach Abbildern anhand des Inhalts sucht. Die Abbilder sind in einer Datenbanktabelle gespeichert (siehe Abb. 4 auf Seite 44). Die Anwendung fordert den Benutzer auf, ein visuelles Beispiel zur Verfügung zu stellen, d. h. ein Abbild, das die suchende Farbe darstellt. Die Anwendung analysiert dann die Farbe im Beispiel und sucht nach Abbildern, deren Farbe am meisten mit dem Beispiel übereinstimmt.

Abb. 7 zeigt ein visuelles Beispiel und die abgerufenen Abbilder, die am weitesten mit dessen Farbe übereinstimmen.

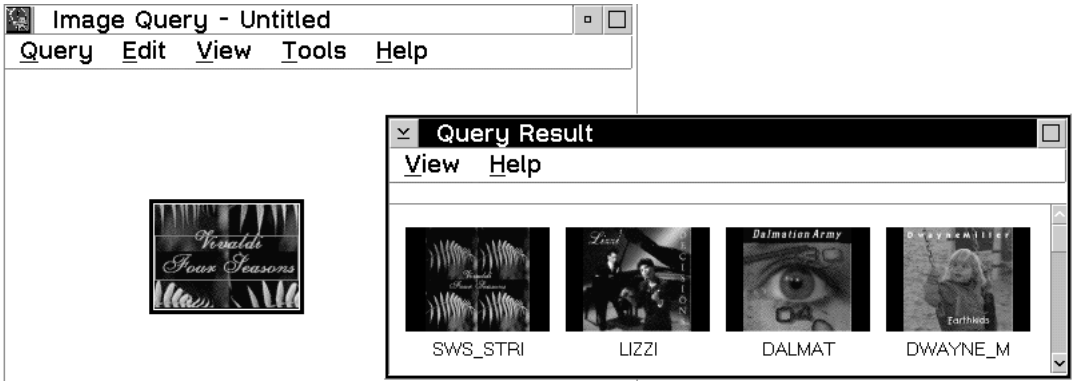


Abbildung 7. Suche von Abbildern anhand des Inhalts. Ein visuelles Beispiel wird verwendet, um nach Abbildern anhand der Durchschnittsfarbe zu suchen. Diese Abbildung wurde dem amerikanischen Buch als Beispiel entnommen und daher nicht übersetzt.

Abb. 8 auf Seite 47 einige Schlüsselelemente der Anwendung. Beachten Sie, dass die Anwendung die QBIC-API `QbQueryCreate` verwendet, um eine QBIC-Abfrage zu erstellen, die APIs `QbQueryAddFeature` und `QbQuerySetFeatureData`, um die Farbauswahl zur Abfrage hinzuzufügen, die API `QbQuerySearch`, um die Abfrage abzusetzen, und die API `QbQueryDelete`, um die Abfrage zu löschen. Die Anwendung verwendet außerdem die grafische API `DBiBrowse`, um die abgerufenen Abbilder anzuzeigen.

```
#include <dmbqbqpi.h>

#define   MaxQueryReturns        10

static  SQLHENV    henv;
static  SQLHDBC    hdbc;
static  SQLHSTMT   hstmt;
static  SQLRETURN  rc;

void main(int argc, char* argv[])
{
 char           line[4000];
 char*          handles[MaxQueryReturns];
 QbQueryHandle  qHandle=0;
 QbResult       results[MaxQueryReturns];
SQLINTEGER     count;
 SQLINTEGER     resultType=qbiArray;

 SQlAllocEnv(&henv);
 SQLAllocConnect(henv, &hdbc);
 rc = SQLConnect(hdbc, (SQLCHAR*)"qtest", SQL_NTS,
              (SQLCHAR*)"", SQL_NTS, (SQLCHAR*)"", SQL_NTS);

 if (argc !=2) {
 printf("usage: query colorname\n");
                exit(1);
 }

 QbImageSource is;
      is.type = qbiSource_AverageColor;

 /* run the get color subroutine */
 getColor(argv[1], is.average.Color);

 QbQueryCreate(&qhandle);
 QbQueryAddFeature(qhandle, "QbColorFeatureClass");
 QbQuerySetFeatureData(qhandle, "QbColorFeatureClass",&is);
 QbQuerySearch(qhandle, "ADS", "ADS_IMAGE", 10, 0, resultType
                           &count, results);
 for (int j = 0; j <count; j++) {
 printf(j,":\n");

 DBiBrowse("usr/local/bin/xv %s", MMDB_PLAY_HANDLE, handles[j],
                MMDB_PLAY_WAIT);
 }
```

Abbildung 8. Anwendung für die Abbildsuche anhand des Inhalts (Teil 1 von 2)

```
QbQueryDelete(qhandle);

 SQLDisconnect(hdbc);
 SQLFreeConnect(hdbc);
 SQLFreeEnv(henv);
}
```

Abbildung 8. Anwendung für die Abbildsuche anhand des Inhalts (Teil 2 von 2)

Betriebsumgebungen

Die Extender Version 7 arbeiten mit DB2 Universal Database Version 7.1 (oder höher) in einer Client/Server-Umgebung. Die Mindestversion und der Mindest-Release-Stand, die/der für die unterstützten Plattformen erforderlich ist, entspricht den Vorgaben für DB2 Universal Database Version 7.1.

Die unterstützten Clientplattformen sind: OS/2, AIX, Windows NT und später Windows 95, Windows 98, Solaris Operating Environment und HP-UX.

Die unterstützten Server sind: OS/2, AIX, Windows NT und später Solaris Operating Environment und HP-UX.

Abb. 9 auf Seite 49 zeigt die unterstützten Plattformen.

Ein weiteres DB2 Extender-Produkt, DB2 Universal Database for z/OS Extenders, unterstützt z/OS-Clients und -Server. Weitere Informationen zu den DB2 Universal Database für z/OS Extendern finden Sie im Handbuch *DB2 Universal Database for z/OS Image, Audio, and Video Extenders Administration and Programming* oder im Handbuch *DB2 Universal Database for z/OS Text Extender Administration and Programming*.

Die DB2 Extender können in einer Einzelpartitionsdatenbankumgebung arbeiten.

Nur EEE: Die Extender können auch in einer Mehrpartitionsdatenbankumgebung auf den folgenden Plattformen arbeiten: AIX, Solaris Operating Environment und Windows NT und später.

Die DB2 Extender müssen mit DB2 Universal Database Enterprise Extended Edition verwendet werden, um in einer Mehrpartitionsdatenbankumgebung arbeiten zu können.

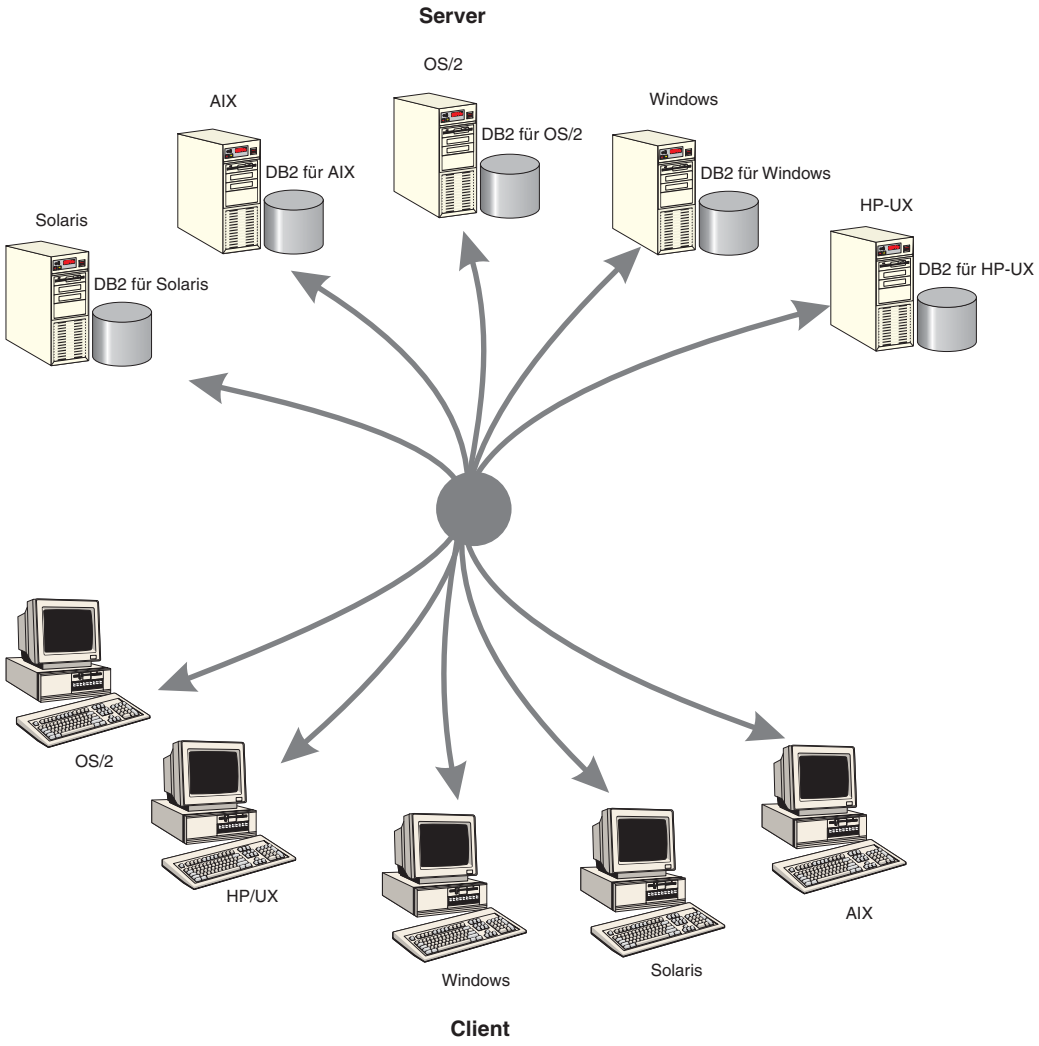


Abbildung 9. DB2 Extender-Plattformen

Beispiele

Kapitel 3. DB2 Extender-Konzepte

In diesem Kapitel werden die Konzepte beschrieben, die Ihnen vor der Verwendung der DB2 Extender geläufig sein sollten.

Weitere Informationen zu objektorientierten Konzepten befinden sich im Handbuch *DB2 Application Development Guide.*

Objektorientierte Konzepte

DB2 unterstützt die **Objektorientierung**, d. h., das Konzept, dass alle Elemente in einer Anwendung, real oder abstrakt, als Objekte dargestellt werden können. Ein solches Objekt umfasst eine Reihe von Operationen und Datenwerte. Beispielsweise kann ein Dokument durch ein Dokumentobjekt dargestellt werden, das die Dokumentdaten und die Operationen umfasst, die mit diesem Dokument ausgeführt werden können, wie z. B. Speichern, Senden und Drucken. Ein Videoclip kann als ein Videoobjekt dargestellt werden, das die Videodaten und die zugehörigen Operationen umfasst, wie etwa die Wiedergabe des Videoclips oder das Suchen eines bestimmten Videovollbildes. Genau wie reale Objekte können Darstellungsobjekte Attribute haben. Beispielsweise können Sie einem Videoobjekt Attribute, wie etwa Komprimierungsart und Abtastrate, geben.

Objekte können zu verschiedenen Objekttypen gruppiert werden. Objekte des gleichen Typs haben die gleichen Attribute und Funktionsweisen, d. h., ihnen sind die gleichen Operationen zugeordnet. Ist beispielsweise der Typ Video mit einem Attribut für die Komprimierungsart definiert, haben alle Objekte vom Typ Video dieses Attribut. Kann ein Objekt vom Typ Video wiedergegeben werden, können alle Objekte vom Typ Video wiedergegeben werden.

Durch die DB2-Unterstützung für die Objektorientierung ist es möglich, Exemplare von Objekttypen in Spalten von Tabellen zu speichern und mit Hilfe von Funktionen in SQL-Anweisungen mit ihnen zu arbeiten. Beispielsweise können Sie Videoobjekte in einer Tabellenspalte speichern und unter Verwendung von SQL-Funktionen mit ihnen arbeiten. Darüber hinaus können die Attribute und Funktionsweisen der gespeicherten Objekte innerhalb Ihrer Anwendungen gemeinsam benutzt werden. Für alle Anwendungen ist die Attributgruppe und Funktionsweise für einen bestimmten Objekttyp gleich.

Videoobjekte sind normalerweise groß und komplex. Das gleiche gilt für Abbild- und Audioobjekte. Als Teil der Unterstützung für die Objektorientierung ermöglicht DB2 das Speichern von großen Objekten (LOBs) in einer Datenbank. DB2 stellt Ihnen außerdem die Möglichkeit zur Verfügung, LOBs mit Hilfe von benutzerdefinierten Typen (UDTs), benutzerdefinierten Funktionen (UDFs) und Auslösern zu definieren und zu bearbeiten.

Große Objekte

DB2 ermöglicht Ihnen, **große Objekte** (LOBs) folgendermaßen in einer Datenbank zu speichern:
- als große binäre Objekte (BLOBs)
- als große Zeichenobjekte (CLOBs)
- als große Doppelbytezeichenobjekte (DBCLOBs)

BLOBs sind binäre Zeichenfolgen. Abbild-, Audio- und Videoobjekte werden als BLOBs in einer DB2-Datenbank gespeichert. CLOBs sind Zeichenfolgen, die aus Einzelbytezeichen bestehen und denen eine Zeichenumsetztabelle zugeordnet ist. Dieser Datentyp wird für Textobjekte verwendet, die Einzelbytezeichen enthalten. DBCLOBs sind Zeichenfolgen, die aus Doppelbytezeichen bestehen und denen eine Zeichenumsetztabelle zugeordnet ist. Dieser Datentyp wird für Textobjekte verwendet, die Doppelbytezeichen enthalten.

Jedes LOB kann bis zu 2 Gigabyte lang sein. Dennoch sind bei DB2 viele LOB-Spalte pro Tabelle zulässig. Sie können bis zu 24 Gigabyte LOB-Speicherbereich pro Zeile und bis zu 4 Terabyte LOB-Speicherbereich pro Tabelle speichern.

Aufgrund seiner Größe wird der LOB-Inhalt nicht direkt in der Tabelle des Benutzers gespeichert, sondern jedes LOB in der Tabelle ist durch einen Deskriptor für große Objekte gekennzeichnet. Mit Hilfe des Deskriptors wird auf das große Objekt zugegriffen, das an einem anderen Ort auf der Platte gespeichert ist. Die DB2 Extender geben Ihnen die zusätzliche Flexibilität zur Speicherung des Inhalts eines LOBs in einer Datei und zum Setzen eines Zeigers von der Datenbank auf diese Datei. Sie legen diese Zuordnung fest, wenn Sie einen DB2 Extender zum Speichern eines Objekts verwenden.

Benutzerdefinierte Typen

Abbild-, Video- und Audioobjekte werden in der Datenbank als BLOBs dargestellt. Ein **benutzerdefinierter Typ** (UDT), auch bekannt als **eindeutiger Typ**, stellt eine Möglichkeit zur Verfügung, zwischen den einzelnen BLOBs zu unterscheiden. Beispielsweise kann ein UDT für Abbildobjekte und ein anderer für Audioobjekte erstellt werden. Obwohl sie als BLOBs gespeichert werden, werden Abbild- und Audioobjekte von BLOBs und auch untereinander unterschieden.

UDTs werden mit einer SQL-Anweisung CREATE DISTINCT TYPE erstellt. Angenommen, Sie entwickeln eine Anwendung, die die geografischen Merkmale auf Landkarten verarbeitet. Sie können einen eindeutigen Typ mit dem Namen `map` für Kartenobjekte wie folgt erstellen:

```
CREATE DISTINCT TYPE map AS BLOB (1M)
```

Das Objekt vom Typ 'map' wird intern als ein 1 Megabyte langes BLOB dargestellt, aber als Objekt mit einem eindeutigen Typ behandelt.

Sie können UDTs wie integrierte SQL-Typen verwenden, um die Daten in Tabellenspalten zu beschreiben. Im folgenden Beispiel wird eine Tabelle mit einer Spalte erstellt, die für die Daten vom Typ 'map' konzipiert ist:

```
CREATE TABLE places
  (locid     INTEGER NOT NULL,
   location  CHAR (50),
   grid      map)
```

Jeder DB2 Extender erstellt einen UDT für seinen Datentyp, d. h. Abbild, Audio und Video.

Benutzerdefinierte Funktionen

Eine **benutzerdefinierte Funktion** (UDF) ist eine Möglichkeit, SQL-Funktionen zu erstellen und zur Gruppe der integrierten Funktionen hinzuzufügen, die mit DB2 geliefert werden. Insbesondere können Sie UDFs erstellen, die Operationen ausführen, die einzigartig für Abbild-, Audio- und Videoobjekte sind. Beispielsweise können Sie UDFs erstellen, um das Komprimierungsformat eines Videos abzurufen oder die Abtastrate eines Tons zurückzugeben. Hierdurch kann die Funktionsweise von Objekten definiert werden, die zu einem bestimmtem Typ gehören. Videoobjekte arbeiten beispielsweise gemäß den Funktionen, die für den Videotyp erstellt wurden, und Abbildobjekte arbeiten gemäß den Funktionen, die für den Abbildtyp erstellt wurden.

UDFs werden mit einer SQL-Anweisung CREATE FUNCTION erstellt. Die Anweisung gibt unter anderem den Datentyp an, für den die UDF angewendet werden kann. Beispielsweise erstellt die folgende Anweisung eine UDF mit dem Namen `map_scale`, die den Maßstab einer Landkarte berechnet. Beachten Sie, dass die UDF 'map' als den Datentyp identifiziert, auf den sie angewendet werden kann. Der Code, der die Funktion implementiert, ist in C geschrieben und wird in der Klausel EXTERNAL NAME angegeben:

```
CREATE FUNCTION map_scale (map)
  RETURNS SMALLINT
  EXTERNAL NAME 'scale!map'
  LANGUAGE C
  PARAMETER STYLE DB2SQL
  NO SQL
  DETERMINISTIC
  NO EXTERNAL ACTION
```

UDFs können in einer SQL-Anweisung wie integrierte Funktionen verwendet werden. Im folgenden Beispiel wird die UDF 'map_scale' in einer SQL-Anweisung verwendet, um den Maßstab einer Karte zurückzugeben, die in einer Tabellenspalte mit dem Namen 'grid' gespeichert ist:

```
SELECT map_scale (grid)
  FROM places
  WHERE location='SAN JOSE, CALIFORNIA'
```

Jeder DB2 Extender erstellt eine Gruppe von UDFs für den jeweiligen Typ, d. h. abbild-, audio- und videospezifische UDFs. Sie können diese UDFs in SQL-Anweisungen verwenden, um Extender-Funktionen anzufordern, wie z. B. Speichern eines Abbilds in einer Tabelle, Abrufen der Vollbildrate eines Videos oder Hinzufügen eines Kommentars zu Audiodaten.

UDF- und UDT-Namen

Der vollständige Name einer DB2-Funktion hat die Syntax *schemaname.funktionsname*, wobei *schemaname* eine Kennung ist, die eine logische Gruppierung für SQL-Objekte zur Verfügung stellt. Der Schemaname für DB2 Extender-UDFs ist MMDBSYS. Der Schemaname MMDBSYS ist außerdem das Qualifikationsmerkmal für die DB2 Extender-UDTs.

Sie können den vollständigen Namen immer verwenden, wenn Sie auf eine UDF oder einen UDT verweisen. Beispielweise gibt MMDBSYS.CONTENT eine UDF an, deren Schemaname MMDBSYS ist und deren Funktionsname CONTENT ist. MMDBSYS.DB2IMAGE gibt einen UDT an, dessen Schemaname MMDBSYS ist und dessen Name für den eindeutigen Typ DB2IMAGE ist.

Sie können den Schemanamen auch weglassen, wenn Sie auf eine UDF oder einen UDT verweisen. In diesem Fall verwendet DB2 den Funktionspfad, um die gewünschte Funktion oder den gewünschten eindeutigen Datentyp zu bestimmen.

Funktionspfad

Der **Funktionspfad** definiert eine geordnete Liste von Schemanamen. DB2 verwendet die Reihenfolge der Schemanamen in der Liste, um Referenzen auf Funktionen und eindeutige Datentypen aufzulösen. Sie können den Funktionspfad angeben, indem Sie die SQL-Anweisung SET CURRENT FUNCTION PATH verwenden. Mit dieser Anweisung wird der Funktionspfad im Sonderregister CURRENT FUNCTION PATH definiert.

Für die DB2 Extender ist es empfehlenswert, das Schema `mmdbsys` zum Funktionspfad hinzuzufügen. Dadurch können Sie DB2 Extender-UDF- und -UDT-Namen eingeben, ohne das Präfix 'mmdbsys' verwenden zu müssen. Es folgt ein Beispiel, wie das Schema 'mmdbsys' zum Funktionspfad hinzugefügt wird:

```
SET CURRENT FUNCTION PATH = mmdbsys, CURRENT FUNCTION PATH
```

Fügen Sie 'mmdbsys' nicht als erstes Schema im Funktionspfad hinzu, wenn Sie sich als mmdbsys anmelden: Wenn Sie sich unter der Benutzer-ID 'mmdbsys' anmelden, ist das erste Schema im Funktionspfad auf 'mmdbsys' gesetzt. Wenn Sie dann versuchen, das erste Schema im Funktionspfad mit Hilfe einer Anweisung SET CURRENT FUNCTION PATH auf 'mmdbsys' zu setzen, beginnt der Funktionspfad mit zwei Schemata 'mmdbsys', was zu einer Fehlerbedingung führt.

Mehrfach belegte Funktionsnamen

Funktionsnamen können **mehrfach belegt** sein. Das heißt, mehrere UDFs, auch in demselben Schema, können den gleichen Namen haben. Zwei Funktionen können jedoch nicht die gleiche **Kennung** haben. Eine Kennung ist ein qualifizierter Funktionsname, der mit den definierten Datentypen aller Funktionsparameter verknüpft ist.

Auslöser

Ein **Auslöser** definiert eine Gruppe von Aktionen, die durch eine Änderung in einer Tabelle aktiviert wird. Auslöser können für folgende Aktionen verwendet werden: Prüfen von Eingabedaten, automatisches Generieren eines Wertes für eine neu eingefügte Zeile, Lesen von anderen Tabellen zu Querverweiszwecken oder Schreiben in andere Tabellen zu Prüfzwecken. Auslöser werden oft zur Überprüfung der Integrität oder zum 'Erzwingen' von Geschäftsregeln verwendet.

Auslöser werden mit einer SQL-Anweisung CREATE TRIGGER erstellt. Die folgende Anweisung erstellt einen Auslöser, der eine Geschäftsregel bezüglich des Teileinventars erzwingt. Der Auslöser bestellt ein Teil neu, wenn weniger als 10 Prozent des maximalen Lagerbestandes vorrätig sind.

```
CREATE TRIGGER reorder
       AFTER UPDATE OF on_hand, max_stocked ON parts
       REFERENCING NEW AS n_row
       FOR EACH ROW MODE DB2SQL

       WHEN (n_row.on_hand < 0.10 * n_row.max_stocked)
       BEGIN ATOMIC
              VALUES(issue_ship_request(n_row.max_stocked -
                                        n_row.on_hand,
                                        n_row.partno));
       END
```

Die DB2 Extender erstellen und verwalten Tabellen zur Verwaltungsunterstützung, um Informationen zu Abbild-, Audio- und Videodaten aufzuzeichnen, die in einer Datenbank gespeichert sind. (Weitere Informationen zu diesen Tabellen befinden sich im Abschnitt „Tabellen zur Verwaltungsunterstützung".) Die Extender verwenden Auslöser, um diese Tabellen zu aktualisieren, wenn Abbild-, Audio- oder Videodaten in eine Datenbank eingefügt, in dieser aktualisiert oder aus dieser gelöscht werden.

Extender-Datenstrukturen

Die Image, Audio und Video Extender erstellen und verwenden Tabellen zur Verwaltungsunterstützung und interne Kennungen, um Abbild-, Audio- und Videodaten zu speichern und auf diese zuzugreifen. Der Image Extender erstellt und verwendet auch QBIC-Kataloge, um anhand des Inhalts auf Abbilder zuzugreifen. Der Video Extender verwendet außerdem Indexdateien und Aufnahmekataloge, um auf Informationen zu Szenenwechseln in einem Video zuzugreifen.

Tabellen zur Verwaltungsunterstützung

Tabellen zur Verwaltungsunterstützung, auch Metadatentabellen genannt, enthalten die Informationen, die die Extender benötigen, um Benutzeranforderungen nach Abbild-, Audio- und Videoobjekten zu verarbeiten. Die Informationen in Tabellen zur Verwaltungsunterstützung werden oft als „Metadaten" bezeichnet.

Wie die Abb. 10 auf Seite 56 zeigt, geben einige der Tabellen zur Verwaltungsunterstützung Benutzertabellen und Spalten an, die für einen Extender aktiviert sind-.Diese Tabellen verweisen auf andere Tabellen zur Verwaltungsunterstützung, die erstellt werden, um Attributinformationen zu Objekten in aktivierten Spalten zu halten. In diesen Tabellen verwalten die Extender Informationen zu Attributen, die für die jeweiligen vom Extender definierten Datentypen einzigartig sind, sowie Informationen zu Attributen, die bei den verschiedenen Extender-Datentypen gleich sind. Beispielsweise pflegt der Image Extender Informationen zur Breite, Höhe und Anzahl an Farben in einem Abbild sowie Informationen zu Attributen, die Abbild-, Audio- und Videoobjekte gemeinsam haben, wie z. B. die Kennung der Person, die das Objekt in die Datenbank importiert oder es zuletzt aktualisiert hat.

Die Tabellen zur Verwaltungsunterstützung können außerdem den Inhalt von gespeicherten Tabellen im BLOB-Format enthalten. Alternativ dazu kann ein Objekt in einer Datei gespeichert werden, auf die durch die Tabellen zur Verwaltungsunterstützung verwiesen wird. Beispielsweise kann ein Videoclip als BLOB in einer Tabelle zur Verwaltungsunterstützung gespeichert werden oder in einer Datei, auf die in der Tabelle verwiesen wird.

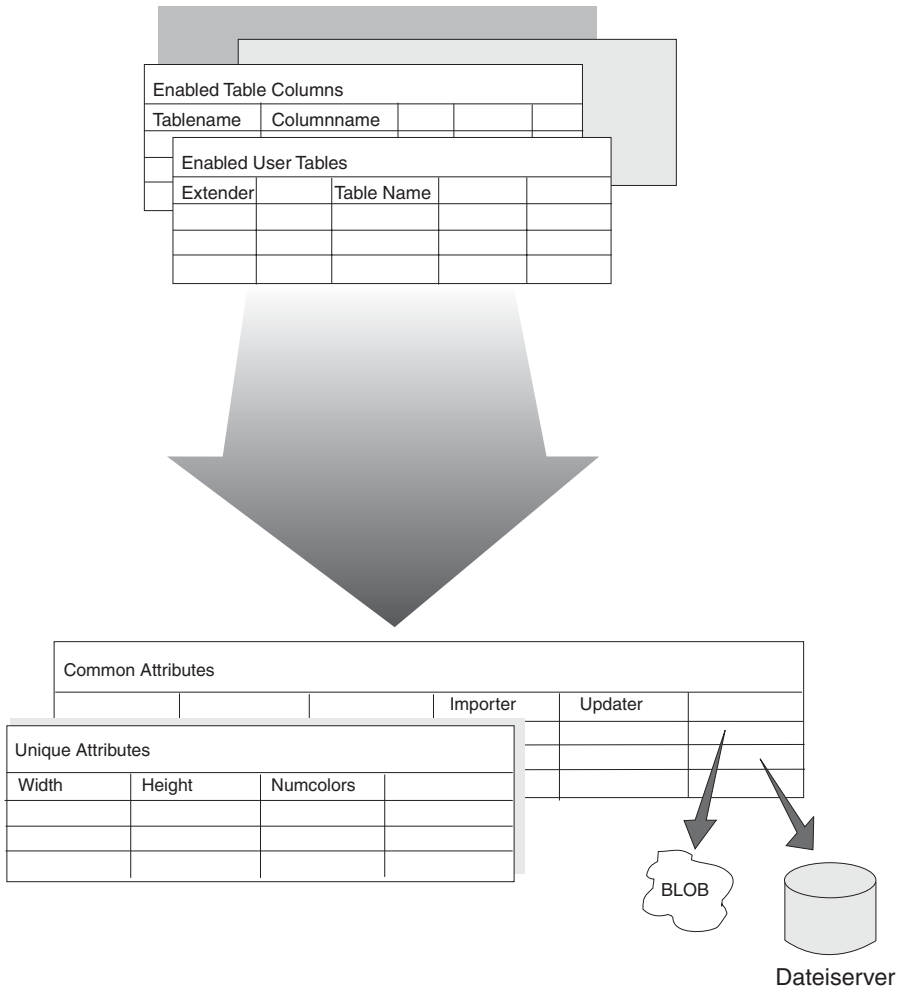


Abbildung 10. Tabellen zur Verwaltungsunterstützung

Kennungen

Beim Speichern eines Abbild-, Audio- oder Videoobjekts in einer Benutzertabelle wird das Objekt nicht tatsächlich in der Tabelle gespeichert. Stattdessen erstellt ein Extender eine Zeichenfolge, eine so genannte **Kennung**, um das Objekt darzustellen, und speichert die Kennung in der Tabelle. Der Extender speichert das Objekt in einer Tabelle zur Verwaltungsunterstützung oder speichert eine Dateikennung in einer Tabelle zur Verwaltungsunterstützung, wenn der Inhalt des Objekts in einer Datei gespeichert wird. Er speichert außerdem die Attribute und die Kennung des Objekts in den Tabellen zur Verwaltungsunterstützung. Auf diese Weise kann der Extender eine Verbindung zwischen der Kennung, die in einer Benutzertabelle gespeichert ist, und den Objektinformationen, die in den Tabellen zur Verwaltungsunterstützung gespeichert sind, herstellen. Abb. 11 auf Seite 57 zeigt die Informationen, die für zwei Abbilder in einer Benutzertabelle gespeichert sind.

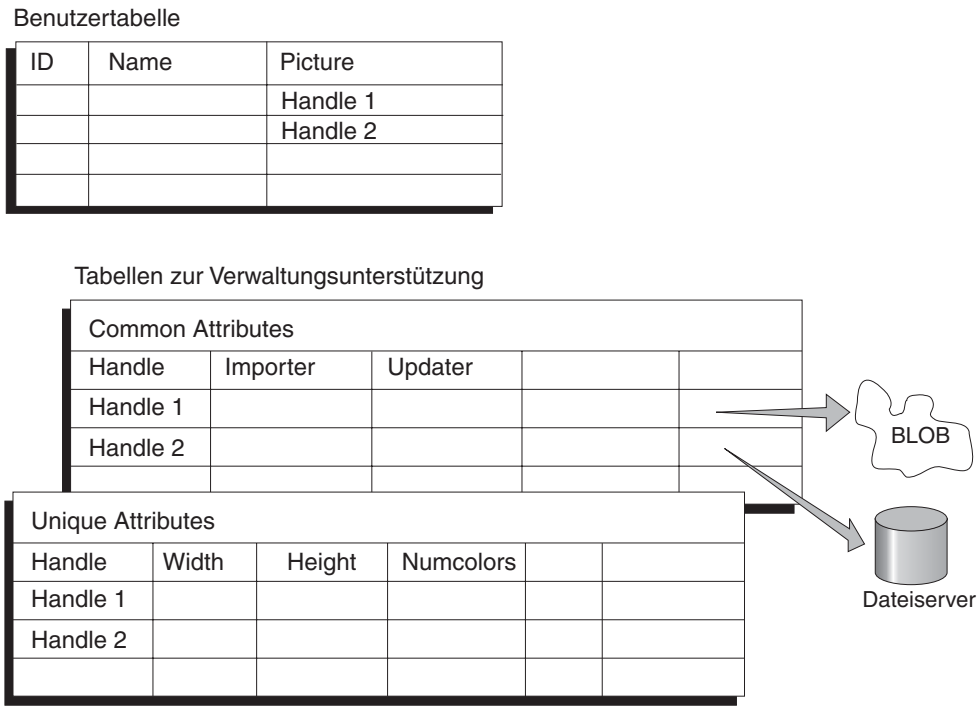


Abbildung 11. Kennungen

QBIC-Kataloge

Ein **QBIC-Katalog** ist eine Dateigruppe, die Daten zu visuellen Merkmalen von Abbildern hält. Der Image Extender verwendet diese Daten, um anhand des Inhalts nach Abbildern zu suchen.

Sie erstellen einen QBIC-Katalog für jede Abbildspalte in einer Benutzertabelle, die Sie für die Suche anhand des Inhalts verfügbar machen wollen. Beim Erstellen eines QBIC-Katalogs geben Sie die Merkmale an, für die der Image Extender Daten analysieren, speichern und später abfragen soll. Nach dem Erstellen des Katalogs können Sie Merkmale zum QBIC-Katalog hinzufügen oder aus dem QBIC-Katalog löschen.

Ein QBIC-Katalog kann Daten für die folgenden Abbildmerkmale enthalten:

Durchschnittsfarbe
: Die Summe der Farbwerte für alle Pixel in einem Abbild, geteilt durch die Anzahl von Pixel in einem Abbild. (Ein Pixel ist das kleinste Element eines Abbilds, dem Farbe und Lichtintensität zugeordnet werden kann.) Bestehen beispielsweise 50 Prozent eines Abbilds aus blauen Pixeln und die anderen 50 Prozent aus roten Pixeln, hat das Abbild die Durchschnittsfarbe Purpurrot. Die Durchschnittsfarbe wird verwendet, um nach Abbildern zu suchen, die eine vorherrschende Farbe haben. Hat ein Abbild eine vorherrschende Farbe, ähnelt die Durchschnittsfarbe dieser Farbe.

Histogrammfarbe
Misst die Verteilung von Farben in einem Abbild gegenüber einem Spektrum von 64 Farben. Für jede der 64 Farben gibt 'Histogrammfarbe' den Prozentsatz von Pixeln in einem Abbild an, die diese Farbe haben. Beispielsweise kann die Histogrammfarbe eines Abbilds 40% weiße Pixel, 50% blaue und 10% rote Pixel angeben. Keines der Pixel in dem Abbild hat eine der restlichen Farben im Histogrammspektrum. 'Histogrammfarbe' wird verwendet, um nach Abbildern zu suchen, die eine Vielzahl von Farben haben.

Positionsgebundene Farbe
Der Wert der Durchschnittsfarbe für Pixel in einem angegebenen Bereich in einem Abbild. Beispielsweise könnte in der rechten oberen Ecke eines Abbilds eine hellgelbe Sonne dargestellt sein. Die positionsgebundene Farbe für diesen Bereich des Abbilds ist hellgelb. Die positionsgebundene Farbe wird verwendet, um nach Abbildern zu suchen, die in einem bestimmten Bereich eine vorherrschende Farbe haben.

Textur
Misst Grobheit, Kontrast und Direktionalität eines Abbilds. Die Grobheit gibt die Größe der sich wiederholenden Elemente in einem Abbild an (z. B. 'Kieselsteine versus Felsbrocken'). Der Kontrast gibt die Helligkeitsvariationen in einem Abbild an (hell versus dunkel). Die Direktionalität gibt an, ob eine Richtung in einem Abbild dominiert (wie etwa die vertikale Richtung eines Palisadenzauns) oder ob keine Richtung dominiert (wie bei einem Abbild von Sand). Textur wird verwendet, um nach Abbildern zu suchen, die ein bestimmtes Muster haben.

Damit nach einem Abbild anhand des Inhalts gesucht werden kann, müssen Sie das Abbild katalogisieren. Beim Katalogisieren eines Abbilds analysiert der Image Extender das Abbild, indem er die Merkmalwerte für das Abbild berechnet, und speichert die Werte in einem QBIC-Katalog.

Wenn Sie nach einem Abbild anhand des Inhalts suchen, gibt Ihre Abfrage ein oder mehrere Merkmale für die Suche (z. B. Durchschnittsfarbe), eine Quelle für jedes Merkmal (z. B. ein Beispielabbild) und eine Zielgruppe von katalogisierten Abbildern an. Der Image Extender berechnet den Merkmalwert der Quelle und vergleicht ihn mit den katalogisierten Merkmalwerten für die Zielabbilder. Er berechnet dann ein Ergebnis, das angibt, wie ähnlich sich die Merkmalwerte der Zielabbilder und der Quelle sind.

Sie können angeben, dass der Image Extender die Abbilder zurückgibt, die der Quelle am ähnlichsten sind. Der Image Extender gibt eine Kennung jedes Abbilds und das Ähnlichkeitsergebnis des Abbilds zurück. Sie können auch angeben, dass der Image Extender nur das Ähnlichkeitsergebnis eines einzelnen Abbilds zurückgibt.

Videoindizes

Ein **Videoindex** ist eine Datei, die der Video Extender verwendet, um eine bestimmte Aufnahme oder ein bestimmtes Vollbild in einem Videoclip zu suchen.

Der Video Extender kann Szenenwechsel in einem Video ermitteln. Ein **Szenenwechsel** ist der Punkt in einem Videoclip, an dem ein deutlicher Unterschied zwischen zwei aufeinander folgenden Vollbildern besteht. Ein Szenenwechsel findet beispielsweise statt, wenn eine Kamera während der Aufzeichnung eines Videos ihren Standpunkt ändert. Die Vollbilder zwischen zwei Szenenwechseln bilden eine **Aufnahme**.

Sie können mit den Suchfunktionen für Szenen, die der Video Extender bietet, eine Aufnahme oder sogar ein einzelnes Vollbild in einem Videoclip suchen. Dazu benötigt der Extender Indexierungsinformationen für die Aufnahme oder das Vollbild. Diese Indexierungsinformationen werden in einer **Indexdatei** gespeichert.

Aufnahmekataloge

Ein **Aufnahmekatalog** wird verwendet, um Daten zu Aufnahmen in einem Videoclip zu speichern. Der Aufnahmekatalog kann in einer Datenbank oder in einer Datei gespeichert werden.

Ein Aufnahmekatalog, der in einer Datei gespeichert ist, enthält die folgenden aufnahmebezogenen Daten:

- Name der Aufnahmekatalogdatei
- Werte zur Steuerung, wie der Video Extender eine Aufnahme ermittelt, z. B. die Mindestanzahl von Vollbildern in einer Aufnahme
- Werte zur Steuerung, wie viele und welche Vollbilder als repräsentative Vollbilder für eine Aufnahme gespeichert werden
- Aufnahmenummer
- Nummer des ersten Vollbilds
- Nummer des letzten Vollbilds
- Repräsentative Vollbildnummer
- Name der Datei, die den Inhalt des repräsentativen Vollbilds enthält

Sie können auf die Daten in der Aufnahmekatalogdatei oder auf eine Sicht des Aufnahmekatalogs, der in der Datenbank gespeichert ist, zugreifen. Die Sicht enthält Spalten für die folgenden aufnahmebezogenen Daten:

- Aufnahmekennung
- Videotabellenname
- Videospaltennamen
- Videokennung
- Videodateiname
- Nummer des ersten Vollbilds
- Nummer des letzten Vollbilds
- Repräsentative Vollbildnummer
- Repräsentative Vollbilddaten
- Kommentar

Konzepte für partitionierte Datenbanken (nur EEE)

Die DB2 Extender können mit DB2 Extended Enterprise Edition arbeiten. Auf diese Weise können sie die Unterstützung für partitionierte Datenbanken nutzen, die unter DB2 Extended Enterprise Edition bereitgestellt wird.

Eine **partitionierte Datenbank** ist eine Datenbank, die über zwei oder mehr unabhängige Maschinen verteilt ist. Für den Endbenutzer und Anwendungsentwickler erscheint die Datenbank wie eine einzelne Datenbank auf einer einzelnen Maschine. Durch die Partitionierung können Anwendungen eine Datenbank effizient nutzen, die zu groß ist, um auf nur einer Maschine ausgeführt werden zu können.

Eine partitionierte Datenbank besteht aus zwei oder mehr Partitionen. Jede Partition wird von ihrem eigenen **Datenbankpartitionsserver** verwaltet. Zu einem Datenbankpartitionsserver gehört ein Datenbankmanager und die Sammlung von Daten und Systemressourcen, die er verwaltet. Normalerweise ist jeder Maschine ein Datenbankpartitionsserver zugeordnet. Es ist jedoch möglich, mehrere Datenbankpartitionsserver auf einer einzelnen Maschine zu haben. Jeder Datenbankpartitionsserver hält einen Teil der gesamten Datenbank. Ein Datenbankpartitionsserver wird in einigen Fällen auch **Knoten** genannt.

Wie die Abb. 12 auf Seite 61 zeigt, können Datenbankpartitionen logisch gruppiert und mit einem Namen versehen werden. Jede Gruppe von Datenbankpartitionen ist als **Knotengruppe** bekannt. Durch das Definieren von Knotengruppen ist es beispielsweise möglich, die Anwendungsabfragen auf ausgewählte Datenbankpartitionen zu begrenzen und somit die Transaktionszeiten zu verbessern. Eine Knotengruppe kann nur eine Datenbankpartition oder auch mehrere Datenbankpartitionen enthalten. Wenn eine Knotengruppe mehrere Datenbankpartitionen enthält, wird sie als **Mehrpartitionsknotengruppe** bezeichnet. Alle Datenbankpartitionen, die zu einer Mehrpartitionsknotengruppe gehören, müssen sich innerhalb derselben Datenbank befinden.

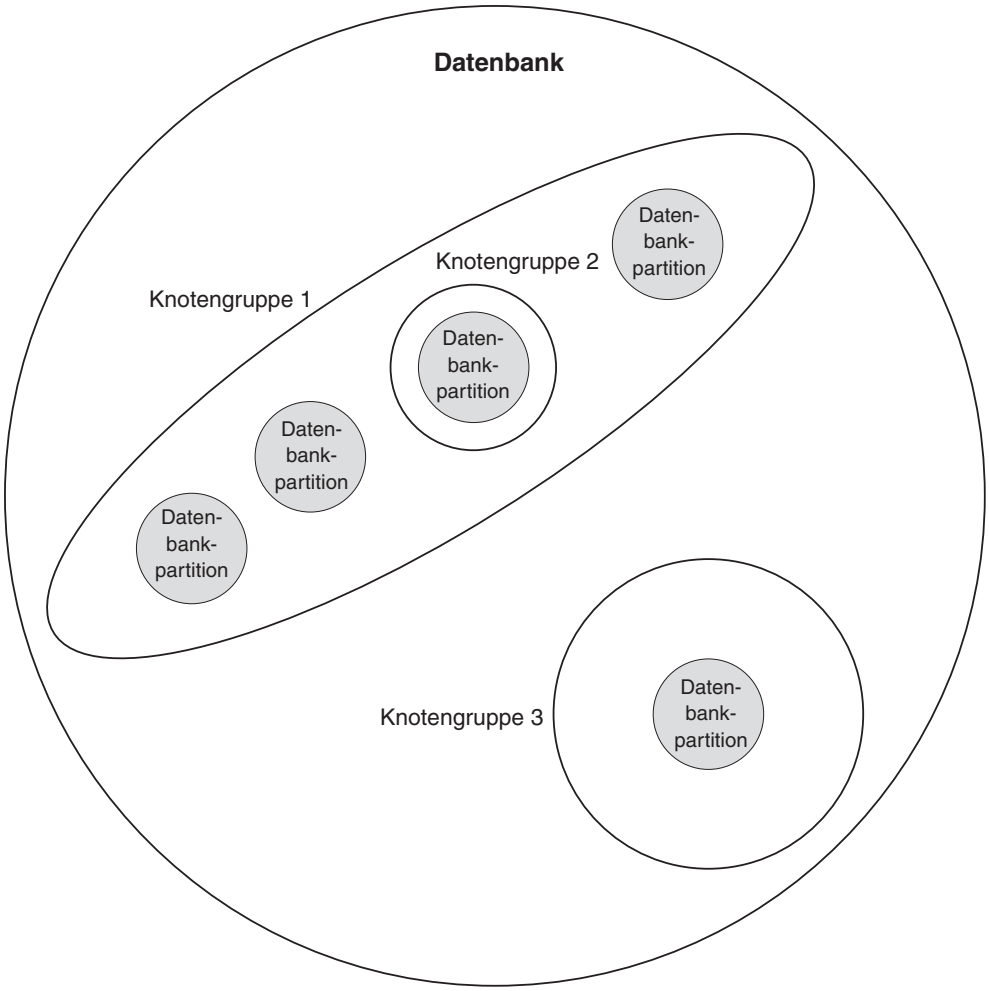


Abbildung 12. Knotengruppen in einer Datenbank

Mit Hilfe der Extender können Sie auf einem partitionierten Datenbanksystem folgende Operationen ausführen:

- Verringern der Ein-/Ausgabeoperationen und Beheben von Engpässen, indem Sie Daten über mehrere Partitionen verteilen.
- Erhöhen der Datenbankgröße, indem Sie weitere Maschinen hinzufügen und die Daten auf diese umverteilen.

Parallelverarbeitung

Eine partitionierte Datenbank kann mehrere CPUs verwenden, um Informationsanforderungen zu verarbeiten. Abruf- und Aktualisierungsanforderungen werden automatisch in Unteranforderungen aufgeteilt und parallel auf den Datenbankpartitionsservern der einzelnen Maschinen ausgeführt.

Um die Leistungsfähigkeit eines partitionierten Datenbanksystems zu illustrieren, nehmen Sie an, dass Sie 100 000 000 Datensätze in einer Datenbank mit nur einer Partition suchen wollen. Dieser Suchvorgang würde erfordern, dass ein einzelner Datenbankmanager 100 000 000 Datensätze sucht. Nehmen Sie jetzt an, dass diese Datensätze in gleichen Teilen auf 20 Datenbankpartitionsservern verteilt wären: jeder Datenbankmanager müsste nur 5 000 000 Datensätze suchen. Wenn jeder Datenbankmanager zur gleichen Zeit und mit der gleichen Geschwindigkeit sucht, würde die erforderliche Zeit für den Suchvorgang nur bei etwa 5 % davon liegen, was ein Einzelpartitionssystem für die Ausführung dieser Task benötigte.

Skalierbarkeit

Wenn Ihre Datenbank größer wird, können Sie Datenbankpartitionsserver zum Datenbanksystem hinzufügen, um die Leistung zu verbessern. Dieser Prozess ist als **Skalieren** des Datenbanksystems bekannt.

Wenn Sie eine Datenbank skalieren, fügen Sie einen Datenbankpartitionsserver hinzu, der seinerseits zu jeder existierenden Datenbank im Datenbanksystem eine Datenbankpartition hinzufügt. Sie können anschließend die neue Datenbankpartition einer existierenden Knotengruppe für diese Datenbank zuordnen. Schließlich können Sie Daten in dieser Knotengruppe neu verteilen, um die neue Datenbankpartition zu nutzen.

DB2 Extender in einer Umgebung für partitionierte Datenbanken verwenden

Durch die Verwendung der DB2 Extender in einer Umgebung für partitionierte Datenbanken können Sie die Funktionen nutzen, die die Bearbeitung von LOBs besonders gut unterstützen. Große Repositorys von LOBs (die jeweils bis zu 2 GB lang sein können) können in einer Datenbank gespeichert werden, seit eine Datenbank über mehrere Maschinen verteilt sein kann.

Die DB2 Extender unterstützen außerdem die Parallelverarbeitung von SQL-Operationen, die von DB2 Extended Enterprise Edition verwaltet werden. Wenn DB2 Extended Enterprise Edition eine Abfrage parallel ausführt, wird jede DB2 Extender UDF in der Abfrage ebenfalls auf den einzelnen Datenbankpartitionen parallel ausgeführt.

Sicherheit und Wiederherstellung

Abbild-, Audio- und Videoobjekte, die als BLOBs in einer DB2-Datenbank gespeichert sind, erhalten die gleiche Sicherheit und den gleichen Wiederherstellungsschutz wie herkömmliche numerische Daten und Zeichendaten. Das gleiche gilt für Informationen, die für diese Daten in Metadatentabellen gespeichert sind. Benutzer müssen über die erforderliche Berechtigung verfügen, um Objekte auswählen, einfügen oder aktualisieren zu können.

Ein Benutzer gibt UDFs aus, um Objekte aus einer Benutzertabelle auszuwählen, einzufügen, zu aktualisieren oder zu löschen. Um die angeforderten Operationen auszuführen, müssen die UDFs auf die Tabellen zur Verwaltungsunterstützung, die Attributinformationen für die Objekte halten, zugreifen und sie gegebenenfalls aktualisieren können. Die Extender ermöglichen den UDFs, diese Operationen für die Tabellen zur Verwaltungsunterstützung auszuführen, wenn der Benutzer die entsprechende Berechtigung für die Benutzertabelle hat.

Für einige extender-bezogene Verwaltungsoperationen ist die Berechtigung DBADM erforderlich. Informationen zur Berechtigung, die für DB2 Extender-Verwaltungs-APIs erforderlich ist, befinden sich in Kapitel 14, „Anwendungsprogrammierschnittstellen", auf Seite 253. Informationen zur Berechtigung, die für die Verwaltungsbefehle der DB2 Extender erforderlich ist, befinden sich in Kapitel 15, „Verwaltungsbefehle für den Client", auf Seite 449.

Wird der Inhalt eines Abbild-, Audio- oder Videoobjekts in einer Datei gespeichert, auf die von der Datenbank verwiesen wird, werden die Metadaten für das Objekt durch DB2 geschützt. Die Datei muss sich in einem Verzeichnis befinden, das von der Benutzergruppe PUBLIC, das heißt von allen Benutzern, gelesen werden kann.

BLOBs und Metadaten werden auf die gleiche Art und Weise gesichert und wiederhergestellt wie andere Daten in DB2. Objektinhalte, die in einer Datei gespeichert sind, können mit Hilfe von Nicht-DB2-Tools gesichert und wiederhergestellt werden. Auch QBIC-Kataloge und Videoindizes können mit Hilfe von Nicht-DB2-Tools gesichert und wiederhergestellt werden. Weitere Informationen zum Sichern eines QBIC-Katalogs finden Sie auf Seite 156. Weitere Informationen zum Sichern eines Videoindexes finden Sie auf Seite 28.

Kapitel 4. Arbeitsweise der Extender

Mit den DB2 Extendern wird Ihnen viel Arbeit bei der Ausführung von Abbild-, Audio- und Videodatenanforderungen abgenommen. Eine gute Möglichkeit, die Arbeitsweise der Extender zu zeigen, ist, sie bei der Verwendung zu analysieren. In diesem Kapitel wird ein Szenario beschrieben, das sich mit dem Image Extender und dem Audio Extender befasst. Operationen des Benutzers und die Art und Weise, wie die Extender antworten, werden erläutert.

Extender-Szenario

Die Personalabteilung eines Unternehmens möchte eine Personaldatenbank (in DB2 für AIX) erstellen, die die Bilder der einzelnen Mitarbeiter enthält.

Eine Datenbank mit Bildern: Wie Abb. 13 zeigt, enthält eine Mitarbeitertabelle die Kennung und den Namen sowie die Bilder der einzelnen Mitarbeiter.

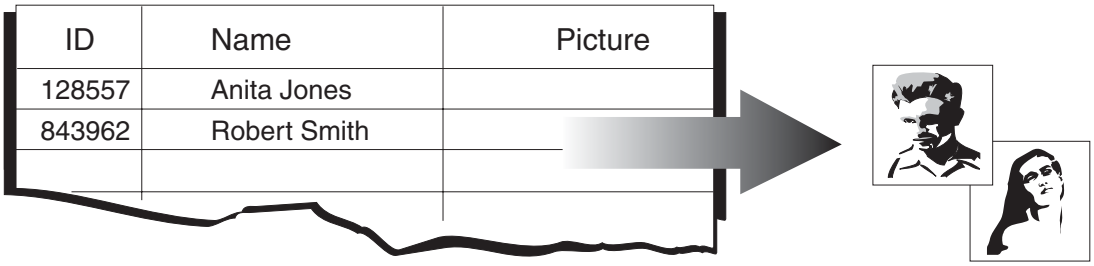


Abbildung 13. Die Mitarbeitertabelle 'employee'

Um die Personaldatenbank für die Abbildverarbeitung vorzubereiten, startet ein Systemadministrator (eine Person mit der Berechtigung SYSADM) die Extender-Services. Der Systemadministrator erstellt dann eine Datenbank und aktiviert diese für die Verwendung durch den Image Extender.

Ein Datenbankadministrator (DBA) oder eine Person mit der entsprechenden Berechtigung erstellt die Mitarbeitertabelle (employee) und aktiviert dann die Tabelle und die Spalte für die Mitarbeiterbilder (picture) für die Verwendung durch den Image Extender.

Eine Datenbank mit Audiodaten: Nachdem die Personaldatenbank und die Mitarbeitertabelle für die Abbildverarbeitung vorbereitet wurden, entschließt sich die Personalabteilung, eine Stimmaufzeichnung der einzelnen Mitarbeiter zur Tabelle hinzuzufügen. Dies wird in Abb. 14 auf Seite 66 gezeigt.

Szenario

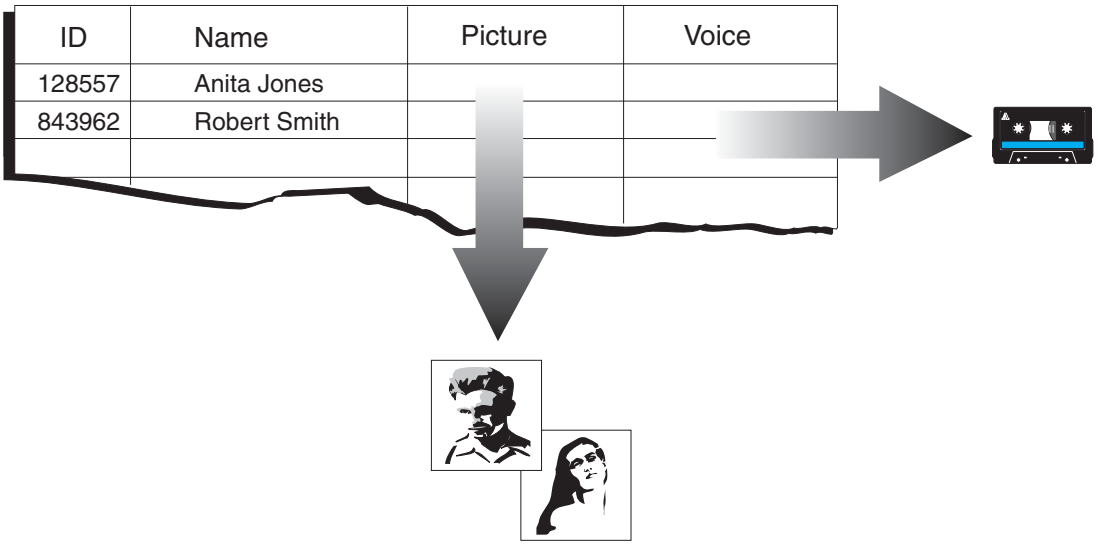


Abbildung 14. Die Mitarbeitertabelle mit einer hinzugefügten Audiospalte

Der Systemadministrator ändert die Tabelle, indem er eine neue Spalte hinzufügt, und aktiviert die Datenbank, Tabelle und Spalte für die Verwendung durch den Audio Extender.

Benutzer in der Personalabteilung können die Tabelle dann verwenden, um Daten einzufügen, auszuwählen und anzuzeigen, zu aktualisieren oder zu löschen.

Extender-Services starten

Die Extender verwenden Services auf dem Server als Teil ihrer Arbeitsgänge. Wenn diese Services noch nicht als eine Funktion der normalen Startoperationen für den Server zur Verfügung stehen, startet der Systemadministrator sie.

Aktionen des Systemadministratoren: Der Systemadministrator meldet sich am AIX-Server als Eigner des Extender-Exemplars an. Er gibt danach folgenden Befehl auf dem Server ein:

```
DMBSTART
```

Ereignis: Die Extender-Services werden für das Extender-Exemplar auf dem Server gestartet. Mit dem Befehl DMBSTART wird außerdem ein DB2-Exemplar gestartet, wenn es noch nicht aktiv ist.

Datenbank vorbereiten

Der Systemadministrator erstellt die Personaldatenbank und aktiviert sie für die Verwendung durch den Image Extender.

Aktionen des Systemadministratoren: Der Systemadministrator erstellt die Personaldatenbank in DB2 für AIX unter Verwendung der folgenden SQL-Anweisung:

```
CREATE DATABASE personnl           /*name of the database*/
       ON /persdb                  /*name of the database directory*/
       WITH "Personnel database"   /*comment*/
```

Der Systemadministrator stellt eine Verbindung zur Datenbank her und aktiviert diese für die Verwendung durch den Image Extender. Der Systemadministrator verwendet den db2ext-Befehlszeilenprozessor, um die folgenden Befehle einzugeben:

```
CONNECT TO personnl
ENABLE DATABASE FOR DB2IMAGE
```

Ereignis: Als Antwort auf den Befehl ENABLE DATABASE führt der Image Extender folgende Aktionen aus:

- Erstellen eines benutzerdefinierten Typs mit dem Namen DB2IMAGE für Abbildobjekte.
- Erstellen von Tabellen zur Verwaltungsunterstützung für Abbildobjekte.
- Erstellen von benutzerdefinierten Funktionen für Abbildobjekte. Die UDFs werden in Tabelle 4 aufgelistet.

Tabelle 4. Benutzerdefinierte Funktionen, die mit dem Image Extender erstellt werden

UDF-Name	Beschreibung
Comment	Abrufen oder Aktualisieren eines Benutzerkommentars
Content	Abrufen oder Aktualisieren des Inhalts eines Abbilds
DB2Image	Speichern des Inhalts eines Abbilds
Filename	Abrufen des Namens der Datei, die das Abbild enthält
Format	Abrufen des Abbildformats (z. B. GIF)
Height	Abrufen der Höhe eines Abbilds in Pixel
Importer	Abrufen der Benutzer-ID der Person, die ein Abbild importiert hat
ImportTime	Abrufen der Zeitmarke, wann ein Abbild importiert wurde
NumColors	Abrufen der Anzahl an Farben, die in einem Abbild verwendet werden
QbScoreFromName	Abrufen des Ähnlichkeitsergebnisses eines Abbilds (unter Verwendung einer benannten Abfrage)
QbScoreFromStr	Abrufen des Ähnlichkeitsergebnisses eines Abbilds (unter Verwendung einer Abfragezeichenfolge)
QbScoreTBFromName	Abrufen einer Tabelle mit Ähnlichkeitsergebnissen für eine Abbildspalte (unter Verwendung einer benannten Abfrage)
QbScoreTBFromStr	Abrufen einer Tabelle mit Ähnlichkeitsergebnissen für eine Abbildspalte (unter Verwendung einer Abfragezeichenfolge)
Replace	Aktualisieren des Inhalts und der Benutzerkommentare für ein Abbild
Size	Abrufen der Größe eines Abbilds in Byte
Thumbnail	Abrufen der Piktogrammversion eines Abbilds
Updater	Abrufen der Benutzer-ID der Person, die ein Abbild aktualisiert hat
UpdateTime	Abrufen der Zeitmarke, wann ein Abbild aktualisiert wurde
Width	Abrufen der Breite eines Abbilds in Pixel

Tabelle vorbereiten

Der Datenbankadministrator erstellt die Mitarbeitertabelle (employee) und aktiviert die Tabelle und die Spalte für die Bilder (picture) für die Verwendung durch den Image Extender.

Aktionen des Datenbankadministratoren: Aus praktischen Gründen fügt der Datenbankadministrator das Schema 'mmdbsys' im aktuellen Funktionspfad hinzu, indem er die folgende SQL-Anweisung verwendet:

```
SET CURRENT FUNCTION PATH = mmdbsys, CURRENT FUNCTION PATH
```

Dadurch ist es möglich, dass UDT- und UDF-Namen ohne das Präfix mit dem Schemanamen 'mmdbsys' angegeben werden. (Das Schema 'mmdbsys' muss nicht das erste Schema im Funktionspfad sein.) Weitere Informationen zu UDT- und UDF-Namen befinden sich im Abschnitt „UDF- und UDT-Namen" auf Seite 53. Der Datenbankadministrator erstellt die Tabelle 'employee'. Er verwendet den DB2-Befehlszeilenprozessor, um die folgende SQL-Anweisung einzugeben:

```
CREATE TABLE   employee            /*name of the table*/
       (id      CHAR(6)            /*employee identification*/
       name    VARCHAR(40)        /*employee name*/
       picture DB2IMAGE)          /*employee picture*/
```

Der Datenbankadministrator verwendet danach den db2ext-Befehlszeilenprozessor, um die folgenden Befehle einzugeben:

```
ENABLE TABLE employee FOR DB2IMAGE
ENABLE COLUMN employee picture FOR DB2IMAGE
```

Ereignis: Als Antwort auf den Befehl ENABLE TABLE führt der Image Extender folgende Aktionen aus:

- Identifizieren der Tabelle 'employee' für die Verwendung.
- Erstellen von Tabellen zur Verwaltungsunterstützung, die Informationen für Abbildobjekte in aktivierten Spalten enthalten.

Als Antwort auf den Befehl ENABLE COLUMN führt der Image Extender folgende Aktionen aus:

- Identifizieren der Spalte 'picture' für die Verwendung.
- Erstellen von Auslösern. Diese Auslöser aktualisieren verschiedene Tabellen zur Verwaltungsunterstützung als Reaktion auf Einfüge-, Aktualisier- und Löschoperationen in der Tabelle 'employee'.

Tabelle ändern

Der Datenbankadministrator fügt eine Audiospalte zur Mitarbeitertabelle hinzu und aktiviert diese für die Verwendung durch den Audio Extender.

Aktionen des Datenbankadministratoren: Der Datenbankadministrator verwendet den db2ext-Befehlszeilenprozessor, um die Personaldatenbank für die Verwendung durch den Audio Extender zu aktivieren:

```
ENABLE DATABASE FOR DB2AUDIO
```

Der Datenbankadministrator gibt anschließend die folgende SQL-Anweisung aus, um die Tabelle 'employee' zu ändern. Der Datenbankadministrator verwendet den DB2-Befehlszeilenprozessor zum Ausgeben der SQL-Anweisung.

```
ALTER TABLE employee              /*name of the table*/
      ADD voice DB2AUDIO          /*employee audio recording*/
```

Der Datenbankadministrator verwendet den db2ext-Befehlszeilenprozessor, um die Tabelle 'employee' und die Spalte 'voice' für die Verwendung durch den Audio Extender zu aktivieren:

```
ENABLE TABLE employee FOR DB2AUDIO
ENABLE COLUMN employee voice FOR DB2AUDIO
```

Ereignis: Als Antwort auf den Befehl ENABLE DATABASE führt der Audio Extender folgende Aktionen aus:

- Erstellen eines benutzerdefinierten Typs mit dem Namen DB2AUDIO für Audioobjekte.
- Erstellen von Tabellen zur Verwaltungsunterstützung für Audioobjekte.
- Erstellen von benutzerdefinierten Funktionen für Audioobjekte. Die UDFs werden in Tabelle 5 aufgelistet.

Tabelle 5. Benutzerdefinierte Funktionen, die mit dem Audio Extender erstellt werden

UDF-Name	Beschreibung
AlignValue	Abrufen der Byte pro Sample eines Tons
BitsPerSample	Abrufen der Anzahl an Bit, die zur Darstellung des Tons verwendet werden
BytesPerSec	Abrufen der Durchschnittsanzahl an Byte pro Sekunde des Tons
Comment	Abrufen oder Aktualisieren eines Benutzerkommentars
Content	Abrufen oder Aktualisieren des Inhalts eines Tons
DB2Audio	Speichern des Inhalts eines Tons
Duration	Abrufen der Spieldauer eines Tons
Filename	Abrufen des Namens der Datei, die den Ton enthält
FindInstrument	Abrufen der Nummer der Tonspur, auf der ein bestimmtes Instrument in einem Ton aufgezeichnet ist
FindTrackName	Abrufen der Spurnummer einer benannten Spur in einer Tonaufzeichnung
Format	Abrufen des Tonformats
GetInstruments	Abrufen der Namen der Instrumente, die in einem Ton aufgezeichnet sind
GetTrackNames	Abrufen der Spurnamen in einem Ton

Tabelle 5. Benutzerdefinierte Funktionen, die mit dem Audio Extender erstellt werden (Forts.)

UDF-Name	Beschreibung
Importer	Abrufen der Benutzer-ID der Person, die einen Ton importiert hat
ImportTime	Abrufen der Zeitmarke, wann ein Ton importiert wurde
NumAudioTracks	Abrufen der Anzahl der aufgezeichneten Spuren in einem Ton
NumChannels	Abrufen der Anzahl an Tonkanälen
Replace	Aktualisieren des Inhalts und der Benutzerkommentare für eine Tonaufzeichnung
SamplingRate	Abrufen der Abtastrate eines Tons
Size	Abrufen der Größe eines Tons in Byte
TicksPerQNote	Abrufen der Anzahl von Taktimpulsen pro Viertelnote, die bei der Aufzeichnung eines Tons verwendet wurde
TicksPerSec	Abrufen der Anzahl von Taktimpulsen pro Sekunde, die bei der Aufzeichnung eines Tons verwendet wurde
Updater	Abrufen der Benutzer-ID der Person, die einen Ton aktualisiert hat
UpdateTime	Abrufen der Zeitmarke, wann ein Ton aktualisiert wurde

Als Antwort auf den Befehl ENABLE TABLE führt der Audio Extender folgende Aktionen aus:

- Identifizieren der Tabelle 'employee' für die Verwendung.
- Erstellen von Tabellen zur Verwaltungsunterstützung, die Informationen für Audioobjekte in aktivierten Spalten enthalten.

Als Antwort auf den Befehl ENABLE COLUMN führt der Audio Extender folgende Aktionen aus:

- Identifizieren der Spalte 'voice' für die Verwendung.
- Erstellen von Auslösern. Diese Auslöser aktualisieren verschiedene Tabellen zur Verwaltungsunterstützung als Reaktion auf Einfüge-, Aktualisier- und Löschoperationen in der Tabelle 'employee'.

Daten in eine Tabelle einfügen

Ein Benutzer fügt einen Datensatz für Anita Jones zur Tabelle 'employee' hinzu. Der Datensatz enthält die Kennung von Anita Jones (128557), den Namen, das Bild und die Tonaufzeichnung. Das Quellenabbild und der Toninhalt befinden sich in Dateien auf dem Server. Das Abbild wird in der Tabelle als BLOB gespeichert, der Inhalt des Tons bleibt in der Serverdatei (der Tabelleneintrag verweist auf die Serverdatei).

Aktionen des Benutzers: Der Benutzer fügt den Datensatz in der Tabelle 'employee' ein, indem er ein Anwendungsprogramm verwendet, das die Anweisungen in Abb. 15 enthält.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvInt_Stor;
long hvExt_Stor;
  EXEC SQL END DECLARE SECTION;

hvInt_Stor = MMDB_STORAGE_TYPE_INTERNAL;
hvExt_Stor = MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
        '128557',                          /*id*/
        'Anita Jones',                     /*name*/
        DB2IMAGE(                          /*Image Extender UDF*/
           CURRENT SERVER,                 /*database server name in*/
                                           /CURRENT SERVER register*/
           '/employee/images/ajones.bmp'   /*image source file*/
           'ASIS',                         /*keep the image format*/
           :hvInt_Stor,                    /*store image in DB as BLOB*/
           'Anita''s picture'),            /*comment*/
        DB2AUDIO(                          /*Audio Extender UDF*/
           CURRENT SERVER,                 /*database server name in*/
                                           /*CURRENT SERVER register*/
           '/employee/sounds/ajones.wav',  /*audio source file*/
           'WAVE',                         /* audio format */
           :hvExt_Stor,                    /*retain content in server file*/
           'Anita''s voice')               /*comment*/
       );
```

Abbildung 15. Daten in eine Tabelle einfügen

Ereignis: Als Antwort auf die UDF DB2Image in der Anweisung INSERT führt der Image Extender folgende Aktionen aus:

- Lesen der Attribute des Abbilds, wie z. B. Höhe, Breite und Anzahl an Farben, aus der Kopfzeile des Quellenabbilds.
- Erstellen einer eindeutigen Kennung für das Abbild und Eintragen der folgenden Informationen in einer Tabelle zur Verwaltungsunterstützung:
  - Kennung für das Abbild
  - Eine Zeitmarke
  - Abbildgröße in Byte
  - Kommentar „Anita's picture" (Anitas Bild)
  - Inhalt des Abbilds

  Die Abbildquelle befindet sich in der Serverdatei mit dem Namen ajones.bmp. Der Inhalt der Datei wird in den Datensatz der Tabelle zur Verwaltungsunterstützung als BLOB eingefügt. Das Format des gespeicherten Abbilds entspricht dem Format des Quellenabbilds; es wird keine Formatumsetzung durchgeführt.
- Speichern eines Datensatzes in einer Tabelle zur Verwaltungsunterstützung. Der Datensatz enthält abbildspezifische Attribute, wie z. B. die Anzahl an Farben im Abbild, und die Piktogrammversion des Abbilds.

Als Antwort auf die UDF DB2Audio in der Anweisung INSERT führt der Audio Extender folgende Aktionen aus:

- Lesen der Attribute eines Tons, wie z. B. Anzahl an Tonspuren und -kanälen, aus der Kopfzeile des Tons.
- Erstellen einer eindeutigen Kennung für den Ton
- Speichern eines Datensatzes in einer Tabelle zur Verwaltungsunterstützung. Der Datensatz enthält:
  - Kennung für den Ton
  - Eine Zeitmarke
  - Tongröße in Byte
  - Kommentar „Anita's voice" (Anitas Stimme)

  Der Toninhalt befindet sich in einer Serverdatei mit dem Namen ajones.wav. Der Datensatz der Tabelle zur Verwaltungsunterstützung verweist auf die Datei.
- Speichern eines Datensatzes in einer anderen Tabelle zur Verwaltungsunterstützung. Der Datensatz enthält tonspezifische Attribute, wie z. B. die Abtastrate des Tons.

Auslöser fügen die Abbild- und Tonattributdaten in den verschiedenen Tabellen zur Verwaltungsunterstützung ein.

Daten aus einer Tabelle auswählen

Ein Benutzer ruft Informationen dazu auf, wann das Abbild und die Tonaufzeichnung von Robert Smith in der Tabelle 'employee' gespeichert wurden.

Aktionen des Benutzers: Der Benutzer ruft die Informationen unter Verwendung eines Anwendungsprogramms ab, das die SQL-Anweisungen in Abb. 16 enthält.

```
EXEC SQL BEGIN DECLARE SECTION;
char[255]  hvImg_Time;
char[255]  hvAud_Time;
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT IMPORTTIME(PICTURE),              /*when image was stored*/
       IMPORTTIME(VOICE)                          /*when audio was stored*/
       INTO :hvImg_Time, :hvAud_Time
  FROM EMPLOYEE
       WHERE NAME='Robert Smith';
```

Abbildung 16. Daten aus einer Tabelle auswählen

Ereignis: Als Antwort auf die UDF ImportTime für die Spalte PICTURE gibt der Image Extender eine Zeitmarke zurück, die das Datum und die Uhrzeit enthält, zu dem bzw. der das Abbild gespeichert wurde. Als Antwort auf die UDF ImportTime für die Spalte VOICE gibt der Audio Extender eine Zeitmarke zurück, die das Datum und die Uhrzeit enthält, zu dem bzw. der die Tonaufzeichnung gespeichert wurde.

Objekte anzeigen und wiedergeben

Ein Benutzer zeigt das Abbild von Robert Smith an und gibt die Tonaufzeichnung von Robert Smith wieder. Das Abbild ist in der Tabelle 'employee' als BLOB gespeichert. Der Inhalt für die Tonaufzeichnung befindet sich in einer Serverdatei.

Aktionen des Benutzers: Der Benutzer fügt den Datensatz in der Tabelle 'voice' ein, indem er ein Anwendungsprogramm verwendet, das die Anweisungen in Abb. 17 enthält.

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_hdl [251];
char hvAud_hdl [251];
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT PICTURE,                              /*Get image handle*/
                VOICE                                 /*Get audio handle*/
       INTO :hvImg_hdl, :hvAud_hdl
  FROM EMPLOYEE
       WHERE NAME='Robert Smith';

rc=DBiBrowse(
        NULL,                                         /*Use default image browser*/
        MMDB_PLAY_HANDLE,                             /*Use handle*/
        hvImg_hdl,                                    /*Image handle*/
        MMDB_PLAY_NO_WAIT);                            /*Run browser independently*/

rc=DBaPlay(
        NULL,                                         /*Use default audio player*/
        MMDB_PLAY_HANDLE,                             /*Use handle*/
        hvAud_hdl,                                    /*Audio handle*/
        MMDB_PLAY_WAIT);                              /*Wait for player to end*/
                                                      /*before continuing*/
```

Abbildung 17. Objekte anzeigen und wiedergeben

Ereignis: DB2 ruft die Kennung des Abbilds und der Tonaufzeichnung von Robert Smith ab. Danach, als Antwort auf die API DBiBrowse, ruft der Image Extender den Abbildinhalt ab, der der abgerufenen Abbildkennung zugeordnet ist. Der Image Extender ruft den Abbildinhalt aus der Datenbank ab und stellt ihn für die Anzeige durch einen Abbildbrowser in eine temporäre Clientdatei. Der Parameter NULL gibt an, dass der Standardabbildbrowser für das Benutzersystem verwendet wird. Der Browser wird unabhängig vom aufrufenden Programm ausgeführt, d. h., das aufrufenden Programm wartet mit der Fortsetzung seiner Verarbeitung nicht darauf, dass der Abbildbrowser seine Ausführung beendet.

Als Antwort auf die API DBaPlay ruft der Audio Extender den Dateinamen des Tons auf, der der abgerufenen Audiokennung zugeordnet ist, und übergibt den Dateinamen an die Audiowiedergabeeinheit. Der Parameter NULL gibt an, dass die Standardeinheit für die Audiowiedergabe für das Benutzersystem verwendet wird. Das aufrufende Programm wartet darauf, dass der Benutzer die Audiowiedergabeeinheit beendet, bevor es mit der Verarbeitung fortfährt.

Daten in einer Tabelle aktualisieren

Anita Jones ersetzt ihr Bild in der Tabelle 'employee' durch ein neueres Bild. Der Inhalt des neueres Bildes befindet sich in einer Serverdatei.

Aktionen des Benutzers: Der Benutzer ersetzt das Bild in der Tabelle 'employee', indem er ein Anwendungsprogramm verwendet, das die SQL-Anweisungen in Abb. 18 enthält.

```
EXEC SQL BEGIN DECLARE SECTION;
  char hvComment [16385];
long hvStorageType;
  EXEC SQL END DECLARE SECTION;

strcpy(hvComment, "Picture taken at Anita's promotion");
hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

EXEC SQL UPDATE EMPLOYEE
       SET PICTURE=REPLACE(
              PICTURE,                      /*image handle*/
              '/myimages/newone.bmp',       /*source image content*/
              'BMP',                        /*source format*/
              :hvStorageType,               /*store image in table as BLOB*/
              :hvComment)                   /*replace comment*/
       WHERE NAME='Anita Jones';
```

Abbildung 18. Daten in einer Tabelle aktualisieren

Ereignis: Als Antwort auf die UDF Replace in der Anweisung UPDATE liest der Image Extender die Attribute des neuen Abbilds. Der Image Extender verwendet die Attribute des neuen Abbilds, um die Attribute zu aktualisieren, die in den Tabellen zur Verwaltungsunterstützung für das alte Abbild gespeichert sind. Die Abbildquelle befindet sich in der Serverdatei mit dem Namen newone.bmp. Der Inhalt der Datei wird in den Datensatz der Tabelle zur Verwaltungsunterstützung als BLOB eingefügt, wodurch der BLOB-Inhalt des alten Abbilds ersetzt wird.

Auslöser ersetzen die Abbildattributdaten in den verschiedenen Tabellen zur Verwaltungsunterstützung.

Daten aus einer Tabelle löschen

Ein Benutzer löscht den Datensatz von Anita Jones aus der Tabelle 'employee'.

Aktionen des Benutzers: Der Benutzer löscht den Datensatz aus der Tabelle 'employee', indem er ein Anwendungsprogramm verwendet, das die folgende SQL-Anweisung enthält:

```
DELETE FROM EMPLOYEE
       WHERE NAME='Anita Jones';
```

Ereignis: Auslöser löschen die Einträge für Anita Jones aus den verschiedenen Tabellen zur Verwaltungsunterstützung.

Daten löschen

Teil 2. Abbild-, Audio- und Videodaten verwalten

Kapitel 5. Überblick zur Verwaltung

Dieses Kapitel bietet einen Überblick über die Verwaltungstasks, die beim Erstellen von Anwendungen ausgeführt werden, die die DB2 Extender verwenden.

Die DB2 Extender bieten die beiden folgenden Möglichkeiten zur Ausführung der meisten Verwaltungstasks:

- Anwendungsprogrammierschnittstellen (APIs). Sie können die DB2 Extender-APIs in Ihre C-Programme einfügen. Referenzinformationen zu diesen APIs befinden sich im Kapitel 14, „Anwendungsprogrammierschnittstellen", auf Seite 253.
- Verwaltungsbefehle. Sie können Verwaltungsbefehle an den db2ext-Befehlszeilenprozessor übergeben. Diese Befehle können nicht von der DB2-Befehlszeile aus ausgeführt werden. Kapitel 15, „Verwaltungsbefehle für den Client", auf Seite 449, enthält Anweisungen zur Eingabe von Verwaltungsbefehlen und weitere Referenzinformationen.

Mit den DB2 Extendern ausführbare Verwaltungstasks

Die Verwaltungstasks können in fünf Kategorien eingeteilt werden:

- Verwalten von Extender-Services. Die DB2 Extender werden auf eigenen Servern unter DB2 ausgeführt. Bevor Anwendungen Extender-Daten verwenden können, startet der Systemadministrator die Extender-Services und der Benutzer stellt eine Verbindung zu der Datenbank her, auf der die Extender-Daten gespeichert sind.
- Vorbereiten von Datenobjekten für Extender-Daten. Sie können Datenbanken, Tabellen und Spalten zur Speicherung von Extender-Daten vorbereiten, indem Sie sie aktivieren. Wenn Sie ein Datenobjekt aktivieren, erstellen und speichern die Extender Tabellen zur Verwaltungsunterstützung (sog. Metadatentabellen), um die Extender-Daten zu verwalten.
- **Nur EEE.** Neuverteilen von Extender-Daten in einer partitionierten Umgebung. Wenn Sie Partitionen in einer partitionierten Datenbank hinzufügen oder freigeben, können Sie die Daten neu verteilen, um die Vorteile der neuen Knotenkonfiguration zu nutzen.
- Überwachen von Datenobjekten und Multimediadateien. Während der Fehlerbehebung in Anwendungen, die mit den DB2 Extendern arbeiten, sollten Sie wissen, welche Datenobjekte für Extender-Daten aktiviert sind. Es ist außerdem sinnvoll, die Korrelation zwischen Benutzertabellen und externen Multimediadateien zu verstehen.
- Bereinigen von Tabellen zur Verwaltungsunterstützung. Beim Arbeiten mit den DB2 Extendern können sich veraltete Einträge in den Tabellen zur Verwaltungsunterstützung ansammeln. Durch das Löschen von veralteten Metadaten kann die Leistung verbessert und Speicherbereich zurückgefordert werden.

In Tabelle 6 auf Seite 80 werden alle Tasks aufgelistet, die bei der Verwaltung von Extender-Daten ausgeführt werden müssen. Sie enthält Angaben darüber, welche Tools zur Ausführung der einzelnen Task zur Verfügung stehen und wo Sie weitere Informationen finden können.

In der Spalte **Extender-API** stellt `x` das dritte Zeichen der einzelnen API-Anweisungen dar. Dieses Zeichen variiert je nach verwendetem Extender:

Zeichen	Extender
a	Audio
i	Image
v	Video

Beispielsweise ist DB**i**EnableTable die API für das Aktivieren einer Tabelle für Abbilddaten, DB**a**EnableTable die API für das Aktivieren einer Tabelle für Audiodaten und DB**v**EnableTable die API für das Aktivieren einer Tabelle für Videodaten. Der Wert `Keine` in der Spalte Extender-API bedeutet, dass keine Extender-API für die Task zur Verfügung steht. Der Wert `Keine` in der Spalte Extender-Befehl bedeutet, dass kein Extender-Befehl für die Task zur Verfügung steht.

Für QBIC sind zusätzliche Verwaltungstasks erforderlich: Wenn Sie die Image Extender-Funktion QBIC (Query by Image Content; Abfrage anhand des Abbildinhalts) verwenden wollen, müssen Sie zusätzliche Verwaltungstasks, wie z. B. das Erstellen eines QBIC-Katalogs, ausführen. Weitere Informationen zu diesen Tasks befinden sich in Kapitel 12, „Abfragen von Abbildern nach Inhalt", auf Seite 153.

Tabelle 6. Verwaltungstasks und -funktionen für die DB2 Extender

Task	Extender-API	Extender-Befehl	Siehe
Verwalten von Extender-Services			
Extender-Services starten	Keine	DMBSTART	S. 3
Status der Extender-Services abrufen	Keine	DMBSTAT	S. 5
Extender-Services stoppen	Keine	DMBSTOP	S. 5
Verbindung zu einer Datenbank herstellen	Keine	CONNECT	S. 3
Extender-Service für Ihre Datenbank starten	Keine	START SERVER	S. 5
Status eines Extender-Services für Ihre Datenbank abrufen	Keine	GET SERVER STATUS	S. 5
Extender-Service für Ihre Datenbank stoppen	Keine	STOP SERVER	S. 5
Extender-Exemplare erstellen und verwalten	Keine	DMBICRT, DMBILIST, DMBIDROP, DMBIMIGR	S. 6
Datenobjekte für Multimedia vorbereiten			
Datenbank aktivieren	DBxEnableDatabase	ENABLE DATABASE	S. 83
Datenbank inaktivieren	DBxDisableDatabase	DISABLE DATABASE	S. 89
Tabelle aktivieren	DBxEnableTable	ENABLE TABLE	S. 86
Tabelle inaktivieren	DBxDisableTable	DISABLE TABLE	S. 89
Spalte aktivieren	DBxEnableColumn	ENABLE COLUMN	S. 88
Spalte inaktivieren	DBxDisableColumn	DISABLE COLUMN	S. 89
Neuverteilen von Extender-Daten in einer partitionierten Umgebung (nur EEE)			
Extender-Daten anhand einer neuen Knotengruppenkonfiguration neu verteilen	DMBRedistribute	REDISTRIBUTE NODEGROUP	S. 18

Tabelle 6. Verwaltungstasks und -funktionen für die DB2 Extender (Forts.)

Task	Extender-API	Extender-Befehl	Siehe
Überwachen von Datenobjekten und Multimediadateien			
Ermitteln, ob Datenbanken aktiviert sind	DBxIsDatabaseEnabled	GET EXTENDER STATUS	S. 92
Ermitteln, ob Tabellen aktiviert sind	DBxIsTableEnabled	GET EXTENDER STATUS	S. 92
Ermitteln, ob Spalten aktiviert sind	DBxIsColumnEnabled	GET EXTENDER STATUS	S. 92
Tabelleneinträge suchen, die auf Dateien in Tabellen verweisen, deren Qualifikationsmerkmal die aktuelle Benutzer-ID ist	DBxIsFileReferenced	Keine	S. 93
Tabelleneinträge suchen, die auf Dateien in allen Tabellen eines bestimmten Qualifikationsmerkmals oder in allen Tabellen in einer Datenbank verweisen	DBxAdminIsFileReferenced	Keine	S. 93
Dateien suchen, auf die durch Tabelleneinträge in Tabellen verwiesen wird, deren Qualifikationsmerkmal die aktuelle Benutzer-ID ist	DBxGetReferencedFiles	GET REFERENCED FILES	S. 94
Dateien suchen, auf die durch Tabelleneinträge in allen Tabellen eines bestimmten Qualifikationsmerkmals oder in allen Tabellen in einer Datenbank verwiesen wird	DBxAdminGetReferencedFiles	GET REFERENCED FILES	S. 94
Unzugängliche Dateien suchen, auf die durch Tabelleneinträge in allen Tabellen verwiesen wird, deren Qualifikationsmerkmal die aktuelle Benutzer-ID ist	DBxGetInaccessibleFiles	GET INACCESSIBLE FILES	S. 95
Unzugängliche Dateien suchen, auf die durch Tabelleneinträge in allen Tabellen eines bestimmten Qualifikationsmerkmals oder in allen Tabellen in einer Datenbank verwiesen wird	DBxAdminGetInaccessibleFiles	GET INACCESSIBLE FILES	S. 95

Tabelle 6. Verwaltungstasks und -funktionen für die DB2 Extender (Forts.)

Task	Extender-API	Extender-Befehl	Siehe
Bereinigen von Tabellen zur Verwaltungsunterstützung (Metadatentabellen)			
Metadatentabellen für eine bestimmte Benutzertabelle oder alle Benutzertabellen bereinigen, deren Qualifikationsmerkmal die aktuelle Benutzer-ID ist	DBxReorgMetadata	REORG	S. 8
Metadatentabellen für alle Benutzertabellen mit einem bestimmten Qualifikationsmerkmal oder alle Benutzertabellen in einer Datenbank bereinigen	DBxAdminReorgMetadata	REORG	S. 8

Reihenfolge von Verwaltungstasks: In der folgenden Liste wird eine geordnete Übersicht der Verwaltungstasks gezeigt, die Sie bei der ersten Verwendung der Extender ausführen. Zum Ausführen einiger Tasks verwenden Sie DB2-Befehle oder -Anweisungen. Andere Tasks führen Sie mit den DB2 Extendern aus. Bei der Reihenfolge wird davon ausgegangen, dass Ihr DB2-System aktiv ist.

Erforderliche Tasks:

1. Die Extender-Services starten.
2. Eine Datenbank erstellen (mit Hilfe von DB2).
3. Die Verbindung zur Datenbank database server herstellen.
4. Die Datenbank aktivieren.
5. Eine Tabelle und Spalte erstellen (mit Hilfe von DB2).
6. Eine Tabelle in der Datenbank aktivieren.
7. Eine Spalte in der Tabelle aktivieren.

Wahlfreie Tasks:

1. Datenobjekte und Multimediadateien überwachen.
2. Den Funktionspfad setzen (mit Hilfe von DB2).
3. Tabellen zur Verwaltungsunterstützung bereinigen.

Beispiele: Bei den meisten Beispielen in den folgenden fünf Kapiteln wird davon ausgegangen, dass die Tasks von einem Systemadministratoren (SYSADM) oder Datenbankadministratoren (DBA) ausgeführt werden. Für einige Tasks ist keine Berechtigung DBA oder SYSADM erforderlich.

In den Beispielen wird davon ausgegangen, dass der DBA das Schema MMDBSYS im aktuellen Funktionspfad hinzugefügt hat. Dadurch kann er UDT-Namen angeben, ohne als Präfix den Schemanamen MMDBSYS anzugeben. Weitere Informationen zu UDT-Namen befinden sich im Abschnitt „UDF- und UDT-Namen" auf Seite 53.

Viele der API-Beispiele in diesem Abschnitt basieren auf dem Beispielanwendungscode, der mit den Extendern geliefert wird. Der Beispielcode befindet sich im Unterverzeichnis SAMPLES auf dem Client.

Kapitel 6. Datenobjekte für Extender-Daten vorbereiten

Sie können Datenbanken, Tabellen und Spalten zur Speicherung von Extender-Daten vorbereiten, indem Sie sie aktivieren. Als Erstes aktivieren Sie die Datenbank, danach eine Tabelle in der Datenbank. Als Letztes aktivieren Sie eine Spalte in der Tabelle.

Sollen nicht länger Extender-Daten in Ihren Datenobjekten enthalten sein, können Sie die Objekte inaktivieren.

Sie können Objekte entweder unter Verwendung der APIs in Ihrem C-Programm oder von der db2ext-Befehlszeile aus aktivieren und inaktivieren. In diesem Kapitel werden Beispiele für beide Methoden gezeigt.

Datenbanken aktivieren

Verwenden Sie die API DBxEnableDatabase (wobei `x` den Wert `a` für Audio, `i` für Abbild (Image) oder `v` für Video hat) oder den Befehl ENABLE DATABASE, um eine Datenbank für einen DB2 Extender zu aktivieren.

Wenn Sie eine Datenbank aktivieren, führt der Extender folgende Aktionen aus:

- Erstellen eines benutzerdefinierten Typs (UDT) DB2`xxxxx` für Ihre Datenobjekte, wobei `xxxxx` entweder für `Image`, `Audio` oder `Video` steht. Die UDT wird zum Definieren einer Spalte in der Benutzertabelle verwendet, die Kennungen für Objekte dieses Typs hält.
- Erstellen von Tabellen zur Verwaltungsunterstützung (sog. Metadatentabellen) für die Datenbank. Bei diesen Tabellen handelt es sich nicht um Benutzertabellen (d. h., Tabellen, in denen Benutzer Geschäftsdaten speichern). Die Extender verwenden sie, um Extender-Daten zu verwalten. Sie dürfen nicht manuell editiert werden.
- Erstellen von benutzerdefinierten Funktionen (UDFs), die dem Extender zugeordnet sind. Die UDFs werden in „Benutzerdefinierte Funktionen" auf Seite 191 aufgelistet.

Beim Aktivieren einer Datenbank müssen Sie außerdem Tabellenbereiche angeben, die die Tabellen zur Verwaltungsunterstützung (und deren Indizes) für die Datenbank halten sollen. Für einen oder mehrere der Tabellenbereiche kann ein Nullwert angegeben werden. In diesem Fall wird ein Standardtabellenbereich verwendet.

Zum Aktivieren einer Datenbank benötigen Sie Datenbankadministratorberechtigung.

Nur EEE: Beim Aktivieren einer Datenbank für einen Extender in einer partitionierten Umgebung sollte der angegebene Tabellenbereich in einer Knotengruppe definiert sein, die alle Knoten im partitionierten Datenbanksystem umfasst. Außerdem sollte sich der Tabellenbereich in derselben Knotengruppe befinden wie die Benutzertabelle.

Beispiele

In den folgenden Beispielen wird eine Datenbank zur Speicherung von Abbilddaten aktiviert, wobei der Standardtabellenbereich verwendet wird.

Verwendung der API: Der Code in Abb. 19 stellt eine Verbindung zu einer bestehenden Datenbank her, bevor sie aktiviert wird. Dieses Beispiel ist unter Verwendung der DB2 Call Level Interface geschrieben. Es enthält unter anderem Definitions- und Fehlerprüfcode. Das vollständige Beispielprogramm befindet sich in der Datei ENABLE.C im Unterverzeichnis SAMPLES.

```
/*---- Set-up -----------------------------------------------*/
#include <stdio.h>
#include <stlib.h>
#include <string.h>
#include "dmbimage.h"       /* image extender function prototypes (DBi) */
#include "utility.h"        /* utility functions */

#define MMDB_ERROR_MSG_TEXT_LEN         1000
#define SERVER_IS_DB2390  (strcmp(dbms, "DB2")==0 || strcmp(dmbs, "DSN06010")==0)

int
main(int argc, char *argv[])
{
        SQLHENV henv = SQL_NULL_HENV;
        SQLHDBC hdbc = SQL_NULL_HDBC;
        SQLHSTMT hstmt = SQL_NULL_HSTMT;
        SQLCHAR uid[18+1];
        SQLCHAR pwd[30+1];
        SQLCHAR dbname[SQL_MAX_DSN_LENGTH+1];
        SQLCHAR buffer[500];
        SQL SMALLINT dbms_sz = 0;
        char    dbms[20];

        SQLRETURN rc = SQL_SUCCESS;
        SQLINTEGER sqlcode = 0;
        char errorMsgText[MMDB_ERROR_MSG_TEXT_LEN+1];
        char *program = "enable;
        char *step;
```

Abbildung 19. Beispielcode zur Aktivierung einer Datenbank (Teil 1 von 3)

```
/*---- Prompt for database name, userid, and password ----*/
        if (argc > 5) ||  (argc >=2 && strcmp(argv[1],"?")== 0))
          {
            printf("Syntax for enable - enabling a DB2 UDB database: \n"
                   "   enable database_name userid password\n");
      exit(0);
          }

        if (argc == 4) {
          strcpy((char *)dbname, argv[1]);
          strcpy((char *)uid   , argv[2]);
          strcpy((char *)pwd   , argv[3]);
          }
        else {
          printf("Enter database name:\n");
  gets((char *) dbName);
          printf("Enter userid:\n");
  gets((char *) uid);
          printf("Enter password:\n");
  gets((char *) pwd);
          }
/*----- connect to the database ------------------------------------------------*/
         rc = cliInitialize(&henv, &hdbc, dbname, uid, pwd);
         cliCheckError(henv, hdbc, SQL_NULL_HSTMT, rc);
         if (rc < 0) goto SERROR;

/*----- find out if application is connected to DB2/UDB or DB2/390?------------*/
         rc = SQLGetInfo(hdbc, SQL_DBMS_NAME, (SQLPOINTER) &dbms,
                         sizeof(dbms), &dbms_sz);
         cliCheckError(henv, hdbc, SQL_NULL_HSTMT, rc);
         if (rc < 0) goto SERROR;
```

Abbildung 19. Beispielcode zur Aktivierung einer Datenbank (Teil 2 von 3)

```
/****** enable server for image extender ***************************************/
     if (!SERVER_IS_DB2390)
           {
            printf("%s: Enabling database......\n", program);
           }
         printf("%s: This may take a few minutes, please wait......\n", program);

     if (!SERVER_IS_DB2390)
           {
            step="DBiEnableDatabase with NULL tablespace"
            rc=DBiEnableDatabase(NULL);
           }
     if (rc < 0) {
                printf("%s: %s failed!\n", program, step);
   printMsg(rc);
            DBiGetError(&sqlcode, errorMsgText);
           if (sqlcode)
             printf("sqlcode=%i, ",sqlcode);
         }else if (rc > 0) {
           printf("%s: %s, warning detected.\n", program, step);
   printMsg(rc);
            DBiGetError(&sqlcode, errorMsgText);
            printf("warning MsgText=%s\n", errorMsgText);
     } else
           printf("%s: %s OK\n", program, step);
/****** end of enable server ****************************************************/
```

Abbildung 19. Beispielcode zur Aktivierung einer Datenbank (Teil 3 von 3)

Verwendung der db2ext-Befehlszeile: In diesem Beispiel besteht bereits eine Verbindung zur Datenbank.

```
enable database for db2image
```

Tabellen aktivieren

Verwenden Sie die API DBxEnableTable (wobei `x` den Wert `a` für Audio, `i` für Abbild (Image) oder `v` für Video hat) oder den Befehl ENABLE TABLE, um eine Tabelle für einen DB2 Extender zu aktivieren.

Beim Aktivieren einer Benutzertabelle müssen Sie außerdem Tabellenbereiche angeben, die die zugehörigen Tabellen zur Verwaltungsunterstützung (und deren Indizes) halten sollen. Für einen oder mehrere der Tabellenbereiche kann ein Nullwert angegeben werden. In diesem Fall wird ein Standardtabellenbereich verwendet.

Nur EEE: Beim Aktivieren einer Tabelle für einen Extender in einer partitionierten Umgebung sollte der angegebene Tabellenbereich in einer Knotengruppe definiert sein, die alle Knoten im partitionierten Datenbanksystem umfasst. Außerdem muss sich der Tabellenbereich in derselben Knotengruppe befinden wie die Benutzertabelle.

Nur EEE: Sie können eine DB2 Extender-Spalte nicht als Partitionierungsschlüsselspalte in einer Umgebung für partitionierte Datenbanken verwenden.

Sie benötigen Steuerungs- oder Änderungsberechtigung für die Benutzertabelle. Die Datenbank muss aktiviert sein, bevor Sie eine Tabelle darin aktivieren.

In den folgenden Beispielen wird eine Tabelle zur Speicherung von Abbilddaten aktiviert, wobei der Standardtabellenbereich verwendet wird. Die Datenbank ist bereits aktiviert.

Verwendung der API: In Abb. 20 auf Seite 87, wird dargestellt, wie Sie mit dem Code vor der Aktivierung die Tabelle erstellen und Änderungen festschreiben können. Das Beispiel enthält unter anderem Fehlerprüfcode. Das vollständige Beispielprogramm befindet sich in der Datei ENABLE.C im Unterverzeichnis SAMPLES.

```
        SQLCHAR szCreate_DB2UDB[]="CREATE TABLE %s(%s mmdbsys.DB2Image,
        %s mmdbsys.DB2Video, %s mmdbsys.DB2Audio, artist varchar(25), title varchar(25)
        stock_no char(11), tw char(10), price char(10))";

        SQLRETURN rc = SQL_SUCCESS;
        SQLINTEGER sqlcode = 0;
        char errorMsgText[MMDB_ERROR_MSG_TEXT_LEN+1];
        char tableName[8+18+1] = "sobay_catalog";
        char audioColumn[18+1] = "music";
        char imageColumn[18+1] = "covers";
        char videoColumn[18+1] = "video";
        char *program = "enable";
        char *step;

/*-----create table ------------------------------------------------*/
       printf("%s: Creating table ......\n", program);
     if (!SERVER_IS_DB2390)
         sprintf((char*) buffer, (char*) szCreate_DB2UDB,
                   tableName, imageColumn, videoColumn, audioColumn):

       rc = SQLAllocStmt(hdbc, &hstmt);
       cliCheckError(SQL_NULL_HENV, hdbc, SQL_NULL_HSTMT, rc);
       rc = SQLExecDirect(hstmt, buffer, SQL_NTS);
         cliCheckError(SQL_NULL_HENV, SQL_NULL_HDBC, hstmt, rc);

/*---- enable table for image extender ---------------------------------*/
        printf("%s: Enabling table......\n", program);
     step="DBiEnableTable";
     if (!SERVER_IS_DB2390)
         rc = DBiEnableTable(NULL,tableName);
           }
     if (rc < 0) {
                printf("%s: %s failed!\n", program, step);
   printMsg(rc);
           DBiGetError(&sqlcode, errorMsgText);
           if (sqlcode)
                  printf("sqlcode=%i, "sqlcode");
                printf("errorMsgText=%s\n", errorMsgText);
    } else if (rc > 0) {
           printf("%s: %s, warning detected.\n", program, step);
   printMsg(rc);
           DBiGetError(&sqlcode, errorMsgText);
           printf("warningMsgText=%s\n", errorMsgText);
     } else
                printf("%s: %s OK\n", program, step)
/*---- end of enable table ---------------------------------------------*/
```

Abbildung 20. Beispielcode zur Aktivierung einer Tabelle

Verwendung der db2ext-Befehlszeile: In diesem Beispiel besteht die Tabelle bereits, und die Datenbank ist aktiviert.

```
enable table employee for db2image
```

Spalten aktivieren

Verwenden Sie die API DBxEnableColumn API (wobei x den Wert a für Audio, i für Abbild (Image) oder v für Video hat) oder den Befehl ENABLE COLUMN, um eine Spalte für einen DB2 Extender zu aktivieren. Wenn Sie die API oder den Befehl ausgeben, geben Sie die entsprechende Tabelle und Spalte an.

Wenn Sie eine Spalte aktivieren, fügt der Extender Informationen zu den Tabellen zur Verwaltungsunterstützung hinzu, die zur Benutzertabelle gehören. Sie benötigen Steuerungs- oder Änderungsberechtigung für die Benutzertabelle, in der sich die Spalte befindet. Sowohl die Datenbank als auch die Tabelle müssen aktiviert sein, bevor Sie die Spalte aktivieren.

In den folgenden Beispielen wird die Spalte PICTURE in der Tabelle EMPLOYEE zur Speicherung von Abbilddaten aktiviert. Die Datenbank und die Tabelle sind bereits aktiviert.

Verwendung der API: Dieses Beispiel enthält unter anderem Fehlerprüfcode. Das vollständige Beispielprogramm befindet sich in der Datei ENABLE.C im Unterverzeichnis SAMPLES.

```
char imageColumn[18+1] = "covers";

/*---- enable column for image extender ----*/
     printf("%s: Enabling columns......\n", program);
     step="DBiEnableColumn";
     rc = DBiEnableColumn(tableName, imageColumn);
     if (rc < 0) {
                 printf("%s: %s failed!\n", program, step);
   printMsg(rc);
            DBiGetError(&sqlcode, errorMsgText);
            if (sqlcode)
               printf("sqlcode=%i, ", sqlcode);
            printf("errorMsgText=%s\n", errorMsgText)

    } else if (rc > 0) {
            printf("%s: %s, warning detected.\n", program, step);
   printMsg(rc);
            DBiGetError(&sqlcode, errorMsgText);
            printf("warningMsgText=%s\n", errorMsgText);
     } else
            printf("%s: %s OK\n", program, step);
/*---- enable column for image extender ----*/
```

Abbildung 21. Beispielcode zur Aktivierung einer Spalte

Verwendung der db2ext-Befehlszeile: In diesem Beispiel besteht die Spalte bereits und die Datenbank und die Tabelle sind aktiviert.

```
enable column employee picture for db2image
```

Datenobjekte inaktivieren

Wenn Sie Extender-Daten aus einer Datenbank, Tabelle oder Spalte löschen, ist es nicht länger erforderlich, dass diese aktiviert ist.Sie haben zwei Möglichkeiten, Datenobjekte zu inaktivieren: mit den DISABLE-Befehlen und den APIs. Weitere Informationen zu Extender-Befehlen befinden sich im Kapitel 15, „Verwaltungsbefehle für den Client", auf Seite 449. Weitere Informationen zu den Extender-APIs befinden sich im Kapitel 14, „Anwendungsprogrammierschnittstellen", auf Seite 253.

Vor dem Freigeben einer Tabelle oder Datenbank, die Extender-Daten enthält, müssen Sie sie inaktivieren und den Server für diese Datenbank stoppen.

Inaktivieren

Kapitel 7. Datenobjekte und Multimediadateien überwachen

Während des Erstellens von und der Fehlerbehebung in Anwendungen, die mit den DB2 Extendern arbeiten, sollten Sie wissen, welche Datenobjekte für Extender-Daten aktiviert sind. Wenn Sie beispielsweise feststellen können, dass eine bestimmte Tabelle für Abbilddateien aktiviert ist, kann Ihre Anwendung erfolgreich Abbilddateien in dieser Tabelle speichern.

Es ist außerdem sinnvoll, die Korrelation zwischen Benutzertabellen und externen Multimediadateien zu verstehen, beispielsweise, welche Tabellen auf eine bestimmte Datei verweisen oder auf welche Dateien von einer bestimmten Tabelle verwiesen wird. Es ist außerdem sinnvoll festzustellen, ob Ihre Tabellen auf Dateien verweisen, die auf dem System nicht mehr existieren.

Sie benötigen die entsprechenden Berechtigungen: Sie müssen Zugriff auf eine Tabelle haben, um Daten in der Tabelle verfolgen zu können. Wenn Sie eine umfassende Überwachungsoperation durchführen wollen, z. B. Ermitteln, welche Einträge in allen Benutzertabellen in der Datenbank auf eine Datei verweisen, benötigen Sie die Berechtigung SYSADM bzw. DBADM oder die Berechtigung SELECT für aktivierte Spalten in allen durchsuchten Benutzertabellen und den zugehörigen Tabellen zur Verwaltungsunterstützung. Wenn Sie keinen Zugriff auf alle Tabellen haben, geben die Extender nur Überwachungsinformationen für die Tabellen zurück, auf die Sie zugreifen können. Außerdem geben sie einen Code zurück, der angibt, dass Sie auf einige der erforderlichen Tabellen keine Zugriffsberechtigung haben.

Status von Datenobjekten prüfen

Sie können überprüfen, ob Datenbanken, Tabellen und Spalten zur Speicherung von Extender-Daten aktiviert sind. Im folgenden Beispiel wird festgestellt, ob die aktuelle Datenbank für den Image Extender aktiviert ist. Zur Datenbank besteht bereits eine Verbindung. Das vollständige Beispielprogramm befindet sich in der Datei API.C im Unterverzeichnis SAMPLES.

Verwendung der API: Der Beispielcode in Abb. 22 enthält Fehlerprüfcode.

```
/*---- Query the database using DBiIsDatabaseEnabled API. ----------*/
step="DBiIsDatabaseEnabled API";
rc = DBiIsDatabaseEnabled(&status);
if (rc < 0) {
        printf("%s: %s FAILED!\n", argv[0], step);
   printMsg(rc);
        DBiGetError(&sqlcode, errorMsgText);
        printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
        fail = TRUE;
} else if (rc > 0) {
        printf("%s: %s, warning detected.\n", argv[0], step);
   printMsg(rc);
        DBiGetError(&sqlcode, errorMsgText);
        printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
} else {
     if (status == 1) {
         printf("%s: \"%s\" database is enabled for Image Extender\n",
                argv[0], dbName);
        printf("%s: %s PASSED\n\n", argv[0], step);
     } else if (status == 0) {
         printf("%s: \"%s\" database is not enabled for Image Extender\n",
                argv[0], dbName);
        printf("%s: %s PASSED\n\n", argv[0], step);
     } else
         printf("%s: %s FAILED, invalid status!\n", argv[0], step);
}
```

Abbildung 22. Beispielcode zum Überprüfen der Datenbankaktivierung

Verwendung der db2ext-Befehlszeile:

```
get extender status
```

Das Überprüfen des Status von Benutzertabellen und Spalten ähnelt dem Überprüfen des Status einer Datenbank. Verwenden Sie die APIs DBxIsTableEnabled und DBxIsColumnEnabled oder den Befehl GET EXTENDER STATUS.

Tabelleneinträge suchen, die auf Dateien verweisen

Sie können überprüfen, welche Einträge in Benutzertabellen auf eine externe Multimediadatei verweisen. Verwenden Sie die API DBxAdminIsFileReferenced, um zu überprüfen, welche Einträge in allen Benutzertabellen oder einer Untergruppe von Benutzertabellen in der aktuellen Datenbank auf eine externe Multimediadatei verweisen. Verwenden Sie die API DBxIsFileReferenced, um zu überprüfen, welche Einträge in einer bestimmten Benutzertabelle auf eine externe Multimediadatei verweisen.

Verwendung der API: Der Beispielcode in Abb. 23 gibt zurück, wie oft und wo auf eine Datei verwiesen wird. Er enthält unter anderem Fehlerprüfcode. Das vollständige Beispielprogramm befindet sich in der Datei API.C im Unterverzeichnis SAMPLES.

```
/*---- Query the database using DBiAdminIsFileReferenced API. ------*/
step="DBiAdminIsFileReferenced API";
rc = DBiAdminIsFileReferenced((char*) uid, filename, &count, &filelist);
if (rc < 0) {
   printf("%s: %s FAILED!\n", program, step);
   printMsg(rc);
        DBiGetError(&sqlcode, errorMsgText);
        printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
} else if (rc > 0) {
   printf("%s: %s, warning detected.\n", program, step);
   printMsg(rc);
        DBiGetError(&sqlcode, errorMsgText);
        printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
} else {
   if (count == 0)
         printf("%s: \"%s\" file is not referenced\n",
                program, filename);
     else {
         printf("%s: \"%s\" file is referenced %d times\n",
                program, filename);
     for (i=0; i < count; i++)
            {
                /* filename is NULL for any IsFileReferenced APIs */

   printf ("filename = %s\n", filelist[i].filename);
   printf ("\tqualifier = %s\n", filelist[i].tqualifier);
   printf ("\ttable = %s\n", filelist[i].tname);
   printf ("\thandle = %s\n", filelist[i].handle);
   printf ("\tcolumn = %s\n", filelist[i].column);
       if (filelist[i].filename)
              free (filelist[i].filename);
           }
         }
   if (filelist)
       free (filelist);
        printf("%s: %s PASSED\n\n", argv[0], step);
}
```

Abbildung 23. Beispielcode zum Überprüfen des Verweises von Benutzertabellen auf eine Datei

Dateien suchen, auf die durch Tabelleneinträge verwiesen wird

Verwenden Sie die API DBxAdminGetReferencedFiles oder den Befehl GET REFERENCED FILES, um die externen Multimediadateien aufzulisten, auf die von allen Benutzertabellen oder einer Untergruppe von Benutzertabellen in der aktuellen Datenbank verwiesen wird. Verwenden Sie die API DBxGetReferencedFiles oder den Befehl GET REFERENCED FILES, um die externen Multimediadateien aufzulisten, auf die in einer bestimmten Tabelle verwiesen wird.

Verwendung der API: Der Beispielcode in Abb. 24 gibt die Anzahl der gefundenen Dateien und eine Liste der Dateien zurück. Das vollständige Beispielprogramm befindet sich in der Datei API.C im Unterverzeichnis SAMPLES.

```
/*---- Query the database using DBiAdminGetReferencedFiles API. ----------*/
step="DBiAdminGetReferencedFilesAPI"
rc = DBiAdminGetReferencedFiles((char*) uid, &count, &filelist);
if (rc < 0) {
   printf("%s: %s FAILED!\n", program, step);
   printMsg(rc);
        DBiGetError(&sqlcode, errorMsgText);
   printf{"sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
} else if (rc > 0) {
   printf("%s: %s, warning detected.\n", program, step);
   printMsg(rc);
        DBiGetError(&sqlcode, errorMsgText);
        printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
} else {
   if (count == 0)
     printf("%s: no referenced files\n", program);
  else {
     printf("%s: %d referenced files\n", program, count);
     for (i=0; i < count; i++)
       {
   printf ("filename = %s\n", filelist[i].filename);
   printf ("\tqualifier = %s\n", filelist[i].tqualifier);
   printf ("\ttable = %s\n", filelist[i].tname);
   printf ("\thandle = %s\n", filelist[i].handle);
   printf ("\tcolumn = %s\n", filelist[i].column);
       if (filelist[i].filename)
               free (filelist[i].filename);
     }
   }
   if (filelist)
       free (filelist);
        printf("%s: %s PASSED\n\n", argv[0], step);
}
```

Abbildung 24. Beispielcode zum Abrufen einer Liste von Dateien mit Verweisen

Verwendung der db2ext-Befehlszeile:

```
get referenced files user anitas for db2image
```

Überprüfen, ob Multimediadateien existieren

Angenommen, ein Benutzer löscht eine Multimediadatei aus dem System, aktualisiert jedoch die Benutzertabelle, die auf diese Datei verweist, nicht. Unter diesen Umständen möchten Sie möglicherweise alle Multimediadateien auflisten, auf die in Ihren Benutzertabellen verwiesen wird, auf die aber nicht zuzugriffen werden kann.

Verwenden Sie die API DBxAdminGetInaccessibleFiles oder den Befehl GET INACCESSIBLE FILES, um die nicht zugänglichen Multimediadateien aufzulisten, auf die von allen Benutzertabellen oder einer Untergruppe von Benutzertabellen in der aktuellen Datenbank verwiesen wird. Verwenden Sie die API DBxGetInaccessibleFiles oder den Befehl GET INACCESSIBLE FILES, um die nicht zugänglichen Multimediadateien aufzulisten, auf die von einer bestimmten Tabelle verwiesen wird.

Auf nicht zugängliche Mediadateien prüfen

Kapitel 8. Berechtigungen für Tabellen zur Verwaltungsunterstützung erteilen und widerrufen

Benutzer verwenden UDFs, um Abbild-, Audio- und Videoobjekte in einer Benutzertabelle auszuwählen, einzufügen, zu aktualisieren oder zu löschen. Um die angeforderten Operationen auszuführen, müssen die UDFs auf die Tabellen zur Verwaltungsunterstützung, die die Attributinformationen für die Objekte enthalten, zugreifen und gegebenenfalls Einfüge-, Aktualisierungs- und Löschoperationen in diesen Tabellen ausführen können. Für den Eigner einer Benutzertabelle erteilen die Extender den UDFs automatisch die Berechtigung, die zur Ausführung der angeforderten Operation benötigt wird. Andere Benutzer als der Tabelleneigner, die ein Objekt in der Benutzertabelle auswählen sollen, müssen die entsprechende Berechtigung für die Tabellen zur Verwaltungsunterstützung erhalten.

Darüber hinaus müssen auch Benutzer, die QBIC-Operationen in Abbildobjekten in einer Benutzertabelle ausführen, über die erforderlichen Berechtigungen für die Tabellen zur Verwaltungsunterstützung verfügen, die im QBIC-Katalog für diese Objekte enthalten sind. Ein Benutzer, der z. B. eine QBICC-Abfrage für eine Spalte von Abbildern ausgibt, muss über die Berechtigung SELECT für die QBIC-Katalogtabellen der Abbildspalte verfügen. Ein Benutzer, der Änderungen am QBIC-Katalog vornimmt, muss über die Berechtigungen SELECT, INSERT, UPDATE und DELETE für die zugehörigen QBIC-Katalogtabellen verfügen.

Der Eigner einer Benutzertabelle oder ein Datenbankadministrator (mit Berechtigung GRANT) für die Datenbank kann den DB2 Extender-Befehl GRANT verwenden, um Berechtigungen für die Tabellen zur Verwaltungsunterstützung zu vergeben. Bei Eingabe des Befehls GRANT müssen Sie die folgenden Informationen angeben:

- Erforderliche Berechtigung, z. B. SELECT oder UPDATE.
- Name des Extenders: DB2IMAGE, DB2AUDIO oder DB2VIDEO. Sie können für alle drei Extender auch ALL angeben.
- Name der Benutzertabelle.
- ID des Benutzers. Sie können der Benutzer-ID das optionale Schlüsselwort USER voranstellen. Außerdem können Sie für alle Benutzer den Wert PUBLIC angeben.

Wenn Sie SELECT angeben, gewähren Sie den angegebenen Benutzern für die Tabellen zur Verwaltungsunterstützung die Berechtigung SELECT. Diese gilt für die angegebenen Extender, die der Benutzertabelle zugeordnet sind. Wenn Sie DB2IMAGE angeben, gewähren Sie die Berechtigung SELECT außerdem für die Tabellen zur Verwaltungsunterstützung für die QBIC-Kataloge, die der Benutzertabelle zugeordnet sind. Mit dem folgenden Befehl können Sie z. B. die Berechtigung SELECT für die Image Extender-Tabellen zur Verwaltungsunterstützung vergeben, die der Tabelle 'employee' zugeordnet sind. Die Berechtigung wird hierbei der Benutzer-ID 'ajones' zugeordnet. Mit dem Befehl wird außerdem der Benutzer-ID 'ajones' die Berechtigung SELECT für die QBIC-Kataloge gewährt, die der Tabelle 'employee' zugeordnet sind:

```
grant select for db2image on employee to ajones
```

Mit dem folgenden Befehl können Sie die Berechtigung SELECT für die Image, Audio und Video Extender-Tabellen zur Verwaltungsunterstützung vergeben, die der Tabelle 'employee' zugeordnet sind. Die Berechtigung wird hierbei allen Benut-

zern zugeordnet. Mit dem Befehl wird außerdem allen Benutzern die Berechtigung SELECT für die QBIC-Kataloge gewährt, die der Tabelle 'employee' zugeordnet sind:

```
grant select for all on employee to public
```

Bei Einfügungs-, Aktualisierungs- und Löschoperationen prüfen die Extender, ob der Benutzer über die erforderlichen Berechtigungen INSERT, UPDATE bzw. DELETE für die Benutzertabelle verfügt. Wenn der Benutzer über die erforderliche Berechtigung verfügt, gewähren die Extender den UDFs den Zugriff auf die Tabellen zur Verwaltungsunterstützung.

Um die Berechtigungen INSERT, UPDATE und DELETE für die QBIC-Katalogtabellen zu vergeben, müssen Sie im Befehl GRANT die Parameter UPDATE und DB2Image angeben. Mit dem folgenden Befehl können Sie z. B. der Benutzer-ID 'ajones' die Berechtigungen INSERT, UPDATE und DELETE für die QBIC-Katalogtabellen erteilen, die der Tabelle 'employee' zugeordnet sind:

```
grant update for db2image on employee to user ajones
```

Wenn ein Benutzer nicht länger auf ein Objekt in einer Benutzertabelle zugreifen muss, kann der Eigner der Benutzertabelle oder ein Datenbankadministrator (mit der Berechtigung GRANT) für die Datenbank die Berechtigung SELECT des Benutzers für die Tabellen zur Verwaltungsunterstützung zurücknehmen. Diese Operation betrifft auch die Tabellen zur Verwaltungsunterstützung, aus denen sich die QBIC-Kataloge zusammensetzen. Verwenden Sie den DB2 Extender-Befehl REVOKE, um die Berechtigungen für die Tabellen zur Verwaltungsunterstützung und für die QBIC-Katalogtabellen zu widerrufen. Das Format des Befehls REVOKE weist Ähnlichkeiten mit dem Format des Befehls GRANT auf. Mit dem folgenden Befehl können Sie z. B. die Berechtigung SELECT für die Image Extender-Tabellen zur Verwaltungsunterstützung zurücknehmen, die der Tabelle 'employee' zugeordnet sind. Die Berechtigung wird hierbei der Benutzer-ID 'ajones' entzogen. Mit dem Befehl wird außerdem die Berechtigung SELECT für die QBIC-Katalogtabellen widerrufen, die der Tabelle 'employee' zugeordnet sind:

```
revoke select for db2image on employee from ajones
```

Sie können auch die Berechtigungen INSERT, UPDATE und DELETE für die Tabellen zur Verwaltungsunterstützung zurücknehmen, aus denen sich die QBIC-Kataloge zusammensetzen. Verwenden Sie hierzu den Parameter UPDATE des Befehls REVOKE. Mit dem folgenden Befehl können Sie z. B. die Berechtigungen INSERT, UPDATE und DELETE für die QBIC-Katalogtabellen zurücknehmen, die der Tabelle 'employee' zugeordnet sind. Die Berechtigungen werden hierbei der Benutzer-ID 'ajones' entzogen.

```
revoke update for db2image on employee from ajones
```

Vergeben Sie die Berechtigungen für einen QBIC-Katalog, nachdem alle Funktionen hinzugefügt wurden: Die Berechtigungen für Tabellen zur Verwaltungsunterstützung, aus denen sich ein QBIC-Katalog zusammensetzt, umfassen die Berechtigungen für die Tabellen zu den QBIC-Funktionen. Dies gilt allerdings nur für die Funktionen, die bereits zum Katalog hinzugefügt wurden. Wenn Sie dem Katalog eine Funktion hinzufügen, nachdem Sie die Katalogberechtigungen vergeben haben, müssen Sie die Berechtigungsvergabe für den Katalog wiederholen. Aus diesem Grund sollten die Berechtigungen für einen QBIC-Katalog erst nach der Katalogerstellung und nach dem Hinzufügen aller benötigten Funktionen vergeben werden.

Teil 3. Programmierung für Abbild-, Audio- und Videodaten

Kapitel 9. Programmierübersicht

In diesem Kapitel erhalten Sie eine Übersicht über die Programmierung für die DB2 Extender. Das Kapitel enthält die Informationen, die Sie benötigen, bevor Sie mit der Programmierung für die Extender beginnen, und beschreibt eine Beispielanwendung, die illustriert, wie Sie Code für einen Extender schreiben.

Extender-UDFs und -APIs verwenden

Die DB2 Extender stellen benutzerdefinierte Funktionen zur Verfügung, mit denen Sie auf Abbild-, Audio- und Videodaten in einer Datenbank zugreifen und die Daten speichern und bearbeiten können. Sie codieren Anforderungen für diese benutzerdefinierten Funktionen (UDFs) in Ihrem Anwendungsprogramm, indem Sie SQL-Anweisungen auf die gleiche Weise verwenden, wie Sie integrierte SQL-Funktionen anfordern. Benutzerdefinierte Funktionen werden wie integrierte Funktionen im Datenbankserver ausgeführt.

Die folgende SQL-Anweisung in einem C-Anwendungsprogramm fordert eine benutzerdefinierte Funktion (UDF) des Image Extenders an. Diese Funktion hat den Namen `DB2Image` und soll verwendet werden, um ein Abbild in einer Datenbanktabelle zu speichern. Der Inhalt des Quellenabbilds befindet sich in einer Serverdatei.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL

EXEC SQL INSERT INTO EMPLOYEE VALUES(
        '128557',                          /*id*/
        'Anita Jones',                     /*name*/
        DB2IMAGE(                          /*Image Extender UDF*/
CURRENT SERVER,                            /*database */
           '/employee/images/ajones.bmp', /*image content*/
           'ASIS',                         /*keep the image format*/
           :hvStorageType,                 /*store image in DB as BLOB*
           'Anita''s picture')             /*comment*/
       );
```

Sie verwenden Extender-Anwendungsprogrammierschnittstellen, um Abbilder anzuzeigen und Audio- oder Videoobjekte wiederzugeben. Sie codieren diese APIs, indem Sie Client-Funktionsaufrufe in der Programmiersprache C verwenden. Die Funktionen werden auf Ihrer Workstation für den Datenbankclient ausgeführt.

Die folgenden Anweisungen in der Programmiersprache C schließen eine API mit dem Namen `DBiBrowse` ein. Die API ruft die Daten für eine Abbildkennung ab und startet einen Browser, um das Abbild anzuzeigen:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_hdl [251];
EXEC SQL END DECLARE SECTION

EXEC SQL SELECT PICTURE INTO :hvImg_hdl
   WHERE NAME= 'Robert Smith';
```

```
rc=DBiBrowse(
             "ib %s",                /*image browser*/
             MMDB_PLAY_HANDLE,       /*use image handle*/
             hvImg_hdl,              /*image handle*/
             MMDB_PLAY_NO_WAIT);     /*run browser independently*/
```

UDFs müssen unter der Benutzer-ID des Exemplars ausgeführt werden: DB2 Extender-UDFs müssen unter derselben Benutzer-ID ausgeführt werden wie das DB2 Extender-Exemplar. Darüber hinaus müssen, wenn Sie ein DB2 Extender-Exemplar erstellen oder ein bestehendes DB2 Extender-Exemplar verwenden, die UDFs unter derselben Benutzer-ID ausgeführt werden wie das DB2-Exemplar.

DB2 muss korrekt konfiguriert sein: Sie müssen DB2 korrekt konfigurieren, um den fehlerfreien Betrieb der DB2 Extender sicherzustellen. Dies gilt besonders für den fehlerfreien Betrieb der DB2 Extender-UDFs. Hierbei muss insbesondere auf die korrekte Einstellung des Datenbankkonfigurationsparameters APP_CTL_HEAP_SZ geachtet werden.

Mit den Extender-UDFs und -APIs ausführbare Tasks

In Tabelle 7 werden die Tasks aufgelistet, die Sie mit den Extender-UDFs und -APIs ausführen können, und wird die Seitenzahl angegeben, unter der sich nähere Informationen zur jeweiligen Task befinden.

Tabelle 7. Mit den DB2 Extender-APIs ausführbare Tasks

Task	Siehe
Abbild-, Audio- oder Videoobjekt speichern	Seite 113
Abbild-, Audio- oder Videoobjekt abrufen	Seite 126
Abbild-, Audio- und Videoattribute abrufen und verwenden	Seite 131
Einem Abbild-, Audio- oder Videoobjekt zugeordnete Kommentare abrufen	Seite 133
Abbild-, Audio- oder Videoobjekt aktualisieren	Seite 133
Abbildobjekt anzeigen	Seite 147
Abbild oder Videovollbild als Piktogramm anzeigen	Seite 150
Audio- oder Videoobjekt wiedergeben	Seite 151
Abbilder nach Inhalt abfragen	Seite 153
Szenenwechsel bei Videoobjekten ermitteln	Seite 19

Beispieltabelle der Beispiele für die Extender

In diesem Kapitel werden Programmierbeispiele dargestellt, die die DB2 Extender verwenden. Für die Beispiele wird angenommen, dass Sie eine Datenbanktabelle mit dem Namen 'employee' erstellt haben und dass die Tabelle Informationen zu Mitarbeitern enthält. Diese Tabelle enthält Spalten für die Personalnummern und Namen der Mitarbeiter. Abhängig vom verwendeten Extender enthält die Tabelle auch eine Spalte für Bilder, gesprochene Nachrichten und Videoclips der Mitarbeiter.

Abb. 25 verdeutlicht die Struktur der Mitarbeitertabelle (Tabelle 'employee') und stellt die SQL-Anweisung dar, die zum Erstellen der Tabelle verwendet wurde.

```
CREATE TABLE  employee(
    id          CHAR(6),

    name        VARCHAR(40),

    picture     DB2Image,
                DB2Audio,


                DB2Video

    sound
```

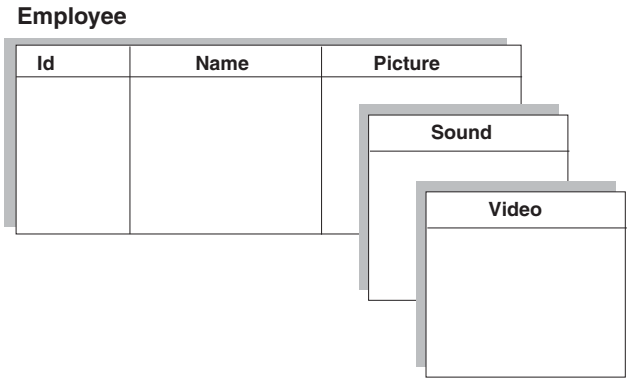


Abbildung 25. In den DB2 Extender-Programmierbeispielen verwendete Tabelle

Vorbereitung für die DB2 Extender-Programmierung

Bevor Sie mit der Entwicklung eines Programms beginnen, das mit den DB2 Extendern arbeitet, sollten Sie mit dem DB2-Anwendungsentwicklungsprozess und den verwendeten Programmiertechniken vertraut sein. Diese werden im Handbuch *DB2 Application Development Guide* beschrieben. Der Entwicklungsprozess für Programme, die die DB2 Extender verwenden, ist zum größten Teil identisch mit dem Prozess für herkömmliche DB2-Anwendungen.

Aufgrund der neuen Datentypen und Funktionen, die von den Extendern definiert werden, unterscheidet sich jedoch der Programmcode der Anwendung von herkömmlichen DB2-Anwendungen. In Abb. 26 auf Seite 104 wird beispielsweise eine in der Programmiersprache C codierte Anwendung dargestellt, die den Image Extender verwendet, um GIF-Abbilder zu identifizieren, die in einer Datenbanktabelle gespeichert sind. Nach dem Auffinden der Abbilder ruft das Programm einen Abbildbrowser auf, um sie anzuzeigen.

Wie dieses Beispiel verdeutlicht, muss eine Anwendung, die einen DB2 Extender verwendet, folgende Funktionen ausführen:

1 Extender-Definitionen einschließen. Im angeführten Beispiel ist die Datei `dmbimage.h` die Includedatei (Kopfdatei) für den Image Extender. Die Includedatei definiert die Konstanten, Variablen und Funktionsprototypen für den Extender.

2 Nach Bedarf Hostvariablen definieren, die Eingaben für oder Ausgaben aus einer benutzerdefinierten Funktion (UDF) oder Eingaben für einen API-Aufruf enthält. Im angeführten Beispiel sind die Variablen `hvFormat`, `hvSize`, `hvWidth`, `hvHeight` und `hvComment` Hostvariablen, die verwendet werden, um die Daten aufzunehmen, die von den UDFs des Image Extenders abgerufen wurden. Die Hostvariable `hvImg_hdl` wird verwendet, um eine Abbildkennung aufzunehmen, das als Eingabe für einen Image Extender-API-Aufruf angegeben wurde.

3 Nach Bedarf UDF-Anforderungen angeben. Im angeführten Beispiel sind die Funktionen `SIZE`, `WIDTH`, `HEIGHT`, `COMMENT` und `FORMAT` Image Extender-UDFs.

4 Nach Bedarf API-Aufrufe definieren. Im angeführten Beispiel ist der Aufruf `DBiBrowse` ein API-Aufruf einer lokalen Funktion in der Programmiersprache C, die die Abbilder, deren Kennungen aus einer Tabelle abgerufen wurden, anzeigt.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sqlenv.h>
#include <sqlcodes.h>
#include <dmbimage.h>  1

int count=0;

long
main(int argc,char *argv[])
{
EXEC SQL BEGIN DECLARE SECTION;  2
         char hvImg_hdl[251];                  /* image handle */
         char hvDBName[19];                    /* database name */
         char hvName[40];                      /* employee name */
         char hvFormat[9];                     /* image format */
         long hvSize;                          /* image size */
         long hvWidth;                         /* image width */
         long hvHeight;                        /* image height */
         struct  {       short len;
                  char data[32700]
         } hvComment;                          /* comment about the image */
 EXEC SQL END DECLARE SECTION;

/*  Connect to database  */
strcpy(hvDBName, argv[1]);                     /* copy the database name */

EXEC SQL CONNECT TO :hvDBName IN SHARE MODE;
/*
 * Set current function path*/
EXEC SQL SET CURRENT FUNCTION PATH = mmdbsys, CURRENT FUNCTION PATH;
```

Abbildung 26. Mit einem DB2 Extender arbeitende Anwendung (Teil 1 von 2)

```
/*
 * Select (query) using Image Extender UDF
 *
 * The SQL statement below finds all images in GIF format.
 */
EXEC SQL DECLARE c1 CURSOR FOR
         SELECT PICTURE, NAME,                3
                SIZE(PICTURE), WIDTH(PICTURE),
                HEIGHT(PICTURE), COMMENT(PICTURE)
  FROM EMPLOYEE
         WHERE PICTURE IS NOT NULL AND
               FORMAT(PICTURE) LIKE 'GIF%'
FOR FETCH ONLY;

EXEC SQL OPEN c1;
for (;;) {
         EXEC SQL FETCH c1 INTO :hvImg_hdl, :hvName, :hvSize,
                                :hvWidth, :hvHeight, :hvComment;         if (SQLCODE != 0)
                   break;

         printf("\nRecord %d:\n", ++count);
         printf("employee name = '%s'\n", hvName);
         printf("image size = %d bytes, width=%d, height=%d\n",
                                               hvSize, hvWidth, hvHeight);
         hvComment.data[Comment.len]='\0';         printf("comment len = %d\n", hvComment.len);
 printf("comment = %s\n", hvComment.data);
/*
* The API call below displays the images
*/
 4  rc=DBiBrowse ("ib %s",MMDB_PLAY_HANDLE,hvImg_hdl,
                          MMDB_PLAY_WAIT);
}

EXEC SQL CLOSE c1;

/* end of program */
```

Abbildung 26. Mit einem DB2 Extender arbeitende Anwendung (Teil 2 von 2)

Extender-Definitionen einschließen

In der Anwendung wird für jeden verwendeten Extender eine Includedatei (Kopfdatei) benötigt. Jede Includedatei definiert Konstanten, Variablen und Funktionsprototypen, die vom Extender verwendet werden. Die Includedateien haben folgende Namen:

Includedatei	Extender
dmbimage.h	Image
dmbqbapi.h	Image (Abfrage nach Abbildinhalt)
dmbaudio.h	Audio
dmbvideo.h	Video
dmbshot.h	Video (Ermittlung von Szenenwechseln)

Die Includedatei wird mit der Anweisung #include in ein C-Programm eingelagert. Die folgende Anweisung lagert beispielsweise die Includedatei für Image Extender ein:

```
#include <dmbimage.h>
```

Namen von benutzerdefinierten Funktionen und Typen angeben

Der vollständige Name einer benutzerdefinierten DB2 Extender-Funktion ist mmdbsys.funktionsname. Der vollständige Name eines benutzerdefinierten DB2 Extender-Typs ist mmdbsys.typname. Hierbei steht mmdbsys für den Schemanamen der Funktion oder des eindeutigen Typs. Der vollständige Name der benutzerdefinierten Funktion "Content" ist beispielsweise mmdbsys.Content; der vollständige Name Datentyps DB2Image, der vom Image Extender erstellt wurde, ist mmdbsys.DB2Image. Die Angabe des Schemanamens mmdbsys kann übergangen werden, wenn der aktuelle Funktionspfad für mmdbsys definiert wurde. Beispiel:

```
SET CURRENT FUNCTION PATH = mmdbsys, CURRENT FUNCTION PATH
SET CURRENT PATH = mmdbsys, CURRENT PATH
```

Große Objekte übertragen

Sie können große Objekte, wie beispielsweise Abbilder, Audioclips und Videoclips, auf verschiedene Weisen zwischen der Anwendung und einer DB2-Datenbank übertragen. Die verwendete Methode hängt davon ab, ob das Objekt aus einer/einem bzw. in eine Datei oder in einen Speicherpuffer übertragen wird. Die verwendete Methode hängt außerdem davon ab, ob die Datei sich auf der Clientmaschine oder dem Datenbankserver befindet.

Objekt zwischen einer Tabelle und einer Serverdatei übertragen

Wenn Sie ein Objekt zwischen einer Datenbanktabelle und einer Serverdatei übertragen, müssen Sie den Dateipfad in der entsprechenden Anforderung für die benutzerdefinierte Extender-Funktion angeben. Da sich die benutzerdefinierte Extender-Funktion und die Datei auf dem Server befinden, kann der Extender die Datei finden. In der folgenden SQL-Anweisung wird beispielsweise ein Abbild, dessen Inhalt sich in einer Serverdatei befindet, in einer Datenbanktabelle gespeichert:

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
      DB2Image(
         CURRENT SERVER,
          '/employee/images/ajones.bmp',
                   'ASIS',
         :hvStorageType,
          'Anita''s picture')
       );
```

Objekt in einen oder aus einem Clientpuffer übertragen

Die Extender können nicht direkt auf einen Speicherpuffer zugreifen. Wenn Sie ein Objekt in einen oder aus einem Puffer auf Ihrer Clientmaschine übertragen wollen, müssen Sie eine Methode verwenden, bei der kein Speicherstandort angegeben wird. Eine Möglichkeit für die Übertragung ist die Verwendung einer Hostvariablen. Mit dieser Methode werden im Allgemeinen Objekte zwischen einer Anwendung und einer DB2-Datenbank übertragen.

Hostvariablen für große Objekte werden auf die gleiche Weise definiert und verwendet wie für herkömmliche Zeichenobjekte und numerische Objekte. Sie deklarieren die Hostvariablen in einem DECLARE-Abschnitt, ordnen ihnen Werte für die Übertragung zu oder greifen auf die Werte zu, die an sie übertragen wurden.

Geben Sie den Datentyp BLOB an, wenn Sie eine Hostvariable für Abbild-, Audio- oder Videodaten deklarieren. Wenn Sie eine benutzerdefinierte Funktion (UDF) verwenden, um ein Objekt zu speichern, abzurufen oder zu ändern, wird die entsprechende Hostvariable als Argument in der UDF-Anforderung angegeben. Verwenden Sie das gleiche Format wie für andere Hostvariablen, die in einer SQL-Anweisung angegeben werden.

Im folgenden Beispiel wird durch die SQL-Anweisungen eine Hostvariable mit dem Namen `hvaudio` deklariert und verwendet, um einen Audioclip an die Datenbank zu übertragen:

```
EXEC SQL BEGIN DECLARE SECTION;
  SQL TYPE IS BLOB (2M) hvaudio;
EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
      DB2Audio(
         CURRENT SERVER,
          :hvaudio,
         'WAVE',
CAST(NULL as LONG VARCHAR),          'Anita''s voice')
       );
```

LOB-Zeiger verwenden

Objekte, wie beispielsweise Audio- und Videoclips, können sehr groß sein, so dass die Verwendung von Hostvariablen möglicherweise nicht die effizienteste Methode ist, sie zu bearbeiten. Ein **LOB-Zeiger** kann eine günstigere Methode sein, LOBs (Large Objects) in Ihren Anwendungen zu bearbeiten.

Ein LOB-Zeiger ist ein kleiner (4 Byte großer) Wert, der in einer Hostvariablen gespeichert ist, und den das Programm verwenden kann, um auf ein wesentlich größeres LOB zu verweisen. Mit Hilfe eines LOB-Zeigers kann das Programm das LOB bearbeiten, als sei das LOB in einer normalen Hostvariablen gespeichert. Der Unterschied liegt darin, dass das LOB nicht zwischen dem Datenbankserver und der Anwendung auf der Clientmaschine übertragen werden muss. Wenn Sie beispielsweise ein LOB in einer Datenbanktabelle auswählen, verbleibt der LOB auf dem Server und der LOB-Zeiger wird an den Client übertragen.

Ein LOB-Zeiger wird in einem DECLARE-Abschnitt deklariert und auf die gleiche Weise verwendet wie eine Hostvariable. Geben Sie den Datentyp BLOB_LOCATOR an, wenn Sie einen LOB-Zeiger für Abbild-, Audio- oder Videodaten deklarieren. Im folgenden Beispiel wird durch die SQL-Anweisungen ein LOB-Zeiger mit dem Namen `video_loc` deklariert, um ein ein Videoclip aus einer Datenbanktabelle abzurufen:

```
EXEC SQL BEGIN DECLARE SECTION;
  SQL TYPE IS BLOB_LOCATOR video_loc;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT(VIDEO)
     INTO :video_loc
  FROM EMPLOYEE
       WHERE NAME='Anita Jones';
```

UDFs verwenden LOB-Zeiger: DB2 Extender-UDFs, die Abbild-, Audio- und Videoobjekte speichern, abrufen und aktualisieren, verwenden LOB-Zeiger. Unter DB2 Extender V1 verwendeten diese UDFs keine LOB-Zeiger und konnten aus diesem Grund keine Objekte verarbeiten, die größer als 2 MB waren. Aufgrund dieser Einschränkung waren Benutzer gezwungen, Objekte, die größer als 2 MB waren, in Segmenten zu übertragen. Da diese UDFs jetzt LOB-Zeiger verwenden, entfällt die Einschränkung auf 2 MB.

Objekt in eine oder aus einer Clientdatei übertragen

Eine Dateireferenzvariable wird verwendet, um Objekte in eine oder aus einer Datei auf einem Client zu übertragen. Durch die Verwendung einer Dateireferenzvariablen ist es nicht mehr notwendig, für ein großes Objekt in Ihrem Anwendungsprogramm Pufferbereich zuzuordnen. Wenn Sie eine Dateireferenzvariable mit einer benutzerdefinierten Funktion (UDF) verwenden, findet die Übergabe des BLOB-Inhalts direkt zwischen Datei und UDF statt.

Eine Dateireferenzvariable wird in einem DECLARE-Abschnitt deklariert und auf die gleiche Weise verwendet wie eine Hostvariable. Geben Sie den Datentyp BLOB_FILE an, wenn Sie eine Dateireferenzvariable für Abbild-, Audio- oder Videodaten deklarieren. Im Gegensatz zu einer Hostvariablen, die den Inhalt eines Objekts enthält, enthält eine Dateireferenzvariable den Namen der Datei. Die Datei kann nicht größer sein, als die für die UDF definierte Größe des BLOB.

Für die Verwendung einer Dateireferenzvariablen für die Ein- und Ausgabe bestehen verschiedene Optionen. Wählen Sie die gewünschte Option aus, indem Sie das Feld FILE_OPTIONS in der Struktur der Dateireferenzvariablen im Programm definieren. Folgende Optionen stehen zur Verfügung:

Option für die Eingabe:

SQL_FILE_READ. Diese Datei kann geöffnet, gelesen und geschlossen werden. Die Länge der Daten in der Datei (in Byte) wird beim Öffnen der Datei festgelegt. Das Feld data_length in der Struktur der Dateireferenzvariablen enthält die Länge der Datei (in Byte).

Optionen für die Ausgabe:

SQL_FILE_CREATE. Diese Option erstellt eine neue Datei, falls sie nicht bereits existiert. Existiert die Datei bereits, wird eine Fehlernachricht ausgegeben. Das Feld data_length in der Struktur der Dateireferenzvariablen enthält die Länge der Datei (in Byte).

SQL_FILE_OVERWRITE. Diese Option erstellt eine neue Datei, falls sie nicht bereits existiert. Existiert die Datei bereits, werden die Daten in der Datei durch die neuen Daten überschrieben. Das Feld data_length in der Struktur der Dateireferenzvariablen enthält die Länge der Datei (in Byte).

SQL_FILE_APPEND. Diese Option fügt die Ausgabe an die Datei an, falls die Datei bereits existiert. Existiert die Datei noch nicht, wird eine neue Datei erstellt. Das Feld data_length in der Struktur der Dateireferenzvariablen enthält nicht die Gesamtlänge der Datei, sondern die Länge der Daten (in Byte), die zur Datei hinzugefügt werden.

Die Anweisungen im folgenden Beispiel deklarieren eine Dateireferenzvariable mit dem Namen Img_file und verwenden sie, um ein Abbild, dessen Inhalt sich in einer Clientdatei befindet, in einer Datenbanktabelle zu speichern. Beachten Sie die Zuordnung SQL_FILE_READ im Feld FILE_OPTIONS:

```
EXEC SQL BEGIN DECLARE SECTION;
  SQL TYPE IS BLOB_FILE Img_file;
  EXEC SQL END DECLARE SECTION;

strcpy (Img_file.name,"/employee/images/ajones.bmp");
Img_file.name_length=strlen(Img_file.name);
Img_file.file_options=SQL_FILE_READ;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
      DB2Image(
         CURRENT SERVER,
          :Img_file,
                  'ASIS',
CAST(NULL as LONG VARCHAR),          'Anita''s picture')
       );
```

Dateinamen beim Übertragen von Objekten angeben

Die DB2 Extender geben Ihnen verschiedene Möglichkeiten, wie Sie Dateinamen beim Speichern, Abrufen und Aktualisieren von Objekten angeben können.

Sie können einen vollständig qualifizierten Dateinamen (d. h. einen vollständigen Pfad, gefolgt vom Dateinamen) für Speicher-, Abruf- und Aktualisierungsoperationen angeben; die Angabe eines relativen Pfadnamens ist jedoch vorteilhafter. Unter AIX, HP-UX und Solaris ist ein relativer Dateiname jeder Dateiname, der nicht mit einem Schrägstrich beginnt. Unter Windows ist ein relativer Dateiname jeder Dateiname, der nicht mit einem Laufwerkbuchstaben, gefolgt von einem Doppelpunkt und einem umgekehrten Schrägstrich beginnt.

Wenn Sie einen relativen Dateinamen angeben, verwenden die Extender die Verzeichnisangaben in den verschiedenen Client- und Serverumgebungsvariablen als Suchpfad, um den Dateinamen aufzulösen. Ein vollständiger Pfadname besteht aus einem führenden Teil, der sich normalerweise auf Mount-Punkte bezieht, und einem abschließenden Pfadnamen, der die benötigte Datei eindeutig identifiziert. Der abschließende Pfadname ist in UDFs angegeben. Umgebungsvariablen liefern eine Liste von führenden Pfadnamen, die durchsucht werden soll, wenn versucht wird, relative Dateinamen aufzulösen. Weitere Informationen zu den Umgebungsvariablen, die von den DB2 Extendern verwendet werden, um Dateinamen aufzulösen, befinden sich in Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521.

Außerdem setzen die Extender, falls erforderlich, die Formate der Dateinamen um. Wird ein Dateiname an den Server übergeben, wird er in das Dateiformat für das Betriebssystem des Servers umgesetzt.

Rückkehrcodes verwenden

Alle eingebetteten SQL-Anweisungen oder DB2 CLI-Aufrufe in Ihrem Programm, einschließlich derer, die DB2 Extender UDFs anfordern, generieren Codes, die angeben, ob die eingebettete SQL-Anweisung oder der DB2 CLI-Aufruf erfolgreich ausgeführt wurde. Andere DB2 Extender-APIs, wie z. B. Verwaltungs-APIs, geben auch Code zurück, der angibt, ob die API erfolgreich oder nicht erfolgreich ausgeführt wurde. Ihr Programm sollte die Codes, die von eingebetteten SQL-Anweisungen, CLI-Aufrufen und APIs zurückgegeben werden, überprüfen und entsprechend auf sie reagieren.

Informationen zur Verwendung dieser Rückkehrcodes befinden sich in Kapitel 16, „Diagnoseinformationen", auf Seite 487.

In Situationen, in denen eine Extender-API ihre Arbeitseinheiten nicht erfolgreich abschließen kann, wird eine ROLLBACK-Operation ausgeführt. Die API gibt außerdem einen Fehlercode zurück. Die ROLLBACK-Operation wird ausgeführt, damit die Datenbank zum vorherigen konsistenten Stand zurückkehren kann. Nähere Einzelheiten finden Sie in Kapitel 14, „Anwendungsprogrammierschnittstellen", auf Seite 253.

Unicode-Unterstützung

Beachten Sie folgende Aspekte bei der Unicode-Unterstützung für die Image, Audio und Video Extender:

- Die einzigen Parameter, die eine Unicode-Zeichenfolge enthalten können, sind die Kommentarfelder in den folgenden UDFs:
  - mmdbsys.db2image( ) Abbild importieren
  - mmdbsys.db2audio( ) Ton importieren
  - mmdbsys.db2video( ) Video importieren
  - mmdbsys.replace( ) Abbild, Ton oder Video ersetzen
  - mmdbsys.comment( ) Kommentar aktualisieren
- Wenn Sie planen, auf eine Unicode-Datenbank zuzugreifen, müssen Sie ein DB2 Extender-Exemplar verwenden, das für die Unterstützung von Unicode eingerichtet ist. Ein Unicode-Exemplar bearbeitet nur Unicode-Datenbanken.

  Damit ein Extender-Exemplar Unicode unterstützt, müssen Sie die Umgebungsvariable DB2CODEPAGE auf 1208 setzen, bevor Sie DMBSTART aufrufen.

Kapitel 10. Objekte speichern, abrufen und aktualisieren

In diesem Kapitel wird beschrieben, wie Sie die benutzerdefinierten Funktionen (UDFs) der DB2 Extender verwenden können, um ein Abbild, einen Ton oder ein Video zu speichern, abzurufen und zu aktualisieren.

Abbild-, Audio- und Videoformate

In Tabelle 8 werden die Formate aufgelistet, in denen Abbild-, Audio- und Videoobjekte gespeichert, abgerufen und aktualisiert werden können. Für Abbildobjekte können Sie angeben, dass der Image Extender das Format des Abbilds beim Speichern, Abrufen oder Aktualisieren umsetzt. (Formate von Audio- und Videoobjekten können beim Speichern, Abrufen oder Aktualisieren nicht umgesetzt werden.)

Die Spalten ″Lesen″ und ″Schreiben″ in der Tabelle geben an, welche Formate gelesen und welche Formate beim Schreiben umgesetzt werden können. Der Eintrag ″x″ in der Spalte ″Lesen″ der Tabelle besagt, dass das entsprechende Objektformat beim Speichern, Abrufen oder Aktualisieren verwendet werden kann. Der Eintrag ″x″ in der Spalte ″Schreiben″ besagt, dass ein Abbildobjekt beim Speichern, Abrufen oder Aktualisieren in das entsprechende Format umgesetzt werden kann. Ein Abbild im BMP-Format kann beispielsweise beim Speichern, Abrufen oder Aktualisieren in das GIF-Format umgesetzt werden. Ein Abbild im JPG-Format kann in das TIF-Format umgesetzt werden. Aber ein Abbild im TIF-Format kann nicht in das JPG-Format umgesetzt werden.

Obwohl die Formatangaben in dieser Tabelle in Großschreibung angegeben sind, wird bei Speicher-, Abruf- und Aktualisierungsanforderungen die Groß- und Kleinschreibung nicht berücksichtigt. So werden beispielsweise die Angaben GIF, gif und Gif gleich behandelt.

Tabelle 8. Von den DB2 Extendern verarbeitbare Formate

Format	Beschreibung	Lesen	Schreiben
	Abbildformate		
_IM	PS/2 Audio Video Connection (AVC)	x	
BMP	OS/2 - Microsoft Windows-Bitmap[2]	x	x
EPS	Eingebundenes PostScript		x
EP2	Eingebundenes PostScript der Ebene 2		x
GIF	Compuserve GIF89a (einschließlich animierte GIFs[3]) und 87	x	x
IMG	IOCA-Abbild	x	x
IPS	Brooktrout FAX-Kartendatei	x	x
JPG	JPEG[4] (JFIF-Format)	x	
PCX	PC Paint-Datei (nur Grauskala)	x	x
PGM	Portable Gray Map (von PBMPLUS)	x	x
PS	PostScript		x
PSC	Komprimiertes PostScript-Abbild		x

Tabelle 8. Von den DB2 Extendern verarbeitbare Formate (Forts.)

Format	Beschreibung	Lesen	Schreiben
PS2	PostScript der Ebene 2 (Farbe)		x
TIF	Alle TIFF 5.0-Formate	x	x
YUV	Digitales Video für YUV	x	x
	Audioformate		
AIF oder AIFF	Audio Interchange File Format	x	
AIFFC	Audio Interchange File Format Compressed	x	
AU	Sun-Audiodateiformat	x	
MIDI	Musical Instrument Digital Interface	x	
MPG1 oder MPEG1	Moving Pictures Expert Group 1	x	
WAV oder WAVE	Wave	x	
	Videoformate		
AVI	Audio/Video Interleaved	x	
MPG1 oder MPEG1	Motion Picture Coding Expert Group 1	x	
MPG2 oder MPEG2	Motion Picture Coding Expert Group 2	x	
QT	Quicktime (AVI)	x	

Umsetzungsoptionen für Abbilder

In Tabelle 9 werden die Umsetzungsoptionen aufgelistet, die Sie (zusätzlich zur Formatumsetzung) beim Speichern, Abrufen oder Aktualisieren für ein Abbild angeben können. Der Image Extender wendet Ihre Angaben auf das Zielabbild an; das Ursprungsabbild bleibt unverändert.

Jede Umsetzungsoption wird als Parameter/Wert-Paar angegeben. Die zulässigen Werte für die einzelnen Parameter werden in der Tabelle aufgelistet.

Tabelle 9. Umsetzungsoptionen für Abbilder

Parameter	Beschreibung	Wert
-b	Anzahl an Bit zur Darstellung des Abbildsamples	1 oder 8 Bit
-s[5]	Maßstabsfaktor	Jeder Dezimalwert, der größer als Null ist. Der Maßstabsfaktor gibt das Größenverhältnis zwischen dem umgesetzten Abbild und dem Original an. Beispielsweise wird bei einem Maßstabsfaktor von 0,5 das Abbild auf die Hälfte der Originalgröße umgesetzt. Bei einem Maßstabsfaktor von 2,0 wird das Abbild auf die doppelte Originalgröße umgesetzt.

2. Das Lesen wird für BMP-Formate unter Windows Version 2, Windows Version 3 und Windows NT unterstützt.

3. Der DB2 Image Extender speichert nur Attributinformationen für das erste Abbild in der animierten GIF-Datei.

4. Die Unterstützung verwendet Software, die teilweise auf der Arbeit der Independent JPEG Group basiert.

Tabelle 9. Umsetzungsoptionen für Abbilder (Forts.)

Parameter	Beschreibung	Wert
-p	Abbildumkehrung. Diese Option ändert die Interpretation eines Abbilds auf der Basis des angegebenen Wertes. Sie ändert nicht das Abbild selbst. Diese Option gilt nur für Abbilder in Schwarzweiß oder Graustufen und gilt nicht für Abbilder im GIF-Format.	0 = Einsen sind schwarz 1 = Einsen sind weiß
-n	Abbildumkehrung. Diese Option ändert ein Abbild, indem Sie Schwarz in Weiß und Weiß in Schwarz umkehrt. Die Option gilt nur für Abbilder in Schwarzweiß oder Graustufen.	Keine
-r[5]	Drehung	0 = 0 Grad (keine Drehung) 1 = 90 Grad (gegen den Uhrzeigersinn) 2 = 90 Grad (mit dem Uhrzeigersinn) 3 = 180 Grad
-x[5]	Breite in Pixel	Anzahl an Pixel
-y[5]	Höhe in Pixel	Anzahl an Pixel
-c	Komprimierungsart	0 = IBM MMR 1 = CCITT Group 3 1-D 2 = CCITT Group 3 2-D (k=2) 3 = CCITT Group 3 2-D (k=4) 4 = CCITT Group 4 6 = TIFF Typ 2 10 = Nicht komprimiert 14 = LZW 15 = TIFF Packbits 25 = JBIG

Abbild-, Audio- oder Videoobjekt speichern

Verwenden Sie die benutzerdefinierte Funktion "DB2Image", "DB2Audio" oder "DB2Video" in einer SQL-Anweisung INSERT, um ein Abbild-, Audio- oder Videoobjekt in einer Datenbank zu speichern.

Sie können ein Objekt speichern, dessen Quelle sich in einem Puffer oder einer Datei auf einer Clientmaschine oder in einer Serverdatei befindet. Für diese Quellen können Sie das Objekt in einer Datenbanktabelle als BLOB (Binary Large Object) oder in einer Datei auf dem Datenbankserver speichern.

Beim Anfordern der UDF (benutzerdefinierte Funktion) müssen Sie Folgendes angeben:

- Den Namen des momentan verbundenen Datenbankservers; diese Angabe ist im Sonderregister CURRENT SERVER enthalten.
- Die Quelle des Objektinhalts; dies kann ein Clientpuffer, eine Clientdatei oder eine Serverdatei sein.

5. Wenn Sie diese Option für ein GIF-Abbild mit Zeilensprungverfahren angeben, sollten Sie auch die Komprimierungsart LZW angeben.

- Ob der Inhalt in einer Datenbanktabelle (als BLOB) oder auf einem Dateiserver gespeichert werden soll.
- Das Format der Quelle.
- Einen Kommentar, der mit dem Objekt gespeichert werden soll (oder einen Nullwert oder eine leere Zeichenfolge, falls Sie keinen Kommentar speichern wollen).

Mit den Image, Audio und Video Extendern können Sie ein Objekt auch speichern, wenn die Extender das Format des Objekts nicht erkennen. Wenn das Format nicht erkannt wird, müssen Sie die Attribute des Objekts angeben. Wenn Sie ein Abbild oder Video mit benutzerdefinierten Attributen speichern, können Sie auch ein Piktogramm speichern. Ein Piktogramm ist eine verkleinerte Version des Abbilds oder des Videos.

Bei Abbildern haben Sie die Möglichkeit, beim Speichern das Format des Abbilds umzusetzen. Wenn Sie die Formatumsetzung anfordern, müssen Sie sowohl das Quellen- als auch das Zielformat des Abbilds angeben. Bei einer Anforderung für eine Formatumsetzung können Sie weitere Änderungen für das Abbild angeben, z. B. eine Ausschnittserstellung oder eine Drehung. Sie geben diese Änderungen mit Hilfe von Umsetzungsoptionen an.

Speicheroperation festschreiben: Nach dem Speichern eines Abbild-, Audio- oder Videoobjekts in einer Datenbank sollten Sie für die Arbeitseinheit eine COMMIT-Operation durchführen. Hierdurch werden die Sperren, die die Extender verwenden, freigegeben, so dass Sie Aktualisierungsoperationen für das gespeicherte Objekt durchführen können.

Formate der UDFs DB2Image, DB2Audio und DB2Video

Die benutzerdefinierten Funktionen (UDFs) ″DB2Image″, ″DB2Audio″ und ″DB2Video″ sind mehrfach belegt. Dies bedeutet, dass sie abhängig von der Verwendung der UDFs unterschiedliche Formate haben. Jede UDF hat die folgenden Formate (die Zeichenfolge xxxxx in den Formaten kann für Image, Audio oder Video stehen):

Format 1: Objekt aus einem Clientpuffer oder einer Clientdatei speichern:

```
DB2xxxxx(
CURRENT SERVER,       /* database name name in CURRENT SERVER REGISTER */ content,
                     /* object content */
     format,                      /* source format */
     target_file,                 /* target file name for storage in file server */
                                  /* or NULL for storage in table as BLOB */
     comment                      /* user comment */
  );
```

Format 2: Objekt aus einer Serverdatei speichern:

```
DB2xxxxx(
CURRENT SERVER,       /* database name in CURRENT SERVER REGISTER */     source_file,
                             /* source file name */
     format,                 /* source format */
     stortype,               /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                             /* in file server*/
                             /* MMDB_STORAGE_TYPE_INTERNAL=store */
                             /* as a BLOB*/
     comment                 /* user comment */
  );
```

Format 3: Objekt mit benutzerdefinierten Attributen aus einem Clientpuffer oder einer Clientdatei speichern:

```
DB2xxxxx(
     CURRENT SERVER,              /* database name in CURRENT SERVER REGISTER */
     content,                     /* object content */
     target_file,                 /* target file name for storage in file server */
                                  /* or NULL for storage in table as BLOB */
     comment,                     /* user comment */
     attrs,                       /* user-supplied attributes */
     thumbnail                    /* thumbnail (image and video only) */
  );
```

Format 4: Objekt mit benutzerdefinierten Attributen aus einer Serverdatei speichern:

```
DB2xxxxx(
     CURRENT SERVER,           /* database name in CURRENT SERVER REGISTER */
     source_file,              /* source file name */
     stortype,                 /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                               /* in file server*/
                               /* MMDB_STORAGE_TYPE_INTERNAL=store */
                               /* as a BLOB*/
     comment,                  /* user comment */
     attrs,                    /* user-supplied attributes */
     thumbnail                 /* thumbnail (image and video only) */
  );
```

Zu der UDF DB2Image gehören zusätzlich die folgenden Formate:

Format 5: Abbild aus einem Clientpuffer oder einer Clientdatei mit Formatumsetzung speichern:

```
DB2Image(
CURRENT SERVER,       /* database name in CURRENT SERVER REGISTER */   content,
                               /* object content */
     source_format,            /* source format */
     target_format,            /* target format */
     target_file,              /* target file name for storage in file server */
                               /* or NULL for storage in table as BLOB */
     comment                   /* user comment */
  );
```

Format 6: Abbild aus einer Serverdatei mit Formatumsetzung speichern:

```
DB2Image(
CURRENT SERVER,       /* database name in CURRENT SERVER REGISTER */  source_file,
                               /* server file name */
     source_format,            /* source format */
     target_format,            /* target format */
     target_file,              /* target file name for storage in file server */
                               /* or NULL for storage in table as BLOB */
     comment                   /* user comment */
  );
```

Format 7: Abbild aus einem Clientpuffer oder einer Clientdatei mit Formatumsetzung und zusätzlichen Änderungen speichern:

```
DB2Image(
CURRENT SERVER,       /* database name in CURRENT SERVER REGISTER */  content,
                              /* object content */
     source_format,           /* source format */
     target_format,           /* target format */
     conversion_options,      /* Conversion options */
     target_file,             /* target file name for storage in file server */
                              /* or NULL for storage in table as BLOB */
     comment                  /* user comment */
  );
```

Format 8: Abbild aus einer Serverdatei mit Formatumsetzung und zusätzlichen Änderungen speichern:

```
DB2Image(
CURRENT SERVER,       /* database name in CURRENT SERVER REGISTER */  source_file,
                              /* server file name */
     source_format,           /* source format */
     target_format,           /* target format */
     conversion_options       /* conversion options */
     target_file,             /* target file name for storage in file server */
                              /* or NULL for storage in table as BLOB */
     comment                  /* user comment */
  );
```

Die folgenden Anweisungen in einem C-Anwendungsprogramm fügen beispielsweise eine Zeile ein, mit der ein Abbild zur Tabelle 'employee' hinzugefügt wird. Das Quellenabbild befindet sich in einer Serverdatei mit dem Namen ajones.bmp. Das Abbild wird in der Tabelle 'employee' als BLOB (Binary Large Object) gespeichert. (Dies entspricht dem oben aufgeführten Format 2.)

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
        '128557',                       /*id*/
        'Anita Jones',                  /*name*/
        DB2IMAGE(                       /*Image Extender UDF*/
CURRENT SERVER,                /*database*/  '/employee/images/ajones.bmp',
                                            /*source file */
           'ASIS',                      /*keep the image format*/
          :hvStorageType                /*store image in DB as BLOB*/
          'Anita''s picture')           /*comment */
       );
```

Die folgenden Anweisungen in einem C-Anwendungsprogramm speichern dieselbe Zeile wie im vorigen Beispiel in der Tabelle 'employee'. In diesem Fall wird jedoch das Abbild beim Speichern vom BMP-Format in das GIF-Format umgesetzt. (Dies entspricht dem oben aufgeführten Format 6.)

```
EXEC SQL INSERT INTO EMPLOYEE VALUES(
        '128557',                       /*id*/
        'Anita Jones',                  /*name*/
        DB2IMAGE(                       /*Image Extender UDF*
CURRENT SERVER,           /*database*/ '/employee/images/ajones.bmp',   /*source file */
          'ASIS',                       /*source image format*/
          'GIF',                        /*target image format*/
           'Anita''s picture')          /*comment*/
       );
```

Wenn Sie ein Abbild-, Audio- oder Videoobjekt speichern, berechnet der Extender die Attribute, wie beispielsweise die Anzahl der im Abbild verwendeten Farben, die Spieldauer des Audioclips oder das Videokomprimierungsformat. Wenn Sie ein Objekt mit einem nicht erkannten Format speichern, müssen Sie diese Attribute als Eingabe für die benutzerdefinierte Funktion angeben. Der Extender speichert diese und andere Attribute, wie beispielsweise Kommentare über das Objekt oder die ID des Benutzers, der das Objekt gespeichert hat, in der Datenbank. Diese Attribute können anschließend in Abfragen verwendet werden.

Objekt speichern, das sich auf dem Client befindet

Verwenden Sie eine Hostvariable oder eine Dateireferenzvariable, um den Inhalt eines Abbild-, Audio- oder Videoobjekts von einem Clientpuffer oder einer Clientdatei auf den Server zu übertragen.

Befindet sich das Objekt in einer Clientdatei, sollten Sie eine Dateireferenzvariable verwenden, um seinen Inhalt zum Speichern auf den Server zu übertragen. Die Anweisungen im folgenden Beispiel für ein C-Anwendungsprogramm definieren eine Dateireferenzvariable mit dem Namen `Audio_file` und verwenden sie, um einen Audioclip, dessen Inhalt sich in einer Clientdatei befindet, zu übertragen. Der Audioclip ist in einer Datenbanktabelle auf dem Server gespeichert. Beachten Sie, dass das Feld `file_option` der Dateireferenzvariablen auf die Option für die Eingabe `SQL_FILE_READ` gesetzt ist. Beachten Sie auch, dass die Dateireferenzvariable als Inhaltsargument für die benutzerdefinierte Funktion ″DB2Audio″ verwendet wird.

```
EXEC SQL BEGIN DECLARE SECTION;
  SQL TYPE IS BLOB_FILE Audio_file;
EXEC SQL END DECLARE SECTION;

strcpy (Audio_file.name, "/employee/sounds/ajones.wav");
Audio_file.name_length= strlen(Audio_file.name);
Audio_file.file_options= SQL_FILE_READ;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2AUDIO(
         CURRENT SERVER,
          :Audio_file,          /* file reference variable */
         'WAVE',
          CAST(NULL as LONG VARCHAR),

          'Anita''s voice')
      );
```

Befindet sich das Objekt in einem Clientpuffer, sollten Sie eine Hostvariable verwenden, die entweder als BLOB oder BLOB_LOCATOR definiert ist, um seinen Inhalt zum Speichern auf den Server zu übertragen. In den folgenden Anweisungen für ein C-Anwendungsprogramm wird eine Hostvariable mit dem Namen `Video_loc` verwendet, um den Inhalt eines Videoclips zum Speichern auf den Server zu übertragen. Der Videoclip wird als BLOB in einer Datenbanktabelle gespeichert. Beachten Sie, dass die Hostvariable als Inhaltsargument für die benutzerdefinierte Funktion ″DB2Video″ verwendet wird.

```
EXEC SQL BEGIN DECLARE SECTION;
  SQL TYPE IS BLOB_LOCATOR Video_loc;
EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2VIDEO(
         CURRENT SERVER,
          :Video_loc,         /* host variable */
         'MPEG1',
          '',
          'Anita''s video')
       );
```

Stellen Sie sicher, dass genügend UDF-Speicher zur Verfügung steht: Wenn Sie ein Objekt speichern, dessen Inhalt sich in einem Clientpuffer befindet, müssen Sie sicherstellen, dass der Parameter UDF_MEM_SZ in der Datenbankmanagerkonfiguration auf einen Wert von mindestens 4 MB gesetzt ist. Sie können den Parameter UDF_MEM_SZ aktualisieren, indem Sie den DB2-Befehl UPDATE DATABASE MANAGER CONFIGURATION verwenden. Weitere Informationen zum Befehl UPDATE DATABASE MANAGER befinden sich im Handbuch *DB2 Command Reference*.

Objekt speichern, das sich auf dem Server befindet

Wenn sich das Abbild-, Audio- oder Videoobjekt, das gespeichert werden soll, in einer Serverdatei befindet, müssen Sie ihren Pfad als Inhaltsargument für die benutzerdefinierte Funktion angeben. Die folgende Anweisung in einem C-Anwendungsprogramm speichert beispielsweise eine Zeile, mit der ein Abbild in die Datenbank eingefügt wird. Der Abbildinhalt befindet sich in einer Datei auf dem Server. Das gespeicherte Abbild verbleibt in der Serverdatei. Von der Datenbank aus wird auf das Abbild verwiesen.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2IMAGE(
         CURRENT SERVER,
          '/employee/images/ajones.bmp', /*source in server file */
          'BMP',
         :hvStorageType,
          'Anita''s picture')
       );
```

Geben Sie den korrekten Pfad an: Wenn Sie ein Objekt speichern, dessen Quelle sich in einer Serverdatei befindet, können Sie den vollständig qualifizierten oder den relativen Namen der Datei angeben. Wenn Sie einen relativen Namen angeben, müssen Sie sicherstellen, dass die entsprechenden Umgebungsvariablen auf dem DB2-Server den korrekten Pfad für die Datei enthalten. Informationen zum Definieren dieser Umgebungsvariablen befinden sich in Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521.

Datenbank- oder Dateispeicherung angeben

Sie können ein Abbild-, Audio- oder Videoobjekt in einer Datenbanktabelle als BLOB (Binary Large Object) oder in einer Serverdatei speichern. Wenn Sie das Objekt in einer Serverdatei speichern, verweist die Datenbank auf die Datei.

Wenn Sie das Objekt aus einem Clientpuffer oder einer Clientdatei speichern, erfolgt die Speicherung als BLOB oder Serverdatei entsprechend den Angaben, die Sie im Parameter target_file machen. Die Angabe eines Dateinamens bedeutet, dass das Objekt in einer Serverdatei gespeichert werden soll. Die Angabe eines Nullwerts oder einer leeren Zeichenfolge bedeutet, dass das Objekt als BLOB in einer Datenbanktabelle gespeichert werden soll. Der Datentyp des Parameters target_file ist LONG VARCHAR. Wenn Sie einen Nullwert angeben, müssen Sie daran denken, ihn mit dem Datentyp LONG VARCHAR zu versehen.

Die folgenden Anweisungen in einem C-Anwendungsprogramm speichern beispielsweise eine Zeile, mit der ein Abbild in eine Datenbanktabelle eingefügt wird. Die Quelle des Abbilds befindet sich in einem Clientpuffer. Das Abbild wird in einer Serverdatei gespeichert. Die Datenbanktabelle zeigt auf die Serverdatei:

```
EXEC SQL BEGIN DECLARE SECTION;
  SQL TYPE IS BLOB_LOCATOR Img_buf
EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2IMAGE(
         CURRENT SERVER,
                  :Img_buf,
                  'ASIS',
          '/employee/images/ajones.bmp',    /* store image in server file */
          'Anita''s picture')
       );
```

Geben Sie die Konstante MMDB_STORAGE_TYPE_INTERNAL an, wenn Sie ein Objekt aus einer Serverdatei als BLOB in einer Datenbanktabelle speichern wollen. Soll das Objekt gespeichert werden und sein Inhalt in der Serverdatei verbleiben, geben Sie die Konstante MMDB_STORAGE_TYPE_EXTERNAL an. MMDB_STORAGE_TYPE_INTERNAL hat einen ganzzahligen Wert von 1. MMDB_STORAGE_TYPE_EXTERNAL hat einen ganzzahligen Wert von 0.

Im folgenden C-Anwendungsprogramm wird beispielsweise ein Audioclip in einer Serverdatei gespeichert. Der Quelleninhalt des Audioclips befindet sich bereits in einer Serverdatei. Die Speicheroperation stellt den Dateinamen in die Datenbank und ermöglicht dadurch den Zugriff von SQL-Anweisungen auf die Datei.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2AUDIO(
         CURRENT SERVER,
          '/employee/sounds/ajones.wav',
         'WAVE',
          :hvStorageType,             /* store audio in server file */
          'Anita''s voice')
       );
```

Format für die Speicherung angeben

Beim Speichern eines Objekts müssen Sie sein Format angeben. Die Formate, die angegeben werden können, sind in Tabelle 8 auf Seite 111 aufgelistet. Die Extender speichern das Abbild-, Audio- oder Videoobjekt im gleichen Format wie die Quelle. Bei Abbildern haben Sie die Möglichkeit anzugeben, dass der Image Extender das Format des gespeicherten Abbilds umsetzen soll. Soll das Abbildformat umgesetzt werden, müssen Sie das Format des Quellenabbilds und des Zielabbilds angeben. Das Zielabbild entspricht dem gespeicherten Abbild.

Format für die Speicherung ohne Umsetzung angeben

Geben Sie das Format der Quelle des Abbild-, Audio- oder Videoobjekts an, wenn das Objekt ohne Formatumsetzung gespeichert werden soll. Die folgende Anweisung in einem C-Anwendungsprogramm speichert beispielsweise ein Bitmap-Abbild (BMP) in einer Datenbanktabelle. Der Inhalt der Quelle befindet sich in einer Serverdatei. Das Zielabbild hat das gleiche Format wie die Quelle.

```
EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2IMAGE(
         CURRENT SERVER,
          '/employee/images/ajones.bmp',
          'BMP',                          /*image in BMP format */
          '',
          'Anita''s picture')
       );
```

Sie können auch einen Nullwert oder eine leere Zeichenfolge als Format angeben. Für den Image Extender ist auch die Zeichenfolge ASIS zulässig. In diesem Fall bestimmt der Extender das Format, indem er die Quelle prüft.

NULL oder ASIS als erkennbare Formate verwenden: Geben Sie einen Nullwert, eine leere Zeichenfolge oder die Zeichenfolge ASIS nur an, wenn der Extender das Format erkennen kann, d. h., wenn es eines der Formate ist, die in Tabelle 8 auf Seite 111 für den Extender aufgeführt sind. Ist dies nicht der Fall, kann der Extender das Format nicht speichern.

Formate und Umsetzungsoptionen für die Speicherung mit Formatumsetzung angeben

Geben Sie das Format für die Quellen- und Zielabbilder an, wenn Sie ein Abbild mit Formatumsetzung speichern wollen. In Tabelle 8 auf Seite 111 ist aufgelistet, welche Formatumsetzungen zulässig sind.

Darüber hinaus können Sie Umsetzungsoptionen für zusätzliche Änderungen angeben, z. B. Drehung oder Komprimierung, die für das gespeicherte Abbild angewendet werden sollen. Die Umsetzungsoption wird über einen Parameter und einen zugeordneten Wert angegeben. Die Parameter und die zulässigen Werte sind in Tabelle 9 auf Seite 112 aufgelistet. Sie können mehrere Änderungen für ein gespeichertes Abbild anfordern, indem Sie mehrere Parameter/Wert-Paare angeben.

Im folgenden Beispiel wird ein Bitmap-Abbild (BMP), dessen Inhalt sich in einer Serverdatei befindet, beim Speichern in einer Datenbanktabelle in das GIF-Format umgewandelt.

```
EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2IMAGE(
         CURRENT SERVER,
          '/employee/images/ajones.bmp',
          'BMP',                          /* source format */
          'GIF',                          /* target format */
          '',
          'Anita''s picture')
       );
```

Im folgenden Beispiel wird das Abbild aus dem vorherigen Beispiel beim Speichern in einer Datenbanktabelle in das GIF-Format umgesetzt. Darüber hinaus wird beim Speichern ein Ausschnitt des Abbilds mit einer Breite von 110 Pixel und einer Höhe von 150 Pixel erstellt. Zusätzlich wird es unter Verwendung der LZW-Komprimierung komprimiert.

```
EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2IMAGE(
         CURRENT SERVER,
          '/employee/images/ajones.bmp',
          'BMP',                          /* source format */
          'GIF',                          /* target format */
          '-x 110 -y 150 -c 14',          /* conversion options */
          '/employee/images/ajones.gif',
          'Anita''s picture')
       );
```

Objekt mit benutzerdefinierten Attributen speichern

Beim Speichern eines Abbild-, Audio- oder Videoobjekts können Sie nicht nur die Formate verwenden, die den Extendern bekannt sind. Sie können auch Ihr eigenes Format angeben. Da den Extendern dieses Format nicht bekannt ist, müssen Sie die Attribute des Quellenobjekts angeben. Ordnen Sie die Attributwerte in einer Attributstruktur zu.Die Attributstruktur muss im Datenfeld der Variablen mit dem Datentyp LONG VARCHAR FOR BIT DATA in der benutzerdefinierten Funktion gespeichert werden.

Der UDF-Code auf dem Server erwartet Daten immer im „Big-Endian"-Format. Das Big-Endian-Format ist ein Format, das von den meisten UNIX-Plattformen verwendet wird. Wenn Sie ein Objekt im „Little-Endian-Format" speichern, müssen Sie die benutzerdefinierten Attributdaten vorbereiten, so dass der UDF-Code auf dem Server das Objekt korrekt verarbeiten kann. Das Little-Endian-Format ist ein Format, das normalerweise auf einer Intel®-Plattform oder einer anderen Mikroprozessorplattform verwendet wird. (Auch wenn Sie das Objekt nicht im Little-Endian-Format speichern, ist es zu empfehlen, die benutzerdefinierten Attributdaten vorzubereiten.) Verwenden Sie die API DBiPrepareAttrs, um Attribute für Abbildobjekte vorzubereiten. Verwenden Sie die API DBaPrepareAttrs, um Attribute von Audioobjekten vorzubereiten. Verwenden Sie die API DBvPrepare-Attrs, um die Attribute von Videoobjekten vorzubereiten.

Die folgenden Anweisungen in einem C-Anwendungsprogramm speichern beispielsweise eine Zeile, mit der ein Abbild in einer Datenbanktabelle eingefügt wird. Das Quellenabbild, das sich in einer Serverdatei befindet, hat ein benutzerdefiniertes Format, eine Höhe von 640 Pixel und eine Breite von 480 Pixel. Beachten Sie, dass die Attribute vorbereitet werden, bevor das Abbild gespeichert wird.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
   struct {
        short len;
        char data[400];
        }hvImgattrs;
EXEC SQL END DECLARE SECTION;

DB2IMAGEATTRS    *pimgattr;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

pimgattr = (DB2IMAGEATTRS *) hvImgattrs.data;
strcpy(pimgattr→format,"FormatI");
```

```
pimgattr→width=640;
pimgattr→height=480;
hvImgattrs.len=sizeof(DB2IMAGEATTRS);

DBiPrepareAttrs(pimgattr);

DBEXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2IMAGE(
         CURRENT SERVER,
          '/employee/images/ajones.bmp',
         :hvStorageType,
          'Anita''s picture',
          :hvImgattrs,                  /* user-specified attributes */
          CAST(NULL as LONG VARCHAR)
       );
```

Die folgende Anweisung in einem C-Anwendungsprogramm speichert eine Zeile, mit der ein Audioclip in einer Datenbanktabelle eingefügt wird. Der Quellenaudioclip befindet sich in einer Serverdatei und hat ein benutzerdefiniertes Format, eine Abtastrate von 44,1 kHz sowie zwei aufgezeichnete Kanäle. Der Audioclip ist nicht MIDI, so dass leere Zeichenfolgen für Spurnamen und Instrumente angegeben werden.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
struct (
       short len;
       char data[600];
}hvAudattr;
EXEC SQL END DECLARE SECTION;

MMDBAudioAttrs          *paudiattr;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

paudioattr=(MMDBAudioAttrs *) hvAudattr.data;
strcpy(paudioAttr→cFormat,"FormatA");
paudioAttr→ulSamplingRate=44100;
paudioAttr→usNumChannels=2;
hvAudattrs.len=sizeof(MMDBAudioAttrs);

DBaPrepareAttrs(paudioAttr);

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2AUDIO(
         CURRENT SERVER,
          '/employee/sounds/ajones.aud',
         :hvStorageType,
          'Anita''s voice',
          :hvAudattr)                  /* user-specified attributes */
       );
```

Piktogramm speichern (nur für Abbild und Video)

Wenn Sie ein Abbild in Ihrem eigenen Format speichern, können Sie auch ein **Piktogramm**, d. h. eine verkleinerte Version des Abbilds, speichern. Sie können die Größe und das Format des Piktogramms steuern. Beim Speichern eines Abbilds in einem Format, das der Image Extender erkennt, wird vom Extender automatisch ein Piktogramm für das Objekt generiert und gespeichert. Der Image Extender erstellt ein Piktogramm im GIF-Format in der Größe 112 x 84 Pixel.

Wenn Sie ein Videoobjekt in Ihrem eigenen Format speichern, können Sie auch ein Piktogramm speichern, das das Videoobjekt symbolisiert. Beim Speichern eines Videoobjekts in einem Format, das der Video Extender erkennt, wird vom Extender automatisch ein generisches Piktogramm für das Objekt gespeichert. Der Video Extender erstellt ein Piktogramm im GIF-Format in der Größe 108 x 78 Pixel.

Geben Sie an Stelle des Piktogramms einen Nullwert oder eine leere Zeichenfolge an, wenn beim Speichern eines Abbild- oder Videoobjekts mit benutzerdefinierten Attributen kein Piktogramm gespeichert werden soll.

Generieren Sie das Piktogramm in Ihrem Programm; die Extender stellen keine APIs zur Verfügung, mit denen Piktogramme generiert werden können. Erstellen Sie im Programm eine Struktur für das Piktogramm, und geben Sie die Piktogrammstruktur in der UDF an.

Die folgenden Anweisungen in einem C-Anwendungsprogramm speichern eine Zeile, mit der ein Videoclip in einer Datenbanktabelle eingefügt wird. Der Quellenvideoclip, dessen Inhalt sich in einer Serverdatei befindet, hat ein benutzerdefiniertes Format. Der Inhalt des Videoclips verbleibt auf dem Server. Auf ihn wird von der Tabelle aus verwiesen. Außerdem wird ein Piktogramm eines repräsentativen Videovollbilds gespeichert.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
struct {
       short len;
      char data[4000];
}hvVidattrs;
   struct {
       short len;
       char data[10000];
       }hvThumbnail;
EXEC SQL END DECLARE SECTION;

MMDBVideoAttrs         *pvideoAttr;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

pvideoAttr=(MMDBVideoAttrs *)hvVidattrs.data;
strcpy(pvideoAttr->cFormat,"Formatv");
hvVidattrs.len=sizeof(MMDBVideoAttrs);

/* Generate thumbnail and assign data in video structure */

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2VIDEO(
         CURRENT SERVER,
          '/employee/videos/ajones.vid',
         :hvStorageType,
```

```
          'Anita''s video',
          :hvVidattrs,
          :hvThumbnail)                    /* Thumbnail*/
       );
```

Kommentar speichern

Ein Kommentar wird mit einem Abbild-, Audio- oder Videoobjekt gespeichert, indem der Kommentar in der UDF-Anforderung angegeben wird. Ein Kommentar ist ein Text mit freiem Format vom Datentyp LONG VARCHAR, der bis zu 32 700 Byte lang sein kann. Geben Sie an Stelle des Kommentars einen Nullwert oder eine leere Zeichenfolge an, wenn beim Speichern eines Objekts kein Kommentar gespeichert werden soll. Wenn Sie einen Nullwert angeben, müssen Sie daran denken, ihn mit dem Datentyp LONG VARCHAR zu versehen.

Die folgenden Anweisungen in einem C-Anwendungsprogramm speichern beispielsweise einen Kommentar mit einem Videoclip.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2VIDEO(
         CURRENT SERVER,
          '/employee/videos/ajones.mpg',
         'MPEG1',
         :hvStorageType,
          'Anita''s video')               /* comment */
       );
```

Die folgenden Anweisungen in einem C-Anwendungsprogramm speichern ein Abbild ohne Kommentar.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2IMAGE(
         CURRENT SERVER,
          '/employee/images/ajones.bmp',
          'GIF',
         :hvStorageType,
          Cast(NULL as LONG VARCHAR)           /* no comment */       );
```

Abbild-, Audio- oder Videoobjekt abrufen

Verwenden Sie die benutzerdefinierte Funktion (UDF) ″Content″ in einer SQL-Anweisung SELECT, um ein Abbild-, Audio- oder Videoobjekt aus einer Datenbanktabelle abzurufen. Sie können das Objekt in einen Clientpuffer, eine Clientdatei oder eine Serverdatei abrufen.

Formate der UDF ″Content″ zum Abrufen

Die benutzerdefinierte Funktion (UDF) ″Content″ ist mehrfach belegt. Dies bedeutet, dass sie abhängig von der Verwendung der UDF unterschiedliche Formate hat. Folgende Formate stehen zur Verfügung:

Format 1: Objekt in einen Clientpuffer oder eine Clientdatei abrufen:

```
Content(
     handle,                    /* object handle */
  );
```

Format 2: Segment eines Objekts in einen Clientpuffer oder eine Clientdatei abrufen:

```
Content(
     handle,                    /* object handle */
     offset,                    /* offset where retrieval begins */
     size                       /* number of bytes to retrieve */
  );
```

Format 3: Objekt in eine Serverdatei abrufen:

```
Content(
     handle,                /* object handle */
     target_file,           /* server file name */
     overwrite              /* 0=Do not overwrite target file if it exists */
                            /* 1=Overwrite target file */

  );
```

Darüber hinaus enthält die benutzerdefinierte Funktion ″Content″ die folgenden Formate für Abbildobjekte:

Format 4: Abbild mit Formatumsetzung in einen Clientpuffer oder eine Clientdatei abrufen:

```
Content(
     handle,                /* object handle */
     target format          /* target format */
       );
```

Format 5: Objekt mit Formatumsetzung in eine Serverdatei abrufen:

```
Content(
     handle,                /* object handle */
     target_file,           /* server file name */
     overwrite,             /* 0=Do not overwrite target file if it exists */
                            /* 1=Overwrite target file */
     target format          /* target format */

  );
```

Format 6: Objekt mit Formatumsetzung und zusätzlichen Änderungen in einen Clientpuffer oder eine Clientdatei abrufen:

```
Content(
     handle,                /* object handle */
     target format,         /* target format */
     conversion_options     /* conversion options */
  );
```

Format 7: Objekt mit Formatumsetzung und zusätzlichen Änderungen in eine Serverdatei abrufen:

```
Content(
     handle,                /* object handle */
     target_file,           /* server file name */
     overwrite,             /* 0=Do not overwrite target file if it exists */
                            /* 1=Overwrite target file */
     target format,         /* target format */
     conversion_options     /* conversion options */
  );
```

Die folgende Anweisung ruft beispielsweise ein Abbild aus der Tabelle 'employee' in eine Datei auf dem Server ab. (Dies entspricht Format 3.)

```
EXEC SQL SELECT CONTENT(                 /* retrieval UDF */
                PICTURE,                 /* image handle */
       '/employee/images/ajones.bmp',    /* target file */
       1)                                /* overwrite target file */
FROM EMPLOYEE
WHERE NAME = 'Anita Jones';
```

Die folgenden Anweisungen in einem C-Anwendungsprogramm rufen ein Abbild aus der Tabelle 'employee' in eine Datei auf dem Server ab. Das Format des Abbilds wird beim Abrufen umgesetzt. (Dies entspricht Format 5.)

```
EXEC SQL BEGIN DECLARE SECTION;
  char hvImg_fname[255];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT(                 /* retrieval UDF */
                PICTURE,                 /* image handle */
       '/employee/images/ajones.bmp',    /* target file */
       1,                                /* overwrite target file */
        'GIF')                           /* target format */
INTO :hvImg_fname
FROM EMPLOYEE
WHERE NAME = 'Anita Jones';
```

Objekt in den Client abrufen

Sie können die benutzerdefinierte Funktion ″Content″ verwenden, um ein Abbild-, Audio- oder Videoobjekt ohne Formatumsetzung in einen Clientpuffer oder eine Clientdatei abzurufen. Darüber hinaus haben Sie bei Abbildern die Möglichkeit anzugeben, dass der Image Extender beim Abrufen das Format des Abbilds umsetzen soll.

Objekt ohne Formatumsetzung in einen Client abrufen

Verwenden Sie einen LOB-Zeiger, um ein Abbild-, Audio- oder Videoobjekt in einen Clientpuffer abzurufen, oder rufen Sie das LOB ab. Verwenden Sie eine Dateireferenzvariable, um ein Abbild-, Audio- oder Videoobjekt in eine Clientdatei abzurufen.

Das Abrufen eines Abbild-, Audio- oder Videoobjekts in einen Clientpuffer unter Verwendung einer Hostvariablen oder in eine Clientdatei unter Verwendung einer Dateireferenzvariablen wird verwendet, wenn der Inhalt des Objekts als BLOB (Binary Large Object) in einer Datenbanktabelle gespeichert ist. Befindet sich der Inhalt in einer Serverdatei, ist es möglicherweise günstiger, den Inhalt von der Server- in die Clientdatei zu kopieren.

Geben Sie die Kennung des Objekts an. Wahlweise können Sie außerdem die relative Adresse (beginnend mit Byte 1), an der die Abrufoperation beginnen soll, und die Anzahl der abzurufenden Byte angeben.

Die folgenden Anweisungen in einem C-Anwendungsprogramm verwenden einen LOB-Zeiger mit dem Namen `audio_loc`, um einen Audioclip in einen Clientpuffer abzurufen.

```
EXEC SQL BEGIN DECLARE SECTION;
  SQL TYPE IS BLOB_LOCATOR audio_loc;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT(
        SOUND)                                  /* audio handle */
     INTO :audio_loc
  FROM EMPLOYEE
     WHERE NAME = 'Anita Jones';
```

Stellen Sie sicher, dass genügend UDF-Speicher zur Verfügung steht: Wenn Sie ein Objekt in einen Clientpuffer abrufen, müssen Sie sicherstellen, dass der Parameter UDF_MEM_SZ in der Datenbankmanagerkonfiguration auf einen Wert von mindestens 4 MB gesetzt ist. Sie können den Parameter UDF_MEM_SZ mit dem DB2-Befehl UPDATE DATABASE MANAGER CONFIGURATION aktualisieren. Informationen zum Befehl UPDATE DATABASE MANAGER befinden sich in der Veröffentlichung *DB2 Command Reference*.

Abbild mit Umsetzung in einen Client abrufen

Verwenden Sie einen LOB-Zeiger, um ein gespeichertes Abbild mit Formatumsetzung in einen Clientpuffer abzurufen, oder rufen Sie das LOB ab. Verwenden Sie eine Dateireferenzvariable, um ein gespeichertes Abbild mit Formatumsetzung in eine Clientdatei abzurufen.

Das Abrufen eines Abbilds in einen Clientpuffer unter Verwendung einer Hostvariablen oder in eine Clientdatei unter Verwendung einer Dateireferenzvariablen wird verwendet, wenn der Inhalt des Abbilds als BLOB (Binary Large Object) in einer Datenbanktabelle gespeichert ist. Befindet sich der Inhalt in einer Serverdatei, ist es möglicherweise günstiger, den Inhalt von der Server- in die Clientdatei zu kopieren.

Wenn Sie ein Abbild mit Formatumsetzung abrufen, müssen Sie sein Zielformat (d. h. das Format, in das umgesetzt wurde) angeben. In Tabelle 8 auf Seite 111 sind die gültigen Formatumsetzungen aufgelistet. Außerdem können Sie Umsetzungsoptionen für zusätzliche Änderungen angeben, z. B. Drehung oder Maßstabsänderung, die für das abgerufene Abbild angewendet werden sollen. In Tabelle 9 auf Seite 112 werden die Umsetzungsoptionen aufgelistet, die Sie angeben können.

Die folgenden Anweisungen in einem C-Anwendungsprogramm rufen beispielsweise ein Abbild in eine Clientdatei ab. Das Quellenabbild hat das Bitmap-Format und ist als BLOB in einer Datenbanktabelle gespeichert. Das abgerufene Abbild wird in das GIF-Format umgesetzt und auf die Dreifache der Originalgröße vergrößert.

```
EXEC SQL BEGIN DECLARE SECTION;
  SQL TYPE IS BLOB_FILE Img_file;
EXEC SQL END DECLARE SECTION;

strcpy (Img_file.name, "/employee/images/ajones.gif");
Img_file.name_length= strlen(Img_file.name);
Img_file.file_options= SQL_FILE_CREATE;

EXEC SQL SELECT CONTENT(
                 PICTURE,                    /* image handle */
        'GIF',                               /* target format */
        '-s 3.0')                            /* conversion options */
     INTO :Img_file,
  FROM EMPLOYEE
     WHERE NAME = 'Anita Jones';
```

Objekt in eine Serverdatei abrufen

Sie können die benutzerdefinierte Funktion "Content" verwenden, um ein Abbild-, Audio- oder Videoobjekt ohne Formatumsetzung in ein Serverdatei abzurufen. Darüber hinaus können Sie die benutzerdefinierte Funktion "Content" verwenden, um ein Abbild mit Formatumsetzung in eine Serverdatei abzurufen.

Geben Sie die Kennung des Objekts, den Zieldateinamen und einen Überschreibungsanzeiger an, wenn Sie ein Abbild-, Audio- oder Videoobjekt ohne Formatumsetzung in eine Datei auf dem Server abrufen. Der Überschreibungsanzeiger teilt dem Extender mit, ob die Zieldatei mit den abgerufenen Daten überschrieben werden soll, falls sie bereits auf dem Server existiert. Existiert die Zieldatei nicht, erstellt sie der Extender auf dem Server.

Bei Angabe eines Überschreibungsanzeigers mit dem Wert 1 überschreibt der Extender die Zieldatei mit den abgerufenen Daten. Bei Angabe eines Überschreibungsanzeigers mit dem Wert 0 überschreibt der Extender die Zieldatei nicht. Daher werden die Daten nicht abgerufen.

Der Überschreibungsanzeiger wird ignoriert, wenn das abzurufende Objekt als BLOB in einer Datenbanktabelle gespeichert ist. In diesem Fall wird die Zieldatei unabhängig von der Angabe im Überschreibungsanzeiger erstellt oder überschrieben.

Wenn Sie ein Objekt in eine Serverdatei abrufen, gibt es den Namen der Serverdatei zurück. Die folgende Anweisung in einem C-Anwendungsprogramm ruft beispielsweise ein Video in eine Datei auf dem Server ab. Der Dateiname der Serverdatei wird in der Hostvariablen hvVid_fname gespeichert.

```
EXEC SQL BEGIN DECLARE SECTION;
struct{
       short len;
      char data[250];
      }hvVid_fname[;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT(
                 VIDEO,                       /* video handle */
        '/employee/videos/ajones.mpg',        /* server file */
        1)                                    /* overwrite target file */
     INTO :hvVid_fname;
  FROM EMPLOYEE
     WHERE NAME = 'Anita Jones';
```

Die benutzerdefinierte Funktion "Content" kann verwendet werden, um ein Objekt ohne Formatumsetzung in eine Serverdatei abzurufen, wenn der Inhalt des Abbilds als BLOB (Binary Large Object) in einer Datenbanktabelle gespeichert ist. Ist das Objekt in einer Serverdatei gespeichert, ist es möglicherweise günstiger, den Inhalt der Quellendatei in die Zieldatei zu kopieren.

Geben Sie die Kennung des Abbilds, den Zieldateinamen, einen Überschreibungsanzeiger für das Ziel und das Zielformat an, wenn Sie ein Abbild mit Formatumsetzung in eine Serverdatei abrufen. In Tabelle 8 auf Seite 111 ist aufgelistet, welche Formatumsetzungen zulässig sind. Sie können als Zielformat auch einen Nullwert, eine leere Zeichenfolge oder die Zeichenfolge ASIS angeben. In diesem Fall hat das abgerufene Abbild das gleiche Format wie die Quelle.

Die folgenden Anweisungen in einem C-Anwendungsprogramm rufen beispielsweise ein Abbild in eine Datei auf dem Server ab. Das Quellenabbild hat das Bitmap-Format und ist als BLOB in einer Datenbanktabelle gespeichert. Das abgerufene Abbild wird in das GIF-Format umgesetzt. Der Dateiname der Serverdatei wird in der Hostvariablen hvImg_fname gespeichert.

```
EXEC SQL BEGIN DECLARE SECTION;
struct{
       short len;
         char [400];
        }hvImg_fname[;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT(
                 PICTURE,                     /* image handle */
        '/employee/images/ajones.gif',        /* target file */
        1,                                    /* overwrite target file */
        'GIF')                                /* target format */
     INTO :hvImg_fname
  FROM EMPLOYEE
     WHERE NAME = 'Anita Jones';
```

Auf die Serverdatei muss Zugriff bestehen: Wenn Sie ein Objekt in eine Serverdatei abrufen, müssen Sie den vollständig qualifizierten Namen der Zieldatei angeben. Alternativ dazu müssen Sie sicherstellen, dass die Umgebungsvariablen DB2IMAGEEXPORT, DB2AUDIOEXPORT und DB2VIDEOEXPORT so gesetzt sind, dass eine unvollständige Angabe des Dateinamens korrekt aufgelöst werden kann.

Attribute abrufen und verwenden

Wenn Sie ein Abbild-, Audio- oder Videoobjekt in einer Datenbank speichern, speichert der Extender auch die Attribute des Objekts in der Datenbank. Wenn Sie ein Objekt aktualisieren, aktualisiert der Extender auch die Attribute, die in der Datenbank gespeichert sind. Diese Attribute können in Abfragen verwendet werden.

Für jedes von den Extendern verwaltete Attribut werden benutzerdefinierte Funktionen erstellt. Daher können Sie benutzerdefinierte Funktionen (UDFs) in SQL-Anweisungen angeben, um auf Objektattribute zuzugreifen und sie zu verwenden. In Tabelle 10 werden die Attribute, die die Extender verwenden, und ihre benutzerdefinierten Funktionen aufgelistet. Außerdem gibt die Tabelle die Objekttypen für jedes Attribut an. Manche dieser Attribute, wie beispielsweise das Format oder der Dateiname eines Objekts, werden für alle Objekttypen verwendet. Diese Attribute sind Abbild-, Audio- und Videoobjekten zugeordnet. Andere Attribute, wie beispielsweise die Abtastrate oder die Komprimierungsart, sind bestimmten Objekttypen (z. B. Audio- oder Videoobjekt) zugeordnet.

Tabelle 10. Von den DB2 Extendern verwaltete Attribute. Auf die Attribute kann über ihre UDFs zugegriffen werden.

Attribut	UDF	Image	Audio	Video
Name der Serverdatei, in der das Objekt gespeichert ist	Filename	x	x	x
Benutzer-ID des Benutzers, der das Objekt gespeichert hat	Importer	x	x	x
Datum und Uhrzeit, zu dem bzw. der das Objekt gespeichert wurde	ImportTime	x	x	x
Größe des Objekts in Byte	Size	x	x	x
Benutzer-ID des Benutzers, der das Objekt zuletzt aktualisiert hat	Updater	x	x	x
Datum und Uhrzeit, zu dem bzw. der das Objekt zuletzt aktualisiert wurde	UpdateTime	x	x	x
Format des Objekts (z. B. GIF oder MPEG1)	Format	x	x	x
Kommentare zum Objekt	Comment	x	x	x
Höhe des Objekts in Pixel	Height	x		x
Breite des Objekts in Pixel	Width	x		x
Anzahl der Farben im Objekt	NumColors	x		
Piktogrammgroßes Abbild des Objekts	Thumbnail	x		x
Anzahl der Byte, die pro Sample in einem Ton oder einer Tonspur eines Videos zurückgegeben werden	AlignValue		x	x
Anzahl der für jedes Sample verwendeten Bit	BitsPerSample		x	x
Anzahl der aufgezeichneten Kanäle	NumChannels		x	x
Dauer in Sekunden	Duration		x	x
Abtastrate (in Samples pro Sekunde)	SamplingRate		x	x
Durchschnittlicher Übertragungswert in Byte pro Sekunde	BytesPerSec		x	

Tabelle 10. Von den DB2 Extendern verwaltete Attribute (Forts.). Auf die Attribute kann über ihre UDFs zugegriffen werden.

Attribut	UDF	Image	Audio	Video
Nummer der Tonspur für das Instrument	FindInstrument		x	
Spurnummer der angegebenen Spur	FindTrackName		x	
Name der aufgezeichneten Instrumente	GetInstruments		x	
Spurnummern und Namen der aufgezeichneten Instrumente	GetTrackNames		x	
Taktimpulse pro Sekunde des Audioclips	TicksPerSec		x	
Taktimpulse pro Viertelnote des Audioclips	TicksPerQNote		x	
Streckungsverhältnis	AspectRatio			x
Videokomprimierungsformat (z. B. MPEG1)	CompressType			x
Vollbilder pro Sekunde Durchsatz	FrameRate			x
Maximaler Durchsatz in Byte pro Sekunde	MaxBytesPerSec			x
Anzahl der Tonspuren	NumAudioTracks		x	x
Anzahl der Vollbilder	NumFrames			x
Anzahl der Videospuren	NumVideoTracks			x

Sie können eine Attribut-UDF im Ausdruck einer SELECT-Klausel oder in der Suchbedingung einer WHERE-Klausel in einer SQL-Anweisung verwenden. Geben Sie bei der Anforderung der UDF den Namen der Spalte in der Datenbanktabelle an, die die Kennung des Objekts enthält.

Die folgende Anweisung verwendet beispielsweise die UDF ″Updater″ in der SELECT-Klausel einer SQL-Anweisung SELECT, um die Benutzer-ID des Benutzers abzurufen, der ein Abbild in der Tabelle 'employee' zuletzt aktualisiert hat:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvUpdatr[30];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT UPDATER(PICTURE)
     INTO :hvUpdatr
  FROM EMPLOYEE
     WHERE NAME = 'Anita Jones';
```

Die folgende Anweisung verwendet die UDF ″Filename″ in der SELECT-Klausel einer SQL-Anweisung SELECT und die UDF ″NumAudioTracks″ in der WHERE-Klausel, um in der Tabelle 'employee' gespeicherte Videos zu suchen, die über Tonspuren verfügen.

```
EXEC SQL BEGIN DECLARE SECTION;
 char hvVid_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(VIDEO)
     INTO :hvVid_fname
  FROM EMPLOYEE
     WHERE NUMAUDIOTRACKS(VIDEO)>0;
```

Kommentar abrufen

Verwenden Sie die UDF "Comment", um mit einem Abbild-, Audio- oder Videoobjekt gespeicherte Kommentare abzurufen. Geben Sie beim Abrufen eines Kommentars für ein Objekt die Spalte in der Datenbanktabelle an, die die Kennung des Objekts enthält. Die folgende Anweisung ruft beispielsweise einen mit einem Audioclip gespeicherten Kommentar in der Tabelle 'employee' ab.

```
EXEC SQL BEGIN DECLARE SECTION;
struct {
       short len;
       char data[32700];
       }hvComment
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT COMMENT(SOUND)
     INTO :hvComment
  FROM EMPLOYEE
     WHERE NAME = 'Anita Jones';
```

Darüber hinaus können Sie die UDF "Comment" als Vergleichselement in der Klausel WHERE einer SQL-Abfrage verwenden. Die folgende Anweisung ruft beispielsweise die Dateinamen aller Abbilder, die mit dem Kommentar „touched up" (retuschiert) versehen wurden, in der Tabelle 'employee' ab.

```
EXEC SQL BEGIN DECLARE SECTION;
struct {
       short len;
      char data[250];
       }hvImg_fname
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
      INTO :hvImg_fname
  FROM EMPLOYEE
     WHERE COMMENT(PICTURE)
           LIKE '%touch%up';
```

Abbild-, Audio- oder Videoobjekt aktualisieren

Verwenden Sie die benutzerdefinierte Funktion (UDF) "Content" in einer SQL-Anweisung UPDATE, um ein Abbild-, Audio- oder Videoobjekt in einer Datenbanktabelle zu aktualisieren. Verwenden Sie die UDF "Replace" in einer SQL-Anweisung UPDATE, um ein Abbild-, Audio- oder Videoobjekt in einer Datenbanktabelle sowie einen dem Objekt zugeordneten Kommentar zu aktualisieren. In beiden Fällen aktualisiert der Extender die Attribute, die dem Objekt zugeordnet sind.

Sie können ein Objekt aktualisieren, das in einer Datenbanktabelle als BLOB oder in einer Serverdatei gespeichert ist. (Dabei verweist bei der zweiten Möglichkeit die Datenbank auf die Datei.) Die Quelle der Aktualisierung kann sich in einem Puffer oder einer Clientdatei oder in einer Serverdatei befinden.

In Tabelle 8 auf Seite 111 werden die Formate aufgelistet, in denen Abbild-, Audio- und Videoobjekte aktualisiert werden können. Sie können jedoch auch ein Objekt aktualisieren, dessen Format vom Extender nicht erkannt wird. In diesem Fall wurden die Attribute des Objekts vom Benutzer beim Speichern des Objekts angegeben. Beim Aktualisieren eines Objekts mit benutzerdefinierten Attributen müssen Sie die aktualisierten Attribute des Objekts angeben.

Verwenden Sie eine ContentA-UDF in einer SQL-Anweisung UPDATE, um ein Abbild-, Audio- oder Videoobjekt mit benutzerdefinierten Attributen in einer Datenbank zu aktualisieren. Verwenden Sie eine ReplaceA-UDF in einer SQL-Anweisung UPDATE, um ein Abbild-, Audio- oder Videoobjekt in einer Datenbanktabelle mit benutzerdefinierten Attributen zu aktualisieren und einen Objektkommentar zu aktualisieren. Beim Aktualisieren eines Objekts mit benutzerdefinierten Attributen müssen Sie die Attribute des Objekts, sein Format und bei Videoobjekten das Kompressionsformat angeben.

Sie können auch das Piktogramm für ein gespeichertes Abbild oder Video aktualisieren.

Aktualisierungsoperation festschreiben: Nach dem Aktualisieren eines Abbild-Audio- oder Videoobjekts in einer Datenbank sollten Sie für die Arbeitseinheit eine COMMIT-Operation durchführen. Hierdurch werden die Sperren, die die Extender verwenden, freigegeben, so dass Sie weitere Aktualisierungsoperationen für das gespeicherte Objekt durchführen können.

Formate der UDF ″Content″ für die Aktualisierung

Die benutzerdefinierte Funktion (UDF) ″Content″ ist mehrfach belegt. Dies bedeutet, dass sie abhängig von der Verwendung der UDF unterschiedliche Formate hat. Folgende Formate stehen zur Verfügung:

Format 1: Objekt aus einem Clientpuffer oder einer Clientdatei aktualisieren:

```
Content(
     handle,                    /* object handle */
     content,                   /* object content */
     source_format,             /* source format */
     target_file                /* target file name for storage in file */
                                /* server or NULL for storage in table as BLOB */
  );
```

Format 2: Objekt aus einer Serverdatei aktualisieren:

```
Content(
     handle,                    /* object handle */
     source_file,               /* server file name */
     source_format,             /* source format */
     stortype                   /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                                /* in file server */
                                /* MMDB_STORAGE_TYPE_INTERNAL=store as a BLOB*/
  );
```

Format 3: Objekt mit benutzerdefinierten Attributen aus einem Clientpuffer oder einer Clientdatei aktualisieren:

```
Content(
     handle,                    /* object handle */
     content,                   /* object content */
     target_file,               /* target file name for storage in file server */
                                /* or NULL for storage in table as BLOB */
     attrs,                     /* user-supplied attributes */
     thumbnail                  /* thumbnail (image and video only) */
  );
```

Format 4: Objekt mit benutzerdefinierten Attributen aus einer Serverdatei aktualisieren:

```
Content(
     handle,                    /* object handle */
     source_file,               /* source file name */
     stortype,                  /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                                /* in file server*/
                                /* MMDB_STORAGE_TYPE_INTERNAL=store */
                                /* as a BLOB*/
     attrs,                     /* user-supplied attributes */
     thumbnail                  /* thumbnail (image and video only) */
  );
```

Für Abbildobjekte verfügt die UDF ″Content″ über die folgenden zusätzlichen Formate:

Format 5: Abbild aus einem Clientpuffer oder einer Clientdatei mit Formatumsetzung aktualisieren:

```
Content(
     handle,                    /* object handle */
     content,                   /* object content */
     source format,             /* source format */
     target format,             /* target format */
     target_file                /* target file name for storage in file server */
                                /* or NULL for storage in table as BLOB */
  );
```

Format 6: Objekt aus einer Serverdatei mit Formatumsetzung aktualisieren:

```
Content(
     handle,                    /* object handle */
     source_file,               /* server file name */
     source format,             /* source format */
     target format,             /* target format */
     target_file                /* target file name for storage in file server */
                                /* or NULL for storage in table as BLOB */
  );
```

Format 7: Abbild aus einem Clientpuffer oder einer Clientdatei mit Formatumsetzung und zusätzlichen Änderungen aktualisieren:

```
Content(
     handle,                    /* object handle */
     content,                   /* object content */
     source format,             /* source format */
     target format,             /* target format */
     conversion_options,        /* conversion options */
     target_file                /* target file name for storage in file server */
                                /* or NULL for storage in table as BLOB */
  );
```

Format 8: Objekt aus einer Serverdatei mit Formatumsetzung und zusätzlichen Änderungen aktualisieren:

```
Content(
     handle,                    /* object handle */
     source_file,               /* server file name */
     source format,             /* source format */
     target format,             /* target format */
     conversion_options,        /* conversion options */
     target_file                /* target file name for storage in file server */
                                /* or NULL for storage in table as BLOB */
  );
```

Die folgenden Anweisungen in einem C-Anwendungsprogramm aktualisieren beispielsweise ein Abbild in der Tabelle 'employee'. Der Quelleninhalt für die Aktualisierung befindet sich in einer Serverdatei mit dem Namen ajones.bmp. Das aktualisierte Abbild wird in der Tabelle 'employee' als BLOB (Binary Large Object) gespeichert. (Dies entspricht dem oben aufgeführten Format 2.)

```
EXEC SQL UPDATE EMPLOYEE
   SET PICTURE=CONTENT(
               PICTURE,                              /*image handle*/
                  '/employee/newimg/ajones.bmp',     /*source file */
            'ASIS',                                  /*keep the image format*/
                  '');                               /*store image in DB as BLOB*/
       WHERE NAME='Anita Jones';
```

Die folgenden Anweisungen in einem C-Anwendungsprogramm aktualisieren das gleiche Abbild wie im vorigen Beispiel. In diesem Fall wird jedoch das Abbild beim Aktualisieren vom BMP-Format in das GIF-Format umgesetzt. (Dies entspricht dem oben aufgeführten Format 6.)

```
EXEC SQL UPDATE EMPLOYEE
   SET PICTURE=CONTENT(
               PICTURE,                             /*image handle*/
                  '/employee/newimg/ajones.bmp',     /*source file */
                  'BMP',                             /*source format*/
                  'GIF',                             /*target format*/
                  '');                               /*store image in DB as BLOB*/
       WHERE NAME='Anita Jones';
```

Formate der UDF "Replace" für die Aktualisierung

Die benutzerdefinierte Funktion (UDF) "Replace" ist mehrfach belegt. Dies bedeutet, dass sie abhängig von der Verwendung der UDF unterschiedliche Formate hat. Folgende Formate stehen zur Verfügung:

Format 1: Objekt aus einem Clientpuffer oder einer Clientdatei aktualisieren und dessen Kommentar ebenfalls aktualisieren:

```
Replace(
     handle,                    /* object handle */
     content,                   /* object content */
     source_format,             /* source format */
     target_file,               /* target file name for storage in file */
     comment                    /* user comment */
  );
```

Format 2: Objekt aus einer Serverdatei aktualisieren und dessen Kommentar ebenfalls aktualisieren:

```
Replace(
     handle,                    /* object handle */
     source_file,               /* server file name */
     source_format,             /* source format */
     stortype,                  /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                                /* in file server*/
                                /* MMDB_STORAGE_TYPE_INTERNAL=store as a BLOB*/
     comment                    /* user comment */
  );
```

Format 3: Objekt mit benutzerdefinierten Attributen aus einer Serverdatei ersetzten und dessen Kommentar ebenfalls ersetzen:

```
Replace(
     handle,                    /* object handle */
     content,                   /* object content */
     target_file,               /* target file name for storage in file */
                                /* or NULL for storage in table as BLOB */
     comment,                   /* user comment */
     attrs,                     /* user-supplied attributes */
     thumbnail                  /* thumbnail */
  );
```

Format 4: Objekt mit benutzerdefinierten Attributen aus einer Serverdatei speichern:

```
Replace(
     handle,                    /* object handle */
     source_file,               /* server file name */
     stortype,                  /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                                /* in file server*/
                                /* MMDB_STORAGE_TYPE_INTERNAL=store as a BLOB*/
     comment,                   /* user comment */
     attrs,                     /* user-supplied attributes */
     thumbnail                  /* thumbnail */
  );
```

Für Abbildobjekte verfügt die UDF ″Replace″ über die folgenden zusätzlichen Formate:

Format 5: Abbild aus einem Clientpuffer oder einer Clientdatei mit Formatumsetzung aktualisieren und seinen Kommentar ebenfalls aktualisieren:

```
Replace(
     handle,                    /* object handle */
     content,                   /* object content */
     source_format,             /* source format */
     target_format,             /* target format */
     target_file,               /* target file name for storage in file server */
                                /* or NULL for storage in table as BLOB */
     comment                    /* user comment */
  );
```

Format 6: Objekt aus einer Serverdatei mit Formatumsetzung aktualisieren und dessen Kommentar ebenfalls aktualisieren:

```
Replace(
     handle,                    /* object handle */
     source_file,               /* server file name */
     source_format,             /* source format */
     target_format,             /* target format */
     target_file,               /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                                /* in file server */
                                /* MMDB_STORAGE_TYPE_INTERNAL=store as a BLOB*/
     comment                    /* user comment */
  );
```

Format 7: Abbild aus einem Clientpuffer oder einer Clientdatei mit Formatumsetzung und zusätzlichen Änderungen aktualisieren und seinen Kommentar ebenfalls aktualisieren:

```
Replace(
     handle,                    /* object handle */
     content,                   /* object content */
     source_format,             /* source format */
     target_format,             /* target format */
     conversion_options,        /* conversion options */
     target_file,               /* target file name for storage in file server */
                                /* or NULL for storage in table as BLOB */
     comment                    /* user comment */
  );
```

Format 8: Objekt aus einer Serverdatei mit Formatumsetzung und zusätzlichen Änderungen aktualisieren und dessen Kommentar ebenfalls aktualisieren:

```
Replace(
     handle,                    /* object handle */
     source_file,               /* server file name */
     source_format,             /* source format */
     target_format,             /* target format */
     conversion_options,        /* conversion options */
     target_file,               /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                                /* in file server */
                                /* MMDB_STORAGE_TYPE_INTERNAL=store as a BLOB*/
     comment                    /* user comment */
  );
```

Die folgenden Anweisungen in einem C-Anwendungsprogramm aktualisieren beispielsweise einen Audioclip in der Tabelle 'employee' und aktualisieren den ihm zugeordneten Kommentar. Der Quelleninhalt für die Aktualisierung befindet sich in einer Serverdatei mit dem Namen ajones.wav. Der aktualisierte Audioclip wird in der Tabelle 'employee' ohne Formatumsetzung als BLOB (Binary Large Object) gespeichert (der Audio Extender unterstützt keine Formatumsetzung). Dies entspricht dem oben aufgeführten Format 2.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

EXEC SQL UPDATE EMPLOYEE
   SET SOUND=REPLACE(
                SOUND,                          /*audio handle*/
                '/employee/newaud/ajones.wav', /*source file */
                'WAV',                          /*keep the audio format*/
                :hvStorageType,                 /*store audio in DB as BLOB*/
                'Anita''s new greeting')        /*user comment*/
       WHERE NAME='Anita Jones';
```

Im folgenden Beispiel werden ein Abbild und der ihm zugeordnete Kommentar aktualisiert. Der Quelleninhalt für die Aktualisierung befindet sich in einer Serverdatei. Das aktualisierte Abbild wird in der Tabelle 'employee' als BLOB (Binary Large Object) gespeichert und beim Aktualisieren vom BMP-Format in das GIF-Format umgesetzt. (Dies entspricht dem oben aufgeführten Format 6.)

```
EXEC SQL UPDATE EMPLOYEE
       SET PICTURE=REPLACE(
              PICTURE,                              /*image handle*/
                 '/employee/newimg/ajones.bmp',     /*source file */
                 'BMP',                             /*source format*/
                 'GIF',                             /*target format*/
                 ''                                 /*store image in DB as BLOB*/
                   'Anita''s new picture')
   WHERE NAME='Anita Jones';                        /* user comment */
```

Objekt vom Client aktualisieren

Verwenden Sie eine Hostvariable oder eine Dateireferenzvariable, um ein Abbild-, Audio- oder Videoobjekt von einem Clientpuffer oder einer Clientdatei zu aktualisieren.

Befindet sich die Quelle für die Aktualisierung in einer Clientdatei, sollten Sie eine Dateireferenzvariable verwenden, um seinen Inhalt zu übertragen. Die Anweisungen im folgenden Beispiel für ein C-Anwendungsprogramm definieren eine Dateireferenzvariable mit dem Namen `Audio_file` und verwenden sie, um einen Audioclip, der als BLOB in einer Datenbanktabelle gespeichert ist, zu aktualisieren. Die Quelle für die Aktualisierung befindet sich in einer Clientdatei. Beachten Sie, dass das Feld `file_options` der Dateireferenzvariablen auf die Option `SQL_FILE_READ`, d. h. auf Eingabe gesetzt ist. Beachten Sie auch, dass die Dateireferenzvariable als Inhaltsargument für die benutzerdefinierte Funktion "Content" verwendet wird.

```
EXEC SQL BEGIN DECLARE SECTION;
  SQL TYPE IS BLOB_FILE Audio_file;
EXEC SQL END DECLARE SECTION;

strcpy (Audio_file.name, "/employee/newsound/ajones.wav");
Audio_file.name_length= strlen(Audio_file.name);
Audio_file.file_options= SQL_FILE_READ;

EXEC SQL UPDATE EMPLOYEE
   SET SOUND=CONTENT(
                SOUND,
                :Audio_file                 /*file reference variable*/
                'WAVE',                     /*keep the image format*/
                CAST(NULL as LONG VARCHAR))

       WHERE NAME='Anita Jones';
```

Befindet sich das Objekt in einem Clientpuffer, sollten Sie eine Hostvariable verwenden, um seinen Inhalt zum Aktualisieren zu übertragen. Im folgenden Beispiel für ein C-Anwendungsprogramm wird eine Hostvariable mit dem Namen `Video_seg` verwendet, um den Inhalt eines Videoclips zum Aktualisieren zu übertragen. Der dem Videoclip zugeordnete Kommentar wird ebenfalls aktualisiert. Der Videoclip wird als BLOB in einer Datenbanktabelle gespeichert.

Beachten Sie, dass die Hostvariable als Inhaltsargument für die benutzerdefinierte Funktion "Replace" verwendet wird.

```
EXEC SQL BEGIN DECLARE SECTION;
  SQL TYPE IS BLOB (2M) Video_seg
EXEC SQL END DECLARE SECTION;

EXEC SQL UPDATE EMPLOYEE
   SET VIDEO=REPLACE(
                VIDEO,
                :Video_seg                  /*host variable*/
          'MPEG1',
                CAST(NULL as LONG VARCHAR),

                'Anita''s new video')
       WHERE NAME='Anita Jones';
```

Stellen Sie sicher, dass genügend UDF-Speicher zur Verfügung steht: Wenn Sie ein Objekt aktualisieren, dessen Inhalt sich in einem Clientpuffer befindet, müssen Sie sicherstellen, dass der Parameter UDF_MEM_SZ in der Datenbankmanagerkonfiguration auf einen Wert von mindestens 4 MB gesetzt ist. Sie können den Parameter UDF_MEM_SZ mit dem DB2-Befehl UPDATE DATABASE MANAGER CONFIGURATION aktualisieren.

Objekt vom Server aktualisieren

Wenn sich der Quelleninhalt für die Aktualisierung eines Abbild-, Audio- oder Videoobjekts in einer Serverdatei befindet, müssen Sie den Dateipfad als Inhaltsargument für die benutzerdefinierte Funktion angeben. Die folgende Anweisung in einem C-Anwendungsprogramm aktualisiert beispielsweise ein Abbild in einer Datenbank. Der Abbildinhalt befindet sich in einer Serverdatei. Die Datenbank zeigt auf die Serverdate. Die Quelle für die Aktualisierung befindet sich ebenfalls in einer Serverdatei.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL UPDATE EMPLOYEE
   SET PICTURE=CONTENT(
                PICTURE,                              /* image handle */
                '/employee/newimg/ajones.bmp',        /* source file */
                  'ASIS',
                :hvStorageType)
       WHERE NAME='Anita Jones';
```

Geben Sie den korrekten Pfad an: Wenn Sie ein Objekt aktualisieren, dessen Quelle sich in einer Serverdatei befindet, können Sie den vollständig qualifizierten oder den relativen Namen der Datei angeben. Wenn Sie einen relativen Namen angeben, müssen Sie sicherstellen, dass die entsprechenden Umgebungsvariablen auf dem DB2-Server den korrekten Pfad für die Datei enthalten. Informationen zum Definieren dieser Umgebungsvariablen befinden sich in Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521.

Datenbank- oder Dateispeicherung für die Aktualisierungen angeben

Sie können ein Abbild-, Audio- oder Videoobjekt aktualisieren, das in einer Datenbanktabelle als BLOB oder in einer Serverdatei (auf die von der Datenbank aus verwiesen wird) gespeichert ist.

Wenn Sie ein Objekt aus einem Clientpuffer oder einer Clientdatei aktualisieren, erfolgt die Speicherung als BLOB oder Serverdatei entsprechend den Angaben, die Sie im Parameter 'filename' machen.Die Angabe eines Dateinamens bedeutet, dass ein Objekt aktualisiert werden soll, dessen Inhalt sich in einer Serverdatei befindet. Wenn Sie keinen Dateinamen angeben, bedeutet dies, dass ein als BLOB in einer Datenbanktabelle gespeichertes Objekt aktualisiert werden soll.

Die folgenden Anweisungen in einem C-Anwendungsprogramm aktualisieren beispielsweise ein Abbild, dessen Inhalt sich in einer Serverdatei befindet. Die Aktualisierungsquelle befindet sich in einem Clientpuffer. Der Kommentar zum Abbild wird ebenfalls aktualisiert.

```
EXEC SQL BEGIN DECLARE SECTION;
  SQL TYPE IS BLOB (2M) Img_buf
EXEC SQL END DECLARE SECTION;

EXEC SQL UPDATE EMPLOYEE
       SET PICTURE=REPLACE(
                 PICTURE,
                   :Img_buf,
                   'ASIS',
                   '/employee/newimg/ajones.bmp',      /*update image in*/
                                                       /*server file*/
                   'Anita''s new picture')
       WHERE NAME='Anita Jones';
```

Geben Sie beim Aktualisieren eines Objekts aus einer Serverdatei die Konstante MMDB_STORAGE_TYPE_INTERNAL an, wenn Sie ein als BLOB in einer Datenbanktabelle gespeichertes Objekt aktualisieren wollen. Wollen Sie ein Objekt aktualisieren, dessen Inhalt sich in der Serverdatei befindet, geben Sie MMD-B_STORAGE_TYPE_EXTERNAL an.

Im folgenden C-Anwendungsprogramm wird beispielsweise ein Audioclip aktualisiert. Der Inhalt des Audioclips befindet sich in einer Serverdatei. Die Quelle für die Aktualisierung befindet sich ebenfalls in einer Serverdatei.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL UPDATE EMPLOYEE
   SET SOUND=CONTENT(
                 SOUND,
                 '/employee/newimg/ajones.wav',
         'WAVE',
                 :hvStorageType)        /*update audio in server file*/
       WHERE NAME='Anita Jones';
```

Format für die Aktualisierung angeben

Beim Aktualisieren eines Objekts müssen Sie sein Format angeben. Die Extender speichern das aktualisierte Abbild-, Audio- oder Videoobjekt im gleichen Format wie die Quelle. Bei Abbildern haben Sie die Möglichkeit anzugeben, dass der Image Extender das Format des aktualisierten Abbilds umsetzen soll. Soll das Abbildformat umgesetzt werden, müssen Sie das Format der Aktualisierungsquelle und des Zielabbilds angeben. Das Zielabbild entspricht dem gespeicherten aktualisierten Abbild.

Format für die Aktualisierung ohne Umsetzung angeben

Geben Sie das Format der Quelle des Abbild-, Audio- oder Videoobjekts an, wenn ein Objekt ohne Formatumsetzung aktualisiert werden soll. Die folgende Anweisung in einem C-Anwendungsprogramm aktualisiert beispielsweise ein Bitmap-Abbild (BMP), dessen Inhalt sich in einer Serverdatei befindet. Das Format des aktualisierten Abbilds wird nicht umgesetzt.

```
EXEC SQL UPDATE EMPLOYEE
   SET PICTURE=CONTENT(
                PICTURE,
                '/employee/newimg/ajones.bmp',
                'BMP',                          /*image format*/
                '')
       WHERE NAME='Anita Jones';
```

Sie können auch einen Nullwert oder eine leere Zeichenfolge als Format angeben. Für den Image Extender ist auch die Zeichenfolge ASIS zulässig. In diesem Fall bestimmt der Extender das Format, indem er die Quelle prüft.

NULL oder ASIS als erkennbare Formate verwenden: Geben Sie einen Nullwert, eine leere Zeichenfolge oder die Zeichenfolge ASIS nur an, wenn der Extender das Format erkennen kann, d. h., wenn es eines der Formate ist, die in Tabelle 8 auf Seite 111 für den Extender aufgeführt sind. Sonst kann der Extender das Objekt nicht aktualisieren.

Formate und Umsetzungsoptionen für die Aktualisierung mit Formatumsetzung angeben

Geben Sie das Format für die Quellen- und Zielabbilder an, wenn Sie ein Abbild mit Formatumsetzung aktualisieren wollen. In Tabelle 8 auf Seite 111 ist aufgelistet, welche Formatumsetzungen zulässig sind.

Darüber hinaus können Sie Umsetzungsoptionen für zusätzliche Änderungen angeben, z. B. Drehung oder Komprimierung, die für das aktualisierte Abbild angewendet werden sollen. Die Umsetzungsoption wird über einen Parameter und einen zugeordneten Wert angegeben. Die Parameter und die zulässigen Werte sind in Tabelle 9 auf Seite 112 aufgelistet. Sie können mehrere Änderungen für das aktualisierte Abbild anfordern, indem Sie mehrere Parameter/Wert-Paare angeben.

Im folgenden Beispiel wird ein Abbild, dessen Inhalt sich in einer Serverdatei befindet, aktualisiert. Die Quelle der Aktualisierung ist im Bitmap-Format (BMP). Das Format wird beim Aktualisieren vom BMP- in das GIF-Format umgesetzt.

```
EXEC SQL UPDATE EMPLOYEE
   SET PICTURE=CONTENT(
                PICTURE,
                '/employee/newimg/ajones.bmp',
                'BMP',                                  /*source format*/
                'GIF',                                  /*target format*/
                '')
       WHERE NAME='Anita Jones';
```

Im folgenden Beispiel wird das gleiche Abbild beim Aktualisieren in das GIF-Format umgesetzt. Darüber hinaus wird das Abbild beim Aktualisieren um 90 Grad im Uhrzeigersinn gedreht.

```
EXEC SQL UPDATE EMPLOYEE
   SET PICTURE=CONTENT(
                PICTURE,
                '/employee/newimg/ajones.bmp',
                'BMP',                                  /*source format*/
                'GIF',                                  /*target format*/
                '-r 1',                                 /* conversion options */
                '')
       WHERE NAME='Anita Jones';
```

Objekt mit benutzerdefinierten Attributen aktualisieren

Wenn Sie ein Abbild-, Audio- oder Videoobjekt aktualisieren, das mit benutzerdefinierten Attributen gespeichert wurde, müssen Sie die Attribute des Inhalts, mit dem aktualisiert wird, angeben. Ordnen Sie die Attributwerte in einer Attributstruktur zu. Die Attributstruktur muss im Datenfeld der Variablen mit dem Datentyp LONG VARCHAR FOR BIT DATA in der benutzerdefinierten Funktion gespeichert werden.

Der UDF-Code auf dem Server erwartet Daten immer im „Big-Endian"-Format. Das Big-Endian-Format ist ein Format, das von den meisten UNIX-Plattformen verwendet wird. Wenn Sie ein Objekt im „Little-Endian-Format" speichern, müssen Sie die benutzerdefinierten Attributdaten vorbereiten, so dass der UDF-Code auf dem Server das Objekt korrekt verarbeiten kann. Das Little-Endian-Format ist ein Format, das normalerweise auf einer Intel-Plattform oder einer anderen Mikroprozessorplattform verwendet wird. (Auch wenn Sie das Objekt nicht im Little-Endian-Format speichern, ist es zu empfehlen, die benutzerdefinierten Attributdaten vorzubereiten.) Verwenden Sie die API DBiPrepareAttrs, um Attribute für Abbildobjekte vorzubereiten. Verwenden Sie die API DBaPrepareAttrs, um Attribute von Audioobjekten vorzubereiten. Verwenden Sie die API DBvPrepareAttrs, um die Attribute von Videoobjekten vorzubereiten.

Die folgenden Anweisungen in einem C-Anwendungsprogramm aktualisieren beispielsweise ein Abbild, dessen Inhalt sich in einer Serverdatei befindet. Das Quellenabbild hat ein benutzerdefiniertes Format und eine Höhe von 640 Pixel sowie eine Breite von 480 Pixel. Beachten Sie, dass die Attribute vorbereitet werden, bevor das Abbild aktualisiert wird.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
   struct {
       short len;
       char data[400];
       }hvImgattrs;
EXEC SQL END DECLARE SECTION;

DB2IMAGEATTRS    *pimgattr;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

pimgattr = (DB2IMAGEATTRS *) hvImgattrs.data;
strcpy(pimgattr→Format,"FormatI");
pimgattr→width=640;
pimgattr→height=480;
hvImgattrs.len=sizeof(DB2IMAGEATTRS);

DBiPrepareAttrs(pimgattr);

EXEC SQL UPDATE EMPLOYEE
```

```
SET PICTURE=REPLACE(
          PICTURE,
          '/employee/newimg/ajones.bmp',
  :hvStorageType,
          'Anita''s new picture',
          :ImgAttrs,                    /*user-supplied attributes*/
   CAST(NULL as LONG VARCHAR))
WHERE NAME='Anita Jones';
```

Piktogramm aktualisieren (nur für Abbild und Video)

Verwenden Sie die benutzerdefinierte Funktion (UDF) ″Thumbnail″, um ein für ein Abbild- oder Videoobjekt gespeichertes Piktogramm zu aktualisieren oder um ein Piktogramm hinzuzufügen, falls dem gespeicherten Abbild- oder Videoobjekt noch keines zugeordnet ist. Geben Sie bei der Verwendung der UDF ″Thumbnail″ die Kennung des Objekts, dessen Piktogramm aktualisiert werden soll, und den Inhalt des aktualisierten oder neuen Piktogramms an.

Generieren Sie das Piktogramm in Ihrem Programm; die Extender stellen keine APIs zur Verfügung, mit denen Piktogramme generiert werden können. Sie können die Größe und das Format des aktualisierten Piktogramms steuern. Erstellen Sie im Programm eine Struktur für das Piktogramm, und geben Sie die Piktogrammstruktur in der UDF an.

Die folgenden Anweisungen in einem C-Anwendungsprogramm aktualisieren beispielsweise das einem gespeicherten Videoclip zugeordnete Piktogramm.

```
EXEC SQL BEGIN DECLARE SECTION;
   struct {
       short len;
       char data[10000];
   }hvThumbnail;
EXEC SQL END DECLARE SECTION;

/*Create thumbnail and store in hvThumbnail*/

EXEC SQL UPDATE employee
   SET picture=Thumbnail(
               picture,
               :hvThumbnail)
   WHERE name='Anita Jones';
```

Sie können ein Piktogramm auch aktualisieren, wenn Sie ein Abbild- oder Videoobjekt mit benutzerdefinierten Attributen aktualisieren. Wenn Sie ein Abbild- oder Videoobjekt mit benutzerdefinierten Attributen aktualisieren, ist es sogar notwendig, dass Sie ein Piktogramm als Eingabe angeben. Geben Sie an Stelle des Piktogramms einen Nullwert oder eine leere Zeichenfolge an, wenn Sie das Piktogramm beim Aktualisieren des Objekts nicht ebenfalls aktualisieren wollen.

Die folgenden Anweisungen in einem C-Anwendungsprogramm aktualisieren einen Videoclip mit benutzerdefinierten Attributen sowie ein dem Videoclip zugeordnetes Piktogramm.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
   struct {
       short len;
       char data[400];
}hvVidattrs;
struct {
       short len;
       char data[10000];
   }hvThumbnail;
```

```
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

MMDBVideoAttrs         *pvideoAttr;
pvideoAttr=(MMDBVideoAttrs *)hvVidattrs.data;
strcpy(pvideoAttr→cformat,"Formatv");
hvVidattrs.len=sizeof(MMDBVideoAttrs);

/* Update video content and thumbnail */

EXEC SQL UPDATE EMPLOYEE
   SET VIDEO=REPLACE(
                VIDEO,
                '/employee/newvid/ajones.mpg',
         :hvStorageType,
                'Anita''s new video',
                :VidAttrs,
                :hvThumbnail)               /*thumbnail*/
       WHERE NAME='Anita Jones';
```

Kommentar aktualisieren

Sie können einen Kommentar separat oder bei der Aktualisierung des zugeordneten Objekts aktualisieren.

Verwenden Sie die benutzerdefinierte Funktion (UDF) "Comment", um einen Kommentar separat zu aktualisieren. Geben Sie den Inhalt des aktualisierten Kommentars und die Tabellenspalte, die die Kennung des Objekts enthält, an. Verwenden Sie eine Hostvariable, um den Inhalt an den Server zu übertragen. Die folgenden Anweisungen deklarieren beispielsweise eine Hostvariable mit dem Namen `hvRemarks` und verwenden sie, um einen bestehenden Kommentar für einen gespeicherten Videoclip zu aktualisieren.

```
EXEC SQL BEGIN DECLARE SECTION;
   struct {
       short len;
         char data [40];
  }hvRemarks;
EXEC SQL END DECLARE SECTION;

/* Get the old comment */

EXEC SQL SELECT COMMENT(VIDEO)
     INTO :hvRemarks
  FROM EMPLOYEE
     WHERE NAME = 'Anita Jones';

/* Append to old comment */

hvRemarks.data[Remarks.len]='\0';
hvRemarks.len=strlen(hvRemarks.data);
strcat (hvRemarks.data, "Updated video");
EXEC SQL UPDATE EMPLOYEE
     SET VIDEO=COMMENT(VIDEO, :hvRemarks)
     WHERE NAME = 'Anita Jones';
```

Verwenden Sie die UDF ″Replace″, um einen Kommentar zu aktualisieren, wenn das zugeordnete Objekt aktualisiert wird. Die folgenden Anweisungen aktualisieren beispielsweise einen Videoclip, der in einer Serverdatei gespeichert ist, sowie den ihm zugeordneten Kommentar.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL UPDATE EMPLOYEE
   SET VIDEO=REPLACE(
                VIDEO,
                '/employee/newvid/ajones.mpg',
         'MPEG1',
         :hvStorageType,
                'Anita''s new video')        /*updated comment*/
       WHERE NAME='Anita Jones';
```

Kapitel 11. Abbild-, Audio- oder Videoobjekt anzeigen oder wiedergeben

In diesem Kapitel wird die Verwendung der DB2 Extender-Anwendungsprogrammierschnittstelle zur Anzeige oder Wiedergabe von Abbild-, Audio- oder Videoobjekten beschrieben, die in einer Datenbank gespeichert sind.

Anzeige- oder Wiedergabe-APIs verwenden

Sie können die Extender-APIs verwenden, um ein Abbild oder ein Videovollbild anzuzeigen, das in einer Datenbank gespeichert ist. Sie können die Version eine Abbilds oder Videovollbilds in Piktogrammgröße oder in normaler Größe anzeigen. Sie können außerdem Extender-APIs verwenden, um Audio- oder Videoobjekte wiederzugeben, die in einer Datenbank gespeichert sind.

Verwenden Sie die folgenden APIs, um Objekte anzuzeigen oder wiederzugeben:

API	Verwendung
DBiBrowse	Anzeigen eines Abbilds oder Videovollbilds
DBaPlay	Wiedergeben eines Audioclips
DBvPlay	Wiedergeben eines Videoclips oder Anzeigen eines Videovollbilds

Bei der Anforderung dieser APIs sind folgende Angaben erforderlich:

- Der Name des Anzeige- oder Wiedergabeprogramms
- Ob das anzuzeigende oder wiederzugebende Objekt in einer Datenbanktabelle als BLOB gespeichert ist oder ob es sich in einer Datei befindet, auf die von der Tabelle aus gezeigt wird
- Der Name der Quellendatei oder die Kennung, die in der Datenbanktabelle gespeichert ist
- Ob das Anwendungsprogramm, bevor es mit der Verarbeitung fortfährt, darauf warten soll, dass der Benutzer das Anzeige- oder Wiedergabeprogramm schließt

Anzeige- oder Wiedergabeprogramm identifizieren

Geben Sie den Namen des Abbildbrowsers, der Audiowiedergabeeinheit oder Videowiedergabeeinheit an, der/die verwendet werden soll. Geben Sie nach dem Namen %s an. Der Extender ersetzt %s durch die Datei, die den Objektinhalt enthält.

Sie können auch einen Nullwert angeben, statt ein bestimmtes Anzeige- oder Wiedergabeprogramm zu benennen. In diesem Fall startet der Extender den Standardabbildbrowser bzw. die Standardaudio- oder -videowiedergabeeinheit, der/die in der Umgebungsvariablen DB2IMAGEBROWSER, DB2AUDIOPLAYER oder DB2VIDEOPLAYER angegeben ist. Weitere Informationen darüber, wie die DB2 Extender Umgebungsvariablen verwenden, befinden sich im Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521.

Beispielsweise startet die folgende Anweisung in einem C-Anwendungsprogramm die Standardaudiowiedergabeeinheit, die in der Umgebungsvariablen DB2AUDIOPLAYER angegeben ist:

```
rc = DBaPlay(
     NULL,                                /* use default audio player */
     MMDB_PLAY_FILE,
     "/employee/sounds/ajones.wav",
     MMDB_PLAY_NO_WAIT
     );
```

Die Umgebungsvariable muss ein Programm angeben: Wenn Sie ein Standardanzeige- oder -wiedergabeprogramm anfordern (durch die Angabe eines Nullwerts), stellen Sie sicher, dass die entsprechende Umgebungsvariable ein Anzeige- oder Wiedergabeprogramm angibt. Ist kein Programm angegeben, gibt die API einen Fehlercode zurück.

BLOB oder Dateiinhalt angeben

Sie können ein Objekt anzeigen oder wiedergeben, das in einer Datenbanktabelle als BLOB gespeichert ist oder dessen Inhalt in einer Datei gespeichert ist (und auf die von der Datenbanktabelle aus gezeigt wird). Wenn das Objekt als BLOB gespeichert ist, geben Sie MMDB_PLAY_HANDLE an. Wenn der Objektinhalt in einer Datei gespeichert ist, geben Sie MMDB_PLAY_FILE an. MMDB_PLAY_HANDLE und MMDB_PLAY_FILE sind Konstanten, die durch die Extender definiert werden.

Beispielsweise gibt die folgende Anweisung in einem C-Anwendungsprogramm ein Video wieder, dessen Inhalt in einer Datei gespeichert ist:

```
rc = DBvPlay(
     "explore %s",
     MMDB_PLAY_FILE,                     /* content in file */
     "/employee/videos/ajones.mpg",
     MMDB_PLAY_NO_WAIT
     );
```

Anzeige- und Wiedergabeprogramme akzeptieren normalerweise die Eingabe aus einer Datei. Wenn Sie MMDB_PLAY_FILE angeben, verwendet der Extender den Wert in den Umgebungsvariablen, um den relativen Dateinamen und Pfad der Datei aufzulösen. Anschließend startet der Extender das Anzeigeprogramm und übergibt ihm den Dateinamen. Wenn Sie MMDB_PLAY_HANDLE angeben, extrahiert der Extender den Dateinamen aus der Kennung (vorausgesetzt, dass der Dateiname nicht Null ist). Wenn der Dateinamen in der Kennung Null ist, wird das Objekt als BLOB gespeichert. Der Extender erstellt eine temporäre Datei auf dem Client und kopiert den Inhalt des Objekts aus der Datenbanktabelle in die Clientdatei. Der Extender startet danach das Programm und übergibt ihm den Namen der Datei (oder der temporären Datei), die den Inhalt enthält.

Beispielsweise rufen die folgenden Anweisungen in einem C-Anwendungsprogramm die Kennung eines Abbilds ab, das als BLOB gespeichert ist, und verwenden die Kennung zur Anzeige des Abbilds:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_hdl [251];
EXEC SQL END DECLARE SECTION;

rc = DBiBrowse(
     "ib %s",
     MMDB_PLAY_HANDLE,                   /* content is BLOB */
     hvImg_hdl,
     MMDB_PLAY_NO_WAIT
     );
```

Der Inhalt muss im Zugriff befindlich sein: Stellen Sie sicher, dass das Anzeige- oder Wiedergabeprogramm auf den Objektinhalt zugreifen kann. Wenn sich der Inhalt in einer Serverdatei befindet, für das Programm aber erforderlich ist, dass sich der Inhalt auf dem Client befindet, kopieren Sie die Datei in eine Clientdatei oder verwenden Sie die UDF Content. Wenn der Inhalt als BLOB gespeichert ist, ruft der Extender ihn automatisch auf den Client ab.

Wartestatusanzeiger angeben

Sie können angeben, ob das Anwendungsprogramm, bevor es mit der Verarbeitung fortfährt (d. h. , bevor die API DBiBrowse, DBaPlay oder DBvPlay einen Code zurückgibt), darauf warten soll, dass der Benutzer das Anzeige- oder Wiedergabeprogramm beendet. Wenn das Anwendungsprogramm warten soll, geben Sie MMDB_PLAY_WAIT an. Wenn das Anwendungsprogramm nicht warten soll, geben Sie MMDB_PLAY_NO_WAIT an. MMDB_PLAY_WAIT und MMDB_PLAY-_NO_WAIT sind Konstanten, die durch die Extender definiert werden.

Wenn Sie MMDB_PLAY_WAIT angeben, wird das Anzeige- oder Wiedergabeprogramm in dem gleichen Thread oder Prozess ausgeführt wie Ihr Anwendungsprogramm. Wenn Sie MMDB_PLAY_NO_WAIT angeben, wird das Anzeige- oder Wiedergabeprogramm unabhängig von Ihrem Anwendungsprogramm in einem eigenen Thread oder Prozess ausgeführt.

Beispielsweise führt die folgende Anweisung dazu, dass das Anwendungsprogramm darauf wartet, dass der Benutzer den Abbildbrowser schließt, bevor es mit der Verarbeitung fortfährt:

```
rc = DBiBrowse(
     "explore %s",
     MMDB_PLAY_FILE,
     "/employee/images/ajones.bmp",
     MMDB_PLAY_WAIT                    /* wait for browser to close */
     );
```

Vorsichtig mit der Angabe DBxPlay und MMDB_PLAY_NO_WAIT umgehen: Wenn Sie die API DBaPlay oder DBvPlay aufrufen, erstellt der Extender eine temporäre Datei, wenn eine der folgenden Bedingungen wahr ist:

- Das Objekt ist als BLOB gespeichert
- Der relative Dateiname kann unter Verwendung der Werte in den Umgebungsvariablen nicht aufgelöst werden
- Auf die Datei kann auf der Clientmaschine nicht zugegriffen werden

Die temporäre Datei wird in dem Verzeichnis erstellt, das durch die Umgebungsvariable TMP angegeben ist. Wenn Sie MMDB_PLAY_WAIT angeben, löscht der Extender die temporäre Datei, nachdem das Objekt wiedergegeben wurde. Wenn Sie jedoch MMDB_PLAY_NO_WAIT angeben, wird die temporäre Datei nicht gelöscht. Sie müssen sie selbst löschen.

Piktogramm eines Abbilds oder Videovollbilds anzeigen

Ein Piktogramm ist eine sehr verkleinerte Version eines gespeicherten Abbilds oder Videovollbilds. Wenn Sie ein Abbild in der Datenbank speichern, speichert der Image Extender ein Piktogramm des Abbilds in einer Attributtabelle. Wenn Sie ein Video in der Datenbank speichern, speichert der Video Extender ein generisches Piktogramm, das das Videoobjekt symbolisiert, in einer Attributtabelle.

Standardmäßig ist die Größe des Abbildpiktogramms, das automatisch vom Image Extender erstellt wird, 112 x 84 Pixel. Die Größe des generischen Videopiktogramms, das der Video Extender einfügt, ist 108 x 78 Pixel. Sowohl das Abbildpiktogramm als auch das generische Videopiktogramm werden im GIF-Format gespeichert. Je nach Dichte der Daten im Abbild oder Videovollbild entspricht dies etwa 4,5 KB bis 5 KB an Daten. Wenn Sie ein Abbild oder Video mit vom Benutzer bereitgestellten Attributen speichern oder aktualisieren, können Sie ein Piktogramm mit gewählter Größe und gewähltem Format angeben.

Verwenden Sie die UDF Thumbnail in einer SQL-Anweisung SELECT, um ein Piktogramm aus der Datenbank abzurufen. Verwenden Sie eine Dateireferenzvariable, um das Piktogramm an eine Datei zu übertragen. Bei der Angabe der UDF müssen Sie den Namen der Spalte in der Datenbanktabelle angeben, die die Abbild- oder Videokennung enthält. Verwenden Sie dann die API DBiBrowse, um das Piktogramm des Abbilds oder Videovollbilds anzuzeigen.

Beispielsweise rufen die folgenden Anweisungen ein Piktogrammabbild ab und zeigen es danach an:

```
long rc, outCount;
char Thumbnail_filename[254];
FILE *file_handle;

EXEC SQL BEGIN DECLARE SECTION;
struct {
         short len
       char data[10000];
         }Thumbnail_buffer;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT THUMBNAIL(PICTURE)
     INTO :Thumbnail_buffer
  FROM EMPLOYEE
     WHERE NAME = 'Anita Jones';

strcpy (Thumbnail_filename,"/tmp/ajones.tmb");
file_handle=fopen(Thumbnail_filename,"wb+");
outCount=fwrite(Thumbnail_buffer.data, 1, Thumbnail_buffer.len, file_handle);
fclose(file_handle);
rc = DBiBrowse(
               NULL,                        /* use the default display program */
               MMDB_PLAY_FILE,              /* thumbnail image in file */
               Thumbnail_filename,          /* thumbnail image content */
               MMDB_PLAY_WAIT);             /* wait for user to finish */
```

Abbild oder Videovollbild in normaler Größe anzeigen

Verwenden Sie die API DBiBrowse, um ein Abbild anzuzeigen, das in einer Datenbanktabelle gespeichert ist. Nähere Einzelheiten zur Verwendung dieser API befinden sich im Abschnitt „Anzeige- oder Wiedergabe-APIs verwenden" auf Seite 147.

Verwenden Sie die API DBvGetNextFrame oder DBvSeekFrame, um ein Videovollbild in normaler Größe abzurufen. Das Vollbild ist im YUV-Format auf einem Puffer gespeichert und kann unter Verwendung der API DBvFrameDatato24BitRGB in das RGB-Format umgesetzt werden. Hängen Sie eine Kopfzeile an das umgesetzte Vollbild an (beispielsweise eine Kopfzeile für einen Dateityp BMP) und schreiben Sie die Kopfzeile und die Vollbilddaten in eine Datei. Verwenden Sie dann die API DBiBrowse, um den Inhalt der Datei anzuzeigen. Das „Szenenwechsel bei Videoobjekten ermitteln" auf Seite 19, enthält nähere Informationen zur Verwendung der APIs DBvGetNextFrame, DBvSeekNextFrame und DBvFrameDatato24BitRGB sowie zur Anzeige eines Videovollbilds.

Ton oder Video wiedergeben

Verwenden Sie die API DBaPlay, um einen Ton wiederzugeben, der in einer Datenbanktabelle gespeichert ist. Verwenden Sie die API DBvPlay, um ein Video wiederzugeben, das in einer Datenbanktabelle gespeichert ist. Nähere Einzelheiten zur Verwendung dieser APIs befinden sich im Abschnitt „Anzeige- oder Wiedergabe-APIs verwenden" auf Seite 147.

Kapitel 12. Abfragen von Abbildern nach Inhalt

Abb. 27 zeigt ein Anwendungsprogramm, mit dem Benutzer nach Abbildern in einer Datenbank suchen können, indem sie ein visuelles Beispiel, d. h. ein Abbild, das eine vorherrschende Farbe oder ein vorherrschendes Texturmuster hat, als Suchkriterium verwenden. Mit einer solchen Anwendung können Benutzer ein Abbild als Eingabe für die Suche zur Verfügung stellen. Die Anwendung gleicht dann die Farbe oder Textur des Quellenabbilds mit der Farbe oder Textur der gespeicherten Abbilder ab und gibt als Ergebnis die Abbilder zurück, deren Farbe oder Textur mit der der Eingabe am meisten übereinstimmt.

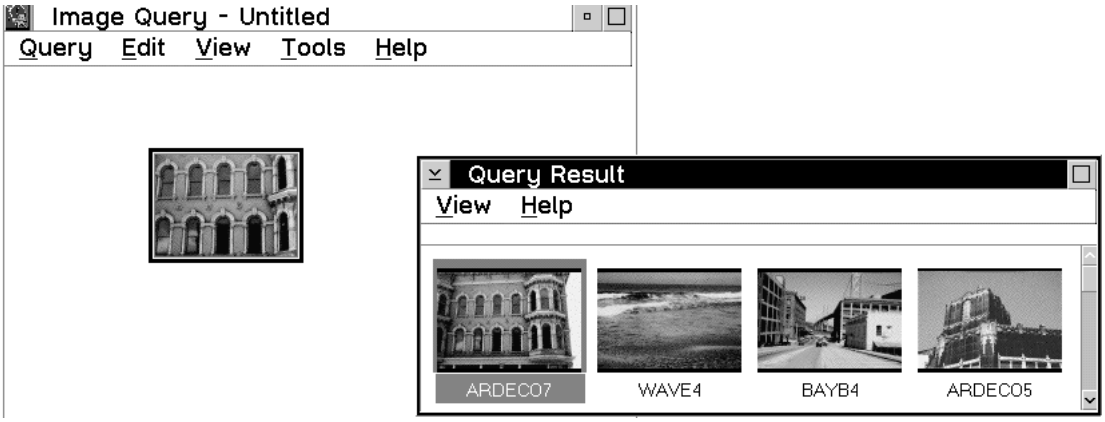


Abbildung 27. Abfrage anhand des Abbildinhalts. Die Farbe oder Textur eines visuellen Beispiels wird für die Suche nach Abbildern verwendet, die in einer Datenbanktabelle gespeichert sind. Diese Abbildung wurde dem amerikanischen Buch als Beispiel entnommen und daher nicht übersetzt.

Diese Fähigkeit, Abbilder anhand ihrer visuellen Merkmale abzufragen, wird **Abfrage anhand des Abbildinhalts (QBIC = Query by Image Content)** genannt [6]. In diesem Kapitel wird beschrieben, wie die APIs und UDFs, die mit dem Image Extender geliefert werden, zum Erstellen von Anwendungen dieser Art verwendet werden. Außerdem wird beschrieben, wie die Befehle und APIs, die mit dem Image Extender geliefert werden, zum Ausführen von QBIC-Verwaltungstasks verwendet werden.

Anhand des Abbildinhalts abfragen

Um ein Abbild anhand des Abbildinhalts abzufragen, führen Sie folgende Schritte aus:

1. Erstellen Sie einen QBIC-Katalog für die Abbilder.
2. Katalogisieren Sie die Abbilder. Das heißt, fügen Sie Einträge für die Abbilder zum Katalog hinzu und speichern Sie die Werte für Abbildmerkmale.

   Im Abschnitt „QBIC-Kataloge" auf Seite 57 befindet sich eine Beschreibung der QBIC-Kataloge und Abbildmerkmale.
3. Erstellen Sie eine Abfrage. Die Abfrage identifiziert die als Suchkriterien zu verwendenden Merkmale, deren Werte und deren Wertigkeiten (d. h., die Gewichtung, die auf die einzelnen Merkmale gelegt wird). Sie können diese Abfrageattribute in einer Zeichenfolge, der so genannten Abfragezeichenfolge, angeben.

6. Der Image Extender enthält Software, die von der University of California, Berkeley, und deren Mitarbeitern entwickelt wurde.

Als Alternative können Sie ein Abfrageobjekt erstellen und diese Attribute dem Abfrageobjekt zuordnen. Sie können dann die Abfragezeichenfolge sichern und erneut verwenden.

4. Führen Sie die Abfrage aus. Wenn Sie die Abfrage ausführen, geben Sie als Eingabe eine Abfragezeichenfolge an oder identifizieren Sie ein Abfrageobjekt für die Abfrage. Geben Sie in jedem Fall die Abbilder an, die gesucht werden sollen. Sie können in beiden Fällen die Abfrage von der DB2-Befehlszeile aus oder innerhalb eines Programms übergeben.

Daraufhin berechnet der Image Extender die Merkmalwerte für die Abfrage. Er vergleicht den Wert mit dem Merkmalwerten, die im QBIC-Katalog für die Zielabbilder gespeichert sind. Anschließend berechnet er dann ein Ergebnis, das angibt, wie ähnlich sich die Merkmalwerte der Zielabbilder und der Quelle sind.

Sie können angeben, dass der Image Extender die Abbilder zurückgibt, deren Merkmalwerte denen der Quelle am ähnlichsten sind. Sie können außerdem angeben, dass der Image Extender die Ähnlichkeitsergebnisse von einem oder mehreren Abbildern zurückgibt.

QBIC-Kataloge verwalten

Bevor Abbilder nach Inhalt abgefragt werden können, müssen sie in einem QBIC-Katalog katalogisiert sein. Ein QBIC-Katalog hält die Daten zu visuellen Merkmalen von Abbildern.

Sie erstellen einen QBIC-Katalog für jede Abbildspalte in einer Benutzertabelle, die Sie für die Abfrage anhand des Inhalts verfügbar machen wollen. Für jede Abbildspalte in einer Benutzertabelle kann nicht mehr als ein QBIC-Katalog definiert sein, mehrere Spalten können nicht denselben QBIC-Katalog gemeinsam benutzen.

Beim Erstellen eines QBIC-Katalogs geben Sie die Merkmale an, für die der Image Extender Daten speichern soll. Sie geben außerdem an, ob der Image Extender ein Abbild automatisch katalogisieren soll. Das automatische Katalogisieren bedeutet, dass der Image Extender automatisch Einträge für ein Abbild im Katalog erstellt, wenn das Abbild in einer Benutzertabelle gespeichert wird. Wenn das Abbild nicht automatisch katalogisiert wird, müssen Sie es manuell katalogisieren. Das bedeutet, dass Sie explizit im Image Extender angeben müssen, dass für das Abbild Einträge im Katalog erstellt werden sollen.

Nach dem Erstellen eines QBIC-Katalogs können Sie folgende Aktionen ausführen:

- Den Katalog für nachfolgende Aktionen öffnen.
- Die Einstellung für das Katalogisieren von 'automatisch' in 'manuell' oder von 'manuell' in 'automatisch' ändern.
- Merkmale zum Katalog hinzufügen. Hierdurch werden die Merkmale identifiziert, für die der Image Extender Daten speichern soll.
- Merkmale aus dem Katalog löschen.
- Informationen zum Katalog abrufen, wie z. B. den Namen der Benutzertabelle und Spalte, die zum Katalog gehören, oder die Merkmale, für die Daten im Katalog gespeichert werden.
- Abbilder manuell im Katalog katalogisieren.
- Ein Abbild aus dem Katalog entfernen (d. h. die Einträge für das Abbild aus dem Katalog löschen).
- Abbilder erneut katalogisieren.

- Den Katalog neu verteilen (d. h., wenn Knoten in einem partitionierten Datenbanksystem hinzugefügt oder gelöscht werden).
- Den Katalog schließen.
- Den Katalog löschen.

Zum Ausführen dieser Tasks, einschließlich Erstellen eines QBIC-Katalogs können Sie die APIs verwenden, die vom Image Extender zur Verfügung gestellt werden. Sie können außerdem viele der Tasks ausführen, indem Sie den db2ext-Befehlszeilenprozessor verwenden.

QBIC-Katalog erstellen

Verwenden Sie die API QbCreateCatalog oder den Befehl CREATE QBIC CATALOG, um einen QBIC-Katalog zu erstellen. Um einen Katalog erstellen zu können, müssen Sie der Eigner der Benutzertabelle sein, deren Abbilder katalogisiert werden.Darüber hinaus müssen Sie über die Berechtigung CREATE TABLE für die Datenbank verfügen, die den Katalog enthalten wird. Die Benutzertabelle und Abbildspalte müssen für den Image Extender aktiviert sein, bevor Sie einen QBIC-Katalog für die Abbilder in dieser Spalte erstellen.

Beim Erstellen eines QBIC-Katalogs müssen Sie

- die Benutzertabelle und Spalte angeben, die die Abbilder enthalten, die katalogisiert werden sollen.
- angeben, ob die Abbilder automatisch katalogisiert werden. Das automatische Katalogisieren bedeutet, dass der Image Extender ein Abbild katalogisiert, nachdem das Abbild in einer Benutzertabelle gespeichert ist. Der Extender prüft in regelmäßigen Abständen, ob ein Abbild auf das Katalogisieren wartet. Sie geben den Zeitraum in Sekunden an, indem Sie den Wert der Umgebungsvariablen DB2CATALOGDELAY setzen. Der Wert kann zwischen 1 und einem extrem großen Wert liegen. Der Standardwert ist 60 Sekunden.

  Das manuelle Katalogisieren bedeutet, dass Sie explizit anfordern müssen, dass der Image Extender ein Abbild katalogisiert. (Weitere Informationen zum manuellen Katalogisieren eines Abbilds befinden sich im Abschnitt „Abbild manuell katalogisieren” auf Seite 161.)

Benutzertabelle und Spalte müssen aktiviert sein: Die Benutzertabelle und Spalte müssen für den Image Extender aktiviert sein, bevor Sie einen QBIC-Katalog für die Abbilder in dieser Spalte erstellen. (Informationen zum Aktivieren von Benutzertabellen und Spalten für den Image Extender befinden sich in Kapitel 6, „Datenobjekte für Extender-Daten vorbereiten”, auf Seite 83.)

Verwendung der API: Wenn Sie die API QbCreateCatalog verwenden, geben Sie das automatische oder manuelle Katalogisieren mit Hilfe des Werts autoCatalog an. Der Wert 1 gibt das automatische Katalogisieren an, der Wert 0 das manuelle Katalogisieren.

Beispielsweise erstellt die folgende Anweisung einen QBIC-Katalog für die Abbilder in der Spalte 'picture' der Tabelle 'employee'. Die Abbilder werden automatisch katalogisiert, wenn sie in der Tabelle 'employee' gespeichert werden:

```
SQLINTEGER autoCatalog=1;                       /* automatic cataloging */

rc=QbCreateCatalog(
                 "employee",                    /* user table */
                 "picture",                     /* image column */
                  autoCatalog);                 /* auto catalog setting */
```

Verwendung der Befehlszeile: Wenn Sie den Befehl CREATE QBIC CATALOG eingeben, geben Sie das automatische Katalogisieren mit der Option ON an. Das manuelle Katalogisieren geben Sie mit der Option OFF an. OFF ist der Standardwert.

Beispielsweise wird mit dem folgenden Befehl der gleiche QBIC-Katalog erstellt wie im vorangegangenen Beispiel mit der API:

```
CREATE QBIC CATALOG employee picture on
```

QBIC-Katalog sichern: Der Image Extender speichert einen QBIC-Katalog in Dateien. Sie sollten diese Dateien in regelmäßigen Abständen sichern für den Fall, dass Sie den Katalog wiederherstellen müssen. Auf einem AIX-, HP-UX- oder Sun Solaris-Server befinden sich die Dateien im Verzeichnis `/home/exemplareigner/dmb/qbic`, wobei *exemplareigner* die Benutzer-ID des Exemplareigners angibt. Auf einem Windows-Server befinden sich die Dateien im Verzeichnis `\ziel\instance\exemplarname\qbic`, wobei *ziel* für das Verzeichnis steht, in dem der Image Extender installiert ist, und *exemplarname* für den Namen des Extender-Exemplars.

QBIC-Katalog öffnen

Sie müssen einen QBIC-Katalog öffnen, um weitere Aktionen ausführen zu können, die den Katalog ändern. Beispielsweise müssen Sie einen QBIC-Katalog öffnen, bevor Sie ein Merkmal zum Katalog hinzufügen.

Um einen QBIC-Katalog zu öffnen, verwenden Sie die API QbOpenCatalog oder den Befehl OPEN QBIC CATALOG. Beim Öffnen eines QBIC-Katalogs müssen Sie

- die Benutzertabelle und die Abbildspalte für den Katalog angeben.
- den Modus angeben, in dem der Katalog geöffnet werden soll (mit dem Befehl OPEN QBIC CATALOG wird der Modus implizit angegeben). Sie können einen Katalog für Leseoperationen öffnen, wie z. B. Suchen nach Abbildern anhand des Inhalts. Sie können einen Katalog auch für Aktualisierungsoperationen öffnen, wie z. B. Hinzufügen eines Merkmals. Sie müssen über die Berechtigung SELECT für die Benutzertabelle verfügen, um den Katalog für Leseoperationen öffnen zu können. Sie müssen über die Berechtigung UPDATE für die Benutzertabelle verfügen, um den Katalog für Aktualisierungsoperationen öffnen zu können.

Falls ein Katalog bereits geöffnet ist: Sie können keinen Katalog für Aktualisierungsoperationen öffnen, wenn der Katalog in einer anderen Sitzung zum Aktualisieren geöffnet ist. Wenn Sie einen QBIC-Katalog öffnen, schließt der Image Extender jeden QBIC-Katalog, der bereits in der aktuellen Sitzung geöffnet ist.

Verwendung der API: Wenn Sie die API QbOpenCatalog verwenden, geben Sie explizit den Modus an, in dem der Katalog geöffnet werden soll. Geben Sie Folgendes an:

- Den API-Parameter qbiRead, um den Katalog für Leseoperationen zu öffnen.
- Den API-Parameter qbiUpdate, um den Katalog für Aktualisierungsoperationen zu öffnen.

QbiRead und QbiUpdate sind Konstanten, die in der Kopfdatei für QBIC, dmbqbapi.h, definiert sind.

Sie müssen außerdem auf die Katalogkennung zeigen. Die Katalogkennung hat den QBIC-spezifischen Datentyp QbCatalogHandle. Dieser Datentyp ist auch in der Datei dmbqbapi.h definiert. Der Image Extender gibt den Wert für die Katalogkennung als Ausgabe der API zurück.

Beispielsweise öffnet der folgende API-Aufruf einen QBIC-Katalog für Leseoperationen:

```
SQLINTEGER mode;
QbCatalogHandle  *CatHdl;

mode=qbiRead;                                   /* open catalog for */
                                                /* read operations */
rc=QbOpenCatalog(
                 "employee",                    /* user table */
                 "picture",                     /* image column */
                 mode,                          /* open catalog mode */
                 &CatHdl);                      /* catalog handle */
```

Verwendung der Befehlszeile: Wenn Sie den Befehl OPEN QBIC CATALOG eingeben, versucht der Image Extender, den Katalog für Aktualisierungsoperationen zu öffnen. Ist der Katalog momentan zum Aktualisieren in einer anderen Sitzung geöffnet, öffnet der Image Extender den Katalog für Leseoperationen.

Beispielsweise wird mit dem folgenden Befehl ein QBIC-Katalog geöffnet. Der Image Extender versucht, ihn für Aktualisierungsoperationen zu öffnen:

```
OPEN QBIC CATALOG employee picture
```

Katalog nach dem Beenden von QBIC-bezogenen Vorgängen schließen: Wenn Sie einen QBIC-Katalog öffnen, ordnet der Image Extender Ressourcen, wie z. B. Hauptspeicher, zu. Schließen Sie den Katalog, wenn Sie die QBIC-bezogenen Vorgänge beendet haben. Dadurch werden die zugeordneten Ressourcen freigegeben.

Einstellung für das automatische Katalogisieren ändern

Verwenden Sie die API QbSetAutoCatalog oder den Befehl SET QBIC AUTOCATALOG, um vom automatischen Katalogisieren zum manuellen Katalogisieren zu wechseln und umgekehrt. Der QBIC-Katalog muss für Aktualisierungsoperationen geöffnet sein, bevor Sie die Katalogeinstellung ändern können.

Die Änderung ist nicht rückwirkend: Wenn Sie die Einstellung zum automatischen Katalogisieren ändern, gilt dies nur für Abbilder, die nach der Änderung zur Benutzertabellenspalte hinzugefügt werden. Abbilder, die bereits in der Benutzertabellenspalte gespeichert sind, sind davon nicht betroffen. Ändern Sie beispielsweise die Einstellung von manuellem Katalogisieren in automatisches Katalogisieren, werden nur die Abbilder automatisch katalogisiert, die nach der Änderung zur Benutzertabellenspalte hinzugefügt werden. Wenn Sie Abbilder katalogisieren wollen, die bereits in der Tabellenspalte sind, müssen Sie sie manuell katalogisieren. (Weitere Informationen zum manuellen Katalogisieren eines Abbilds befinden sich im Abschnitt „Abbild manuell katalogisieren” auf Seite 161.)

Verwendung der API: Wenn Sie die API QbSetAutoCatalog verwenden, geben Sie die Kennung des QBIC-Katalogs an (die Kennung wird zurückgegeben, wenn Sie den Katalog mit Hilfe der API QbOpenCatalog öffnen). Geben Sie außerdem den autoCatalog-Wert 1 für das automatische Katalogisieren oder den Wert 0 für das manuelle Katalogisieren an.

Im folgenden Beispiel ist das manuelle Katalogisieren für einen QBIC-Katalog angegeben, der den Abbildern in der Spalte 'picture' der Tabelle 'employee' zugeordnet ist. Beachten Sie, dass der QBIC-Katalog zunächst für Aktualisierungsoperationen geöffnet wird.

```
SQLINTEGER mode;
SQLINTEGER autoCatalog=0;                       /* manual cataloging */

QbCatalogHandle  *CatHdl;

mode=qbiUpdate;                                 /* open catalog for */
                                                /* update */
/* Open a QBIC catalog */
rc=QbOpenCatalog(
                 "employee",                    /* user table */
                 "picture",                     /* image column */
                 mode,                          /* open catalog mode */
                 &CatHdl);                      /* catalog handle */

/* Change the auto catalog setting */
rc=QbSetAutoCatalog(
                CatHdl,                         /* catalog handle */
                  autoCatalog);                 /* auto catalog flag */
```

Verwendung der Befehlszeile: Wenn Sie den Befehl SET QBIC AUTOCATALOG eingeben, geben Sie das automatische Katalogisieren mit der Option ON an. Das manuelle Katalogisieren geben Sie mit der Option OFF an. Der Befehl gilt für den momentan geöffneten Katalog.

Beispielsweise schaltet der folgende Befehl das automatische Katalogisieren für den momentan geöffneten QBIC-Katalog aus (OFF).

```
SET QBIC AUTOCATALOG off
```

Merkmal zu einem QBIC-Katalog hinzufügen

Verwenden Sie die API QbAddFeature oder den Befehl ADD QBIC FEATURE, um ein Merkmal zu einem QBIC-Katalog hinzuzufügen. Sie müssen mindestens ein Merkmal zu einem QBIC-Katalog hinzufügen, bevor Sie ein Abbild darin katalogisieren können. Der QBIC-Katalog muss für Aktualisierungsoperationen geöffnet sein, bevor Sie ein Merkmal hinzufügen können.

Wenn Sie ein Merkmal zu einem Katalog hinzufügen, geben Sie den Namen des hinzuzufügenden Merkmals an (Merkmalnamen werden in Tabelle 11 aufgelistet).

Tabelle 11. QBIC-Merkmalnamen

Merkmalname	Beschreibung
QbColorFeatureClass	Durchschnittsfarbe
QbColorHistogramFeatureClass	Histogrammfarbe
QbDrawFeatureClass	Positionsgebundene Farbe
QbTextureFeatureClass	Textur

Abbilder müssen möglicherweise erneut katalogisiert werden: Wenn Sie ein Merkmal zu einem QBIC-Katalog hinzufügen, speichert der Image Extender die Daten zum neuen Merkmal für die bereits katalogisierten Abbilder nicht automatisch, auch wenn das automatische Katalogisieren eingeschaltet ist. Um die Daten zu einem neuen Merkmal für bereits katalogisierte Abbilder einzufügen, müssen Sie die Abbilder erneut katalogisieren (siehe „Abbild erneut katalogisieren" auf Seite 163).

Verwendung der API: Wenn Sie die API QbAddFeature verwenden, müssen Sie die Kennung des QBIC-Katalogs zusätzlich zum Merkmalnamen angeben. Beachten Sie die Verwendung der Konstanten qbiMaxFeatureName für die Länge des Merkmalnamens. Die Konstante ist in der Kopfdatei für QBIC, dmbqbapi.h, mit dem Wert 50 definiert.

Im folgenden Beispiel wird die API QbAddFeature verwendet, um das Merkmal 'Histogrammfarbe' zu einem QBIC-Katalog hinzuzufügen:

```
char                featureName[qbiMaxFeatureName];

QbCatalogHandle  CatHdl;

strcpy(featureName,"QbColorHistogramFeatureClass");

rc=QbAddFeature(
                CatHdl,                     /* catalog handle */
                featureName);               /* feature name */
```

Verwendung der Befehlszeile: Der Befehl ADD QBIC FEATURE gilt für den momentan geöffneten Katalog. Im folgenden Beispiel wird der Befehl verwendet, um das Merkmal 'positionsgebundene Farbe' zum momentan geöffneten Katalog hinzuzufügen:

```
ADD QBIC FEATURE QbDrawFeatureClass
```

Merkmal aus einem QBIC-Katalog löschen

Verwenden Sie die API QbRemoveFeature oder den Befehl REMOVE QBIC FEATURE, um ein Merkmal aus einem QBIC-Katalog zu löschen. Der Image Extender löscht die Katalogtabelle für das Merkmal. Daraus ergibt sich, dass Daten für dieses Merkmal nicht gespeichert werden, wenn Sie ein Abbild katalogisieren. Der QBIC-Katalog muss für Aktualisierungsoperationen geöffnet sein, bevor Sie ein Merkmal löschen können.

Wenn Sie ein Merkmal aus einem Katalog löschen, geben Sie den Namen des zu löschenden Merkmals an.

Verwendung der API: Wenn Sie die API QbRemoveFeature verwenden, müssen Sie die Kennung des QBIC-Katalogs zusätzlich zum Merkmalnamen angeben.

Im folgenden Beispiel wird die API QbRemoveFeature verwendet, um das Merkmal 'Histogrammfarbe' aus einem QBIC-Katalog zu löschen:

```
char                featureName[qbiMaxFeatureName];

QbCatalogHandle  CatHdl;

strcpy(featureName,"QbColorHistogramFeatureClass");

rc=QbRemoveFeature(
                CatHdl,                     /* catalog handle */
                featureName);               /* feature name */
```

Verwendung der Befehlszeile: Der Befehl REMOVE QBIC FEATURE gilt für den momentan geöffneten Katalog. Im folgenden Beispiel wird der Befehl verwendet, um das Merkmal 'positionsgebundene Farbe' aus dem momentan geöffneten QBIC-Katalog zu löschen:

```
REMOVE QBIC FEATURE QbDrawFeatureClass
```

Informationen zu einem QBIC-Katalog abrufen

Sie können die folgenden Informationen zu einem QBIC-Katalog abrufen:

- Den Namen der Benutzertabelle und Abbildspalte, die dem Katalog zugeordnet sind.
- Die Anzahl an Merkmalen, für die Daten im Katalog gespeichert werden, und die Merkmalnamen.
- Ob der Image Extender Abbilder automatisch katalogisiert, wenn sie in der Benutzertabelle gespeichert werden.

Verwenden Sie die API QbGetCatalogInfo, um die Benutzertabelle und Spaltennamen, die Anzahl an Merkmalen und die Einstellung für das automatische Katalogisieren abzurufen. Verwenden Sie die API QbListFeatures, um die Merkmalnamen abzurufen. Sie können auch den Befehl GET QBIC CATALOG INFO verwenden, um alle Informationen abzurufen.

Der QBIC-Katalog muss geöffnet sein, bevor Sie Informationen abrufen können.

Verwendung der API: Wenn Sie die API QbGetCatalogInfo verwenden, müssen Sie die Kennung des QBIC-Katalogs angeben. Sie müssen außerdem auf eine Struktur zeigen, in der der Image Extender die Kataloginformationen zurückgibt. Die Struktur der Kataloginformationen ist in der Kopfdatei für QBIC, dmbqapi.h, wie folgt definiert:

```
typedef struct{
          char       tableName[qbiMaxTableName+1]    /* user table */
          char       columnName[qbiMaxColumnName+1]  /* image column */
          SQLINTEGER featureCount;                   /* number of features */
          SQLINTEGER autoCatalog;                    /* auto catalog flag */
} QbCatalogInfo;
```

Wenn Sie den API-Aufruf eingeben, müssen Sie einen Puffer zuordnen, der die zurückgegebenen Merkmalnamen enthalten soll. Ein Leerzeichen trennt Merkmalnamen, die im Puffer gespeichert werden, voneinander. Sie müssen außerdem die Katalogkennung und die Größe des Puffers für die zurückgegebenen Merkmalnamen angeben. Um die benötigte Puffergröße zu schätzen, können Sie die Merkmalanzahl verwenden, die von der API QbGetCatalogInfo zurückgegeben wird, und sie mit der Länge des längsten Merkmalnamens multiplizieren. Sie können die Konstante qbiMaxFeatureName als Größe des längsten Merkmalnamens verwenden.

Die API-Aufrufe in den folgenden Beispielen rufen Informationen zu einem QBIC-Katalog ab. Beachten Sie, wie die Merkmalanzahl, die von der API QbGetCatalogInfo zurückgegeben wird, und die Namenskonstante qbiMaxFeature verwendet werden, um die Puffergröße für die API QbListFeatures zu berechnen:

```
long  bufSize;
long  count;
char  *featureNames;

QbCatalogHandle  CatHdl;
QbCatalogInfo    catInfo;

/* Get user table name, image column name, feature count, */
/* and auto catalog setting */

rc=QbGetCatalogInfo(
               CatHdl,                    /* catalog handle */
               &catInfo);                 /* catalog info. structure */

/* List feature names */
```

```
bufSize=catInfo.featureCount*qbiMaxFeatureName;
featureNames=malloc(bufSize);

rc=QbListFeatures(
               CatHdl,                      /* catalog handle */
               bufSize                      /* size of buffer */
               count,                       /* feature count */
               featureNames);               /* buffer for feature names */
```

Verwendung der Befehlszeile: Der Befehl GET QBIC CATALOG INFO gilt für den momentan geöffneten Katalog. Im folgenden Beispiel wird der Befehl verwendet, um Informationen zu dem momentan geöffneten QBIC-Katalog abzurufen:

```
GET QBIC CATALOG INFO
```

Abbild manuell katalogisieren

Wenn Sie einen Katalog erstellen, geben Sie an, ob der Image Extender ein Abbild automatisch katalogisieren soll, wenn das Abbild in einer Benutzertabelle gespeichert wird. Wenn ein Abbild nicht automatisch katalogisiert wird, müssen Sie es manuell katalogisieren, nachdem es in der Benutzertabelle gespeichert wurde. Sie können ein einzelnes Abbild oder eine ganze Spalte mit Abbildern katalogisieren.

Einzelnes Abbild manuell katalogisieren

Verwenden Sie die API QbCatalogImage, um ein einzelnes Abbild manuell zu katalogisieren. Sie können ein Abbild nicht mit Hilfe eines Befehls katalogisieren, da es keine Möglichkeit gibt, ein einzelnes Abbild in der Befehlszeile zu identifizieren. Wenn Sie die API verwenden, geben Sie die Katalogkennung und die Abbildkennung an (Sie können die Abbildkennung aus der Benutzertabelle abrufen). Der QBIC-Katalog muss geöffnet sein, bevor Sie ein Abbild manuell katalogisieren können.

Beispielsweise rufen die folgenden Anweisungen eine Abbildkennung aus einer Benutzertabelle ab und katalogisieren danach das Abbild:

```
/* Retrieve the image handle */

EXEC SQL BEGIN DECLARE SECTION;
char Img_hdl[251];
EXEC SQL END DECLARE SECTION;

QbCatalogHandle  CatHdl;

EXEC SQL SELECT PICTURE INTO :Img_hdl
  FROM EMPLOYEE
       WHERE NAME='Anita Jones';
/* Catalog the image*/

rc=QbCatalogImage(
               CatHdl,                      /* catalog handle */
               Img_hdl);                    /* image handle */
```

Spalte mit Abbildern manuell katalogisieren

Verwenden Sie die API QbCatalogColumn oder den Befehl CATALOG QBIC COLUMN, um eine Spalte mit Abbildern manuell zu katalogisieren. Der Image Extender katalogisiert nur Abbilder in der Spalte, die neu hinzugefügt, aktualisiert oder gelöscht wurden, nachdem die Spalte zuletzt katalogisiert wurde. Der Image Extender katalogisiert diese Abbilder für alle Merkmale im Katalog. Der QBIC-Katalog muss für Aktualisierungsoperationen geöffnet sein, bevor Sie eine Spalte mit Abbildern manuell katalogisieren können.

Verwendung der API: Wenn Sie die API QbCatalogColumn verwenden, geben Sie die Katalogkennung an. Der Image Extender verwendet die Abbilder in der Benutzertabellenspalte, die dem angegebenen Katalog zugeordnet ist.

Beispielsweise werden mit dem folgenden API-Aufruf die nicht katalogisierten Abbilder in einer Benutzertabellenspalte katalogisiert, die dem angegebenen Katalog zugeordnet ist. Die Abbilder werden für alle Merkmale im Katalog katalogisiert:

```
QbCatalogHandle  CatHdl;

rc=QbCatalogColumn(
                CatHdl);                              /* catalog handle */
```

Verwendung der Befehlszeile: Verwenden Sie den Befehl CATALOG QBIC COLUMN, um eine Spalte mit Abbildern manuell zu katalogisieren. Sie können den Befehl auch verwenden, um Abbilder erneut zu katalogisieren (siehe „Abbild erneut katalogisieren" auf Seite 163). Geben Sie die Parameter FOR und NEW an. (FOR und NEW sind Standardparameter.)

Im folgenden Beispiel wird der Befehl verwendet, um die nicht katalogisierten Abbilder in der Tabellenspalte zu katalogisieren, die dem momentan geöffneten Katalog zugeordnet ist. Die Abbilder werden für alle Merkmale im Katalog katalogisiert:

```
CATALOG QBIC COLUMN FOR NEW
```

Abbild entkatalogisieren

Das Entkatalogisieren eines Abbilds bedeutet, dass die Einträge für das Abbild aus einem QBIC-Katalog gelöscht werden. Verwenden Sie die API QbUncatalogImage, um ein Abbild zu entkatalogisieren. Sie können ein Abbild nicht mit Hilfe eines Befehls entkatalogisieren, da es keine Möglichkeit gibt, ein einzelnes Abbild in der Befehlszeile zu identifizieren. Wenn Sie die API verwenden, geben Sie die Katalogkennung und die Abbildkennung an (Sie können die Abbildkennung aus der Benutzertabelle abrufen). Der QBIC-Katalog muss für Aktualisierungsoperationen geöffnet sein, bevor Sie ein Abbild entkatalogisieren können.

Beispielsweise rufen die folgenden Anweisungen eine Abbildkennung aus einer Benutzertabelle ab und entkatalogisieren danach das Abbild:

```
/* Retrieve the image handle */

EXEC SQL BEGIN DECLARE SECTION;
char Img_hdl[251];
EXEC SQL END DECLARE SECTION;

QbCatalogHandle  CatHdl;

EXEC SQL SELECT PICTURE INTO :Img_hdl
  FROM EMPLOYEE
       WHERE NAME='Anita Jones';
/* Uncatalog the image */

rc=QbUncatalogImage(
                CatHdl,                       /* catalog handle */
                Img_hdl);                     /* image handle */
```

Abbild erneut katalogisieren

Wenn Sie ein Abbild katalogisieren, analysiert der Image Extender die Merkmale des Abbilds, die für den QBIC-Katalog identifiziert wurden, und speichert die Werte für diese Merkmale im Katalog. Wenn Sie ein Merkmal zu einem QBIC-Katalog hinzufügen, analysiert der Image Extender nicht automatisch das neue Merkmal für bereits katalogisierte Abbilder. Um Werte für das neue Merkmal zum Katalog hinzuzufügen, müssen Sie alle Abbilder erneut katalogisieren.

Verwenden Sie die API QbReCatalogColumn oder den Befehl CATALOG QBIC COLUMN, um die Abbilder in einem QBIC-Katalog erneut zu katalogisieren. Der Image Extender löscht alle Merkmaldaten, die sich momentan im Katalog befinden. Anschließend analysiert er die Abbilder für alle Merkmale, einschließlich aller neuen Merkmale, und katalogisiert die Abbilder. Der QBIC-Katalog muss geöffnet sein, bevor Sie Abbilder erneut katalogisieren können.

Verwendung der API: Wenn Sie die API QbReCatalogColumn verwenden, geben Sie die Katalogkennung an.

Im folgenden Beispiel werden die Abbilder in einem QBIC-Katalog erneut analysiert:

```
QbCatalogHandle  CatHdl;

rc=QbReCatalogColumn(
                CatHdl);                              /* catalog handle */
```

Verwendung der Befehlszeile: Verwenden Sie den Befehl CATALOG QBIC COLUMN, um Abbilder erneut zu katalogisieren. Der Befehl gilt für den momentan geöffneten Katalog. Sie können den Befehl auch verwenden, um Abbilder manuell zu katalogisieren (siehe „Abbild manuell katalogisieren" auf Seite 161).

Wenn Sie den Befehl eingeben, geben Sie die Parameter FOR und ALL an. Dadurch wird dem Image Extender angegeben, dass alle Abbilder erneut katalogisiert werden sollen.

Im folgenden Beispiel werden die katalogisierten Abbilder im momentan geöffneten QBIC-Katalog erneut katalogisiert:

```
CATALOG QBIC COLUMN FOR ALL
```

QBIC-Katalog neu verteilen (nur EEE)

Verwenden Sie die API DMBRedistribute oder den Befehl REDISTRIBUTE NODEGROUP, um QBIC-Merkmaldaten neu zu verteilen, wenn ein Knoten zu einer Knotengruppe hinzugefügt oder aus einer Knotengruppe gelöscht wird. Der Befehl stellt die QBIC-Merkmaldaten auf den gleichen Knoten wie die entsprechenden Benutzerdaten.

Wenn der Neuverteilungsprozess einen Fehler zurückgibt, können Sie den Befehl mit oder ohne den Parameter CONTINUE erneut ausführen, je nachdem, welche Anweisungen in der Antwort auf den Befehl angegeben werden. Mit dieser Option wird das System angewiesen, an der Stelle fortzufahren, an der es gestoppt wurde, und nicht am Anfang erneut zu starten. Der Parameter CONTINUE sollte nicht verwendet werden, wenn der Befehl REDISTRIBUTE NODEGROUP das erste Mal nach Ausführung des DB2-Befehls REDISTRIBUTE verwendet wird.

Um die Datenintegrität zu gewährleisten, müssen Sie die einzelnen Knotengruppen nacheinander neu verteilen. Warten Sie, bis die Neuverteilung für eine Knotengruppe beendet ist, bevor Sie die nächste starten.

Verwendung der API: Das folgende Beispiel zeigt, wie die QBIC-Merkmaldaten in der Knotengruppe 'groupone' neu verteilt werden:

```
#include <dmbrdst.h>

rc = DMBRedistribute(groupone,"continue");
```

Verwendung der Befehlszeile: Das folgende Beispiel zeigt, wie mit dem Befehl REDISTRIBUTE NODEGROUP Daten für den Knoten 'my_nodegroup' unter Verwendung des Parameters CONTINUE neu verteilt werden:

```
redistribute nodegroup my_nodegroup continue
```

QBIC-Katalog schließen

Verwenden Sie die API QbCloseCatalog oder den Befehl CLOSE QBIC CATALOG, um einen QBIC-Katalog zu schließen. Der QBIC-Katalog muss geöffnet sein, bevor Sie ihn schließen können.

Verwendung der API: Wenn Sie den API-Aufruf QbCloseCatalog eingeben, geben Sie die Katalogkennung an. Beispiel:

```
QbCatalogHandle  CatHdl;

rc=QbCloseCatalog(
                CatHdl);                          /* catalog handle */
```

Verwendung der Befehlszeile: Der Befehl CLOSE QBIC CATALOG gilt für den momentan geöffneten Katalog. Im folgenden Beispiel wird der Befehl verwendet, um den momentan geöffneten QBIC-Katalog zu schließen:

```
CLOSE QBIC CATALOG
```

QBIC-Katalog löschen

Beim Löschen eines QBIC-Katalogs werden alle Merkmaldaten in den Katalogtabellen gelöscht. Daraus ergibt sich, dass die zugehörigen Abbilder nicht länger für die Abfrage anhand des Inhalts zur Verfügung stehen. Um einen QBIC-Katalog zu löschen, müssen Sie über die Berechtigung ALTER oder CONTROL für die Tabelle verfügen, die dem Katalog zugeordnet ist. Der QBIC-Katalog muss geöffnet sein, bevor Sie ihn löschen können.

Verwenden Sie die API QbDeleteCatalog oder den Befehl DELETE QBIC CATALOG, um einen QBIC-Katalog zu löschen. Wenn Sie einen QBIC-Katalog löschen, geben Sie die Benutzertabelle und Spalte an, die dem Katalog zugeordnet sind.

Verwendung der API: Im folgenden Beispiel wird die API QbDeleteCatalog verwendet, um einen QBIC-Katalog zu löschen:

```
rc=QbDeleteCatalog(
                "employee",                       /* user table */
                 "picture");                      /* image column */
```

Verwendung der Befehlszeile: Der Befehl DELETE QBIC CATALOG gilt für den momentan geöffneten Katalog. Im folgenden Beispiel wird der Befehl verwendet, um den momentan geöffneten QBIC-Katalog zu löschen:

```
DELETE QBIC CATALOG employee picture
```

Programm für QBIC-Katalogbeispiel

Abb. 28 auf Seite 166 zeigt Teile eines in C geschriebenen Programms, mit dem ein QBIC-Katalog erstellt wird. Das Programm katalogisiert außerdem eine Spalte mit Abbildern im QBIC-Katalog. Das vollständige Programm befindet sich in der Datei QBCATDMO.C im Unterverzeichnis SAMPLES. Bevor Sie das vollständige Programm ausführen können, müssen Sie die Beispielprogramme ENABLE und POPULATE ausführen (die sich auch im Unterverzeichnis SAMPLES befinden). Weitere Informationen zu Beispielprogrammen befinden sich im Anhang B, „Beispielprogramme und Multimediadateien", auf Seite 529.

Beachten Sie die folgenden Punkte im Programm:

1. Die Kopfdatei dmbqbapi wird eingeschlossen.
2. Eine Verbindung zur Datenbank wird hergestellt.
3. Der Katalog wird erstellt. Das automatische Katalogisieren ist ausgeschaltet (OFF).
4. Der Katalog wird für Aktualisierungsoperationen geöffnet.
5. Das Merkmal 'Durchschnittsfarbe' wird zum Katalog hinzugefügt.
6. Eine Spalte mit Abbildern wird katalogisiert.
7. Der Katalog wird geschlossen.

```
#include <sql.h>
#include <sqlcli.h>
#include <sqlcli1.h>
#include <dmbqbqpi.h>  1
#include <stdio.h>

/*************************************************************************/
/*  Define the function prototypes                                       */
/*************************************************************************/

void printError(SQLHSTMT hstmt);
void createCatalog();
void openCatalog();
void closeCatalog();
void addFeature();
void catalogImageColumn();

QbCatalogHandle cHdl = 0;

static  SQLHENV    henv;
static  SQLHDBC    hdbc;
static  SQLHSTMT   hstmt;
static  SQLRETURN  rc;
char tableName[] = "sobay_catalog";
char columnName[] = "covers";

SQLCHAR uid[18+1];
SQLCHAR pwd[30+1];
SQLCHAR dbnName[SQL_MAX_DSN_LENGTH+1];

void main ()
{
/*---- prompt for database name, userid, and password ----*/
  printf("Enter database name:\n");
  gets((char *) dbName);
  printf("Enter userid:\n");
  gets((char *) pwd);
/* set up the SQL CLI environment */
 SQlAllocEnv(&henv);
 SQLAllocConnect(henv, &hdbc);
 rc = SQLConnect(hdbc,dbname,SQL_NTS,uid,SQL_NTS,pwd,SQL_NTS);  2
 if (rc != SQL_SUCCESS)
 {
  printError(SQL_NULL_HSTMT);
               exit(1);
 }
```

Abbildung 28. Programm für QBIC-Katalogbeispiel (Teil 1 von 4)

```
  createCatalog();
   openCatalog();
   addFeature();
 getCatalogInfo();
 listFeatures();
 catalogImageColumn();
 closeCatalog();

 SQLDisconnect(hdbc);
 SQLFreeConnect(hdbc);
 SQLFreeEnv(henv);
}
/*********************************************************************/
void createCatalog()
{
 SQLINTEGER autoCatalog = 0;
 SQLINTEGER retLen;
   SQLINTEGER errCode = 0;
   char errMsg[500];

   QbCreateCatalog( 3
         (char *) tableName,
         (char *) columnName,
         autoCatalog,
         0
        );

    DBiGetError(&errCode, errMsg);
    if(errCode) printf("Error code is %d Error Message %s", errCode, errMsg);
}
/*********************************************************************/
void openCatalog()
{
   SQLINTEGER errCode = 0;
   char errMsg[500];
 SQLINTEGER mode = qbiUpdate;

   QbOpenCatalog( 4
         (char *) tableName,
         (char *) columnName,
         mode,
         &cHdl
        );

    DBiGetError(&errCode, errMsg);
    if(errCode) printf("Error code is %d Error Message %s", errCode, errMsg);
}
```

Abbildung 28. Programm für QBIC-Katalogbeispiel (Teil 2 von 4)

```
/*********************************************************************/
void addFeature()
{
 SQLINTEGER errCode=0;
 char errMsg[5
 if(cHdl) /* if we have an open catalog, else do nothing */
 {
   char featureName*lbrk.] = "QbColorFeatureClass"; 5
     QbaddFeature(
         cHdl,
         featureName
        );

    DBiGetError(&errCode, errMsg);
    if(errCode) printf("Error code is %d Error Message %s", errCode, errMsg);
 }
   else
 {
               exit(1);
 }
}
/*********************************************************************/
void catalogImageColumn()
{
   SQLINTEGER errCode = 0;
   char errMsg[500];

 if(cHdl) /* if we have an open catalog, else do nothing */
 {
  SQLRETURN rc;
    QbCatalogColumn( 6
         cHdl,
        );

    DBiGetError(&errCode, errMsg);
    if(errCode) printf("Error code is %d Error Message %s", errCode, errMsg);

   else
 {
               exit(1);
 }
}
```

Abbildung 28. Programm für QBIC-Katalogbeispiel (Teil 3 von 4)

```
/*********************************************************************/
void closeCatalog()
{
 if(cHdl) /* if we have an open catalog, else do nothing */
 {
    QbCloseCatalog( 7
         cHdl,
        );
 }
}
/*********************************************************************/
```

Abbildung 28. Programm für QBIC-Katalogbeispiel (Teil 4 von 4)

Abfragen erstellen

Wenn Sie Abbilder nach Inhalt abfragen, identifizieren Sie die Eingabe für die Abfrage und eine Zielgruppe von katalogisierten Abbildern. Bei der Eingabe für die Abfrage werden die Namen der in der Abfrage zu verwendenden Merkmale, die Merkmalwerte und die Merkmalwertigkeiten (d. h., die Gewichtung, die auf die einzelnen Merkmale gelegt wird) angegeben.

Sie haben zwei Möglichkeiten, diese Eingabe zu liefern:

- Geben Sie in Ihrer Abfrage eine Abfragezeichenfolge an. Die Abfragezeichenfolge ist eine Zeichenfolge, die die Merkmale, die Merkmalwerte und die Merkmalwertigkeiten für die Abfrage angibt.
- Erstellen Sie ein Abfrageobjekt und verweisen Sie in Ihrer Abfrage darauf. Das Abfrageobjekt gibt die Merkmale und die Merkmalwertigkeiten an. Außerdem identifiziert es eine Datenquelle für die einzelnen Merkmale. Die Datenquelle liefert den Wert für die einzelnen Merkmale.

Abfragezeichenfolge angeben

Sie können eine Abfragezeichenfolge verwenden, um die Merkmale, die Merkmalwerte und die Merkmalwertigkeiten für die Abfrage anzugeben. Eine **Abfragezeichenfolge** ist eine Zeichenfolge im Format *merkmalname wert*, wobei *merkmalname* ein QBIC-Merkmalname ist und *wert* ein Wert, der dem Merkmal zugeordnet ist.

Sie können mehrere Merkmale in einer Abfrage angeben. Sie können dann ein Name/Wert-Paar für jedes Merkmal angeben, wie im Abschnitt „Merkmalwert" beschrieben. Die einzelnen Paare werden durch die Klausel AND getrennt. Wenn Sie mehrere Merkmale in einer Abfrage angeben, können Sie einem oder mehreren Merkmalen auch eine Wertigkeit zuordnen, wie im Abschnitt „Merkmalwertigkeit" auf Seite 171 beschrieben. Die Abfragezeichenfolge hat dann das Format *merkmalname wert wertigkeit*, wobei *wertigkeit* die Wertigkeit angibt, die dem Merkmal zugeordnet ist.

Der Image Extender bietet eine API (QbQueryStringSearch) und zwei UDFs (QbScoreFromStr und QbScoreTBFromStr), die eine Abfragezeichenfolge verwenden. Wenn Sie eine Abfrage eingeben, verwenden Sie die entsprechende API oder UDF, und geben Sie die Abfragezeichenfolge als Eingabeparameter an. (Der Abschnitt „Abfragen anhand des Abbildinhalts ausführen" auf Seite 179 enthält weitere Einzelheiten.)

Merkmalwert

Geben Sie für jedes Merkmal in der Abfrage einen Merkmalwert in der Abfragezeichenfolge ein.

Wenn eine Abfrage innerhalb eines DB2-Befehls übergeben wird, müssen bestimmte Dateinamenkonventionen befolgt werden, damit die Abfrage korrekt ausgeführt wird. Sie müssen Dateinamen, die Leerzeichen oder abschließende spitze Klammern (>) enthalten, in doppelte Anführungszeichen einschließen; andere Dateinamen können wahlweise in doppelte Anführungszeichen eingeschlossen werden. Wenn Sie Anführungszeichen um einen Dateinamen verwenden, muss davor ein Escapezeichen (\) stehen. Wenn die Abfrage nicht innerhalb eines DB2-Befehls übergeben wird, ist es nicht erforderlich, mit den Anführungszeichen voranstehende Escapezeichen anzugeben.

Im folgenden Beispiel wird eine Abfragezeichenfolge innerhalb eines DB2-Befehls übergeben:

```
db2 "select image_id from table
(mmdbsys.QbScoreTBFromStr
('texture file=<server,patterns/ptrn07.gif>',
'fabric',
'swatch_img',
10))
as T1"
```

In Tabelle 12 werden die Werte aufgelistet, die Sie für die einzelnen Merkmale angeben können. Direkt unter jedem Merkmalnamen ist eine Kurzversion angegeben, die stattdessen verwendet werden kann.

Tabelle 12. Merkmalnamen, die in einer Abfragezeichenfolge angegeben werden können

Merkmalname	Wert
averageColor, average oder QbColorFeatureClass	color=<*Rwert*, *Gwert*, *Bwert*> Jeder Farbwert ist eine ganze Zahl zwischen 0 und 255, die den Rotwert (*Rwert*), Grünwert (*Gwert*) und Blauwert (*Bwert*) des Abbilds angibt. file=<*dateiadresse*, *dateiname*> Die *dateiadresse* für eine Serverdatei lautet `server`. Der *dateiname* ist der vollständige Dateipfad, der in dem Format angegeben werden muss, das für das System, auf dem sich die Datei befindet, gilt, oder ein relativer Dateiname. Die DB2 Extender lösen den relativen Dateinamen mit Hilfe von Umgebungsvariablen auf (siehe „Umgebungsvariablen zum Auflösen von Dateinamen verwenden" auf Seite 521).
histogram, histogramcolor oder QbColorHistogramFeatureClass	histogram=<(*hist_wert*, *Rwert*, *Gwert*, *Bwert*>), ... Jeder Histogrammfarbwert wird in einer Klausel angegeben, die den Prozentsatz (1 bis 100) dieser Farbe im Histogramm (*hist_wert*) sowie den Rotwert (*Rwert*), Grünwert (*Gwert*) und Blauwert (*Bwert*) dieser Farbe identifiziert. file=<*dateiadresse*, *dateiname*> Die *dateiadresse* für eine Serverdatei lautet `server`. Der *dateiname* ist der vollständige Dateipfad, der in dem Format angegeben werden muss, das für das System, auf dem sich die Datei befindet, gilt, oder ein relativer Dateiname. Die DB2 Extender lösen den relativen Dateinamen mit Hilfe von Umgebungsvariablen auf.
draw, positional oder QbDrawFeatureClass	file=<*dateiadresse*, *dateiname*> handle=<*abbildkennung*> Die *dateiadresse* für eine Serverdatei lautet `server`. Der *dateiname* ist der vollständige Dateipfad, der in dem Format angegeben werden muss, das für das System, auf dem sich die Datei befindet, gilt, oder ein relativer Dateiname. Die DB2 Extender lösen den relativen Dateinamen mit Hilfe von Umgebungsvariablen auf.

Tabelle 12. Merkmalnamen, die in einer Abfragezeichenfolge angegeben werden können (Forts.)

Merkmalname	Wert
texture oder QbTextureFeatureClass	file=<*dateiadresse*, *dateiname*> handle=<*abbildkennung*> Die *dateiadresse* für eine Serverdatei lautet `server`. Der *dateiname* ist der vollständige Dateipfad, der in dem Format angegeben werden muss, das für das System, auf dem sich die Datei befindet, gilt, oder ein relativer Dateiname. Die DB2 Extender lösen den relativen Dateinamen mit Hilfe von Umgebungsvariablen auf.

Merkmalwertigkeit

Wenn Sie mehrere Merkmale in einer Abfragezeichenfolge angeben, können Sie für ein oder mehrere Merkmale auch eine Wertigkeit angeben. Die Wertigkeit eines Merkmals gibt die Gewichtung an, die der Image Extender auf das Merkmal legt, wenn er Ähnlichkeitsergebnisse berechnet und Ergebnisse für eine Abfrage anhand des Abbildinhalts zurückgibt. Je höher die Wertigkeit, die Sie für ein Merkmal angeben, desto größer ist die Gewichtung dieses Merkmals in der Abfrage. Die Wertigkeit ist eine reelle Zahl, die größer als 0,0 ist, beispielsweise 2,5 oder 10,0. Wenn Sie keine Wertigkeit in einer Abfragezeichenfolge angeben, verwendet der Image Extender die Standardwertigkeit für das Merkmal. Die Zuordnung einer Wertigkeit ist ohne Bedeutung, wenn das Merkmal das einzige Merkmal ist, das in einer Abfragezeichenfolge angegeben ist. (Dieses Merkmal erhält in jedem Fall die volle Wertigkeit in der Abfrage.)

Die Wertigkeit eines Merkmals steht in Relation zu anderen Merkmalen, die in der Abfrage angegeben sind. Nehmen Sie als Beispiel an, Sie geben in einer Abfragezeichenfolge die Merkmale 'Durchschnittsfarbe' und 'Textur' an und geben außerdem einen Wertigkeitswert von 2,0 für die Durchschnittsfarbe an. Dadurch wird der Image Extender angewiesen, der Durchschnittsfarbe eine doppelt so hohe Gewichtung zu geben wie der Textur.

Beispiele

Die folgende Abfragezeichenfolge gibt die Durchschnittsfarbe Rot an:

```
averageColor color=<255, 0, 0>
```

Die folgende Abfragezeichenfolge gibt ein Histogramm an, das aus 10 % Rot, 50 % Grün und 40 % Blau besteht:

```
histogram histogram=<(10, 255, 0, 0), (50, 0, 255, 0),
                                      (40, 0, 0, 255)>
```

Die folgende Abfragezeichenfolge gibt eine Durchschnittsfarbe und eine Textur an. Der Wert für die Textur wird durch ein Abbild in einer Serverdatei bereitgestellt. Die Wertigkeit der Textur ist doppelt so hoch wie die der Durchschnittsfarbe:

```
averageColor color=<30, 200, 25> and
  texture file=<server, "\patterns\pattern7.gif"> weight=2.0
```

Abfrageobjekt verwenden

Sie können ein Abfrageobjekt verwenden, um die Merkmale, die Merkmalwerte und die Merkmalwertigkeiten für die Abfrage anzugeben. Sie können das Abfrageobjekt erstellen und darin Merkmale hinzufügen. Anschließend können Sie eine Datenquelle für die einzelnen Merkmale angeben. Die Datenquelle liefert einen Wert für die einzelnen Merkmale. Beispielsweise kann die Datenquelle ein Abbild in einer Datei sein. Wenn die Durchschnittsfarbe das passende Merkmal ist, wird die Durchschnittsfarbe des Abbilds dem Abfrageobjekt zugeordnet. Wenn Sie mehrere Merkmale zu einem Abfrageobjekt hinzufügen, können Sie einem oder mehreren Merkmalen eine Wertigkeit zuordnen.

Der Image Extender stellt drei APIs (QbQuerySearch, QbQueryStringSearch und QbQueryNameSearch) und zwei UDFs (QbScoreFromName und QbScoreTBFromName) zur Verfügung, die ein Abfrageobjekt verwenden. Wenn Sie eine Abfrage eingeben, verwenden Sie die entsprechende API oder UDF, und geben Sie das Abfrageobjekt als Eingabeparameter an. (Der Abschnitt „Abfragen anhand des Abbildinhalts ausführen" auf Seite 179 enthält weitere Einzelheiten.)

Abfrageobjekt erstellen

Verwenden Sie die API QbQueryCreate, um ein Abfrageobjekt zu erstellen. Daraufhin gibt der Image Extender eine Kennung für das Abfrageobjekt zurück. Die Kennung hat den QBIC-spezifischen Datentyp QbQueryHandle, der in der Kopfdatei für QBIC, dmbqbapi.h, definiert ist.

Wenn Sie die API verwenden, müssen Sie auf die Kennung für das Abfrageobjekt zeigen. Sie müssen außerdem die Kennung in APIs angeben, die andere Operationen im Abfrageobjekt ausführen, wie z. B. Hinzufügen eines Merkmals.

Beispielsweise erstellt der folgende API-Aufruf ein Abfrageobjekt:

```
QbQueryHandle  qHandle;

rc=QbQueryCreate(
             &qHandle);                          /* query object handle */
```

Merkmal zu einem Abfrageobjekt hinzufügen

Sie identifizieren die Abbildmerkmale, die der Image Extender abfragen soll, indem Sie die Merkmale zu einem Abfrageobjekt hinzufügen.

Verwenden Sie die API QbQueryAddFeature, um ein Merkmal zu einem Abfrageobjekt hinzuzufügen. Wenn Sie die API verwenden, geben Sie die Kennung für das Abfrageobjekt an. Geben Sie außerdem den Namen des Merkmals an. Sie können nur ein Merkmal in der API angeben. Sie müssen für jedes Merkmal, das Sie zu einem Abfrageobjekt hinzufügen wollen, eine separate API aufrufen.

Im folgenden Beispiel wird die API QbQueryAddFeature verwendet, um das Merkmal 'Durchschnittsfarbe' zu einem Abfrageobjekt hinzuzufügen:

```
char                featureName[qbiMaxFeatureName];
QbQueryHandle  qHandle;

rc=QbQueryAddFeature(
               qHandle,                          /* query object handle */
               "QbColorFeatureClass");           /* feature name */
```

Datenquelle für ein Merkmal in einem Abfrageobjekt angeben

Verwenden Sie die API QbQuerySetFeatureData, um die Datenquelle für ein Merkmal in einem Abfrageobjekt anzugeben. Folgende Datenquellen sind möglich:

- Ein katalogisiertes oder entkatalogisiertes Abbild in einer Spalte einer Benutzertabelle
- Eine Abbilddatei auf einer Client-Workstation
- Ein Abbild in einem Puffer auf einer Client-Workstation

Darüber hinaus können Sie explizit Daten für das Merkmal 'Durchschnittsfarbe' oder 'Histogrammfarbe' angeben. Beispielsweise können Sie die Werte rot, grün und blau für eine Durchschnittsfarbe angeben.

Wenn Sie die API verwenden,

- geben Sie die Kennung für das Abfrageobjekt ein.
- geben Sie den Namen des Merkmals an.
- zeigen Sie auf die Struktur QbImageSource (Einzelheiten siehe Seite 173).

Strukturen für Datenquellen verwenden: Verschiedene Strukturen werden verwendet, um Informationen zu Datenquellen für ein Abfrageobjekt zur Verfügung zu stellen. Zu diesen Strukturen gehören:

- QbImageSource
- QbColor
- QbHistogramColor

QbImageSource: Die Struktur QbImageSource gibt den Quellentyp für ein Merkmal in einem Abfrageobjekt an. Die Struktur ist in der Kopfdatei (Includedatei) für QBIC, `dmbqbapi.h`, wie folgt definiert:

```
typedef struct{
          SQLINTEGER    type;
          union {
          char       imageHandle[MMDB_BASE_HANDLE_LEN+1];
QbImageFile       clientFile;
QbImageBuffer     buffer;
                QbSampleSource    reserved;
                QbColor           averageColor;
                QbHistogramColor  histogramColor[qbiHistogramCount];
          };
} QbImageSource;
```

Das Feld für den Typ in der Struktur QbImageSource gibt den Typ der Quelle an. Sie können den Wert in dem Feld wie folgt setzen:

Wert	Bedeutung
qbiSource_ImageHandle	Die Quelle ist in einer Benutzertabellenspalte
qbiSource_ClientFile	Die Quelle ist in einer Client-Workstation-Datei
qbiSource_Buffer	Die Quelle ist in einem Client-Workstation-Puffer
qbiSource_ServerFile	Die Quelle ist in der Serverdatei
qbiSource_AverageColor	Die Quelle ist eine Angabe zur Durchschnittsfarbe
qbiSource_HistogramColor	Die Quelle ist eine Angabe zur Histogrammfarbe

Diese Einstellungen sind nur für das entsprechende Merkmal gültig. Beispielsweise ist 'qbiSource_AverageColor' nur für das Merkmal 'Durchschnittsfarbe' gültig.

Wenn Sie das Feld für den Typ auf 'qbiSource_ServerFile' setzen, verwenden Sie 'clientFile' für den Namen und den Typ der Datei auf dem Server.

Je nach Quellentyp prüft der Image Extender auch andere von Ihnen angegebene Informationen. Dies wird in Tabelle 13 gezeigt.

Tabelle 13. Prüfobjekte des Image Extender in QbImageSource

Quelle	Prüfobjekte des Image Extender	Ort der Angabe
Benutzertabelle	Abbildkennung	Feld 'imageHandle' von QbImageSource
Datei	Name der Datei Format der Datei	Feld 'clientFile' von QbImageSource
Puffer	Name der Datei	QbImageBuffer (Einzelheiten zur Verwendung dieser Struktur finden Sie auf Seite 174
Angabe zur Durchschnittsfarbe	Farbwerte rot, grün und blau	Struktur QbColor (Einzelheiten zur Verwendung dieser Struktur finden Sie auf Seite 174)
Angabe zur Histogrammfarbe	Farbwerte und -prozentsätze	Struktur QbHistogramColor (Einzelheiten zur Verwendung dieser Struktur finden Sie auf Seite 175)

QbImageBuffer: Verwenden Sie die Struktur QbImageBuffer, um Format, Länge und Inhalt eines Abbilds anzugeben, wenn die Datenquelle in einem Puffer ist. Die Struktur ist in der Kopfdatei für QBIC, dmbqbapi.h, wie folgt definiert:

```
typedef struct{
          char          format[qbiImageFormatLength+1];
          SQLINTEGER    length;
          char*         image;
} QbImageBuffer;
```

QbColor: Verwenden Sie die Struktur QbColor, um die Werte rot, grün und blau für eine Durchschnittsfarbe anzugeben, wenn die Datenquelle eine Angabe zur Durchschnittsfarbe ist. Die Struktur ist in der Kopfdatei für QBIC, dmbqbapi.h, wie folgt definiert:

```
typedef struct{
          SQLUSMALLINT    red;          /*0 off - 65535 (fully on) */
          SQLUSMALLINT    green;        /*0 off - 65535 (fully on) */
          SQLUSMALLINT    blue;         /*0 off - 65535 (fully on) */
} QbColor;
```

Setzen Sie die Werte in QbColor, um den Anteil an roten, grünen und blauen Pixel anzugeben, der bei der Berechnung des Durchschnittswerts eingeschlossen werden soll. Die Werte liegen im Bereich von 0 bis 65535. Der Wert 0 bedeutet, dass der Eintrag ignoriert wird.

QbHistogramColor: Verwenden Sie die Struktur QbHistogramColor, um die einzelnen Farbkomponenten einer Histogrammfarbspezifikation anzugeben. Die vollständige Spezifikation für eine Histogrammfarbe ist in einem Bereich von QbHistogramColor-Strukturen enthalten. Jede Struktur enthält einen Farbwert und einen Prozentsatz. Der Farbwert besteht aus Pixel-Werten für rot, grün und blau. Der Prozentsatz gibt den Anteil dieser Farbe an, der im Zielabbild erforderlich ist.

Die Struktur ist in der Kopfdatei für QBIC, dmbqbapi.h, wie folgt definiert:

```
typedef struct{
          QbColor         color;
          SQLUSMALLINT    percentage;   /*0 - 100 */
} QbHistogramColor;
```

Setzen Sie die Werte in QbColor, um den Anteil an roten, grünen und blauen Pixel für die Farbe anzugeben. Die Werte liegen im Bereich von 0 bis 65535. Setzen Sie den Prozentsatz, um den Anteil der angegebenen Farbe, der im Zielabbild erforderlich ist, anzugeben. Der Wert kann im Bereich von 1 bis 100 liegen. Die Summe der Prozentsätze für die Farbkomponenten in einer Histogrammfarbe muss 100 oder weniger betragen.

Examples: Die API im folgenden Beispiel gibt die Datenquelle für das Merkmal 'Histogrammfarbe' in einem Abfrageobjekt an. Die Datenquelle ist eine Datei auf der Client-Workstation.

```
char                featureName[qbiMaxFeatureName];
QbQueryHandle  qHandle;
QbImageSource       imgSource;

imgSource.type=qbiSource_ClientFile;strcpy(imgSource.clientFile.fileName,
"/tmp/image.gif");strcpy(imgSource.clientFile.format,"GIF");
rc=QbQuerySetFeatureData(
                qHandle,                            /* query object handle */
                "QbColorHistogramFeatureClass",     /* feature name */
                &imgSource);                        /* feature data source */
```

Im folgenden Beispiel ist die Datenquelle eine Angabe zur Durchschnittsfarbe rot:

```
char                featureName[qbiMaxFeatureName];
QbColor             avgColor;
QbImageSource       imgSource;

imgSource.type=qbSource_AverageColor;
avgColor.red=255;
avgColor.green=0;
avgColor.blue=0;
strcpy(featureName,"QbColorFeatureClass");

rc=QbQuerySetFeatureData(
                qHandle,                            /* query object handle */
                featureName,                        /* feature name */
                &imgSource);                        /* feature data source */
```

Wertigkeit eines Merkmals in einem Abfrageobjekt festlegen

Wenn Sie mehr als ein Merkmal zu einem Abfrageobjekt hinzugefügt haben, können Sie die Wertigkeit angeben, die ein oder mehrere Merkmale in einer Abfrage erhalten sollen. Verwenden Sie die API QbQuerySetFeatureWeight, um die Wertigkeit eines Merkmals anzugeben. Die Wertigkeit eines Merkmals gibt die Gewichtung an, die der Image Extender auf das Merkmal legt, wenn er Ähnlichkeitsergebnisse berechnet und Ergebnisse für eine Abfrage anhand des Abbildinhalts zurückgibt. Je höher die Wertigkeit, die Sie für ein Merkmal angeben, desto größer ist die Gewichtung dieses Merkmals im Abfrageobjekt.

Sie können eine Wertigkeit für ein oder mehrere Merkmale in einem Abfrageobjekt angeben, obwohl Sie bei jeder Verwendung der API QbQuerySetFeatureWeight nur die Wertigkeit für jeweils ein Merkmal angeben können. Wenn Sie keine Wertigkeit für ein Merkmal in einem Abfrageobjekt angeben, verwendet der Image Extender die Standardwertigkeit für das Merkmal. Die Zuordnung einer Wertigkeit für ein Merkmal ist ohne Bedeutung, wenn das Merkmal das einzige Merkmal in einem Abfrageobjekt ist. (Dieses Merkmal erhält in jedem Fall die volle Wertigkeit im Abfrageobjekt.)

Wenn Sie die API verwenden,

- geben Sie die Kennung für das Abfrageobjekt ein.
- geben Sie den Merkmalnamen ein.
- zeigen Sie auf die Merkmalwertigkeit. Sie können die Wertigkeit auf eine reelle Zahl setzen, die größer als 0 ist, beispielsweise 2,5 oder 10,0. Je höher der angegebene Wert, desto größer ist die Gewichtung dieses Merkmals. Die Einstellung ändert jede Wertigkeit, die zuvor für das Merkmal im Abfrageobjekt festgelegt wurde.

Im folgenden Beispiel enthält das Abfrageobjekt das Merkmal 'Durchschnittsfarbe' und mindestens ein weiteres Merkmal. Die API QbQuerySetFeatureWeight wird verwendet, um eine Wertigkeit für das Merkmal 'Durchschnittsfarbe' im Abfrageobjekt anzugeben:

```
char                featureName[qbiMaxFeatureName];
double              weight;
QbQueryObjectHandle qoHandle;

strcpy(featureName,"QbColorFeatureClass");
weight=5.00;

rc=QbQuerySetFeatureWeight(
                qoHandle,                       /* query object handle */
                featureName,                    /* feature name */
                &weight);                       /* feature weight */
```

Abfragezeichenfolge sichern und erneut verwenden

Abfrageobjekte sind 'vergänglich', solange sie nicht gesichert sind. Sie existieren nur während einer einzigen Datenbankverbindung.Sie können die Abfragezeichenfolge aus einer Abfrage sichern, um sie erneut im Programm oder über Programmaufrufe hinweg zu verwenden oder sogar, nachdem die aktuelle Datenbankverbindung unterbrochen wurde.

Der Image Extender bietet die API QbQueryGetString, die die Abfragezeichenfolge aus einem Abfrageobjekt zurückgibt. Sie können dann diese Abfragezeichenfolge als Eingabe für die API QbQueryStringSearch oder für die UDFs QbScoreFromStr und QbScoreTBFromStr in anderen Abfragen anhand des Abbildinhalts verwenden (siehe „Abfragen anhand des Abbildinhalts ausführen" auf Seite 179).

Die Abfragezeichenfolge wird erstellt, wenn Sie die Abfrage mit Hilfe einer der folgenden APIs erstellen:

- QbQueryCreate
- QbQueryAddFeature
- QbQuerySetFeatureData
- QbQuerySetFeatureWeight
- QbQueryRemoveFeature

Nachdem Sie die Abfrage erstellt haben, können Sie die API QbQueryGetString aufrufen, um die Zeichenfolge abzurufen. Sie können diese Abfragezeichenfolge in Aufrufen innerhalb desselben Programms verwenden oder sie in einer Datei sichern, um sie in nachfolgenden Aufrufen Ihrer Anwendung oder in anderen Datenbankverbindungen zu verwenden. Nachdem Sie die Verwendung der Abfragezeichenfolge, die von QbQueryGetString zurückgegeben wurde, abgeschlossen haben, müssen Sie den Bereich explizit freigeben.

Im folgenden Beispiel wird die API QbQueryGetString verwendet, um die Abfragezeichenfolge aus einem Abfrageobjekt abzurufen:

```
SQLRETURN rc;
char* qryString;
QbQueryHandle  qHandle;

.....           /* Here you create and use the query */

rc = QbQueryGetString(qHandle, &qryString);
if ( rc == 0) {
   ...         /* Use the query string as input here  */
   free((void *)qryString);
   qryString=(char *)0;
}
```

Einschränkung: Wenn Sie eine Clientdatei verwenden, um die Datenquelle für ein Merkmal anzugeben, gibt die Abfragezeichenfolge nicht die Merkmaldaten wieder.

Informationen zu einem Abfrageobjekt abrufen

Sie können feststellen, welche Merkmale (falls vorhanden) zu einem Abfrageobjekt hinzugefügt wurde. Sie können außerdem die aktuelle Wertigkeit eines Merkmals feststellen.

API	Zum Abrufen
QbQueryGetFeatureCount	der Anzahl von Merkmalen in einem Abfrageobjekt
QbQueryListFeatures	der Namen von Merkmalen in einem Abfrageobjekt

Wenn Sie die API QbQueryGetFeatureCount verwenden, geben Sie die Kennung des Abfrageobjekts an. Sie müssen außerdem auf einen Zähler zeigen. Der Image Extender gibt die Merkmalanzahl im Zähler zurück.

Im folgenden Beispiel wird die API QbQueryGetFeatureCount API verwendet, um die Anzahl von Merkmalen in einem Abfrageobjekt festzustellen:

```
SQLINTEGER     count;
QbQueryHandle  qHandle;

rc=QbQueryGetFeatureCount(
                qHandle,                          /* query object handle */
                &count);                          /* feature count */
```

Wenn Sie die API QbQueryListFeatures aufrufen, müssen Sie einen Puffer zuordnen, der den zurückgegebenen Merkmalnamen enthalten soll. Sie müssen außerdem die Katalogkennung und die Größe des Puffers für den zurückgegebenen Merkmalnamen angeben.

Im folgenden Beispiel wird die API QbQueryListFeatures verwendet, um die Namen der einzelnen Merkmale in einem Abfrageobjekt abzurufen:

```
SQLINTEGER     retCount,bufSize;
char*          featureName;
QbQueryHandle  qHandle;

bufSize=qbiMaxFeatureName;
featureName=(char*)malloc(bufSize);

rc=QbQueryListFeatures(
                qHandle,                          /* query object handle */
                bufSize                           /* size of buffer */
                &retCount,                        /* feature count */
                featureName);                     /* buffer for feature names */
```

Merkmal aus einem Abfrageobjekt löschen

Löschen Sie ein Merkmal aus einem Abfrageobjekt mit Hilfe der API QbQueryRemoveFeature. Wenn Sie die API verwenden, geben Sie die Kennung für das Abfrageobjekt und den Namen des Merkmals an.

Im folgenden Beispiel wird die API QbQueryRemoveFeature verwendet, um das Merkmal 'Histogrammfarbe' aus einem Abfrageobjekt zu löschen:

```
char                featureName[qbiMaxFeatureName];
QbQueryHandle  qHandle;

strcpy(featureName,"QbColorHistogramFeatureClass");

rc=QbQueryRemoveFeature(
                qHandle,                          /* query object handle */
                featureName);                     /* feature name */
```

Abfrageobjekt löschen

Löschen Sie ein nicht benanntes Abfrageobjekt mit Hilfe der API QbQueryDelete. Der Image Extender löscht die Abfrage aus der Datenbank, zu der momentan eine Verbindung besteht.

Wenn Sie die API QbQueryDelete verwenden, geben Sie die Kennung für das Abfrageobjekt an.

Im folgenden Beispiel wird die API QbQueryDelete verwendet, um ein Abfrageobjekt zu löschen:

```
QbQueryHandle  qHandle;

rc=QbQueryDelete(
           qHandle);                              /* query object handle */
```

Wenn Sie eine benannte Abfrage verwendet haben, löschen Sie das Abfrageobjekt mit Hilfe der API QbQueryNameDelete.

Abfragen anhand des Abbildinhalts ausführen

Nach dem Katalogisieren von Abbildern können Sie ein oder mehrere Abbilder nach Inhalt abfragen. Wenn Sie ein Abbild nach Inhalt abfragen, identifizieren Sie die Eingabe für die Abfrage und eine Zielgruppe von katalogisierten Abbildern. Sie können die Eingabe in einer Abfragezeichenfolge (siehe „Abfragezeichenfolge angeben" auf Seite 169) oder in einem Abfrageobjekt (siehe „Abfrageobjekt verwenden" auf Seite 172) angeben.

Wenn Sie eine Abfragezeichenfolge verwenden, können Sie die Abfrage von der DB2-Befehlszeile aus oder innerhalb eines Programms übergeben. Wenn Sie ein Abfrageobjekt verwenden, übergeben Sie die Abfrage innerhalb eines Programms aus, indem Sie auf die Kennung verweisen.

Der Image Extender vergleicht die Merkmalwerte, die in der Abfrage angegeben sind, mit denen der Zielabbilder und berechnet ein Ähnlichkeitsergebnis für jedes Abbild. Dieses Ergebnis zeigt, wie sehr der Merkmalwerte des Zielabbilds den Merkmalwerten ähneln, die in der Abfrage angegeben sind.

Sie können Abbilder abrufen, deren Merkmalwerte denen der Abfrage am ähnlichsten sind. Sie können außerdem ein einzelnes katalogisiertes Abbild abfragen und dessen Ähnlichkeitsergebnis abrufen, oder die Ähnlichkeitsergebnisse für alle katalogisierten Abbilder in einer Tabellenspalte abrufen.

Abbilder abfragen

Der Image Extender stellt drei APIs zur Verfügung, um die katalogisierten Abbilder in einer Tabellenspalte abzurufen. Die APIs unterscheiden sich nur insofern, dass sie eine Abfragezeichenfolge oder ein Abfrageobjekt als Eingabe benötigen:

API	Eingabe
QbQueryStringSearch	Abfragezeichenfolge
QbQuerySearch	Kennung für Abfrageobjekt
QbQueryNameSearch	Name des Abfrageobjekts

In jeder der drei APIs können Sie außerdem die folgenden Operationen ausführen:

- Angeben der Benutzertabelle und Spalte, die die zu durchsuchenden Abbilder enthält. Die Abbilder müssen in einem QBIC-Katalog katalogisiert sein.
- Angeben der maximalen Anzahl von Ergebnissen, die zurückgegeben werden sollen.
- Setzen eines Zeigers auf eine Struktur, die den Bereich der Abfrage angibt. Setzen Sie den Zeiger auf 0, auf den Wert NULL oder auf eine leere Zeichenfolge. Damit wird angegeben, dass alle katalogisierten Abbilder in der Tabellenspalte durchsucht werden sollen.
- Angeben der Konstanten qbiArray, um anzuzeigen, dass die Ergebnisse in einem Bereich gespeichert werden sollen. Die Konstante qbiArray ist in der Kopfdatei für QBIC, dmbqbapi.h, definiert.

Sie können außerdem auf einen Bereich von Ausgabestrukturen zeigen, der die Ergebnisse der Suche enthalten soll. Daraufhin gibt der Image Extender in diesen Strukturen die Kennungen der Zielabbilder zurück, deren Merkmalwerte denen der Abfrage am ähnlichsten sind. Er gibt außerdem ein Ähnlichkeitsergebnis für jedes Abbild zurück, das angibt, wie sehr der Merkmalwert des Abbilds dem der Abfrage ähnelt. Die Struktur ist in der Kopfdatei für QBIC, dmbqbapi.h, wie folgt definiert:

```
typedef struct{
          char       imageHandle[MMDB_BASE_HANDLE_LEN+1];
          SQLDOUBLE  SCORE
} QbResult;
```

Sie müssen einen Bereich zuordnen, der groß genug ist, um die maximale angegebene Anzahl von Ergebnissen zu speichern, und Sie müssen in der API auf den Bereich zeigen. Sie müssen außerdem auf einen Zähler zeigen. Der Image Extender setzt den Wert des Zählers auf die Anzahl der Ergebnisse, die er zurückgibt.

Im folgenden Beispiel wird die API QbQueryStringSearch verwendet, um die katalogisierten Abbilder in einer Tabellenspalte anhand des Inhalts abzufragen. Beachten Sie, dass der Zeiger auf den Abfragebereich auf den Wert Null gesetzt ist.

```
QbResult       returns[MaxQueryReturns];
SQLINTEGER     maxResults=qbiMaxQueryReturns;
SQLINTEGER     count;
QbQueryHandle  qHandle;
QbResult       results[qbiMaxQueryReturns];

rc=QbQueryStringSearch(
          "QbColorFeatureClass color=<255, 0, 0>" /*query string */
                "employee",                       /* user table */
                "picture",                        /* image column */
                maxResults,                       /* maximum number of results */
                0,                                /* query scope pointer * /
                qbiArray,                         /* store results in an array */
                &count,                           /* count of returned images */
                results);                         /* array of returned results */
```

Es folgt eine Anforderung, die die API QbQuerySearch verwendet. Beachten Sie, dass die Kennung für das Abfrageobjekt als Eingabe angegeben ist.

```
QbResult       returns[MaxQueryReturns];
SQLINTEGER     maxResults=qbiMaxQueryReturns;
SQLINTEGER     count;
QbQueryHandle  qHandle;
QbResult       results[qbiMaxQueryReturns];

rc=QbQuerySearch(
                 qHandle,                      / query object handle */
                "employee",                    /* user table */
                "picture",                     /* image column */
                maxResults,                    /* maximum number of results */
                0,                             /* query scope pointer * /
                qbiArray,                      /* store results in an array */
                &count,                        /* count of returned images */
                results);                      /* array of returned results */
```

Ähnlichkeitsergebnisse für Abbilder abrufen

Der Image Extender stellt vier UDFs zur Verfügung, die Sie in einer SQL-Anweisung verwenden können, um das Ähnlichkeitsergebnis eines katalogisierten Abbilds in einer Tabellenspalte abzurufen. Das Ähnlichkeitsergebnis ist ein Gleitkommawert mit doppelter Genauigkeit im Bereich von 0,0 bis gegen Unendlichkeit. Je niedriger das Ähnlichkeitsergebnis, desto mehr stimmen die Merkmalwerte des Abbilds mit den Merkmalwerten überein, die in der Abfrage angegeben sind. Ein Ähnlichkeitsergebnis von 0,0 bedeutet, dass das Abbild exakt übereinstimmt.

Zu den UDFs gehören:

- QbScorefromStr
- QbScoreTBfromStr
- QbScoreFromName
- QbScoreTBFromName

Empfehlung: Verwenden Sie die UDF QbScoreFromStr, um das Ähnlichkeitsergebnis eines einzelnen katalogisierten Abbilds abzurufen. Verwenden Sie die UDF QbScoreTBFromStr, um das Ähnlichkeitsergebnis von mehreren katalogisierten Abbildern in einer Tabellenspalte abzurufen.

Ähnlichkeitsergebnis für ein einzelnes Abbild abrufen

Verwenden Sie die UDF QbScoreFromStr, um das Ähnlichkeitsergebnis eines einzelnen katalogisierten Abbilds in einer Tabellenspalte abzurufen. Geben Sie eine Abfragezeichenfolge als Eingabe für die UDF QbScoreFromStr an. Wenn Sie die UDF QbScoreFromName verwenden, geben Sie den Namen des Abfrageobjekts als Eingabe für die UDF QbSco- reFromName an. Geben Sie bei beiden UDFs auch den Namen der Tabellenspalte an, die das Zielabbild enthält.

In der folgenden Abfrage wird die UDF QbScoreFromStr verwendet, um die katalogisierten Abbilder in einer Tabellenspalte zu suchen, deren Ähnlichkeitsergebnis für die Durchschnittsfarbe sehr nah an Rot liegt.

```
SELECT name, description
decimal (QbScoreFromStr(swatch_img,
            'QbColorFeatureClass color=<255, 0, 0>'),   /* query string *
                  10, 5) AS score
FROM fabric                                             /* table column */
ORDER BY score
```

Ähnlichkeitsergebnis für mehrere Abbilder abrufen

Verwenden Sie die UDF QbScoreTBFromStr, um das Ähnlichkeitsergebnis von mehreren katalogisierten Abbildern in einer Tabellenspalte abzurufen. Sie können die UDF QbScoreTBFromName verwenden, wenn Sie eine benannte Abfrage haben. Beide UDFs geben eine zweispaltige Tabelle mit Abbildkennungen und Ähnlichkeitsergebnissen zurück; die Zeilen in der Tabelle sind aufsteigend nach Ähnlichkeitsergebnis sortiert. Der Name der Kennungsspalte in der Ergebnistabelle ist IMAGE_ID; der Name der Ähnlichkeitsergebnisspalte ist SCORE.

Geben Sie eine Abfragezeichenfolge als Eingabe für die UDF QBScoreTBFromStr an. Geben Sie den Namen eines Abfrageobjekts als Eingabe für die UDF QbScoreTBFromName an. Geben Sie bei beiden UDFs auch den Namen der Tabelle und der Spalte an, die die Zielabbilder enthält. Sie können außerdem die maximale Anzahl an Zeilen angeben, die in der Ergebnistabelle zurückgegeben werden sollen. Wenn Sie keine maximale Anzahl an Ergebniszeilen angeben, gibt die UDF für jedes katalogisierte Abbild in der Zieltabellenspalte eine Zeile zurück.

In der folgenden Abfrage wird die UDF QbScoreTBFromStr verwendet, um die zehn katalogisierten Abbilder in einer Tabellenspalte zu suchen, deren Textur der eines Abbilds in einer Serverdatei am ähnlichsten ist.

```
SELECT name, description
FROM fabric
WHERE CAST (swatch_img as varchar(250)) IN
 SELECT CAST (image_id as varchar(25)) FROM TABLE
 (QbScoreTBFromStr
 (QbTextureFeatureClass file=<server,"patterns/ptrn07.gif">'   /*query string */

           'fabric',                                          /* table */
     'swatch_img',                                            /* table column */
     10))                                          /* maximum number of results */
  AS T1));
```

Programm für QBIC-Abfragebeispiel

Abb. 29 auf Seite 183 zeigt Teile eines in C geschriebenen Programms, mit dem eine QBIC-Abfrage erstellt und ausgeführt wird.Der Code in der Abbildung fragt Abbilder nach Durchschnittsfarbe ab. Er fordert den Benutzer zur Eingabe des Namens einer Farbe oder Abbilddatei auf. Der Benutzer kann auch ein Abbild, das von einer Abfrage zurückgegeben wird, als ein Beispielabbild für eine nachfolgende Abfrage verwenden. Das Programm verwendet danach die benannte Farbe oder die Farbe des Abbilds als Durchschnittsfarbe, um eine Spalte mit Abbildern abzufragen.

Das vollständige Programm befindet sich in der Datei QBICDEMO.C im Unterverzeichnis SAMPLES. Das vollständige Programm kann verwendet werden, um Abbilder nach Histogrammfarbe oder positionsgebundener Farbe sowie nach Durchschnittsfarbe abzufragen. Um das vollständige Programm ausführen zu können, müssen Sie die Beispielprogramme ENABLE, POPULATE und QBCATDMO ausführen (die sich auch im Unterverzeichnis SAMPLES befinden). Weitere Informationen zu Beispielprogrammen befinden sich im Anhang B, „Beispielprogramme und Multimediadateien”, auf Seite 529.

Beachten Sie folgende Punkte in Abb. 29 auf Seite 183:

1 Die Kopfdatei dmbqbapi wird eingeschlossen.

2 Der Benutzer wird zur Eingabe von Datenbankinformationen aufgefordert.

3 Eine Verbindung zur Datenbank wird hergestellt.

4 Ein Abfrageobjekt wird erstellt.

5 Ein Merkmal wird zu einem Abfrageobjekt hinzugefügt.

6 Der Benutzer wird zur Eingabe des Eingabetyps aufgefordert (Farbname, Abbilddatei oder zuvor abgerufenes Abbild).

7 Die Datenquelle für das Merkmal wird angegeben. Die Datenquelle ist eine explizite Angabe für die Durchschnittsfarbe.

8 Die Abfrage wird ausgeführt. Der Image Extender durchsucht die gesamte Spalte mit Abbildern. Außerdem wird 10 als maximale Anzahl von Abbildern angegeben, die zurückgegeben werden sollen.

9 Das nächste Abbild aus der Gruppe der zurückgegebenen Abbilder wird angezeigt. Weitere Informationen zur Anzeige von Abbildern befinden sich im Abschnitt „Abbild oder Videovollbild in normaler Größe anzeigen” auf Seite 151.

10 Das Abfrageobjekt wird gelöscht.

Das Unterverzeichnis SAMPLES enthält ein weiteres Programm, das die Erstellung und Verwendung einer QBIC-Abfrage demonstriert. Das Programm, `QbicQry.java`, zeigt eine Möglichkeit, die Suchkriterien für eine QBIC-Abfrage grafisch anzugeben. Beispielsweise bietet das Programm ein Farbauswahlmenü, mit dem die Durchschnittsfarbe ausgewählt werden kann. Das Programm setzt die Auswahl in eine Abfragezeichenfolge um.

```
#include <sql.h>
#include <sqlcli.h>
#include <sqlcli1.h>
#include <dmbqbqpi.h>  1
#include <stdio.h>
#include <string.h>
#include <malloc.h>
#include <color.h>
#include <ctype.h>

#define   MaxQueryReturns        10

#define   MaxDatabaseNameLength  SQL_SH_IDENT
#define   MaxUserIdLength        SQL_SH_IDENT
#define   MaxPasswordLength      SQL_SH_IDENT
#define   MaxTableNameLength     SQL_LG_IDENT
#define   MaxColumnNameLength    SQL_LG_IDENT

static  char       databaseName[MaxDatabaseNameLength+1];
static  char       userid[MaxUserIdLength+1];
static  char       password[MaxPasswordLength+1];

static  char       tableName[MaxTableNameLength+1];
static  char       columnName[MaxColumnNameLength+1];

static  char       line[4000];

static  QbResult   results[MaxQueryReturns];
static  long       currentImage   = -1;
static  long       imageCount  = 0;

static  char*      tableAndColumn;

static  QbQueryHandle averageHandle = 0;
static  QbQueryHandle histogramHandle = 0;
static  QbQueryHandle drawHandle = 0;
static  QbQueryHandle lastHandle = 0;

static  SQLHENV    henv;
static  SQLHDBC    hdbc;
static  SQLHSTMT   hstmt;
static  SQLRETURN  rc;

static  char*    listQueries =
 "SELECT NAME,DESCRIPTION FROM MMDBSYS.QBICQUERIES ORDER BY NAME";

static  char*    menu[] = {
```

Abbildung 29. Programm für QBIC-Abfragebeispiel (Teil 1 von 6)

```
/*
 12345678901234567890123456789012345678901234567890123456789012345678901234567890 */
"",
"+-------------------------------------------------------------------------------+",
"| AVERAGE COLOR colorname                                                       |",
"| AVERAGE FILE filename format                                                  |",
"| AVERAGE LAST                                                                  |",
"| Press Enter to display the next image in the series                           |",
"+-------------------------------------------------------------------------------+",
"",
        0
};

static  char*       help[] = {
"",
"AVERAGE     Execute an average color query",
"  COLOR       Specifies the color to query for",
"  FILE        Specifies the file to compute the average color from",
"  LAST        Specifies the last displayed image be used to compute the color",
"  NEXT        Displays the next image from the current query or nothing if",
"                all of the image have been displayed."
"",
">>pause<<",
         0
};
/*****************************************************************************/
/* doNext()                                                                  */
/*****************************************************************************/
static void doNext(void)
{
 int ret;
   if (currentImage < imageCount)
      currentImage++;
   if (currentImage < imageCount)
      ret = DBiBrowse("/usr/local/bin/xv %s", MMDB_PLAY_HANDLE,
            results[currentImage].imageHandle, MMDB_PLAY_NO_WAIT);  9
}
```

Abbildung 29. Programm für QBIC-Abfragebeispiel (Teil 2 von 6)

```
/*****************************************************************************/
/* doAverage()                                                               */
/*****************************************************************************/
static void doAverage(void)
{
   QbQueryHandle qohandle = 0; QbImageSource is; char* type;
   char* arg1; char* arg2;

   type = nextWord(0);
   if (abbrev(type, "color", 1)) {
      is.type = qbiSource_AverageColor;
      arg1 = nextWord(0);
      if (arg1 == 0) {
         printf("AVERAGE COLOR command requires a colorname argument.\n");
      return;
      }
      if (getColor(arg1, &is.averageColor) == 0) {
         printf("The colorname entered was not recognized.\n");
      return;
      }
   }
   else if (abbrev(type, "file", 1)) {
      is.type = qbiSource_ClientFile;
      arg1 = nextWord(0);
      if (arg1 == 0) {
         printf("AVERAGE FILE command requires a filename argument.\n");
      return;
      }
      arg2 = nextWord(0);
      if (arg2 == 0) {
         printf("AVERAGE FILE command requires a file format argument.\n");
      return;
      }
      strcpy(is.clientFile.fileName, arg1);
      strcpy(is.clientFile.format, arg2);
   }
   else if (abbrev(type, "last", 1)) {
      is.type = qbiSource_ImageHandle;
      if (0 <= currentImage &&; currentImage < imageCount)
         strcpy(is.imageHandle, results[currentImage]imageHandle);
      else {
         printf("No last image for AVERAGE LAST command\n");
      return;
      }
   }
```

Abbildung 29. Programm für QBIC-Abfragebeispiel (Teil 3 von 6)

```
   else {
      printf("AVERAGE command only supports COLOR, FILE, and LAST types.\n");
      return;
   }

   _QbQuerySetFeatureData(averageHandle, "QbColorFeatureClass", &is); 7
   _QbQuerySearch(averageHandle, tableAndColumn, "IMAGE",
         MaxQueryReturns, 0, 0, &imageCount, results); 8
   lastHandle = averageHandle;

   currentImage = -1;
}
/*****************************************************************************/
/* commandLoop()                                                             */
/*****************************************************************************/
void commandLoop(void)
{
   int   done = 0;

   while (!done) { 6
      displayText(menu);
      printf("%d", currentImage + 1);
      if (0 <= currentImage &&; currentImage < imageCount)
            printf("  %8.6f", results[currentImage].score);
      printf("> ");
   gets(line);
      done = processCommand(line);
   }
}
```

Abbildung 29. Programm für QBIC-Abfragebeispiel (Teil 4 von 6)

```
/*****************************************************************************/
/* main()                                                                    */
/*****************************************************************************/
void main(void)
{
   char*    inst;
   int      i;

   printf("\n\n");
   printf("Please enter: database_name [user_id] [password] "\n"); 2
   gets(line);
   if (copyWord(line, databaseName, sizeof(databaseName)) == 0)
      exit(0);
   copyWord(0, userid, sizeof(userid));
   copyWord(0, password, sizeof(password));
   printf("\n");

   if (SQLAllocEnv(&henv) != SQL_SUCCESS)
      sqlError(SQL_NULL_HSTMT);
   if (SQLAllocConnect(henv, &hdbc) != SQL_SUCCESS)
      sqlError(SQL_NULL_HSTMT);
   if (SQLConnect(hdbc, 3
                  (SQLCHAR*)databaseName,
                  SQL_NTS,
                  (SQLCHAR*)userid,
                  SQL_NTS,
                  (SQLCHAR*)password,
                  SQL_NTS) != SQL_SUCCESS)
      sqlError(SQL_NULL_HSTMT);

   printf("Initializing . . .\n");
```

Abbildung 29. Programm für QBIC-Abfragebeispiel (Teil 5 von 6)

```
      inst = getenv("DB2INSTANCE");
   if (inst != 0 &&; strcmp(inst, "keeseyt") == 0)
      tableAndColumn = "KEESEY.TEST";
   else
      tableAndColumn = "QBICDEMO.TEST";

   _QbQueryCreate(&averageHandle); 4
   _QbQueryAddFeature(averageHandle, "QbColorFeatureClass");
   _QbQueryCreate(&histogramHandle);
   _QbQueryAddFeature(histogramHandle, "QbColorHistogramFeatureClass");
   _QbQueryCreate(&drawHandle);
   _QbQueryAddFeature(drawHandle, "QbDrawFeatureClass"); 5

   commandLoop();

   _QbQueryDelete(drawHandle);
   _QbQueryDelete(histogramHandle); 10
   _QbQueryDelete(averageHandle);

   if (SQLDisconnect(hdbc) != SQL_SUCCESS)
      sqlError(SQL_NULL_HSTMT);
   if (SQLFreeConnect(hdbc) != SQL_SUCCESS)
      sqlError(SQL_NULL_HSTMT);
   if (SQLFreeEnv(henv) != SQL_SUCCESS)
      sqlError(SQL_NULL_HSTMT);
}
```

Abbildung 29. Programm für QBIC-Abfragebeispiel (Teil 6 von 6)

Teil 4. Referenz

Kapitel 13. Benutzerdefinierte Typen und Funktionen

In diesem Kapitel erhalten Sie Referenzinformationen zu den UDTs (benutzerdefinierten Typen) und UDFs (benutzerdefinierten Funktionen), die durch die DB2 Extender erstellt werden.

Schema

Die Extender verwenden das Schema MMDBSYS für alle objektrelationalen Objekte einschließlich UDTs und UDFs.

Benutzerdefinierte Typen

In Tabelle 14 werden die benutzerdefinierten Typen, die durch die DB2 Extender erstellt werden, aufgelistet und beschrieben. Außerdem werden die DB2-Quellendatentypen für jeden UDT aufgelistet.

Tabelle 14. Benutzerdefinierte Typen, die durch die DB2 Extender erstellt werden

UDT	Quellendatentyp	Beschreibung
DB2IMAGE	VARCHAR(250)	Abbildkennung. Eine Zeichenfolge variabler Länge, die Informationen enthält, die für den Zugriff auf ein Abbildobjekt erforderlich sind. Abbildkennungen werden in einer Benutzertabellenspalte gespeichert, die für den Image Extender aktiviert ist.
DB2AUDIO	VARCHAR(250)	Audiokennung. Eine Zeichenfolge variabler Länge, die Informationen enthält, die für den Zugriff auf ein Audioobjekt erforderlich sind. Audiokennungen werden in einer Benutzertabellenspalte gespeichert, die für den Audio Extender aktiviert ist.
DB2VIDEO	VARCHAR(250)	Videokennung. Eine Zeichenfolge variabler Länge, die Informationen enthält, die für den Zugriff auf ein Videoobjekt erforderlich sind. Videokennungen werden in einer Benutzertabellenspalte gespeichert, die für den Video Extender aktiviert ist.

Benutzerdefinierte Funktionen

In diesem Abschnitt erhalten Sie Referenzinformationen zu den DB2 Extendern. Die UDFs werden in alphabetischer Reihenfolge aufgelistet.

Die folgenden Informationen werden für die einzelnen UDFs zur Verfügung gestellt:

- Die Extender, die die UDF zur Verfügung stellen
- Eine Kurzbeschreibung
- Die Kopfdatei für die UDF
- Die SQL-Syntax der UDF
- Eine Beschreibung, einschließlich Datentyp, der UDF-Parameter
- Der durch die UDF zurückgegebene Wert, einschließlich Datentyp
- Beispiele für die Verwendung

In Tabelle 15 werden die UDFs aufgelistet und die Extender identifiziert, die die einzelnen UDFs zur Verfügung stellen. Außerdem enthält die Tabelle Verweise auf weitere Informationen zu den einzelnen UDFs. Die UDFs in dieser Tabelle können in eingebetteten SQL-Anweisungen oder in DB2 CLI-Aufrufen codiert werden.

Tabelle 15. DB2 Extender-UDFs

UDF	Beschreibung	Image	Audio	Video	Siehe Seite
AlignValue	Gibt die Anzahl an Byte pro Sample in einem WAVE-Ton oder in einer Tonspur eines Videos zurück.		x	x	195
AspectRatio	Gibt das Streckungsverhältnis der ersten Spur eines MPEG1- und MPEG2-Videos zurück.			x	196
BitsPerSample	Gibt die Anzahl an Datenbit zurück, die zur Darstellung der einzelnen Beispiele von WAVE- oder AIFF-Audiodaten in Tönen oder in einer Tonspur eines Videos verwendet werden.		x	x	197
BytesPerSec	Gibt die Datenübertragungsgeschwindigkeit, in Durchschnittsbyte pro Sekunde, für einen WAVE-Ton an.		x		198
Comment	Gibt einen Kommentar zurück, der mit einem Abbild, Ton oder Video gespeichert wird, oder aktualisiert ihn.	x	x	x	199
CompressType	Gibt das Komprimierungsformat eines Videos, z. B. MGEG-1, zurück.			x	201
Content	Ruft den Inhalt eines Abbilds, Tons oder Videos aus der Datenbank ab oder aktualisiert ihn.	x	x	x	202
DB2Audio	Speichert den Inhalt eines Tons in einer Datenbanktabelle.		x		207
DB2Image	Speichert den Inhalt eines Abbilds in einer Datenbanktabelle.	x			210
DB2Video	Speichert den Inhalt eines Videos in einer Datenbanktabelle.			x	214
Duration	Gibt die Dauer (d. h. die Spieldauer in Sekunden) eines WAVE- oder AIFF-Tons oder eines Videos zurück.		x	x	217
Filename	Gibt den Namen der Serverdatei zurück, die den Inhalt eines Abbilds, Tons oder Videos enthält.	x	x	x	218
FindInstrument	Gibt die Spurnummer des ersten Auftretens eines bestimmten Instruments in einem MIDI-Ton zurück.		x		219
FindTrackName	Gibt die Nummer einer angegebenen benannten Spur in einem MIDI-Ton zurück.		x		220
Format	Gibt das Format eines Abbilds, Tons oder Videos zurück.	x	x	x	221
FrameRate	Gibt den Durchsatz eines Videos in Vollbildern pro Sekunde zurück.			x	222

Tabelle 15. DB2 Extender-UDFs (Forts.)

UDF	Beschreibung	Image	Audio	Video	Siehe Seite
GetInstruments	Gibt die Instrumentennamen aller Instrumente in einem MIDI-Ton zurück.		x		223
GetTrackNames	Gibt die Namen aller Spuren in einem MIDI-Ton zurück.		x		224
Height	Gibt die Höhe eines Abbilds oder Videovollbilds in Pixeln zurück.	x		x	225
Importer	Gibt die Benutzer-ID der Person zurück, die ein Abbild, einen Ton oder ein Video in einer Datenbanktabelle gespeichert hat.	x	x	x	226
ImportTime	Gibt eine Zeitmarke zurück, die angibt, wann ein Abbild, Ton oder Video in einer Datenbanktabelle gespeichert wurde.	x	x	x	227
MaxBytesPerSec	Gibt den maximalen Durchsatz eines Videos in Byte pro Sekunde zurück.			x	228
NumAudioTracks	Gibt die Anzahl an Tonspuren in einem Video oder MIDI-Ton zurück.		x	x	229
NumChannels	Gibt die Anzahl an aufgezeichneten Tonkanälen in einem WAVE- oder AIFF-Ton oder einem Video zurück.		x	x	230
NumColors	Gibt die Anzahl an Farben in einem Abbild zurück.	x			231
NumFrames	Gibt die Anzahl an Vollbildern in einem Video zurück.			x	232
NumVideoTracks	Gibt die Anzahl an Videospuren in einem Video zurück.			x	233
QbScoreFromName	Gibt das Ähnlichkeitsergebnis eines Abbilds (unter Verwendung eines Abfrageobjekts) zurück. (Ersetzt QbScore.)	x			234
QbScoreFromStr	Gibt das Ähnlichkeitsergebnis eines Abbilds (unter Verwendung einer Abfragezeichenfolge) zurück.	x			236
QbScoreTBFromName	Gibt eine Tabelle von Ähnlichkeitsergebnissen von einer Abbildspalte (unter Verwendung eines benannten Abfrageobjekts) zurück.	x			237
QbScoreTBFromStr	Gibt eine Tabelle von Ähnlichkeitsergebnissen von einer Abbildspalte (unter Verwendung einer Abfragezeichenfolge) zurück.	x			239
Replace	Aktualisiert den Inhalt eines Abbilds, Tons oder Videos, das/der in einer Datenbank gespeichert ist, und aktualisiert dessen Kommentar.	x	x	x	241

Tabelle 15. DB2 Extender-UDFs (Forts.)

UDF	Beschreibung	Image	Audio	Video	Siehe Seite
SamplingRate	Gibt die Abtastrate eines WAVE- oder AIFF-Tons oder einer Tonspur in einem Video in Samples pro Sekunde zurück.		x	x	244
Size	Gibt die Größe eines Abbilds, Tons oder Videos in Byte zurück.	x	x	x	245
Thumbnail	Gibt eine Piktogrammversion eines Abbilds oder Videovollbilds zurück, das in einer Datenbank gespeichert ist, oder aktualisiert die Version.	x		x	246
TicksPerQNote	Gibt die Taktgeschwindigkeit eines aufgezeichneten MIDI-Tons in Impulsen pro Viertelnote zurück.		x		248
TicksPerSec	Gibt die Taktgeschwindigkeit eines aufgezeichneten MIDI-Tons in Impulsen pro Sekunde zurück.		x		249
Updater	Gibt die Benutzer-ID der Person zurück, die ein Abbild, einen Ton oder ein Video zuletzt in einer Datenbanktabelle aktualisiert hat.	x	x	x	250
UpdateTime	Gibt eine Zeitmarke zurück, die angibt, wann ein Abbild, Ton oder Video zuletzt in einer Datenbanktabelle aktualisiert wurde.	x	x	x	251
Width	Gibt die Breite eines Abbilds oder Videovollbilds in Pixeln zurück.	x		x	252

AlignValue

Image	Audio	Video
	X	X

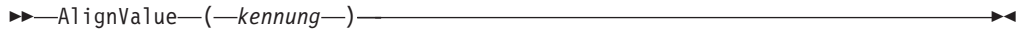
Gibt die Anzahl an Byte pro Sample in einem WAVE-Ton oder in einer Tonspur eines Videos zurück. Ein WAVE-Ton kann seine Daten unter Verwendung von einem Byte pro Sample (8-Bit-Mono, die so genannte „Byteausrichtung"), unter Verwendung von zwei Byte pro Sample (8-Bit-Stereo oder 16-Bit-Mono, die so genannte „Wortausrichtung") oder unter Verwendung von vier Byte pro Sample (16-Bit-Stereo, die so genannte „Doppelwortausrichtung") speichern.

Kopfdatei

Audio dmbaudio.h

Video dmbvideo.h

Syntax



Parameter (Datentyp)

kennung (DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Tons enthält.

Rückgabewerte (Datentyp)

Byte pro Sample eines WAVE-Tons oder einer Tonspur in einem Video (SMALLINT). Mögliche Werte sind:

1 Byteausrichtung

2 Wortausrichtung

4 Doppelwortausrichtung

Nullwert Ton in anderen Formaten

Beispiele

Abrufen der Dateinamen aller Audiodaten, die in der Spalte 'sound' der Tabelle 'employee' gespeichert sind, bei denen die Wortausrichtung ausgeführt wird:

```
EXEC SQL BEGIN DECLARE SECTION;
 char hvAud_fname[251];
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (SOUND)
     INTO :hvAud_fname
  FROM EMPLOYEE
      WHERE ALIGNVALUE(SOUND) = 2;
```

Abrufen der Byte pro Sample einer Tonspur in einem Video, das in der Spalte 'video' der Tabelle 'employee' für Anita Jones gespeichert ist:

```
EXEC SQL BEGIN DECLARE SECTION;
      short hvAlign_val;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT ALIGNVALUE(VIDEO)
      INTO :hvAlign_val
      FROM EMPLOYEE
      WHERE NAME='Anita Jones';
```

AspectRatio

Image	Audio	Video
		X

Gibt das Streckungsverhältnis der ersten Spur eines MPEG-Videos zurück.

Kopfdatei
dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Videos enthält.

Rückgabewerte (Datentyp)
Streckungsverhältnis der ersten Spur des MPEG-Videos oder ein Nullwert für Videos in anderen Formaten (SMALLINT)

Beispiele
Abrufen des Streckungsverhältnisses des Videos, das für Robert Smith in der Spalte 'video' der Tabelle 'employee' gespeichert ist:

```
EXEC SQL BEGIN DECLARE SECTION;
      short hvAsp_ratio;
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT ASPECTRATIO(VIDEO)
      INTO :hvAsp_ratio
  FROM EMPLOYEE
      WHERE NAME='Robert Smith';
```

BitsPerSample

Image	Audio	Video
	X	X

Gibt die Anzahl an Datenbit zurück, die zur Darstellung der einzelnen Beispiele von WAVE- oder AIFF-Audiodaten in Tönen oder in einer Tonspur eines Videos verwendet werden.

Kopfdatei

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Anzahl von Datenbit, die zur Darstellung der einzelnen Beispiele eines Videos oder eines WAVE- oder AIFF-Tons (SMALLINT) verwendet werden. Ein Nullwert wird für Töne in anderen Formaten zurückgegeben.

Beispiele

Abrufen der Dateinamen aller WAVE-Töne, die in der Spalte 'sound' in der Tabelle 'employee' gespeichert sind, deren Bit pro Sample-Wert gleich 8 ist:

```
EXEC SQL BEGIN DECLARE SECTION;
 char hvAud_fname[251];
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (SOUND)
     INTO :hvAud_fname
  FROM EMPLOYEE
      WHERE FORMAT(SOUND)='WAVE'
      AND BITSPERSAMPLE(SOUND) = 8;
```

BytesPerSec

Image	Audio	Video
	X	

Gibt die Datenübertragungsgeschwindigkeit, in Durchschnittsbyte pro Sekunde, für einen WAVE-Ton an.

Kopfdatei
dmbaudio.h

Syntax

Parameter (Datentyp)

kennung (DB2AUDIO)
Name der Spalte oder Hostvariable, die die Kennung des Tons enthält.

Rückgabewerte (Datentyp)
Datenübertragungsgeschwindigkeit (INTEGER). Ein Nullwert wird für Töne in anderen Formaten zurückgegeben.

Beispiele
Abrufen der Dateinamen aller Töne, die in der Spalte 'sound' in der Tabelle 'employee' gespeichert sind, deren Wert für die Durchschnittsbyte pro Sekunde kleiner als 44100 ist:

```
EXEC SQL BEGIN DECLARE SECTION;
 char hvAud_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(SOUND)
      INTO :hvAud_fname
      FROM EMPLOYEE
      WHERE BYTESPERSEC(SOUND) < 44100;
```

Comment

Image	Audio	Video
X	X	X

Gibt einen Kommentar zurück, der mit einem Abbild, Ton oder Video gespeichert wird, oder aktualisiert ihn.

Kopfdatei

Image dmbimage.h

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Abrufen des Kommentars

Syntax

Aktualisieren des Kommentars

Parameter (Datentyp)

kennung (DB2IMAGE, DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds, Tons oder Videos enthält.

neuer_kommentar (LONG VARCHAR)
Neuer Kommentar für die Aktualisierung. Durch einen Nullwert oder eine leere Zeichenfolge wird der bestehende Kommentar gelöscht.

Rückgabewerte (Datentyp)

Beim Aktualisieren die Kennung des Abbilds, Tons oder Videos (DB2IMAGE, DB2AUDIO oder DB2VIDEO). Beim Abrufen der Kommentar (LONG VARCHAR).

Beispiele

Abrufen der Dateinamen aller Abbilder aus der Spalte 'picture' der Tabelle 'employee', bei denen das Wort „confidential" im zugeordneten Kommentar vorkommt:

```
EXEC SQL BEGIN DECLARE SECTION;
 char hvImg_fname[255;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
     INTO :hvImg_fname
     FROM EMPLOYEE
     WHERE COMMENT(PICTURE)
        LIKE '%confidential%';
```

Aktualisieren des Kommentars, der dem Videoclip von Anita Jones in der Spalte 'video' der Tabelle 'employee' zugeordnet ist:

```
EXEC SQL BEGIN DECLARE SECTION;
struct{
       short len;
      char data[4000];
      }hvRemarks;
  EXEC SQL END DECLARE SECTION;

/* Get the old comment */

EXEC SQL SELECT COMMENT(VIDEO)
     INTO :hvRemarks
  FROM EMPLOYEE
     WHERE NAME = 'Anita Jones';

/* Update the comment */

hvRemarks.data[hvRemarks.len]='\0';
strcat (hvRemarks.data, "Updated video");
hvRemarks.len=strlen(hvRemarks.data);

EXEC SQL UPDATE EMPLOYEE
     SET VIDEO=COMMENT(VIDEO, :hvRemarks)
     WHERE NAME = 'Anita Jones';
```

CompressType

Image	Audio	Video
		X

Gibt das Komprimierungsformat eines Videos, z. B. MGEG-1, zurück.

Kopfdatei
dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Videos enthält.

Rückgabewerte (Datentyp)
Komprimierungsformat des Videos (VARCHAR(8)).

Beispiele
Abrufen der Namen aller Videos, die in der Spalte 'video' der Tabelle 'employee' gespeichert sind, deren Komprimierungsformat MPEG-1 ist:

```
EXEC SQL BEGIN DECLARE SECTION;
 char hvVid_fname[251];
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (VIDEO)
     INTO :hvVid_fname
  FROM EMPLOYEE
     WHERE COMPRESSTYPE(VIDEO) = 'MPEG1';
```

Content

Image	Audio	Video
X	X	X

Ruft den Inhalt eines Abbilds, Tons oder Videos aus der Datenbank ab oder aktualisiert ihn. Der Inhalt kann in einen Clientpuffer, eine Clientdatei oder eine Serverdatei abgerufen werden.

Kopfdatei

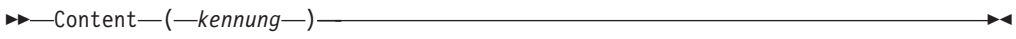
Image dmbimage.h

Audio dmbaudio.h

Video dmbvideo.h

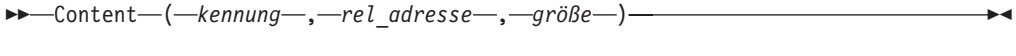
Syntax

Abrufen des Inhalts in einen Puffer oder eine Clientdatei



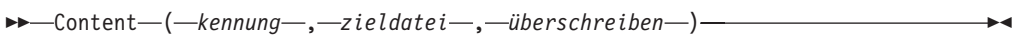
Syntax

Abrufen eines Inhaltsegments in einen Puffer oder eine Clientdatei



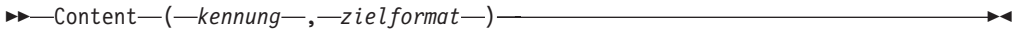
Syntax

Abrufen des Inhalts in eine Serverdatei



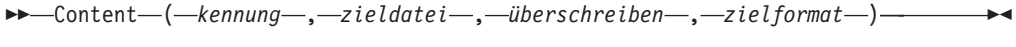
Syntax

Abrufen des Inhalts in einen Puffer oder eine Clientdatei mit Formatumsetzung (nur Abbilder)



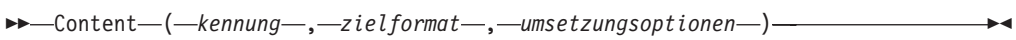
Syntax

Abrufen des Inhalts in eine Serverdatei mit Formatumsetzung (nur Abbilder)



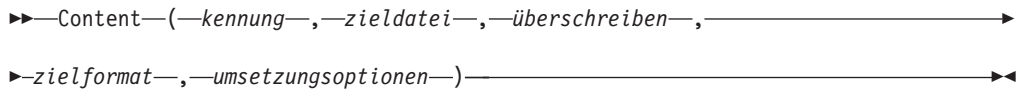
Syntax

Abrufen des Inhalts in einen Puffer oder eine Clientdatei mit Formatumsetzung und zusätzlichen Änderungen (nur Abbilder)



Syntax

Abrufen des Inhalts in eine Serverdatei mit Formatumsetzung und zusätzlichen Änderungen (nur Abbilder)

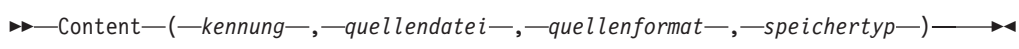


Syntax

Aktualisieren des Inhalts eines Puffers oder einer Clientdatei

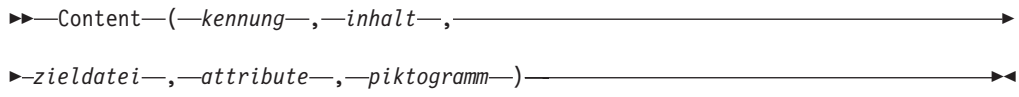
Syntax

Aktualisieren des Inhalts einer Serverdatei



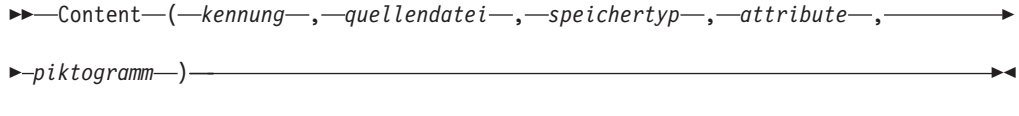
Syntax

Aktualisieren des Inhalts eines Puffers oder einer Clientdatei mit benutzerdefinierten Attributen



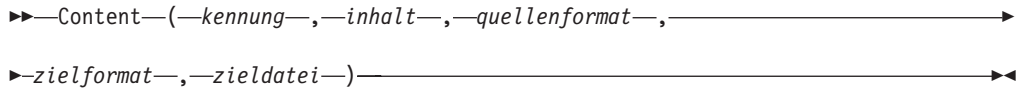
Syntax

Aktualisieren des Inhalts einer Serverdatei mit benutzerdefinierten Attributen



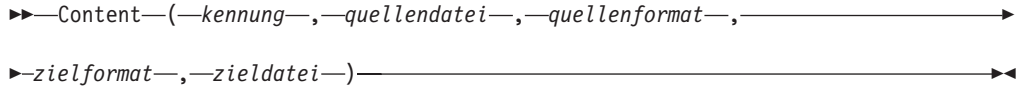
Syntax

Aktualisieren des Inhalts eines Puffers oder einer Clientdatei mit Formatumsetzung (nur Abbilder)



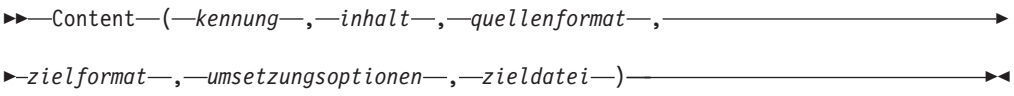
Syntax

Aktualisieren des Inhalts einer Serverdatei mit Formatumsetzung (nur Abbilder)



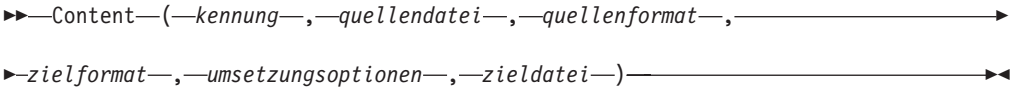
Syntax

Aktualisieren des Inhalts eines Puffers oder einer Clientdatei mit Formatumsetzung und zusätzlichen Änderungen (nur Abbilder)



Syntax

Aktualisieren des Inhalts einer Serverdatei mit Formatumsetzung und zusätzlichen Änderungen (nur Abbilder)



Parameter (Datentyp)

kennung (DB2IMAGE, DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds, Tons oder Videos enthält.

rel_adresse (INTEGER)
Die relative Startadresse (Ursprung 1) eines Abbilds, Tons oder Videos, das/der abgerufen werden soll.

größe (INTEGER)
Anzahl an Byte eines Abbilds, Tons oder Videos, das/der abgerufen werden soll.

quellendatei (LONG VARCHAR)
Der Name der Datei, die den Inhalt zum Aktualisieren des Abbilds, Tons oder Videos enthält.

zieldatei (LONG VARCHAR)
Beim Abrufen der Name der Datei, in die das Abbild, der Ton oder das Video abgerufen werden soll. Beim Aktualisieren der Name der Datei, die das Abbild, den Ton oder das Video enthält, das aktualisiert werden soll.

speichertyp (INTEGER)
Ein Wert, der angibt, wo das aktualisierte Abbild, der aktualisierte Ton oder das aktualisierte Video gespeichert werden soll. Die Konstante MMDB_STORAGE_TYPE_INTERNAL (Wert=1) gibt an, dass das aktualisierte Objekt in der Datenbank als BLOB gespeichert wird. Die Konstante MMDB_STORAGE_TYPE_EXTERNAL (Wert=0) gibt an, dass das aktualisierte Objekt in einer Serverdatei gespeichert wird.

überschreiben (INTEGER)
Ein Wert, der angibt, ob die Zieldatei, wenn sie bereits existiert, überschrieben wird. Der Wert kann 0 oder 1 sein. Der Wert 0 bedeutet, dass die Zieldatei nicht überschrieben wird (im Prinzip findet kein Abrufen statt). Der Wert 1 bedeutet, dass die Zieldatei überschrieben wird, wenn sie bereits existiert.

zielformat (VARCHAR(8))
Das Format des Abbilds nach dem Abrufen oder Aktualisieren. Das Format des Quellenabbilds wird entsprechend umgesetzt. Beim Abrufen eines Abbilds in eine Serverdatei muss das Zielformat, wenn die Zieldatei der Quellendatei entspricht, dasselbe Format sein wie das Quellenformat. Für das MPG1-Format können Sie MPG1, mpg1, MPEG1 bzw. mpeg1 angeben. Für das MPG2-Format können Sie MPG2, mpg2, MPEG2 bzw. mpeg2 angeben.

umsetzungsoptionen (VARCHAR(100))
Ein Wert, der Änderungen, wie z. B. Drehung und Komprimierung, angibt, die beim Abrufen oder Aktualisieren des Abbilds angewendet werden sollen. Unterstützte Umsetzungsoptionen befinden sich in Tabelle 9 auf Seite 112.

inhalt (BLOB(2G) AS LOCATOR)
Die Hostvariable, die den Inhalt für die Aktualisierung des Abbilds, Tons oder Videos enthält. Die Hostvariable kann vom Typ BLOB, BLOB_FILE oder BLOB_LOCATOR sein. DB2 stuft den Datentyp des Inhalts in BLOB_LOCATOR um und übergibt den LOB-Zeiger an die UDF Content.

quellenformat (VARCHAR(8))
Das Format der Quelle für die Aktualisierung des Abbilds, Tons oder Videos. Ein Nullwert oder eine leere Zeichenfolge kann angegeben werden, bzw. nur für Abbilder die Zeichenfolge ASIS. Bei diesen drei Angaben versucht der Extender, das Format automatisch zu bestimmen. Für das MPG1-Format können Sie MPG1, mpg1, MPEG1 bzw. mpeg1 angeben. Für das MPG2-Format können Sie MPG2, mpg2, MPEG2 bzw. mpeg2 angeben.

attribute (LONG VARCHAR FOR BIT)
Die Attribute des Abbilds, Tons oder Videos.

piktogramm (LONG VARCHAR FOR BIT DATA)
Ein Piktogramm des Abbilds oder Videovollbilds (nur Abbild und Video).

Rückgabewerte (Datentyp)

Der Inhalt des abgerufenen Abbilds, Tons oder Videos, wenn er in einen Puffer abgerufen wird (BLOB(2G) AS LOCATOR). Wenn der Inhalt in eine Datei abgerufen wird, VARCHAR(254).

Beim Aktualisieren die Kennung des zu aktualisierenden Abbilds, Tons oder Videos (DB2IMAGE, DB2AUDIO oder DB2VIDEO).

Beispiele

Abrufen des Abbilds, das für Anita Jones in der Spalte 'picture' der Tabelle 'employee' gespeichert ist, in eine Serverdatei:

```
struct{
       short len;
      char data[250];
      }hvImg_fname;
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT (PICTURE,
       '/employee/images/ajones.bmp',1)
      INTO :hvImg_fname
  FROM EMPLOYEE
       WHERE NAME='Anita Jones';
```

Abrufen des 1-MB-Audioclips, der für Robert Smith in der Spalte 'sound ' der Tabelle 'employee' gespeichert ist, in einen Clientpuffer:

```
EXEC SQL BEGIN DECLARE SECTION;
     SQL TYPE IS BLOB_LOCATOR audio_loc;
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT (SOUND, 1, 1000000)
      INTO :audio_loc
  FROM EMPLOYEE
      WHERE NAME='Robert Smith';
```

Aktualisieren des Abbilds für Anita Jones in der Spalte 'picture' der Tabelle 'employee'; Umsetzen des Formats des Abbilds von BMP in GIF und Verkleinern des Abbilds auf 50 % der Originalgröße:

```
EXEC SQL UPDATE EMPLOYEE
  SET picture = CONTENT(PICTURE,
       '/employee/newimg/ajones.bmp',
          'BMP',
          'GIF',
        '-s 0.5',
       '');
       WHERE NAME='Anita Jones';
```

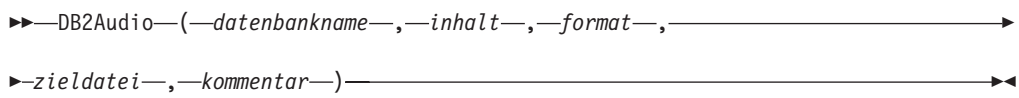
DB2Audio

Image	Audio	Video
	X	

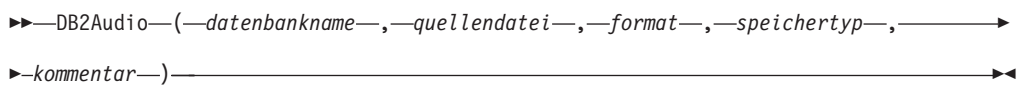
Speichert den Inhalt eines Tons in einer Datenbanktabelle. Die Tonquelle kann ein Clientpuffer, eine Clientdatei oder eine Serverdatei sein.Der Ton kann in der Datenbanktabelle als BLOB oder in einer Serverdatei (auf die durch die Datenbanktabelle verwiesen wird) gespeichert werden. Die Tonquelle kann ein unterstütztes Format haben, wobei der DB2Audio Extender deren Speicherattribute identifiziert; es kann aber auch ein nicht unterstütztes Format haben, wobei die Attribute in der UDF angegeben werden müssen.

Kopfdatei
dmbaudio.h

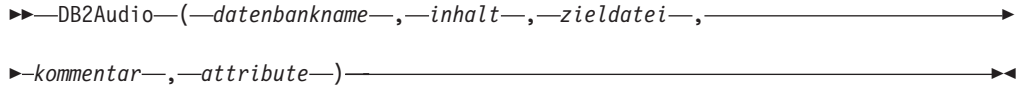
Syntax
Speichern des Inhalts eines Puffers oder einer Clientdatei



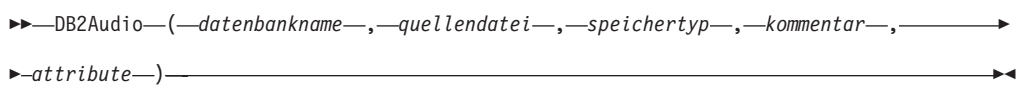
Syntax
Speichern des Inhalts einer Serverdatei



Syntax
Speichern des Inhalts eines Puffers oder einer Clientdatei mit benutzerdefinierten Attributen



Syntax
Speichern des Inhalts einer Serverdatei mit benutzerdefinierten Attributen



Parameter (Datentyp)

datenbankname (VARCHAR(18))
Der Name der Datenbank, zu der momentan eine Verbindung besteht, wie durch das Sonderregister CURRENT SERVER angegeben.

inhalt (BLOB(2G) AS LOCATOR)
Die Hostvariable, die den Inhalt des Tons enthält. Die Hostvariable kann

vom Typ BLOB, BLOB_FILE oder BLOB_LOCATOR sein. DB2 stuft den Datentyp des Inhalts in BLOB_LOCATOR um und übergibt den LOB-Zeiger an die UDF DB2Audio.

format (VARCHAR(8))
Das Format des Quellentons. Ein Nullwert oder eine leere Zeichenfolge kann angegeben werden, wobei der Audio Extender versucht, das Quellenformat automatisch zu bestimmen. Der Ton wird in dem gleichen Format gespeichert wie die Quelle. Unterstützte Audioformate befinden sich in Tabelle 8 auf Seite 111.

zieldatei (LONG VARCHAR)
Der Name der Zielserverdatei (zum Speichern in eine Serverdatei) oder ein Nullwert bzw. eine leere Zeichenfolge (zum Speichern in eine Datenbanktabelle als BLOB). Die Zieldatei kann mit einem vollständig qualifizierten Namen angegeben werden. Wenn der Name nicht als vollständig qualifizierter Name angegeben wird, werden die Umgebungsvariablen DB2AUDIOSTORE und DB2MMSTORE auf dem Server verwendet, um die Datei zu lokalisieren.

quellendatei (LONG VARCHAR)
Der Name der Quellenserverdatei. Der Quellendateiname kann ein vollständig qualifizierter Name oder ein nicht vollständig qualifizierter Name sein. Die Angabe eines Nullwerts oder einer leeren Zeichenfolge ist nicht gültig. Wenn der Name nicht als vollständig qualifizierter Name angegeben wird, werden die Umgebungsvariablen DB2AUDIOPATH und DB2MMPATH auf dem Server verwendet, um die Datei zu lokalisieren.

speichertyp (INTEGER)
Ein Wert, der angibt, wo der Ton gespeichert wird. Die Konstante MMDB_STORAGE_TYPE_INTERNAL (Wert=1) gibt an, dass der Ton in der Datenbank als BLOB gespeichert wird. Die Konstante MMDB_STORAGE_TYPE_EXTERNAL (Wert=0) gibt an, dass der Toninhalt in einer Serverdatei (auf die von der Datenbank aus gezeigt wird) gespeichert wird.

kommentar (LONG VARCHAR)

Zusammen mit dem Ton kann ein Kommentar gespeichert werden.

attribute (LONG VARCHAR FOR BIT DATA)
Die Attribute des Tons.

Rückgabewerte (Datentyp)

Kennung des Tons (DB2AUDIO).

Beispiele

Einfügen eines Datensatzes in die Tabelle 'employee', der einen Audioclip für Anita Jones enthält. Die Tonquelle befindet sich auf einem Clientpuffer. Speichern des Audioclips in der Tabelle als BLOB:

```
EXEC SQL BEGIN DECLARE SECTION;
     SQL TYPE IS BLOB (5M) aud_seg;
  EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2AUDIO(
         CURRENT SERVER,
         :aud_seg,
```

```
         'WAVE',
         CAST(NULL as LONG VARCHAR),

         'Anita''s voice'));
```

Einfügen eines Datensatzes in die Tabelle 'employee', der einen Audioclip für Robert Smith enthält. Die Tonquelle befindet sich in einer Serverdatei. Der Datensatz in der Tabelle 'employee' zeigt auf die Datei.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
  EXEC SQL END DECLARE SECTION;

hvStorageType = MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '384779',
      'Robert Smith',
        DB2AUDIO(
         CURRENT SERVER,
         '/employee/sounds/rsmith.wav',
         'WAV',
         :hvStorageType,
         'Robert''s voice'));
```

Einfügen eines Datensatzes in die Tabelle 'employee', der einen Audioclip für Anita Jones enthält. Speichern des Audioclips als BLOB. Der Quellenaudioclip, der sich in einer Serverdatei befindet, hat ein benutzerdefiniertes Format, eine Abtastrate von 44,1 KHz, und zwei aufgezeichnete Kanäle.

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
struct {
       short len;
       char data[600];
}hvAudattr;
  EXEC SQL END DECLARE SECTION;

MMDBAudioAttrs          *paudiattr;

hvStorageType = MMDB_STORAGE_TYPE_INTERNAL;

paudioattr=(MMDBAudioAttrs *) hvAudattr.data;
strcpy(paudioAttr→cFormat,"cFormatA");
paudioAttr→ulSamplingRate=44100;
paudioAttr→usNumChannels=2;
hvAudattrs.len=sizeof(MMDBAudioAttrs);

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2AUDIO(
         CURRENT SERVER,
          '/employee/sounds/ajones.aud',
         :hvStorageType,
               'Anita"s voice',
               :hvAudattr)
       );
```

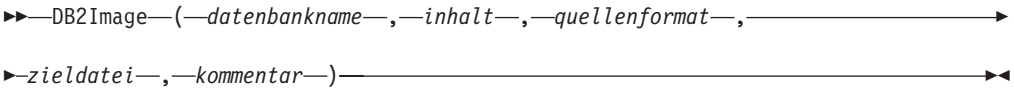
DB2Image

Image	Audio	Video
X		

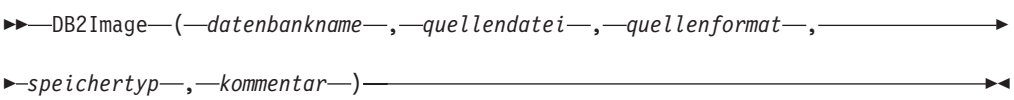
Speichert den Inhalt eines Abbilds in einer Datenbanktabelle. Die Abbildquelle kann ein Clientpuffer, eine Clientdatei oder eine Serverdatei sein.Das Abbild kann in der Datenbanktabelle als BLOB oder in einer Serverdatei (auf die durch die Datenbanktabelle verwiesen wird) gespeichert werden. Die Abbildquelle kann ein unterstütztes Format haben, wobei der DB2Image Extender deren Speicherattribute identifiziert; es kann aber auch ein nicht unterstütztes Format haben, wobei die Attribute in der UDF angegeben werden müssen.

Kopfdatei
dmbimage.h

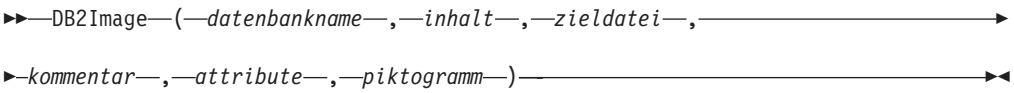
Syntax
Speichern des Inhalts eines Puffers oder einer Clientdatei



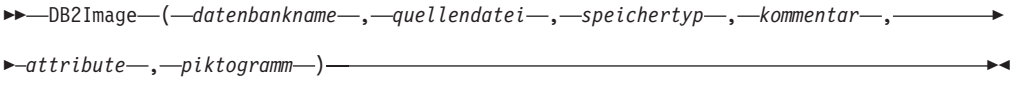
Syntax
Speichern des Inhalts einer Serverdatei



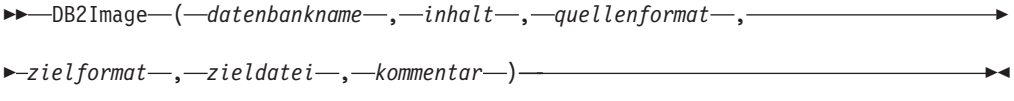
Syntax
Speichern des Inhalts eines Puffers oder einer Clientdatei mit benutzerdefinierten Attributen



Syntax
Speichern des Inhalts einer Serverdatei mit benutzerdefinierten Attributen

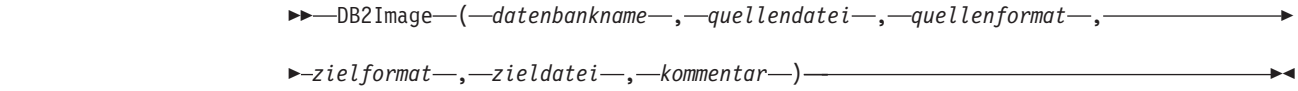


Syntax
Speichern des Inhalts eines Puffers oder einer Clientdatei mit Formatumsetzung



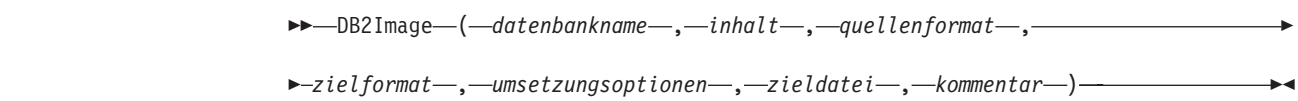
Syntax

Speichern des Inhalts einer Serverdatei mit Formatumsetzung



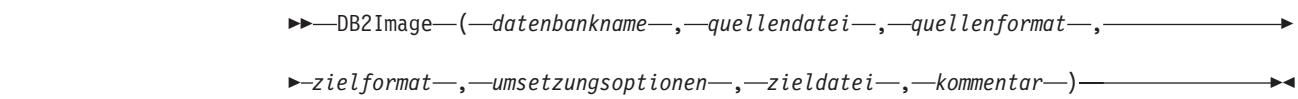
Syntax

Speichern des Inhalts eines Puffers oder einer Clientdatei mit Formatumsetzung und zusätzlichen Änderungen



Syntax

Speichern des Inhalts einer Serverdatei mit Formatumsetzung und zusätzlichen Änderungen



Parameter (Datentyp)

datenbankname (VARCHAR(18))
Der Name der Datenbank, zu der momentan eine Verbindung besteht, wie durch das Sonderregister CURRENT SERVER angegeben.

inhalt (BLOB(2G) AS LOCATOR)
Die Hostvariable, die den Inhalt des Abbilds enthält. Die Hostvariable kann vom Typ BLOB, BLOB_FILE oder BLOB_LOCATOR sein. DB2 stuft den Datentyp des Inhalts in BLOB_LOCATOR um und übergibt den LOB-Zeiger an die UDF DB2Image.

quellenformat (VARCHAR(8))
Das Format des Ursprungsabbilds. Ein Nullwert, eine leere Zeichenfolge oder die Zeichenfolge ASIS kann angegeben werden. Bei jeder dieser Angaben versucht der Image Extender, das Quellenformat automatisch zu bestimmen. Das Abbild wird in dem gleichen Format gespeichert wie die Quelle. Unterstützte Abbildformate befinden sich in Tabelle 8 auf Seite 111.

zielformat (VARCHAR(8))
Das Format des Abbilds nach dem Speichern. Das Format des Quellenabbilds wird entsprechend umgesetzt.

zieldatei (LONG VARCHAR)
Der Name der Zielserverdatei (zum Speichern in eine Serverdatei) oder ein Nullwert bzw. eine leere Zeichenfolge (zum Speichern in eine Datenbanktabelle als BLOB). Der Zieldateiname kann ein vollständig qualifizierter Name sein. Wenn der Name nicht als vollständig qualifizierter Name angegeben wird, werden die Umgebungsvariablen DB2IMAGESTORE und DB2MMSTORE auf dem Server verwendet, um die Datei zu lokalisieren. Wenn das Abbild mit Formatumsetzung gespeichert wird, muss der Pfad für die Zieldatei in den Umgebungsvariablen DB2IMAGEPATH und DB2MMPATH angegeben werden.

quellendatei (LONG VARCHAR)
Der Name der Quellenserverdatei. Der Quellendateiname kann ein vollständig qualifizierter Name oder ein nicht vollständig qualifizierter Name sein. Die Angabe eines Nullwerts oder einer leeren Zeichenfolge ist nicht gültig. Wenn der Name nicht als vollständig qualifizierter Name angegeben wird, werden die Umgebungsvariablen DB2IMAGEPATH und DB2MMPATH auf dem Server verwendet, um die Datei zu lokalisieren.

speichertyp (INTEGER)
Ein Wert, der angibt, wo das Abbild gespeichert wird. Die Konstante MMDB_STORAGE_TYPE_INTERNAL (Wert=1) gibt an, dass das Abbild in der Datenbank als BLOB gespeichert wird. Die Konstante MMDB_STORAGE_TYPE_EXTERNAL (Wert=0) gibt an, dass der Abbildinhalt in einer Serverdatei (auf die von der Datenbank aus gezeigt wird) gespeichert wird.

kommentar (LONG VARCHAR)
Zusammen mit dem Abbild kann ein Kommentar gespeichert werden.

attribute (LONG VARCHAR FOR BIT DATA)
Die Attribute des Abbilds.

piktogramm (LONG VARCHAR FOR BIT DATA)
Ein Piktogramm des Abbilds.

umsetzungsoptionen (VARCHAR(100))
Ein Wert, der Änderungen, wie z. B. Drehung und Komprimierung, angibt, die beim Speichern des Abbilds angewendet werden sollen. Unterstützte Umsetzungsoptionen befinden sich in Tabelle 9 auf Seite 112.

Rückgabewerte (Datentyp)
Kennung des Abbilds (DB2IMAGE).

Beispiele
Einfügen eines Datensatzes in die Tabelle 'employee', der einen Videoclip für Anita Jones enthält. Die Quelle des Abbilds befindet sich in einem Clientpuffer. Speichern des Abbilds in der Tabelle als BLOB:

```
EXEC SQL BEGIN DECLARE SECTION
     SQL TYPE IS BLOB (2M) hvImg
  EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2IMAGE(
         CURRENT SERVER,
         :hvImg,
                  'ASIS',
         CAST(NULL as LONG VARCHAR),

         'Anita''s picture'));
```

Einfügen eines Datensatzes in die Tabelle 'employee', der einen Videoclip für Robert Smith enthält. Die Abbildquelle befindet sich in einer Serverdatei. Der Datensatz in der Tabelle 'employee' zeigt auf die Datei. Umsetzen des Formats des Abbilds von BMP in GIF bei der Speicherung. Erstellen eines Ausschnitts des Abbilds mit einer Breite von 110 Pixel und einer Höhe von 150 Pixel und Komprimieren des Abbilds durch Verwendung der Komprimierung vom Typ LZW:

```
EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '384779',
      'Robert Smith',
        DB2IMAGE(
```

```
        CURRENT SERVER,
        '/employee/pictures/rsmith.bmp',
         'BMP',
         'GIF',
        '-x 110 -y 150 -c 14',
        '',
        'Robert"s picture'));
```

Einfügen eines Datensatzes in die Tabelle 'employee', der einen Videoclip für Robert Smith enthält. Das Quellenabbild, das sich in einer Serverdatei befindet, hat ein benutzerdefiniertes Format, eine Höhe von 640 Pixel und eine Breite von 480 Pixel. Speichern des Abbilds als BLOB:

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
struct {
       short len;
       char data[400];
}hvImgattrs;
  EXEC SQL END DECLARE SECTION;

DB2IMAGEATTRS    *pimgattr;

hvStorageType = MMDB_STORAGE_TYPE_INTERNAL;

pimgattr = (DB2IMAGEATTRS *) hvImgattrs.data;
strcpy(pimgattrt→cFormat,"FormatI");
pimgattr→width=640;
pimgattr→height=480;
hvImgattrs.len=sizeof(DB2IMAGEATTRS);

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2IMAGE(
         CURRENT SERVER,
          '/employee/images/ajones.bmp',
         :hvStorageType,
          'Anita''s picture',
          :hvImgattrs,
          CAST(NULL as LONG VARCHAR))
       );
```

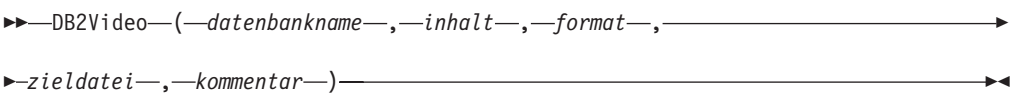
DB2Video

Image	Audio	Video
		X

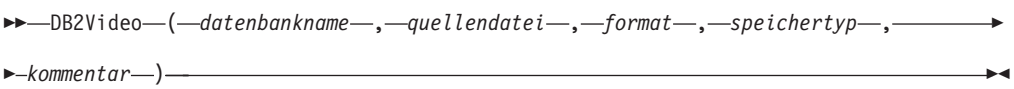
Speichert den Inhalt eines Videos in einer Datenbanktabelle. Die Videoquelle kann ein Clientpuffer, eine Clientdatei oder eine Serverdatei sein.Das Video kann in der Datenbanktabelle als BLOB oder in einer Serverdatei (auf die durch die Datenbanktabelle verwiesen wird) gespeichert werden. Die Videoquelle kann ein unterstütztes Format haben, wobei der DB2Video Extender deren Speicherattribute identifiziert; es kann aber auch ein nicht unterstütztes Format haben, wobei die Attribute in der UDF angegeben werden müssen.

Kopfdatei
dmbvideo.h

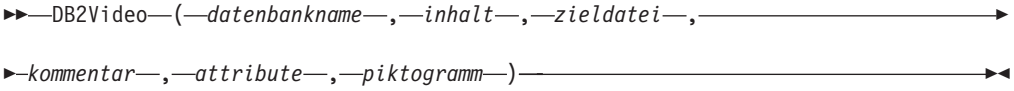
Syntax
Speichern des Inhalts eines Puffers oder einer Clientdatei



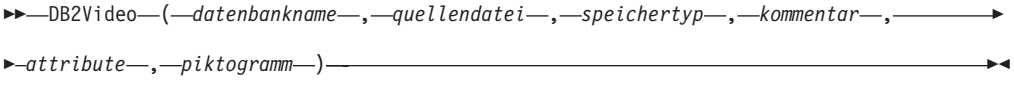
Syntax
Speichern des Inhalts einer Serverdatei



Syntax
Speichern des Inhalts eines Puffers oder einer Clientdatei mit benutzerdefinierten Attributen



Syntax
Speichern des Inhalts einer Serverdatei mit benutzerdefinierten Attributen



Parameter (Datentyp)

datenbankname (VARCHAR(18))
Der Name der Datenbank, zu der momentan eine Verbindung besteht, wie durch das Sonderregister CURRENT SERVER angegeben.

inhalt (BLOB(2G) AS LOCATOR)
Die Hostvariable, die den Inhalt des Videos enthält. Die Hostvariable kann vom Datentyp BLOB, BLOB_FILE oder BLOB_LOCATOR sein. DB2 stuft den Datentyp des Inhalts auf BLOB_LOCATOR um und übergibt den LOB-

Zeiger an die UDF DB2Video. Befindet sich der Inhalt auf einem Clientpuffer, muss der Puffer mindestens die ersten 640 KB des Inhalts enthalten, um sicherzustellen, dass die gesamte Videokopfzeile gelesen wird.

format (VARCHAR(8))
Das Format des Quellenvideos. Wenn ein Nullwert oder eine leere Zeichenfolge angegeben wird, versucht der Video Extender, das Quellenformat automatisch zu bestimmen.Das Video wird in dem gleichen Format gespeichert wie die Quelle. Unterstützte Videoformate befinden sich in Tabelle 8 auf Seite 111. Für das MPG1-Format können Sie MPG1, mpg1, MPEG1 bzw. mpeg1 angeben. Für das MPG2-Format können Sie MPG2, mpg2, MPEG2 bzw. mpeg2 angeben.

zieldatei (LONG VARCHAR)
Der Name der Zielserverdatei (zum Speichern in eine Serverdatei) oder ein Nullwert bzw. eine leere Zeichenfolge (zum Speichern in eine Datenbanktabelle als BLOB). Die Serverdatei muss mit einem vollständig qualifizierten Namen angegeben werden. Wenn der Dateiname nicht als vollständig qualifizierter Name angegeben wird, werden die Umgebungsvariablen DB2VIDEOSTORE und DB2MMSTORE auf dem Server verwendet, um die Datei zu lokalisieren.

quellendatei (LONG VARCHAR)
Der Name der Quellenserverdatei. Der Name kann ein vollständig qualifizierter Name oder ein nicht vollständig qualifizierter Name sein. Die Angabe eines Nullwerts oder einer leeren Zeichenfolge ist nicht gültig. Wenn der Name nicht als vollständig qualifizierter Name angegeben wird, werden die Umgebungsvariablen DB2VIDEOPATH und DB2MMPATH auf dem Server verwendet, um die Datei zu lokalisieren.

speichertyp (INTEGER)
Ein Wert, der angibt, wo das Video gespeichert wird. Die Konstante MMDB_STORAGE_TYPE_INTERNAL (Wert=1) gibt an, dass das Video in der Datenbank als BLOB gespeichert wird. Die Konstante MMDB_STORAGE_TYPE_EXTERNAL (Wert=0) gibt an, dass der Videoinhalt in einer Serverdatei (auf die von der Datenbank aus gezeigt wird) gespeichert wird.

kommentar (LONG VARCHAR)
Zusammen mit dem Video kann ein Kommentar gespeichert werden.

attribute (LONG VARCHAR FOR BIT DATA)
Die Attribute des Videos.

piktogramm (LONG VARCHAR FOR BIT DATA)
Ein Piktogrammabbild, das das Video darstellt.

Rückgabewerte (Datentyp)

Kennung des Videos (DB2VIDEO).

Beispiele

Einfügen eines Datensatzes in die Tabelle 'employee', der einen Videoclip für Anita Jones enthält. Die Quelle des Videos befindet sich in einem Clientpuffer. Speichern des Videoclips in der Tabelle als BLOB:

```
EXEC SQL BEGIN DECLARE SECTION
     SQL TYPE IS BLOB (8M) vid;
  EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2VIDEO(
```

```
         CURRENT SERVER,
         :vid,
                'MPEG1',
CAST(NULL as LONG VARCHAR),         'Anita''s video'));
```

Einfügen eines Datensatzes in die Tabelle 'employee', der einen Videoclip für Robert Smith enthält. Die Videoquelle befindet sich in einer Serverdatei. Der Datensatz in der Tabelle 'employee' zeigt auf die Datei:

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
  EXEC SQL END DECLARE SECTION;

hvStorageType = MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '384779',
      'Robert Smith',
        DB2VIDEO(
         CURRENT SERVER,
         '/employee/videos/rsmith.mpg',
                'MPEG1',
         :hvStorageType,
         'Robert''s video'));
```

Einfügen eines Datensatzes in eine Datenbanktabelle, der einen Videoclip enthält. Der Quellenvideoclip, der sich in einer Serverdatei befindet, hat ein benutzerdefiniertes Format. Behalten Sie den Videoinhalt in der Serverdatei bei (der Datensatz in der der Datenbanktabelle zeigt auf die Datei). Speichern eines Piktogramms, das das Video darstellt:

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
struct {
       short len;
       char data[400];
}hvVidattrs;
struct {
       short len;
       char data[10000];
}hvThumbnail;
  EXEC SQL END DECLARE SECTION;

MMDBVideoAttrs         *pvideoAttr;

hvStorageType = MMDB_STORAGE_TYPE_EXTERNAL;

pvideoAttr=(MMDBVideoAttrs *)hvVidattrs.data;
strcpy(pvideoAttr→cFormat,"Formatv");
pvideoAttr.len=sizeof(MMDBVideoAttrs);
.
.
.
/* Generate thumbnail and assign data   */
/* in video structure                   */
.
.
.
EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
        DB2VIDEO(
         CURRENT SERVER,
          '/employee/videos/ajones.vid',
         :hvStorageType,
          'Anita''s video',
          :hvVidattrs,
                :hvThumbnail)
       );
```

Duration

Image	Audio	Video
	X	X

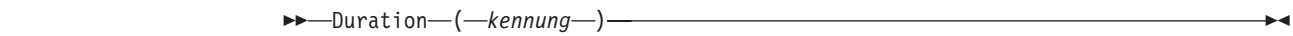
Gibt die Dauer (d. h. die Spieldauer in Sekunden) eines WAVE- oder AIFF-Tons oder eines Videos zurück.

Kopfdatei

Audio dmbaudio.h

Video dmbvideo.h

Syntax



Parameter (Datentyp)

kennung (DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Dauer (in Sekunden) eines Videos oder die Dauer (in Sekunden) eines Tons im WAVE-, AIFF- oder benutzerdefinierten Format (INTEGER). Ein Nullwert wird für Töne in anderen Formaten zurückgegeben.

Beispiele

Anzeigen der Dauer aller Videos, die in der Spalte 'video' der Tabelle 'employee' gespeichert sind:

```
EXEC SQL BEGIN DECLARE SECTION;
long hvDur_vid;
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT DURATION(VIDEO)
     INTO :hvDur_vid
     FROM EMPLOYEE;
```

Filename

Image	Audio	Video
X	X	X

Gibt den Namen der Serverdatei zurück, die den Inhalt eines Abbilds, Tons oder Videos enthält, wenn der Objektinhalt in einer Datei gespeichert wird (auf die von einer Datenbanktabelle aus gezeigt wird). Wenn das Abbild, der Ton oder das Video in einer Datenbanktabelle als BLOB gespeichert ist, wird ein Nullwert zurückgegeben.

Kopfdatei

Image dmbimage.h

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2IMAGE, DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds, Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Dateiname der Serverdatei, wenn sich der Objektinhalt in einer Serverdatei befindet (VARCHAR(250)); Nullwert, wenn das Objekt als BLOB gespeichert ist.

Beispiele

Anzeigen des Dateinamens des Videos für den Eintrag 'Robert Smith' in der Tabelle 'employee':

```
EXEC SQL BEGIN DECLARE SECTION;
char hvVid_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (VIDEO)
     INTO :hvVid_fname
  FROM EMPLOYEE
     WHERE NAME='Robert Smith';
```

FindInstrument

Image	Audio	Video
	X	

Gibt die Spurnummer des ersten Auftretens eines bestimmten Instruments in einem MIDI-Ton zurück.

Kopfdatei
dmbaudio.h

Syntax

Parameter (Datentyp)

kennung (DB2AUDIO)
Name der Spalte oder Hostvariable, die die Kennung des Tons enthält.

instrument (VARCHAR(255))
Name des Instruments, nach dem gesucht werden soll. Der Audio Extender sucht nach einem Instrument, dessen Name exakt mit dem angegebenen Namen übereinstimmt.

Rückgabewerte (Datentyp)
Nummer der Spur, die das erste Auftreten eines angegebenen Instrumentennamens enthält (SMALLINT). Der Wert -1 wird zurückgegeben, wenn kein Instrument mit dem angegebenen Namen gefunden wird. NULL wird für Töne in anderen Formaten zurückgegeben.

Beispiele
Suchen des ersten Auftretens von PIANO in der MIDI-Tonaufzeichnung für Robert Smith, die in der Spalte 'sound' der Tabelle 'employee' gespeichert ist:

```
EXEC SQL BEGIN DECLARE SECTION;
      short hvInstr;
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FINDINSTRUMENT(SOUND, 'PIANO')
      INTO :hvInstr
  FROM EMPLOYEE
      WHERE NAME = 'Robert Smith';
```

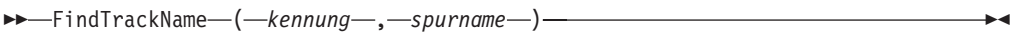
FindTrackName

Image	Audio	Video
	X	

Gibt die Nummer einer angegebenen benannten Spur in einem MIDI-Ton zurück.

Kopfdatei
dmbaudio.h

Syntax



Parameter (Datentyp)

kennung (DB2AUDIO)
Name der Spalte oder Hostvariable, die die Kennung des Tons enthält.

spurname (VARCHAR(255))
Name der Spur, nach der gesucht werden soll. Der Audio Extender sucht nach einer Spur, deren Name exakt mit dem angegebenen Namen übereinstimmt.

Rückgabewerte (Datentyp)
Nummer der benannten Spur des angegebenen Instruments (SMALLINT). Der Wert -1 wird zurückgegeben, wenn keine Spur mit dem angegebenen Namen gefunden wird. Ein Nullwert wird für Töne in anderen Formaten zurückgegeben.

Beispiele
Feststellen, ob eine Spur mit dem Namen WELCOME in der MIDI-Tonaufzeichnung für Robert Smith existiert. Die Tonaufzeichnung ist in der Spalte 'sound' der Tabelle 'employee' gespeichert:

```
EXEC SQL BEGIN DECLARE SECTION;
      short hvTrack;
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FINDTRACKNAME(SOUND,
          'WELCOME')
      INTO :hvTrack
  FROM EMPLOYEE
      WHERE NAME = 'Robert Smith';
```

Format

Image	Audio	Video
X	X	X

Gibt das Format eines Abbilds, Tons oder Videos zurück.

Kopfdatei

Image dmbimage.h

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2IMAGE, DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds, Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Format des Abbilds, Tons oder Videos (VARCHAR(8). Informationen zu den unterstützten Abbild-, Audio- und Videoformaten befinden sich in Tabelle 8 auf Seite 111.

Beispiele

Abrufen der Namen aller Mitarbeiter, deren Abbilder in der Spalte 'picture' der Tabelle 'employee' im GIF-Format gespeichert sind:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvName[30];
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT NAME
     INTO :hvName
  FROM EMPLOYEE
     WHERE FORMAT(PICTURE)='GIF';
```

FrameRate

Image	Audio	Video
		X

Gibt den Durchsatz eines Videos in Vollbildern pro Sekunde zurück.

Kopfdatei
dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Videos enthält.

Rückgabewerte (Datentyp)
Vollbildrate des Videos (SMALLINT). Ein Nullwert wird zurückgegeben, wenn die Durchsatzrate variabel ist.

Beispiele
Abrufen der Vollbildrate des Videos, das in der Spalte 'video' der Tabelle 'employee' für Anita Jones gespeichert ist:

```
EXEC SQL BEGIN DECLARE SECTION;
short hvFm_rate;
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FRAMERATE (VIDEO)
  FROM EMPLOYEE
      INTO :hvFm_rate
       WHERE NAME='Anita Jones';
```

GetInstruments

Image	Audio	Video
	X	

Gibt die Instrumentennamen aller Instrumente in einem MIDI-Ton zurück.

Kopfdatei
dmbaudio.h

Syntax

Parameter (Datentyp)

kennung (DB2AUDIO)
Name der Spalte oder Hostvariable, die die Kennung des Tons enthält.

Rückgabewerte (Datentyp)
Instrumentennamen aller Instrumente im MIDI-Ton (VARCHAR(1536)). Die Werte werden in der Reihenfolge der Spurnummer zurückgegeben (z. B. PIANO; TRUMPET; BASS). Das Ergebnis wird in *n* Felder aufgeteilt, wobei *n* die Anzahl von Spuren im MIDI-Ton ist. Wenn eine Spur kein zugeordnetes Instrument hat, ist das entsprechende Feld leer. Für andere Tonformate als MIDI wird ein Nullwert zurückgegeben.

Beispiele
Suchen aller Instrumente (d. h. Spurnummern und Instrumentennamen) in der MIDI-Tonaufzeichnung für Robert Smith. Die Tonaufzeichnung ist in der Spalte 'sound' der Tabelle 'employee' gespeichert:

```
EXEC SQL BEGIN DECLARE SECTION;
 char hvAud_Instr[1536];
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT GETINSTRUMENTS(SOUND)
      INTO :hvAud_Instr
  FROM EMPLOYEE
      WHERE NAME = 'Robert Smith';
```

GetTrackNames

Image	Audio	Video
	X	

Gibt die Namen aller Spuren in einem MIDI-Ton zurück.

Kopfdatei
dmbaudio.h

Syntax

Parameter (Datentyp)

kennung (DB2AUDIO)
Name der Spalte oder Hostvariable, die die Kennung des Tons enthält.

Rückgabewerte (Datentyp)
Name aller Spuren im MIDI-Ton (VARCHAR(1536)). Die Werte werden in der Reihenfolge der Spurnummer zurückgegeben (z. B. PIANO TUNE; TRUMPET FANFARE). Das Ergebnis wird in *n* Felder aufgeteilt, wobei *n* die Anzahl von Spuren im MIDI-Ton ist. Wenn eine Spur keinen Namen hat, ist das entsprechende Feld leer. Für andere Tonformate als MIDI wird ein Nullwert zurückgegeben.

Beispiele
Abrufen aller Spurnummern und Spurnamen in der MIDI-Tonaufzeichnung für Robert Smith, die in der Spalte 'sound' der Tabelle 'employee' gespeichert ist:

```
EXEC SQL BEGIN DECLARE SECTION;
 char hvTracks[1536];
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT GETTRACKNAMES(SOUND)
      INTO :hvTracks
  FROM EMPLOYEE
      WHERE NAME = 'Robert Smith';
```

Height

Image	Audio	Video
X		X

Gibt die Höhe eines Abbilds oder Videovollbilds in Pixeln zurück.

Kopfdatei

Image dmbimage.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2IMAGE or DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds oder Videos enthält.

Rückgabewerte (Datentyp)

Höhe in Pixeln (INTEGER).

Beispiele

Abrufen der Dateinamen aller Abbilder in der Spalte 'picture' der Tabelle 'employee', die kürzer als 500 Pixel sind:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
     INTO :hvImg_fname
  FROM EMPLOYEE
     WHERE HEIGHT(PICTURE)<500;
```

Importer

Image	Audio	Video
X	X	X

Gibt die Benutzer-ID der Person zurück, die ein Abbild, einen Ton oder ein Video in einer Datenbanktabelle gespeichert hat.

Kopfdatei

Image dmbimage.h

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2IMAGE, DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds, Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Benutzer-ID der Person, die das Objekt in der Datenbank gespeichert hat (CHAR(8)).

Beispiele

Abrufen der Namen aller Dateien für Töne, die durch den Benutzer mit der Benutzer-ID rsmith in der Spalte 'sound' der Tabelle 'employee' gespeichert wurden:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvAud_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (SOUND)
     INTO :hvAud_fname
  FROM EMPLOYEE
     WHERE IMPORTER(SOUND)='rsmith';
```

ImportTime

Image	Audio	Video
X	X	X

Gibt eine Zeitmarke zurück, die angibt, wann ein Abbild, Ton oder Video in einer Datenbanktabelle gespeichert wurde.

Kopfdatei

Image dmbimage.h

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2IMAGE, DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds, Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Zeitmarke, wann das Abbild, der Ton oder das Video gespeichert wurde (TIMESTAMP).

Beispiele

Abrufen der Namen aller Dateien für Abbilder, die vor mehr als einem Jahr in der Spalte 'picture' der Tabelle 'employee' gespeichert wurden:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
     INTO :hvImg_fname
  FROM EMPLOYEE
     WHERE(CURRENT TIMESTAMP -
        IMPORTTIME(PICTURE))>365;
```

MaxBytesPerSec

Image	Audio	Video
		X

Gibt den maximalen Durchsatz eines Videos in Byte pro Sekunde zurück.

Kopfdatei
dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Videos enthält.

Rückgabewerte (Datentyp)
Durchsatz des Videos (INTEGER). Ein Nullwert wird zurückgegeben, wenn die Durchsatzrate variabel ist.

Beispiele
Abrufen des maximalen Durchsatzes eines Videos, das in der Spalte 'video' der Tabelle 'employee' für Anita Jones gespeichert ist:

```
EXEC SQL BEGIN DECLARE SECTION;
long hvMax_BytesPS;
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT MAXBYTESPERSEC(VIDEO)
     INTO :hvMax_BytesPS
  FROM EMPLOYEE
       WHERE NAME='Anita Jones';
```

NumAudioTracks

Image	Audio	Video
	X	X

Gibt die Anzahl an Tonspuren in einem Video oder MIDI-Ton zurück.

Kopfdatei

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Anzahl an Tonspuren im Video oder MIDI-Ton (SMALLINT). Ein Nullwert wird für Töne in anderen Formaten zurückgegeben.

Beispiele

Abrufen der Namen aller Videodateien aus der Spalte 'video' der Tabelle 'employee', die keine Tonspuren enthalten:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvVid_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (VIDEO)
     INTO :hvVid_fname
  FROM EMPLOYEE
     WHERE NUMAUDIOTRACKS(VIDEO) = 0;
```

NumChannels

Image	Audio	Video
	X	X

Gibt die Anzahl an aufgezeichneten Tonkanälen in einem WAVE- oder AIFF-Ton oder einem Video zurück.

Kopfdatei

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Anzahl an aufgezeichneten Tonkanälen in einem Video oder einem WAVE- oder AIFF-Ton (SMALLINT). Ein Nullwert wird für Töne in anderen Formaten zurückgegeben.

Beispiele

Abrufen der Namen aller Audiodateien aus der Spalte Spalte 'sound' der Tabelle 'employee', die in Stereo (d. h. mit 2 Kanälen) aufgezeichnet wurden:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvAud_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (SOUND)
     INTO :hvAud_fname
  FROM EMPLOYEE
     WHERE NUMCHANNELS(SOUND) = 2;
```

NumColors

Image	Audio	Video
X		

Gibt die Anzahl an Farben in einem Abbild zurück.

Kopfdatei
dmbimage.h

Syntax

Parameter (Datentyp)

kennung (DB2IMAGE)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds enthält.

Rückgabewerte (Datentyp)
Anzahl von Farben in Abbildern (INTEGER).

Beispiele
Abrufen der Namen von Abbilddateien aus der Spalte 'picture' der Tabelle 'employee' für Abbilder, die weniger als 16 Farben haben:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
     INTO :hvImg_fname
  FROM EMPLOYEE
     WHERE NUMCOLORS(PICTURE) < 16;
```

NumFrames

Image	Audio	Video
		X

Gibt die Anzahl an Vollbildern in einem Video zurück.

Kopfdatei
dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Videos enthält.

Rückgabewerte (Datentyp)
Anzahl von Vollbildern in Videos (INTEGER). Ein Nullwert wird zurückgegeben, wenn die Durchsatzrate variabel ist.

Beispiele
Abrufen der Anzahl von Vollbildern in dem Video, das in der Spalte 'video' der Tabelle 'employee' für Robert Smith gespeichert ist:

```
EXEC SQL BEGIN DECLARE SECTION;
long hvNum_Frames;
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT NUMFRAMES (VIDEO)
     INTO :hvNum_Frames
  FROM EMPLOYEE
     WHERE NAME='Robert Smith';
```

NumVideoTracks

Image	Audio	Video
		X

Gibt die Anzahl an Videospuren in einem Video zurück.

Kopfdatei
dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Videos enthält.

Rückgabewerte (Datentyp)
Anzahl von Videospuren (SMALLINT).

Beispiele
Abrufen der Dateinamen aller Videos aus der Spalte 'video' der Tabelle 'employee', die mehr als eine Videospur haben:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvVid_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (VIDEO)
     INTO :hvVid_fname
  FROM EMPLOYEE
     WHERE NUMVIDEOTRACKS(VIDEO) > 1;
```

QbScoreFromName

Image	Audio	Video
X		

Gibt das Ähnlichkeitsergebnis eines Abbilds zurück, wobei es sich um eine Zahl handelt, die angibt, wie weit die Merkmale eines Abbilds mit denen eines Abfrageobjekts übereinstimmen. Der QBIC-Katalog, der der Spalte zugeordnet ist, zu der die Abbildkennung gehört, wird verwendet, um das Ähnlichkeitsergebnis des Abbilds zu berechnen. Je niedriger das Ähnlichkeitsergebnis, desto mehr stimmen die Merkmale des Abbilds mit denen des angegebenen Abfrageobjekts überein. (QbScoreFromName ersetzt QbScore; QbScore wird jedoch noch akzeptiert.)

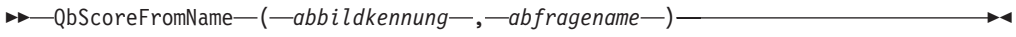
Anmerkungen:

1. **Nur EEE:** QbScoreFromName wird nicht in einer Umgebung für partitionierte Datenbanken unterstützt. Verwenden Sie stattdessen die UDF QbScoreFromStr, nachdem Sie die API QbQueryGetString verwendet haben, um die Abfragezeichenfolge abzurufen.
2. QbScoreFromName wird in zukünftigen Releases für Umgebungen für nicht partitionierte Datenbanken nicht mehr unterstützt. Um eine Abfrage erneut zu verwenden, sollten Sie die API QbQueryGetString verwenden, um die Abfragezeichenfolge abzurufen, und diese Zeichenfolge für die spätere Verwendung in Ihrer Anwendung sichern.

Kopfdatei

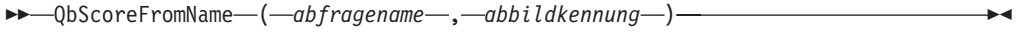
Keine

Syntax



Syntax

Veraltete Version



Parameter (Datentyp)

abbildkennung (DB2Image)
Die Kennung für das Abbild.

abfragename (varchar(18))
Der Name des Abfrageobjekts.

Rückgabewerte (Datentyp)

Das Ähnlichkeitsergebnis des Abbilds (DOUBLE). Das Ähnlichkeitsergebnis kann im Bereich von 0,0 bis gegen unendlich liegen. Je niedriger das Ähnlichkeitsergebnis, desto mehr stimmen die Merkmalwerte des Zielabbilds mit den Merkmalwerten überein, die in der Abfrage angegeben sind. Ein Ähnlichkeitsergebnis von 0,0 gibt eine exakte Übereinstimmung an. Ein Ähnlichkeitsergebnis von Null bedeutet, dass das Abbild nicht katalogisiert wurde; die veraltete Version dieser UDF gibt das Ähnlichkeitsergebnis -1 zurück, wenn das Abbild nicht katalogisiert wurde.

Beispiele

Suchen der katalogisierten Abbilder in einer Tabellenspalte, deren Durchschnittsfarbe sehr nah an Rot liegt:

```
EXEC SQL BEGIN DECLARE SECTION;
char Img_fnd[100];
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT NAME
  INTO :Img_fnd
 FROM FABRIC
  WHERE (QBSCOREFROMNAME(SWATCH_IMG,
         'fshavgcol'))<0.1;
```

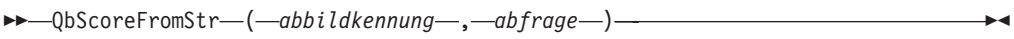
QbScoreFromStr

Image	Audio	Video
X		

Gibt das Ähnlichkeitsergebnis eines Abbilds zurück, wobei es sich um eine Zahl handelt, die angibt, wie weit die Merkmale eines Abbilds mit denen einer Abfragezeichenfolge übereinstimmen. Der QBIC-Katalog, der der Spalte zugeordnet ist, zu der die Abbildkennung gehört, wird verwendet, um das Ähnlichkeitsergebnis des Abbilds zu berechnen. Je niedriger das Ähnlichkeitsergebnis, desto mehr stimmen die Merkmale des Abbilds mit denen der Abfragezeichenfolge überein.

Kopfdatei
Keine

Syntax



Syntax
Veraltete Version

Parameter (Datentyp)

abbildkennung (DB2Image)
Die Kennung für das Abbild.

abfrage (VARCHAR(1024))
Die Abfragezeichenfolge.

Rückgabewerte (Datentyp)
Das Ähnlichkeitsergebnis des Abbilds (DOUBLE). Das Ähnlichkeitsergebnis kann im Bereich von 0,0 bis gegen unendlich liegen. Je niedriger das Ähnlichkeitsergebnis, desto mehr stimmen die Merkmalwerte des Zielabbilds mit den Merkmalwerten überein, die in der Abfrage angegeben sind. Ein Ähnlichkeitsergebnis von 0,0 gibt eine exakte Übereinstimmung an. Ein Ähnlichkeitsergebnis von Null bedeutet, dass das Abbild nicht katalogisiert wurde; die veraltete Version dieser UDF gibt das Ähnlichkeitsergebnis -1 zurück, wenn das Abbild nicht katalogisiert wurde.

Beispiele
Suchen der katalogisierten Abbilder in einer Tabellenspalte, deren Durchschnittsfarbe sehr nah an Rot liegt:

```
SELECT name
FROM fabric
WHERE (QbScoreFromStr(Swatch_Img,
       'QbColorFeatureClass color=<255, 0, 0>'))<0.1
```

QbScoreTBFromName

Image	Audio	Video
X		

Gibt eine Tabelle von Ähnlichkeitsergebnissen für eine Abbildspalte zurück. Jedes Ähnlichkeitsergebnis ist eine Zahl, die angibt, wie weit die Merkmale eines Abbilds mit denen des Abfrageobjekts übereinstimmen. Der QBIC-Katalog, der der angegebenen Tabelle und Spalte zugeordnet ist, zu der die Abbildkennung gehört, wird verwendet, um das Ähnlichkeitsergebnis der einzelnen Abbilder zu berechnen. Je niedriger das Ähnlichkeitsergebnis für die einzelnen Abbilder, desto mehr stimmen die Merkmale des jeweiligen Abbilds mit denen des Abfrageobjekts überein.

Anmerkungen:

1. **Nur EEE:** QbScoreTBFromName wird nicht in einer Umgebung für partitionierte Datenbanken unterstützt. Verwenden Sie stattdessen die UDF QbScoreFromStr, nachdem Sie die API QbQueryGetString verwendet haben, um die Abfragezeichenfolge abzurufen.
2. QbScoreTBFromName wird in Zukunft für Umgebungen für nicht partitionierte Datenbanken nicht mehr unterstützt. Um eine Abfrage erneut zu verwenden, sollten Sie die API QbQueryGetString verwenden, um die Abfragezeichenfolge abzurufen, und diese Zeichenfolge für die spätere Verwendung in Ihrer Anwendung sichern.

Kopfdatei

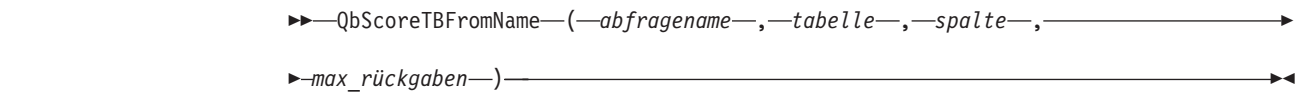
Keine

Syntax

Zurückgeben der Ähnlichkeitsergebnisse für alle katalogisierten Abbilder in einer Spalte

Syntax

Zurückgeben der Ähnlichkeitsergebnisse für eine bestimmte Anzahl von katalogisierten Abbildern in einer Spalte



Parameter (Datentyp)

abfragename (VARCHAR(18))
Der Name des Abfrageobjekts.

tabelle (CHAR(18))
Der qualifizierte Name der Tabelle, die die Abbildspalte enthält. Sie können einen nicht qualifizierten Tabellennamen verwenden, wenn das Tabellenschema mit der Benutzer-ID übereinstimmt, die zum Starten der DB2 Extender-Services verwendet wurde.

spalte (CHAR(18))
Der Name der Abbildspalte.

max_rückgaben (INTEGER)
Die maximale Anzahl an Kennungen, die die Ergebnistabelle zurückgeben soll. Wenn kein Wert angegeben ist, ist die maximale Anzahl an zurückgegebenen Kennungen 100.

Rückgabewerte (Datentyp)

Tabelle mit Abbildkennungen und Ähnlichkeitsergebnissen für die Abbilder in der Spalte. Die Ergebnistabelle hat zwei Spalten: IMAGE_ID (DB2Image) für die Abbildkennungen und SCORE (DOUBLE) für die Ähnlichkeitsergebnisse. Die Ergebnistabelle ist in aufsteigender Reihenfolge nach Ähnlichkeitsergebnis sortiert. Das Ähnlichkeitsergebnis kann im Bereich von 0,0 bis gegen unendlich liegen. Je niedriger das Ähnlichkeitsergebnis, desto mehr stimmen die Merkmalwerte des Zielabbilds mit den Merkmalwerten überein, die in der Abfrage angegeben sind. Ein Ähnlichkeitsergebnis von 0,0 gibt eine exakte Übereinstimmung an. Ein Ähnlichkeitsergebnis von -1 bedeutet, dass das Abbild nicht katalogisiert wurde.

Beispiele

Vergleichen der Textur der Abbilder in einer Tabellenspalte mit der Textur, die im Abfrageobjekt angegeben ist; Rückgabe der Abbildkennungen und deren Ähnlichkeitsergebnisse:

```
SELECT name, description
INTO :hvName, :hvDesc
FROM fabric
WHERE CAST (swatch_img as varchar(250)) IN
 (SELECT CAST (image_id as varchar(250)) FROM TABLE
 (QbScoreTBFromName
     'fstxtr',
     'clothes.fabric',
     'swatch_img'))
 AS T1));
```

QbScoreTBFromStr

Image	Audio	Video
X		

Gibt eine Tabelle von Ähnlichkeitsergebnissen für eine Abbildspalte zurück. Jedes Ähnlichkeitsergebnis ist eine Zahl, die angibt, wie weit die Merkmale eines Abbilds mit denen der Abfragezeichenfolge übereinstimmen. Der QBIC-Katalog, der der Tabelle und Spalte zugeordnet ist, zu der die Abbildkennung gehört, wird verwendet, um das Ähnlichkeitsergebnis der einzelnen Abbilder zu berechnen. Je niedriger das Ähnlichkeitsergebnis für ein Abbild, desto mehr stimmen die Merkmale des Abbilds mit denen der Abfragezeichenfolge überein.

Kopfdatei

Keine

Syntax

Zurückgeben der Ähnlichkeitsergebnisse für alle katalogisierten Abbilder in einer Spalte

Syntax

Zurückgeben der Ähnlichkeitsergebnisse für eine bestimmte Anzahl von katalogisierten Abbildern in einer Spalte

Parameter (Datentyp)

abfrage (VARCHAR(1024))
Die Abfragezeichenfolge.

tabelle (CHAR(18))
Der qualifizierte Name der Tabelle, die die Abbildspalte enthält. Sie können einen nicht qualifizierten Tabellennamen verwenden, wenn das Tabellenschema mit der Benutzer-ID übereinstimmt, die zum Starten der DB2 Extender-Services verwendet wurde.

spalte (CHAR(18))
Die abzufragende Abbildspalte.

max_rückgaben (INTEGER)
Die maximale Anzahl an Kennungen, die die Ergebnistabelle zurückgeben soll. Wenn kein Wert angegeben ist, ist die maximale Anzahl an zurückgegebenen Abbildkennungen 100.

Rückgabewerte (Datentyp)

Tabelle mit Abbildkennungen und Ähnlichkeitsergebnissen für die Abbilder in der Spalte. Die Ergebnistabelle hat zwei Spalten: IMAGE_ID (DB2Image) für die Abbildkennungen und SCORE (DOUBLE) für die Ähnlichkeitsergebnisse. Die Ergebnistabelle ist in aufsteigender Reihenfolge nach Ähnlichkeitsergebnis sortiert. Das Ähnlichkeitsergebnis kann im Bereich von 0,0 bis gegen unendlich liegen. Je niedriger das Ähnlichkeitsergebnis, desto mehr stimmen die Merkmalwerte des Zielabbilds mit den Merkmalwerten überein, die in der Abfrage angegeben sind. Ein Ähnlichkeitsergebnis von 0,0 gibt eine exakte Übereinstimmung an. Ein Ähnlichkeitsergebnis von -1 bedeutet, dass das Abbild nicht katalogisiert wurde.

Beispiele

Suchen der zehn katalogisierten Abbilder in einer Tabellenspalte, deren Textur der eines Abbilds in einer Serverdatei am ähnlichsten ist:

```
SELECT name, description
FROM fabric
WHERE CAST (swatch_img as varchar(250)) IN
 (SELECT CAST (image_id as varchar(250)) FROM TABLE
 (QbScoreTBFromStr
 (QbTextureFeatureClass file=<server,"patterns/ptrn07.gif">'
      'clothes.fabric',
      'swatch_img',
      10))
  AS T1));
```

Replace

Image	Audio	Video
X	X	X

Aktualisiert den Inhalt eines Abbilds, Tons oder Videos, das/der in einer Datenbank gespeichert ist, und aktualisiert dessen Kommentar.

Kopfdatei

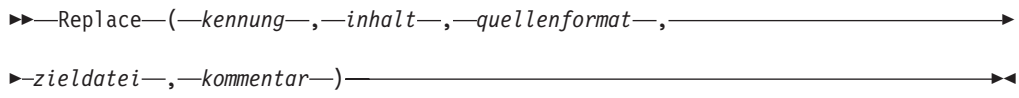
Image dmbimage.h

Audio dmbaudio.h

Video dmbvideo.h

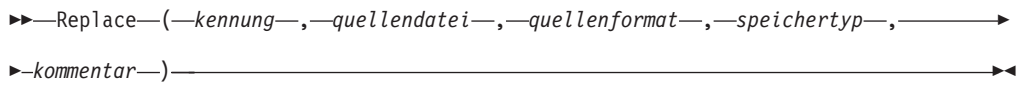
Syntax

Aktualisieren des Inhalts eines Puffers oder einer Clientdatei und Aktualisieren des Kommentars



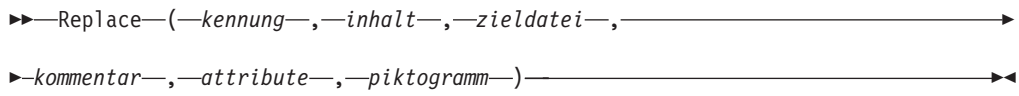
Syntax

Aktualisieren des Inhalts einer Serverdatei und Aktualisieren des Kommentars



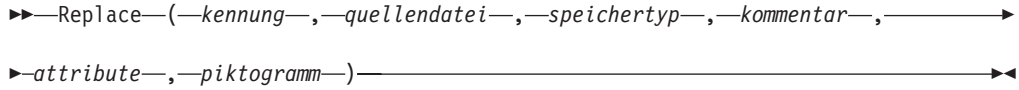
Kopfdatei

Aktualisieren des Inhalts eines Puffers oder einer Clientdatei mit benutzerdefinierten Attributen und Aktualisieren des Kommentars



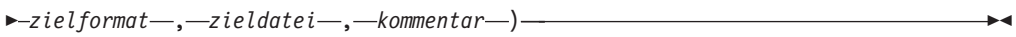
Kopfdatei

Aktualisieren des Inhalts einer Serverdatei mit benutzerdefinierten Attributen und Aktualisieren des Kommentars



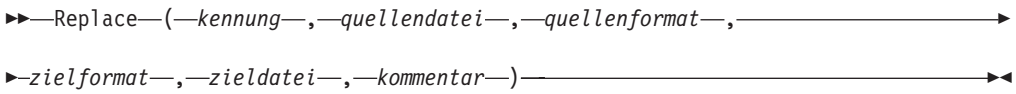
Syntax

Aktualisieren des Inhalts eines Puffers oder einer Clientdatei mit Formatumsetzung und Aktualisieren des Kommentars (nur Abbilder)



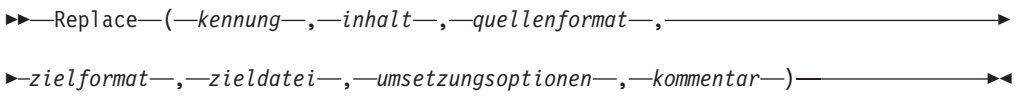
Syntax

Aktualisieren des Inhalts einer Serverdatei mit Formatumsetzung und Aktualisieren des Kommentars (nur Abbilder)



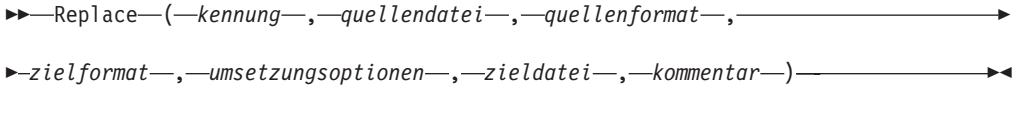
Syntax

Aktualisieren des Inhalts eines Puffers oder einer Clientdatei mit Formatumsetzung und zusätzlichen Änderungen und Aktualisieren des Kommentars (nur Abbilder)



Syntax

Aktualisieren des Inhalts einer Serverdatei mit Formatumsetzung und zusätzlichen Änderungen und Aktualisieren des Kommentars (nur Abbilder)



Parameter (Datentyp)

kennung (DB2IMAGE, DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds, Tons oder Videos enthält.

quellendatei (LONG VARCHAR)
Der Name der Datei, die den Inhalt zum Aktualisieren des Abbilds, Tons oder Videos enthält.

zieldatei (LONG VARCHAR)
Der Name der Datei, die den Inhalt des Abbilds, Tons oder Videos enthält, das/der aktualisiert werden soll.

zielerstellung (INTEGER)
Ein Wert, der angibt, ob eine Zieldatei erstellt werden soll, wenn sich der Quelleninhalt in einer Serverdatei befindet. Der Wert kann 0 oder 1 sein. Der Wert 0 bedeutet, dass die Zieldatei nicht erstellt wird (im Prinzip findet kein Abrufen statt). Der Wert 1 bedeutet, dass die Zieldatei erstellt wird (existiert sie bereits, bedeutet dieser Wert, dass die Datei überschrieben wird). Wenn der Quelleninhalt ein BLOB ist, wird die Zieldatei in jedem Fall erstellt (existiert die Datei bereits, wird sie überschrieben).

zielformat (VARCHAR(8))
Das Format des Abbilds nach dem Abrufen. Das Format des Quellenabbilds wird entsprechend umgesetzt. Wenn der Inhalt mit Formatumsetzung aktualisiert wird, muss der Pfad für die Zieldatei in den Umgebungsvariablen DB2IMAGEPATH und DB2MMPATH angegeben werden. Für das MPG1-Format können Sie MPG1, mpg1, MPEG1 bzw. mpeg1 angeben. Für das MPG2-Format können Sie MPG2, mpg2, MPEG2 bzw. mpeg2 angeben.

inhalt (BLOB(2G) AS LOCATOR)
Die Hostvariable, die den Inhalt für die Aktualisierung des Abbilds, Tons oder Videos enthält. Die Hostvariable kann vom Typ BLOB, BLOB_FILE oder BLOB_LOCATOR sein. DB2 stuft den Datentyp auf BLOB_LOCATOR um und übergibt den LOB-Zeiger an die UDF Replace.

quellenformat (VARCHAR(8))
Das Format der Quelle für die Aktualisierung des Abbilds, Tons oder Videos. Ein Nullwert oder eine leere Zeichenfolge kann angegeben werden, bzw. nur für Abbilder die Zeichenfolge ASIS. Bei diesen drei Angaben versucht der Extender, das Format automatisch zu bestimmen. Für das MPG1-Format können Sie MPG1, mpg1, MPEG1 bzw. mpeg1 angeben. Für das MPG2-Format können Sie MPG2, mpg2, MPEG2 bzw. mpeg2 angeben.

kommentar (LONG VARCHAR)
Ein Kommentar.

attribute (LONG VARCHAR FOR BIT DATA)
Die Attribute des Abbilds, Tons oder Videos.

piktogramm (LONG VARCHAR FOR BIT DATA)
Ein Piktogramm des Abbilds oder Videovollbilds (nur Abbild und Video).

umsetzungsoptionen (VARCHAR(100))
Ein Wert, der Änderungen, wie z. B. Drehung und Komprimierung, angibt, die beim Aktualisieren des Abbilds angewendet werden sollen. Unterstützte Umsetzungsoptionen befinden sich in Tabelle 9 auf Seite 112.

Rückgabewerte (Datentyp)

Die Kennung des zu aktualisierenden Abbilds, Tons oder Videos (DB2IMAGE, DB2AUDIO oder DB2VIDEO).

Beispiele

Aktualisieren des Abbilds für Anita Jones in der Spalte 'picture' der Tabelle 'employee', Umsetzen des Formats des Abbilds von BMP in GIF und Aktualisieren des Kommentars:

```
EXEC SQL BEGIN DECLARE SECTION;
  long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType = MMDB_STORAGE_TYPE_INTERNAL;
EXEC SQL UPDATE EMPLOYEE
  SET PICTURE = REPLACE(PICTURE,
        '/employee/newimg/ajones.bmp',
        'BMP',
        'GIF',
        :hvStorageType,
        'Anita''s new picture')
  WHERE NAME='Anita Jones';
```

SamplingRate

Image	Audio	Video
	X	X

Gibt die Abtastrate eines WAVE- oder AIFF-Tons oder einer Tonspur in einem Video in Samples pro Sekunde zurück.

Kopfdatei

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Abtastrate des Videos oder WAVE- oder AIFF-Tons (INTEGER). Ein Nullwert wird für Töne in anderen Formaten zurückgegeben.

Beispiele

Abrufen der Dateinamen aller Töne aus der Spalte 'sound' der Tabelle 'employee', deren Abtastrate 44,1 KHz ist:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvAud_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (SOUND)
     INTO :hvAud_fname
  FROM EMPLOYEE
     WHERE SAMPLINGRATE(SOUND) = 44100;
```

Size

Image	Audio	Video
X	X	X

Gibt die Größe eines Abbilds, Tons oder Videos in Byte zurück.

Kopfdatei

Image dmbimage.h

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2IMAGE, DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds, Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Größe des Abbilds, Tons oder Videos in Byte (INTEGER).

Beispiele

Abrufen der Dateinamen aller Abbilder in der Spalte 'picture' der Tabelle 'employee', die größer als 310 KB sind:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
     INTO :hvImg_fname
  FROM EMPLOYEE
     WHERE SIZE(PICTURE) > 310000;
```

Thumbnail

Image	Audio	Video
X		X

Gibt eine Piktogrammversion eines Abbilds oder Videovollbilds zurück, das in einer Datenbank gespeichert ist, oder aktualisiert die Version.

Kopfdatei

Image dmbimage.h

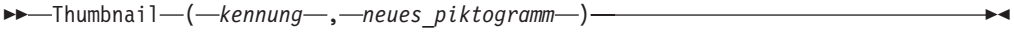
Video dmbvideo.h

Syntax

Abrufen eines Piktogramms

Syntax

Aktualisieren eines Piktogramms



Parameter (Datentyp)

kennung (DB2IMAGE or DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds oder Videos enthält.

neues_piktogramm (LONG VARCHAR FOR BIT DATA)
Quelleninhalt für das Aktualisieren des Piktogramms.

Rückgabewerte (Datentyp)

Beim Abrufen der Inhalt des abgerufenen Piktogramms (LONG VARCHAR FOR BIT DATA), beim Aktualisieren die Kennung des Abbilds oder Videos (DB2IMAGE oder DB2VIDEO).

Beispiele

Abrufen des Piktogramms des Abbilds von Anita Jones, das in der Tabelle 'employee' gespeichert ist:

```
EXEC SQL BEGIN DECLARE SECTION;
 struct{
       short len;
       char data [32000];
         }hvThumbnail;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT THUMBNAIL(PICTURE)
     INTO :hvThumbnail
     FROM EMPLOYEE
     WHERE NAME = 'Anita Jones';
```

Aktualisieren des Piktogramms, das dem Video von Anita Jones in der Tabelle 'employee' zugeordnet ist:

```
EXEC SQL BEGIN DECLARE SECTION;
struct {
       short len;
       char data[10000];
}hvThumbnail;
  EXEC SQL END DECLARE SECTION;

/*  Create thumbnail and    */
/*  store in hvThumbnail    */

EXEC SQL UPDATE EMPLOYEE
   SET VIDEO=THUMBNAIL(
                VIDEO,
               :hvThumbnail)
       WHERE NAME='Anita Jones';
```

TicksPerQNote

Image	Audio	Video
	X	

Gibt die Taktgeschwindigkeit eines aufgezeichneten MIDI-Tons in Impulsen pro Viertelnote zurück.

Kopfdatei
dmbaudio.h

Syntax

Parameter (Datentyp)

kennung (DB2AUDIO)
Name der Spalte oder Hostvariable, die die Kennung des Tons enthält.

Rückgabewerte (Datentyp)
Anzahl von Taktimpulsen pro Viertelnote des MIDI-Tons (SMALLINT). Ein Nullwert wird für Töne in anderen Formaten zurückgegeben.

Beispiele
Abrufen der Dateinamen aller MIDI-Töne in der Spalte 'sound' der Tabelle 'employee', die mit einer höheren Geschwindigkeit als 200 Taktimpulse pro Viertelnote aufgezeichnet wurden:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvAud_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (SOUND)
     INTO :hvAud_fname
  FROM EMPLOYEE
      WHERE FORMAT(SOUND)='MIDI'
      AND TICKSPERQNOTE(SOUND)>200;
```

TicksPerSec

Image	Audio	Video
	X	

Gibt die Taktgeschwindigkeit eines aufgezeichneten MIDI-Tons in Impulsen pro Sekunde zurück.

Kopfdatei
dmbaudio.h

Syntax

Parameter (Datentyp)

kennung (DB2AUDIO)
Name der Spalte oder Hostvariable, die die Kennung des Tons enthält.

Rückgabewerte (Datentyp)
Anzahl von Taktimpulsen pro Sekunde des MIDI-Tons (SMALLINT). Ein Nullwert wird für Töne in anderen Formaten zurückgegeben.

Beispiele
Abrufen der Dateinamen aller MIDI-Töne in der Spalte 'sound' der Tabelle 'employee', die mit einer niedrigeren Geschwindigkeit als 50 Taktimpulse pro Sekunde aufgezeichnet wurden:

```
EXEC SQL BEGIN DECLARE SECTION;
 char hvAud_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(SOUND)
      INTO :hvAud_fname
      FROM EMPLOYEE
      WHERE FORMAT(SOUND)='MIDI'
      AND TICKSPERSEC(SOUND)<50;
```

Updater

Image	Audio	Video
X	X	X

Gibt die Benutzer-ID der Person zurück, die ein Abbild, einen Ton oder ein Video zuletzt in einer Datenbanktabelle aktualisiert hat.

Kopfdatei

Image dmbimage.h

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2IMAGE, DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds, Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Benutzer-ID der Person, die das Abbild, den Ton oder das Video zuletzt aktualisiert hat (CHAR(8)).

Beispiele

Abrufen der Benutzer-ID der Person, die zuletzt das Video aktualisiert hat, das in der Spalte 'video' der Tabelle 'employee' für Robert Smith gespeichert ist:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvUpdater[30];
  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT UPDATER(VIDEO)
     INTO :hvUpdater
  FROM EMPLOYEE
     WHERE NAME='rsmith';
```

UpdateTime

Image	Audio	Video
X	X	X

Gibt eine Zeitmarke zurück, die angibt, wann ein Abbild, Ton oder Video zuletzt in einer Datenbanktabelle aktualisiert wurde.

Kopfdatei

Image dmbimage.h

Audio dmbaudio.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2IMAGE, DB2AUDIO oder DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds, Tons oder Videos enthält.

Rückgabewerte (Datentyp)

Zeitmarke, wann das Abbild, der Ton oder das Video zuletzt aktualisiert wurde (TIMESTAMP).

Beispiele

Abrufen der Namen von Dateien für Abbilder in der Spalte 'picture' der Tabelle 'employee', die in den letzten zwei Tagen aktualisiert wurden:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
     INTO :hvImg_fname
  FROM EMPLOYEE
     WHERE(CURRENT TIMESTAMP -
       UPDATETIME(PICTURE))< 2;
```

Width

Image	Audio	Video
X		X

Gibt die Breite eines Abbilds oder Videovollbilds in Pixeln zurück.

Kopfdatei

Image dmbimage.h

Video dmbvideo.h

Syntax

Parameter (Datentyp)

kennung (DB2IMAGE or DB2VIDEO)
Name der Spalte oder Hostvariable, die die Kennung des Abbilds oder Videos enthält.

Rückgabewerte (Datentyp)

Breite in Pixeln (INTEGER).

Beispiele

Abrufen der Dateinamen aller Abbilder in der Spalte 'picture' der Tabelle 'employee', die schmaler als 300 Pixel sind:

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_fname[251];  EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
     INTO :hvImg_fname
  FROM EMPLOYEE
     WHERE WIDTH(PICTURE)<300;
```

Kapitel 14. Anwendungsprogrammierschnittstellen

In diesem Kapitel erhalten Sie Referenzinformationen zu den Verwaltungs-APIs (Anwendungsprogrammierschnittstellen) von DB2 Extender. Die APIs werden in alphabetischer Reihenfolge aufgelistet.

Die folgenden Informationen werden für die einzelnen APIs zur Verfügung gestellt:

- Der Extender, der die API liefert
- Eine Kurzbeschreibung
- Die für diese API benötigte Berechtigung
- Die Bibliotheksdatei für die API
- Die Includedatei (Kopfdatei) für die API
- Die C-Syntax für den API-Aufruf
- Eine Beschreibung der API-Parameter
- Werte, die von der API zurückgegeben werden
- Beispiele für die Verwendung

DBaAdminGetInaccessibleFiles

Image	Audio	Video
	X	

Gibt die Namen der nicht zugänglichen Dateien zurück, auf die in Audiospalten von Benutzertabellen verwiesen wird. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaAdminGetInaccessibleFiles(
      char *qualifier,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank durchsucht.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von nicht zugänglichen Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

SQL_ERROR und andere SQL-Rückkehrcodes
Fehler von DB2 zurückgegeben.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

Beispiele

Auflisten aller nicht zugänglichen Dateien, auf die in Audiospalten von Tabellen verwiesen wird, deren Eigner der Benutzer mit der Benutzer-ID rsmith ist:

```
#include <dmbaudio.h>

long idx;

rc = DBaAdminGetInaccessibleFiles("rsmith",
       &count, &filelist);
```

DBaAdminGetReferencedFiles

Image	Audio	Video
	X	

Gibt die Namen der Dateien zurück, auf die in Audiospalten von Benutzertabellen verwiesen wird. Wenn eine Datei nicht zugänglich ist (beispielsweise, wenn der Dateiname unter Verwendung der Angaben für die Umgebungsvariable nicht aufgelöst werden kann), wird dem Dateinamen ein Stern vorangestellt. Diese API verwendet das Feld FILENAME der Datenstruktur FILEREF nicht und setzt es daher auf NULL. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur freigeben.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaAdminGetReferencedFiles(
      char *qualifier,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank durchsucht.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

Beispiele

Auflisten aller Dateien, auf die in Audiospalten in Tabellen verwiesen wird, deren Eigner der Benutzer mit der Benutzer-ID ajones ist:

```
#include <dmbaudio.h>

long idx;

rc = DBaAdminGetReferencedFiles("ajones",
       &count, &fileList);
```

DBaAdminIsFileReferenced

Image	Audio	Video
	X	

Gibt eine Liste mit Einträgen zu Audiospalten in Benutzertabellen zurück, die auf eine angegebene Datei verweisen. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaAdminIsFileReferenced(
      char *qualifier,
      char *fileName,
      long *count,
      FILEREF *(*tableList)
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank durchsucht.

fileName (Eingabe)
Der Name der Datei, auf die verwiesen wird.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

tableList (Ausgabe)
Eine Liste mit Tabelleneinträgen, die auf die angegebene Datei verweisen.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

Beispiele

Auflisten der Einträge in Audiospalten in allen Tabellen in der aktuellen Datenbank, die auf die Datei /audios/asmith.wav verweisen:

```
#include <dmbaudio.h>
long idx;

rc = DBaAdminIsFileReferenced(NULL,
       "/audios/asmith.wav",
       &count, &tableList);
```

DBaAdminReorgMetadata

Image	Audio	Video
	X	

„Bereinigt" audiobezogene Metadatentabellen, z. B.:

- Fordert Speicherbereich zurück, der nicht länger in Audiometadatentabellen verwendet wird.
- Löscht in Audiometadatentabellen Verweise auf Audiodateien, die nicht mehr existieren.

Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaAdminReorgMetadata(
      char *qualifier
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal bereinigt. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank bereinigt.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

Beispiele

Bereinigen von Metadatentabellen für Audiospalten in Tabellen, deren Eigner der Benutzer mit der Benutzer-ID rsmith ist:

```
#include <dmbaudio.h>


rc = DBaAdminReorgMetadata("rsmith");
```

DBaDisableColumn

Image	Audio	Video
	X	

Inaktiviert eine Spalte für Audiodaten (DB2Audio-Daten), so dass sie keine Audiodaten speichern kann. Der Inhalt von Spalteneinträgen wird auf NULL gesetzt und die dieser Spalte zugeordneten Metadaten werden gelöscht. Alle Auslöser, die durch den Audio Extender für diese Spalte definiert wurden, werden auch gelöscht. Neue Zeilen können in der Tabelle eingefügt werden, die die inaktivierte Zeile enthält, und die neuen Zeilen können Daten vom Typ DB2Audio enthalten, es gibt allerdings keine Metadaten (in den Tabellen zur Verwaltungsunterstützung), die den neuen Zeilen zugeordnet sind. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird. Es ist zu empfehlen, dass Sie nach dem Aufrufen dieser API eine SQL-Anweisung COMMIT ausgeben.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaDisableColumn(
     char *tableName,
      char *colName,
      );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die die Audiospalte enthält.

colName (Eingabe)
Der Name der Audiospalte.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Inaktivieren der Spalte 'sound' in der Tabelle 'employee' für Audiodaten (DB2Audio-Daten):

```
#include <dmbaudio.h>

rc = DBaDisableColumn("employee", "sound");
```

DBaDisableDatabase

Image	Audio	Video
	X	

Inaktiviert eine Datenbank für Audiodaten (DB2Audio-Daten), so dass sie keine Audiodaten speichern kann. Alle Tabellen in der Datenbank, die für DB2Audio definiert ist, werden ebenfalls inaktiviert. Die Metadaten und UDFs, die durch den Audio Extender für die Datenbank definiert wurden, werden gelöscht. Neue Zeilen können in den Tabellen in der Datenbank eingefügt werden, die mit dem Typ DB2Audio definiert sind, es gibt allerdings keine Metadaten (in den Tabellen zur Verwaltungsunterstützung), die den neuen Zeilen zugeordnet sind. Es ist zu empfehlen, dass Sie nach dem Aufrufen dieser API eine SQL-Anweisung COMMIT ausgeben.

Autorisierung

DBADM, SYSADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaDisableDatabase(
      );
```

Parameter

DBaDisableDatabase hat keine Parameter.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Inaktivieren der aktuellen Datenbank für Audiodaten (DB2Audio-Daten):

```
#include <dmbaudio.h>


rc = DBaDisableDatabase();
```

DBaDisableTable

Image	Audio	Video
	X	

Inaktiviert eine Tabelle für Audiodaten (DB2Audio-Daten), so dass sie keine Audiodaten speichern kann. Alle Spalten in der Tabelle, die für DB2Audio definiert ist, werden ebenfalls inaktiviert. Einige der Metadaten, die durch den Audio Extender für die Tabelle definiert wurden, werden gelöscht. Neue Zeilen können in den Tabellen eingefügt werden, die mit dem Typ DB2Audio definiert sind, es gibt allerdings keine Metadaten (in den Tabellen zur Verwaltungsunterstützung), die den neuen Zeilen zugeordnet sind. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird. Es ist zu empfehlen, dass Sie nach dem Aufrufen dieser API eine SQL-Anweisung COMMIT ausgeben.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaDisableTable(
      char *tableName
      );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die eine Audiospalte enthält.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Inaktivieren der Tabelle 'employee' für Audiodaten (DB2Audio-Daten):

```
#include <dmbaudio.h>


rc = DBaDisableTable("employee");
```

DBaEnableColumn

Image	Audio	Video
	X	

Aktiviert eine Spalte für Audiodaten (DB2Audio-Daten). Die API definiert und verwaltet die Abhängigkeiten zwischen dieser Spalte und den Metadatentabellen. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird. Es ist zu empfehlen, dass Sie nach dem Aufrufen dieser API eine SQL-Anweisung COMMIT ausgeben.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Die Verwendungsberechtigung ist außerdem für Tabellenbereiche und Pufferpools erforderlich, die in den API-Parametern angegeben werden.

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaEnableColumn(
     char *tableName,
      char *colName,
      );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die die Audiospalte enthält.

colName (Eingabe)
Der Name der Audiospalte.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_WARN_ALREADY_ENABLED
Spalte bereits aktiviert.

MMDB_RC_WRONG_SIGNATURE
Datentyp für die angegebene Spalte nicht korrekt. Benutzerdefinierter Datentyp MMDBSYS.DB2AUDIO wird erwartet.

MMDB_RC_COLUMN_DOESNOT_EXIST
Spalte ist in der angegebenen Tabelle nicht definiert.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_NOT_ENABLED
Datenbank oder Tabelle nicht aktiviert.

Beispiele

Aktivieren der Spalte 'sound' in der Tabelle 'employee' für Audiodaten (DB2Audio-Daten):

```
#include <dmbaudio.h>

rc = DBaEnableColumn("employee", "sound");
```

DBaEnableDatabase

Image	Audio	Video
	X	

Aktiviert eine Datenbank für Audiodaten (DB2Audio-Daten). Diese API wird einmal pro Datenbank aufgerufen. Sie definiert einen benutzerdefinierten DB2-Typ, DB2Audio, für den Datenbankmanager. Sie erstellt außerdem alle UDFs, die DB2Audio-Daten bearbeiten. Es ist zu empfehlen, dass Sie nach dem Aufrufen dieser API eine SQL-Anweisung COMMIT ausgeben.

Autorisierung

DBADM, SYSADM, SYSCTRL

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaEnableDatabase(
     char *tableSpace
      );
```

Parameter

tableSpace (Eingabe)

Der Name des Tabellenbereichs, bei dem es sich um eine Gruppe von Behältern handelt, in denen Verwaltungstabellen gespeichert werden. Die Angabe zum Tabellenbereich besteht aus den folgenden drei Teilen: *datats, indexts, longts*. Dabei ist *datats* der Tabellenbereich, in dem Metadatentabellen erstellt werden, *indexts* ist der Tabellenbereich, in dem Indizes für die Metadatentabellen erstellt werden, und *longts* ist der Tabellenbereich, in dem die Werte von langen Spalten in Metadatentabellen (z. B. Spalten, die die Datentypen LONG VARCHAR und LOB enthalten) gespeichert werden. Wenn Sie für einen Teil der Angabe zum Tabellenbereich einen Nullwert angeben, wird der Standardtabellenbereich für diesen Teil verwendet.

Nur EEE: Die Tabellenbereiche, die beim Aktivieren einer Datenbank für einen Extender angegeben werden, sollten in einer Knotengruppe definiert sein, die alle Knoten im partitionierten Datenbanksystem umfasst.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_WARN_ALREADY_ENABLED
Die Datenbank ist bereits aktiviert.

MMDB_RC_API_NOT_SUPPORTED_FOR_SERVER
Der Server, zu dem die Verbindung besteht, unterstützt diesen Befehl nicht.

MMDB_WARN_NOT_ALL_NODES
Der angegebene Tabellenbereich enthält nicht alle Knoten für den Extender. **(Nur EEE)**

MMDB_RC_NOT_SAME_NODEGROUP
Die angegebenen Tabellenbereiche befinden sich nicht in derselben Knotengruppe. **(Nur EEE)**

Beispiele

Aktivieren der aktuellen Datenbank für Audiodaten (DB2Audio-Daten) im Tabellenbereich MYTS: Verwenden der Standardwerte für die Tabellenbereiche für den Index und lange Spalten:

```
#include <dmbaudio.h>


rc = DBaEnableDatabase("myts,,");
```

Aktivieren der aktuellen Datenbank für Audiodaten (DB2Audio-Daten). Verwenden der Standardtabellenbereiche:

```
#include <dmbaudio.h>


rc = DBaEnableDatabase(NULL);
```

DBaEnableTable

Image	Audio	Video
	X	

Aktiviert eine Tabelle für Audiodaten (DB2Audio-Daten). Diese API wird einmal pro Tabelle aufgerufen. Sie erstellt Metadatentabellen, um Attribute für Audiospalten in einer Tabelle zu speichern und zu verwalten. Um die Möglichkeit des Sperrens auszuschließen, sollte die Anwendung die Transaktionen festschreiben, bevor diese API aufgerufen wird. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird. Es ist zu empfehlen, dass Sie nach dem Aufrufen dieser API eine SQL-Anweisung COMMIT ausgeben.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaEnableTable(
      char *tableSpace,
      char *tableName
      );
```

Parameter

tableSpace (Eingabe)
Der Name des Tabellenbereichs, bei dem es sich um eine Gruppe von Behältern handelt, in denen Verwaltungstabellen gespeichert werden.Die Angabe zum Tabellenbereich besteht aus den folgenden drei Teilen: *datats, indexts, longts*. Dabei ist *datats* der Tabellenbereich, in dem Metadatentabellen erstellt werden, *indexts* ist der Tabellenbereich, in dem Indizes für die Metadatentabellen erstellt werden, und *longts* ist der Tabellenbereich, in dem die Werte von langen Spalten in Metadatentabellen (z. B. Spalten, die die Datentypen LONG VARCHAR und LOB enthalten) gespeichert werden. Wenn Sie für einen Teil der Angabe zum Tabellenbereich einen Nullwert angeben, wird der Standardtabellenbereich für diesen Teil verwendet.

Wenn Sie für einen Teil der Angabe zum Tabellenbereich einen Nullwert angeben, wird der Standardtabellenbereich für diesen Teil verwendet.

Nur EEE: Der angegebene Tabellenbereich sollte sich in derselben Knotengruppe befinden wie die Benutzertabelle.

tableName (Eingabe)
Der Name der Tabelle, die eine Audiospalte enthalten soll.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_WARN_ALREADY_ENABLED
Tabelle ist bereits aktiviert.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_TABLE_DOESNOT_EXIST
Tabelle existiert nicht.

MMDB_RC_TABLESPACE_NOT_SAME_NODEGROUP
Der angegebene Tabellenbereich befindet sich nicht in derselben Knotengruppe wie die Benutzertabelle. **(Nur EEE)**

Beispiele

Aktivieren der Tabelle 'employee' für Audiodaten (DB2Audio-Daten) im Tabellenbereich MYTS. Verwenden der Standardwerte für den Index und lange Tabellenbereiche:

```
#include <dmbaudio.h>


rc = DBaEnableTable("myts,,",
       "employee");
```

Aktivieren der Tabelle 'employee' für Audiodaten (DB2Audio-Daten). Verwenden der Standardtabellenbereiche:

```
#include <dmbaudio.h>


rc = DBaEnableTable(NULL,
       "employee");
```

DBaGetError

Image	Audio	Video
	X	

Gibt eine Beschreibung des letzten Fehlers zurück. Rufen Sie diese API auf, wenn eine beliebige andere API einen Fehlercode zurückgibt.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaGetError(
      SQLINTEGER *sqlcode,
      char *errorMsgText
      );
```

Parameter

sqlcode (Ausgabe)
Der generische SQL-Fehlercode.

errorMsgText (Ausgabe)
Der SQL-Fehlernachrichtentext.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

Beispiele

Abrufen des letzten Fehlers und Speichern des SQL-Fehlercodes in `errCode` und des Nachrichtentextes in `errMsg`:

```
#include <dmbaudio.h>


rc = DBaGetError(&errCode, &errMsg);
```

DBaGetInaccessibleFiles

Image	Audio	Video
	X	

Gibt die Namen der nicht zugänglichen Dateien zurück, auf die in Audiospalten von Benutzertabellen verwiesen wird. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

Berechtigung SELECT für aktivierte Audiospalten in allen durchsuchten Benutzertabellen und zugehörigen Tabellen zur Verwaltungsunterstützung

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaGetInaccessibleFiles(
     char *tableName,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, wird diese Tabelle nach Verweisen auf nicht zugängliche Dateien durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von nicht zugänglichen Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten aller nicht zugänglichen Dateien, auf die in Audiospalten in der Tabelle 'employee' verwiesen wird:

```
long idx;
#include <dmbaudio.h>


rc = DBaGetInaccessibleFiles("employee",
       &count, &filelist);
```

DBaGetReferencedFiles

Image	Audio	Video
	X	

Gibt die Namen der Dateien zurück, auf die in Audiospalten von Benutzertabellen verwiesen wird. Wenn eine Datei nicht zugänglich ist (beispielsweise, wenn der Dateiname unter Verwendung der Angaben für die Umgebungsvariable nicht aufgelöst werden kann), wird dem Dateinamen ein Stern vorangestellt. Diese API verwendet das Feld FILENAME der Datenstruktur FILEREF nicht und setzt es daher auf NULL. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur freigeben.

Autorisierung

Berechtigung SELECT für aktivierte Audiospalten in allen durchsuchten Benutzertabellen und zugehörigen Tabellen zur Verwaltungsunterstützung

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaGetReferencedFiles(
     char *tableName,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, wird diese Tabelle nach Verweisen auf Dateien durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen in der Datenbank durchsucht, deren Eigner der Benutzer mit der aktuellen Benutzer-ID ist.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten aller Dateien, auf die in Audiospalten in der Tabelle 'employee' verwiesen wird:

```
#include <dmbaudio.h>
long idx;

rc = DBaGetReferencedFiles("employee",
       &count, &filelist);
```

DBaIsColumnEnabled

Image	Audio	Video
	X	

Stellt fest, ob eine Spalte für Audiodaten (DB2Audio-Daten) aktiviert wurde. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

SYSADM, DBADM, Tabelleneigner oder SELECT für die Benutzertabelle

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaIsColumnEnabled(
     char *tableName,
      char *colName,
short  *status
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname.

colName (Eingabe)
Der Name einer Spalte.

status (Ausgabe)
Gibt an, ob die Spalte aktiviert ist. Dieser Parameter gibt einen numerischen Wert zurück. Der Extender gibt außerdem eine Konstante zurück, die den Status angibt. Folgende Werte und Konstanten sind möglich:

1 MMDB_IS_ENABLED

0 MMDB_IS_NOT_ENABLED

-1 MMDB_INVALID_DATATYPE

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Feststellen, ob die Spalte 'sound' in der Tabelle 'employee' für Audiodaten aktiviert ist:

```
#include <dmbaudio.h>

rc = DBaIsColumnEnabled("employee",
       "sound", &status);
```

DBaIsDatabaseEnabled

Image	Audio	Video
	X	

Stellt fest, ob eine Datenbank für Audiodaten (DB2Audio-Daten) aktiviert wurde. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaIsDatabaseEnabled(
short  *status
      );
```

Parameter

status (Ausgabe)
Gibt an, ob die Datenbank aktiviert ist. Dieser Parameter gibt einen numerischen Wert zurück. Der Extender gibt außerdem eine Konstante zurück, die den Status angibt. Folgende Werte und Konstanten sind möglich:

1 MMDB_IS_ENABLED

0 MMDB_IS_NOT_ENABLED

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Feststellen, ob die Datenbank 'personnl' für Audiodaten aktiviert ist:

```
#include <dmbaudio.h>


rc = DBaIsDatabaseEnabled(&status);
```

DBaIsFileReferenced

Image	Audio	Video
	X	

Gibt eine Liste von Tabelleneinträgen zurück, die auf eine angegebene Datei verweisen. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

Berechtigung SELECT für aktivierte Audiospalten in allen durchsuchten Benutzertabellen und zugehörigen Tabellen zur Verwaltungsunterstützung

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaIsFileReferenced(
     char *tableName,
      char *fileName,
      long *count,
      FILEREF *(*tableList)
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, wird diese Tabelle nach Verweisen auf die angegebene Datei durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen durchsucht, deren Eigner der Benutzer mit der aktuellen Benutzer-ID ist.

fileName (Eingabe)
Der Name der Datei, auf die verwiesen wird.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

tableList (Ausgabe)
Eine Liste mit Tabelleneinträgen, die auf die angegebene Datei verweisen.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten der Einträge in Audiospalten der Tabelle 'employee', die auf die Datei /audios/ajones.wav verweisen:

```
#include <dmbaudio.h>
long idx;

rc = DBaIsFileReferenced(NULL,
       "/audios/ajones.wav",
       &count, &tableList);
```

DBaIsTableEnabled

Image	Audio	Video
	X	

Stellt fest, ob eine Tabelle für Audiodaten (DB2Audio-Daten) aktiviert wurde. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaIsTableEnabled(
     char *tableName,
short  *status
      );
```

Parameter

tableName (Eingabe)
Ein Tabellenname.

status (Ausgabe)
Gibt an, ob die Tabelle aktiviert ist. Dieser Parameter gibt einen numerischen Wert zurück. Der Extender gibt außerdem eine Konstante zurück, die den Status angibt. Folgende Werte und Konstanten sind möglich:

1 MMDB_IS_ENABLED

0 MMDB_IS_NOT_ENABLED

-1 MMDB_INVALID_DATATYPE

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Feststellen, ob die Tabelle 'employee' für Audiodaten (DB2Audio-Daten) aktiviert ist:

```
#include <dmbaudio.h>

rc = DBaIsTableEnabled("employee", &status);
```

DBaPlay

Image	Audio	Video
	X	

Öffnet die Audiowiedergabeeinheit auf dem Client und gibt einen Audioclip wieder. Der Clip kann in einer Audiospalte oder einer externen Datei gespeichert sein:

- Ist der Audioclip in einer externen Datei gespeichert, können Sie entweder den Namen der Datei oder die Audiokennung an diese API übergeben. Die API verwendet die Clientumgebungsvariable DB2AUDIOPATH, um die Dateiadresse aufzulösen. Auf die Datei muss von der Client-Workstation aus zugegriffen werden können.
- Ist der Audioclip in einer Spalte gespeichert, müssen Sie die Audiokennung an die API übergeben. Die Anwendung muss mit der Datenbank verbunden sein und über Lesezugriff auf die Tabelle verfügen, in der der Audioclip gespeichert ist.

Wenn der Audioclip in einer Spalte gespeichert ist, erstellt der Extender eine temporäre Datei und kopiert den Inhalt des Objekts aus der Spalte in die Datei. Der Extender erstellt möglicherweise auch eine temporäre Datei, wenn der Audioclip in einer externen Datei gespeichert ist und wenn sein relativer Dateinamen nicht unter Verwendung der Werte in Umgebungsvariablen aufgelöst werden kann oder wenn die Datei auf der Clientmaschine nicht zugänglich ist. Die temporäre Datei wird in dem Verzeichnis erstellt, das in der Umgebungsvariablen DB2AUDIOTEMP angegeben ist. Der Extender gibt dann den Audioclip aus der temporären Datei wieder.

Autorisierung

Auswahlberechtigung (SELECT) für die Benutzertabelle, wenn ein Audioclip in einer Spalte wiedergegeben wird.

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

Wiedergabe von Audiodaten, die in einer Spalte gespeichert sind

```
long DBaPlay(
      char *playerName,
      MMDB_PLAY_HANDLE,
      DB2Audio *audioHandle,
      waitFlag
      );
```

Syntax

Wiedergabe von Audiodaten, die als Datei gespeichert sind

```
long DBaPlay(
      char *playerName,
     MMDB_PLAY_FILE,
      char *fileName,
      waitFlag
      );
```

Parameter

playerName (Eingabe)
Der Name der Audiowiedergabeeinheit. Ist dieser Wert auf NULL gesetzt, wird die Standardeinheit für die Audiowiedergabe verwendet, die in der Umgebungsvariablen DB2AUDIOPLAYER angegeben ist.

MMDB_PLAY_HANDLE (Eingabe)
Eine Konstante, die angibt, dass die Audiodaten als BLOB gespeichert sind.

MMDB_PLAY_FILE (Eingabe)
Eine Konstante, die angibt, dass die Audiodaten als Datei gespeichert sind, auf die vom Client aus zugegriffen werden kann.

audioHandle (Eingabe)
Die Kennung für die Audiodaten. Dieser Parameter muss übergeben werden, wenn Sie einen Audioclip in einer Spalte wiedergeben. Stellt die Audiokennung eine externe Datei dar, wird die Clientumgebungsvariable DB2VIDEOPATH verwendet, um die Dateiadresse aufzulösen.

fileName (Eingabe)
Der Name der Datei, die die Audiodaten enthält.

waitFlag (Eingabe)
Eine Konstante, die angibt, ob das Programm vor dem Fortfahren wartet, bis der Benutzer die Wiedergabeeinheit schließt. MMDB_PLAY_WAIT führt die Wiedergabeeinheit auf demselben Thread aus wie Ihre Anwendung. MMDB_PLAY_NO_WAIT führt die Wiedergabeeinheit auf einem separaten Thread aus.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Wiedergeben von Audiodaten, die durch 'audioHandle' identifiziert werden. Ausführen der Standardwiedergabeeinheit auf demselben Thread wie die Anwendung:

```
#include <dmbaudio.h>


rc = DBaPlay(NULL, MMDB_PLAY_HANDLE,
       audioHandle, MMDB_PLAY_WAIT);
```

DBaPrepareAttrs

Image	Audio	Video
	X	

Bereitet die vom Benutzer angegebenen Audioattribute vor. Diese API wird verwendet, wenn ein Audioobjekt mit vom Benutzer angegebenen Attributen gespeichert oder aktualisiert wird. Der UDF-Code, der auf dem Server aktiv ist, erwartet Daten immer im „Big-Endian"-Format. Hierbei handelt es sich um ein Format, das von den meisten UNIX-Plattformen verwendet wird. Wenn ein Audioobjekt in einem „Little-Endian"-Format, das heißt von einem Nicht-UNIX-Client, gespeichert oder aktualisiert wird, muss die API DBaPrepare verwendet werden, bevor die Speicher- bzw. Aktualisierungsanforderung gestellt wird.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
void DBaPrepareAttrs(
      MMDBAudioAttrs *audAttr
      );
```

Parameter

audAttr (Eingabe)
Die vom Benutzer angegebenen Audioattribute.

Beispiele

Vorbereiten der vom Benutzer angegebenen Audioattribute:

```
#include <dmbaudio.h>


DBaPrepareAttrs(&imgattr);
```

DBaReorgMetadata

Image	Audio	Video
	X	

„Bereinigt" audiobezogene Metadatentabellen, z. B.:

- Fordert Speicherbereich zurück, der nicht länger in Audiometadatentabellen verwendet wird.
- Löscht in Audiometadatentabellen Verweise auf Audiodateien, die nicht mehr existieren.

Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

ALTER, CONTROL, SYSADM, SYSCTRL, SYSMAINT, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbaudiolib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

Kopfdatei

dmbaudio.h

Syntax

```
long DBaReorgMetadata(
      char *tableName
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, werden die Audiometadatentabellen bereinigt, die der angegebenen Benutzertabelle zugeordnet sind. Wird ein Nullwert angegeben, werden Metadatentabellen für Audiospalten in allen Tabellen bereinigt, deren Eigner der Benutzer mit der aktuellen Benutzer-ID ist.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Bereinigen von Metadatentabellen für Audiospalten in der Tabelle 'employee':

```
#include <dmbaudio.h>

rc = DBaReorgMetadata("employee");
```

DBiAdminGetInaccessibleFiles

Image	Audio	Video
X		

Gibt die Namen der nicht zugänglichen Dateien zurück, auf die in Abbildspalten von Benutzertabellen verwiesen wird. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiAdminGetInaccessibleFiles(
      char *qualifier,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank durchsucht.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von nicht zugänglichen Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten aller nicht zugänglichen Dateien, auf die in Abbildspalten von Tabellen verwiesen wird, deren Eigner der Benutzer mit der Benutzer-ID rjones ist:

```
#include <dmbimage.h>

long idx;

rc = DBiAdminGetInaccessibleFiles
       ("rjones", &count, &filelist);
```

DBiAdminGetReferencedFiles

Image	Audio	Video
X		

Gibt die Namen der Dateien zurück, auf die in Abbildspalten von Benutzertabellen verwiesen wird. Wenn eine Datei nicht zugänglich ist (beispielsweise, wenn der Dateiname unter Verwendung der Angaben für die Umgebungsvariable nicht aufgelöst werden kann), wird dem Dateinamen ein Stern vorangestellt. Diese API verwendet das Feld FILENAME der Datenstruktur FILEREF nicht und setzt es daher auf NULL. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur freigeben.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiAdminGetReferencedFiles(
      char *qualifier,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank durchsucht.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten aller Dateien, auf die in Abbildspalten in Tabellen verwiesen wird, deren Eigner der Benutzer mit der Benutzer-ID ajones ist:

```
#include <dmbimage.h>

long idx;

rc = DBiAdminGetReferencedFiles("ajones",
       &count, &filelist);
```

DBiAdminIsFileReferenced

Image	Audio	Video
X		

Gibt eine Liste mit Einträgen zu Abbildspalten in Benutzertabellen zurück, die auf eine angegebene Datei verweisen. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiAdminIsFileReferenced(
      char *qualifier,
      char *fileName,
      long *count,
      FILEREF *(*tableList)
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank durchsucht.

fileName (Eingabe)
Der Name der Datei, auf die verwiesen wird.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

tableList (Ausgabe)
Eine Liste mit Tabelleneinträgen, die auf die angegebene Datei verweisen.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten der Einträge in Abbildspalten in allen Tabellen in der aktuellen Datenbank, die auf die Datei /images/asmith.bmp verweisen:

```
#include <dmbimage.h>

long idx;

rc = DBiAdminIsFileReferenced(NULL,
       "/images/asmith.bmp",
       &count, &tableList);
```

DBiAdminReorgMetadata

Image	Audio	Video
X		

„Bereinigt" abbildbezogene Metadatentabellen:

- Fordert Speicherbereich zurück, der nicht länger in Abbildmetadatentabellen verwendet wird.
- Löscht in Abbildmetadatentabellen Verweise auf Abbilddateien, die nicht mehr existieren.

Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiAdminReorgMetadata(
      char *qualifier
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal bereinigt. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank bereinigt.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

Beispiele

Bereinigen von Metadatentabellen für Abbildspalten in Tabellen, deren Eigner der Benutzer mit der Benutzer-ID rsmith ist:

```
#include <dmbimage.h>


rc = DBiAdminReorgMetadata("rsmith");
```

DBiBrowse

Image	Audio	Video
X		

Öffnet den Abbildbrowser auf dem Client und zeigt ein Abbild an. Das Abbild kann in einer Abbilddatei oder einer externen Datei gespeichert sein:

- Ist das Abbild in einer externen Datei gespeichert, können Sie entweder den Namen der Datei oder die Abbildkennung an diese API übergeben. Die API verwendet die Clientumgebungsvariable DB2IMAGEPATH, um die Dateiadresse aufzulösen. Auf die Datei muss von der Client-Workstation aus zugegriffen werden können.
- Ist das Abbild in einer Spalte gespeichert, müssen Sie die Abbildkennung an die API übergeben. Die Anwendung muss mit der Datenbank verbunden sein und über Lesezugriff auf die Tabelle verfügen, in der das Abbild gespeichert ist.

Kann der Browser nicht direkt auf das Abbild zugreifen, erstellt der Extender eine temporäre Datei in dem Verzeichnis, das in der Umgebungsvariablen DB2IMAGETEMP angegeben ist. Der Extender zeigt dann das Abbild von der temporären Datei an.

Autorisierung

Auswahlberechtigung (SELECT) für die Benutzertabelle, wenn ein Abbild in einer Spalte angezeigt wird.

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

Anzeigen eines Abbilds, das in einer Spalte gespeichert ist

```
long DBiBrowse(
      char *browserName,
      MMDB_PLAY_HANDLE,
      DB2Image *imageHandle,
      waitFlag
      );
```

Syntax

Anzeigen eines Abbilds, das als Datei gespeichert ist

```
long DBiBrowse(
      char *browserName,
     MMDB_PLAY_FILE,
      char *fileName,
      waitFlag
      );
```

Parameter

browserName (Eingabe)
Der Name des Abbildbrowsers. Ist dieser Wert auf NULL gesetzt, wird der Standardabbildbrowser verwendet, der in der Umgebungsvariablen DB2IMAGEBROWSER angegeben ist.

MMDB_PLAY_HANDLE (Eingabe)
Eine Konstante, die angibt, dass das Abbild als BLOB gespeichert ist.

MMDB_PLAY_FILE (Eingabe)
Eine Konstante, die angibt, dass das Abbild als Datei gespeichert ist, auf die vom Client aus zugegriffen werden kann.

imageHandle (Eingabe)
Die Kennung für das Abbild. Dieser Parameter muss übergeben werden, wenn Sie ein Abbild in einer Spalte anzeigen. Stellt die Abbildkennung eine externe Datei dar, wird die Clientumgebungsvariable DB2IMAGEPATH verwendet, um die Dateiadresse aufzulösen.

fileName (Eingabe)
Der Name der Datei, die das Abbild enthält.

waitFlag (Eingabe)
Eine Konstante, die angibt, ob das Programm vor dem Fortfahren wartet, bis der Benutzer den Browser schließt. MMDB_PLAY_WAIT führt den Browser auf demselben Thread aus wie Ihre Anwendung. MMDB_PLAY-_NO_WAIT führt den Browser auf einem separaten Thread aus.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Anzeigen des Abbilds, das durch 'imageHandle' identifiziert wird. Ausführen des Standardbrowsers auf demselben Thread wie die Anwendung:

```
#include <dmbimage.h>


rc = DBiBrowse(NULL, MMDB_PLAY_HANDLE,
       imageHandle, MMDB_PLAY_WAIT);
```

DBiDisableColumn

Image	Audio	Video
X		

Inaktiviert eine Spalte für Abbilder (DB2Image-Daten), so dass sie keine Abbilddaten speichern kann. Der Inhalt von Spalteneinträgen wird auf NULL gesetzt und die dieser Spalte zugeordneten Metadaten werden gelöscht. Der QBIC-Katalog, der dieser Spalte zugeordnet ist, wird ebenfalls gelöscht. Alle Auslöser, die durch den Image Extender für diese Spalte definiert wurden, werden auch gelöscht. Neue Zeilen können in der Tabelle eingefügt werden, die die inaktivierte Zeile enthält, und die neuen Zeilen können Daten vom Typ DB2Image enthalten, es gibt allerdings keine Metadaten (in den Tabellen zur Verwaltungsunterstützung), die den neuen Zeilen zugeordnet sind. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiDisableColumn(
     char *tableName,
      char *colName,
      );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die die Abbildspalte enthält.

colName (Eingabe)
Der Name der Abbildspalte.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Inaktivieren der Spalte 'picture' in der Tabelle 'employee' für Abbilder (DB2Image-Daten):

```
#include <dmbimage.h>

rc = DBiDisableColumn("employee",
       "picture");
```

DBiDisableDatabase

Image	Audio	Video
X		

Inaktiviert eine Datenbank für Abbilder (DB2Image-Daten), so dass sie keine Abbilddaten speichern kann. Alle Tabellen in der Datenbank, die für DB2Image definiert ist, werden ebenfalls inaktiviert. Die Metadaten und UDFs, die durch den Image Extender für die Datenbank definiert wurden, werden gelöscht. Neue Zeilen können in den Tabellen in der Datenbank eingefügt werden, die mit dem Typ DB2Image definiert sind, es gibt allerdings keine Metadaten (in den Tabellen zur Verwaltungsunterstützung), die den neuen Zeilen zugeordnet sind.

Autorisierung

DBADM, SYSADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiDisableDatabase(
      );
```

Parameter

DBiDisableDatabase hat keine Parameter.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Inaktivieren der aktuellen Datenbank für Abbilder (DB2Image-Daten):

```
#include <dmbimage.h>


rc = DBiDisableDatabase();
```

DBiDisableTable

Image	Audio	Video
X		

Inaktiviert eine Tabelle für Abbilder (DB2Image-Daten), so dass sie keine Abbilddaten speichern kann. Alle Spalten in der Tabelle, die für DB2Image definiert ist, werden ebenfalls inaktiviert. Einige der Metadaten, die durch den Image Extender für die Tabelle definiert wurden, werden gelöscht. Alle QBIC-Kataloge, die den Abbildspalten in der Tabelle zugeordnet sind, werden ebenfalls gelöscht. Neue Zeilen können in den Tabellen eingefügt werden, die mit dem Typ DB2Image definiert sind, es gibt allerdings keine Metadaten (in den Tabellen zur Verwaltungsunterstützung), die den neuen Zeilen zugeordnet sind. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiDisableTable(
      char *tableName
      );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die eine Abbildspalte enthält.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Inaktivieren der Tabelle 'employee' für Abbilder (DB2Image-Daten):

```
#include <dmbimage.h>

rc = DBiDisableTable("employee");
```

DBiEnableColumn

Image	Audio	Video
X		

Aktiviert eine Spalte für Abbilder (DB2Image-Daten). Die API definiert und verwaltet die Abhängigkeiten zwischen dieser Spalte und den Metadatentabellen. Bevor diese API aufgerufen werden kann, muss die Anwendung mit einer Datenbank verbunden und die Benutzertabelle muss festgeschrieben worden sein.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiEnableColumn(
     char *tableName,
      char *colName,
      );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die die Abbildspalte enthält.

colName (Eingabe)
Der Name der Abbildspalte.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_WARN_ALREADY_ENABLED
Spalte bereits aktiviert.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_WRONG_SIGNATURE
Datentyp für die angegebene Spalte nicht korrekt. Benutzerdefinierter Typ MMDBSYS.DB2IMAGE wird erwartet.

MMDB_RC_COLUMN_DOESNOT_EXIST
Spalte ist in der angegebenen Tabelle nicht definiert.

MMDB_RC_NOT_ENABLED
Datenbank oder Tabelle nicht aktiviert.

Beispiele

Aktivieren der Spalte 'picture' in der Tabelle 'employee' für Abbilder:

```
#include <dmbimage.h>


rc = DBiEnableColumn("employee",
       "picture");
```

DBiEnableDatabase

Image	Audio	Video
X		

Aktiviert eine Datenbank für Abbilder (DB2Image-Daten). Diese API wird einmal pro Datenbank aufgerufen. Sie definiert einen benutzerdefinierten DB2-Typ, DB2Image, für den Datenbankmanager. Sie erstellt außerdem alle UDFs, die DB2Image-Daten bearbeiten.

Autorisierung

DBADM, SYSADM, SYSCTRL

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiEnableDatabase(
     char *tableSpace
      );
```

Parameter

tableSpace (Eingabe)

Der Name des Tabellenbereichs, bei dem es sich um eine Gruppe von Behältern handelt, in denen Verwaltungstabellen gespeichert werden. Die Angabe zum Tabellenbereich besteht aus den folgenden drei Teilen: *datats, indexts, longts*. Dabei ist *datats* der Tabellenbereich, in dem Metadatentabellen erstellt werden, *indexts* ist der Tabellenbereich, in dem Indizes für die Metadatentabellen erstellt werden, und *longts* ist der Tabellenbereich, in dem die Werte von langen Spalten in Metadatentabellen (z. B. Spalten, die die Datentypen LONG VARCHAR und LOB enthalten) gespeichert werden. Wenn Sie für einen Teil der Angabe zum Tabellenbereich einen Nullwert angeben, wird der Standardtabellenbereich für diesen Teil verwendet.

Nur EEE: Die Tabellenbereiche, die beim Aktivieren einer Datenbank für einen Extender angegeben werden, sollten in einer Knotengruppe definiert sein, die alle Knoten im partitionierten Datenbanksystem umfasst.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_WARN_ALREADY_ENABLED
Die Datenbank ist bereits aktiviert.

MMDB_RC_API_NOT_SUPPORTED_FOR_SERVER
Der Server, zu dem die Verbindung besteht, unterstützt diesen Befehl nicht.

MMDB_WARN_NOT_ALL_NODES
Der angegebene Tabellenbereich enthält nicht alle Knoten für den Extender. **(Nur EEE)**

MMDB_RC_NOT_SAME_NODEGROUP
Die angegebenen Tabellenbereiche befinden sich nicht in derselben Knotengruppe. **(Nur EEE)**

Beispiele

Aktivieren der aktuellen Datenbank für Abbilder (DB2Image-Daten) im Tabellenbereich MYTS. Verwenden der Standardwerte für die Tabellenbereiche für den Index und lange Spalten:

```
#include <dmbimage.h>

rc = DBiEnableDatabase("myts,,");
```

Aktivieren der aktuellen Datenbank für Abbilder (DB2Image-Daten). Verwenden der Standardtabellenbereiche:

```
#include <dmbimage.h>


rc = DBiEnableDatabase(NULL);
```

DBiEnableTable

Image	Audio	Video
X		

Aktiviert eine Tabelle für Abbilder (DB2Image-Daten). Diese API wird einmal pro Tabelle aufgerufen. Sie erstellt Metadatentabellen, um Attribute für Abbildspalten in einer Tabelle zu speichern und zu verwalten. Um die Möglichkeit des Sperrens auszuschließen, sollte die Anwendung die Transaktionen festschreiben, bevor diese API aufgerufen wird. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiEnableTable(
      char *tableSpace,
      char *tableName
      );
```

Parameter

tableSpace (Eingabe)
Der Name des Tabellenbereichs, bei dem es sich um eine Gruppe von Behältern handelt, in denen Verwaltungstabellen gespeichert werden.Die Angabe zum Tabellenbereich besteht aus den folgenden drei Teilen: *datats, indexts, longts*. Dabei ist *datats* der Tabellenbereich, in dem Metadatentabellen erstellt werden, *indexts* ist der Tabellenbereich, in dem Indizes für die Metadatentabellen erstellt werden, und *longts* ist der Tabellenbereich, in dem die Werte von langen Spalten in Metadatentabellen (z. B. Spalten, die die Datentypen LONG VARCHAR und LOB enthalten) gespeichert werden. Wenn Sie für einen Teil der Angabe zum Tabellenbereich einen Nullwert angeben, wird der Standardtabellenbereich für diesen Teil verwendet.

Wenn Sie für einen Teil der Angabe zum Tabellenbereich einen Nullwert angeben, wird der Standardtabellenbereich für diesen Teil verwendet.

Nur EEE: Der angegebene Tabellenbereich sollte sich in derselben Knotengruppe befinden wie die Benutzertabelle.

tableName (Eingabe)
Der Name der Tabelle, die eine Abbildspalte enthalten soll.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_WARN_ALREADY_ENABLED
Tabelle ist bereits aktiviert.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_TABLE_DOESNOT_EXIST
Tabelle existiert nicht.

MMDB_RC_TABLESPACE_NOT_SAME_NODEGROUP
Der angegebene Tabellenbereich befindet sich nicht in derselben Knotengruppe wie die Benutzertabelle. **(Nur EEE)**

Beispiele

Aktivieren der Tabelle 'employee' für Abbilder (DB2Image-Daten) im Tabellenbereich MYTS. Verwenden der Standardwerte für den Index und lange Tabellenbereiche:

```
#include <dmbimage.h>


rc = DBiEnableTable("myts,,",
       "employee");
```

Aktivieren der Tabelle 'employee' für Abbilder (DB2Image-Daten). Verwenden der Standardtabellenbereiche:

```
#include <dmbimage.h>


rc = DBiEnableTable(NULL,
       "employee");
```

DBiGetError

Image	Audio	Video
X		

Gibt eine Beschreibung des letzten Fehlers zurück. Rufen Sie diese API auf, wenn eine beliebige andere API einen Fehlercode zurückgibt.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiGetError(
      SQLINTEGER *sqlcode,
      char *errorMsgText
      );
```

Parameter

sqlcode (Ausgabe)
Der generische SQL-Fehlercode.

errorMsgText (Ausgabe)
Der SQL-Fehlernachrichtentext.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

Beispiele

Abrufen des letzten Fehlers und Speichern des SQL-Fehlercodes in `errCode` und des Nachrichtentextes in `errMsg`:

```
#include <dmbimage.h>


rc = DBiGetError(&errCode, &errMsg);
```

DBiGetInaccessibleFiles

Image	Audio	Video
X		

Gibt die Namen der nicht zugänglichen Dateien zurück, auf die in Abbildspalten von Benutzertabellen verwiesen wird. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

Berechtigung SELECT für aktivierte Abbildspalten in allen durchsuchten Benutzertabellen und zugehörigen Tabellen zur Verwaltungsunterstützung

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiGetInaccessibleFiles(
     char *tableName,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, wird diese Tabelle nach Verweisen auf nicht zugängliche Dateien durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von nicht zugänglichen Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten aller nicht zugänglichen Dateien, auf die in Abbildspalten in der Tabelle 'employee' verwiesen wird:

```
#include <dmbimage.h>

long idx;

rc = DBiGetInaccessibleFiles("employee",
       &count, &filelist);
```

DBiGetReferencedFiles

Image	Audio	Video
X		

Gibt die Namen der Dateien zurück, auf die in Abbildspalten von Benutzertabellen verwiesen wird. Wenn eine Datei nicht zugänglich ist (beispielsweise, wenn der Dateiname unter Verwendung der Angaben für die Umgebungsvariable nicht aufgelöst werden kann), wird dem Dateinamen ein Stern vorangestellt. Diese API verwendet das Feld FILENAME der Datenstruktur FILEREF nicht und setzt es daher auf NULL. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur freigeben.

Autorisierung

Berechtigung SELECT für aktivierte Abbildspalten in allen durchsuchten Benutzertabellen und zugehörigen Tabellen zur Verwaltungsunterstützung

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiGetReferencedFiles(
     char *tableName,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, wird diese Tabelle nach Verweisen auf Dateien durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen durchsucht, deren Eigner der Benutzer mit der aktuellen Benutzer-ID ist.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten aller Dateien, auf die in Abbildspalten in der Tabelle 'employee' verwiesen wird:

```
#include <dmbimage.h>
long idx;

rc = DBiGetReferencedFiles("employee",
       &count, &filelist);
```

DBiIsColumnEnabled

Image	Audio	Video
X		

Stellt fest, ob eine Spalte für Abbilder (DB2Image-Daten) aktiviert wurde. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

SYSADM, DBADM, Tabelleneigner oder SELECT für die Benutzertabelle

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiIsColumnEnabled(
     char *tableName,
      char *colName,
short  *status
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname.

colName (Eingabe)
Der Name einer Spalte.

status (Ausgabe)
Gibt an, ob die Spalte aktiviert ist. Dieser Parameter gibt einen numerischen Wert zurück. Der Extender gibt außerdem eine Konstante zurück, die den Status angibt. Folgende Werte und Konstanten sind möglich:

1 MMDB_IS_ENABLED

0 MMDB_IS_NOT_ENABLED

-1 MMDB_INVALID_DATATYPE

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_WARN_ALREADY_ENABLED
Spalte bereits aktiviert.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Feststellen, ob die Spalte 'picture' in der Tabelle 'employee' für Abbilder aktiviert ist:

```
#include <dmbimage.h>

rc = DBiIsColumnEnabled("employee",
       "picture", &status);
```

DBiIsDatabaseEnabled

Image	Audio	Video
X		

Stellt fest, ob eine Datenbank für Abbilder (DB2Image-Daten) aktiviert wurde. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiIsDatabaseEnabled(
short  *status
      );
```

Parameter

status (Ausgabe)
Gibt an, ob die Datenbank aktiviert ist. Dieser Parameter gibt einen numerischen Wert zurück. Der Extender gibt außerdem eine Konstante zurück, die den Status angibt. Folgende Werte und Konstanten sind möglich:

1 MMDB_IS_ENABLED

0 MMDB_IS_NOT_ENABLED

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Feststellen, ob die Datenbank 'personnl' für Abbilder aktiviert ist:

```
#include <dmbimage.h>


rc = DBiIsDatabaseEnabled(&status);
```

DBiIsFileReferenced

Image	Audio	Video
X		

Gibt eine Liste von Tabelleneinträgen in Abbildspalten zurück, die auf eine angegebene Datei verweisen. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

Berechtigung SELECT für aktivierte Abbildspalten in allen durchsuchten Benutzertabellen und zugehörigen Tabellen zur Verwaltungsunterstützung

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiIsFileReferenced(
     char *tableName,
      char *fileName,
      long *count,
      FILEREF *(*tableList)
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, wird diese Tabelle nach Verweisen auf die angegebene Datei durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen durchsucht, deren Eigner der Benutzer mit der aktuellen Benutzer-ID ist.

fileName (Eingabe)
Der Name der Datei, auf die verwiesen wird.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

tableList (Ausgabe)
Eine Liste mit Tabelleneinträgen, die auf die angegebene Datei verweisen.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten der Einträge in Abbildspalten der Tabelle 'employee', die auf die Datei /images/ajones.bmp verweisen:

```
#include <dmbimage.h>
long idx;

rc = DBiIsFileReferenced(NULL,
       "/images/ajones.bmp",
       &count, &tableList);
```

DBiIsTableEnabled

Image	Audio	Video
X		

Stellt fest, ob eine Tabelle für Abbilder (DB2Image-Daten) aktiviert wurde. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiIsTableEnabled(
     char *tableName,
short  *status
      );
```

Parameter

tableName (Eingabe)
Ein Tabellenname.

status (Ausgabe)
Gibt an, ob die Tabelle aktiviert ist. Dieser Parameter gibt einen numerischen Wert zurück. Der Extender gibt außerdem eine Konstante zurück, die den Status angibt. Folgende Werte und Konstanten sind möglich:

1 MMDB_IS_ENABLED

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Feststellen, ob die Tabelle 'employee' für Abbilder aktiviert ist:

```
#include <dmbimage.h>


rc = DBiIsTableEnabled("employee",
       &status);
```

DBiPrepareAttrs

Image	Audio	Video
X		

Bereitet die vom Benutzer angegebenen Abbildattribute vor. Diese API wird verwendet, wenn ein Abbildobjekt mit vom Benutzer angegebenen Attributen gespeichert oder aktualisiert wird. Der UDF-Code, der auf dem Server aktiv ist, erwartet Daten immer im „Big-Endian"-Format. Hierbei handelt es sich um ein Format, das von den meisten UNIX-Plattformen verwendet wird. Wenn ein Abbildobjekt in einem „Little-Endian"-Format, das heißt von einem Nicht-UNIX-Client, gespeichert oder aktualisiert wird, muss die API DBiPrepare verwendet werden, bevor die Speicher- bzw. Aktualisierungsanforderung gestellt wird.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
void DBiPrepareAttrs(
      MMDBImageAttrs *imgAttr
      );
```

Parameter

imgAttr (Eingabe)
Die vom Benutzer angegebenen Attribute des Abbilds.

Beispiele

Vorbereiten der vom Benutzer angegebenen Abbildattribute:

```
#include <dmbimage.h>


DBiPrepareAttrs(&imgattr);
```

DBiReorgMetadata

Image	Audio	Video
X		

„Bereinigt" abbildbezogene Metadatentabellen, z. B.:

- Fordert Speicherbereich zurück, der nicht länger in Abbildmetadatentabellen verwendet wird.
- Löscht in Abbildmetadatentabellen Verweise auf Abbilddateien, die nicht mehr existieren.

Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

ALTER, CONTROL, SYSADM, SYSCTRL, SYSMAINT, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

Kopfdatei

dmbimage.h

Syntax

```
long DBiReorgMetadata(
      char *tableName
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, werden die Abbildmetadatentabellen bereinigt, die der angegebenen Benutzertabelle zugeordnet sind. Wird ein Nullwert angegeben, werden Metadatentabellen für Abbildspalten in allen Tabellen bereinigt, deren Eigner der Benutzer mit der aktuellen Benutzer-ID ist.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Bereinigen von Metadatentabellen für Abbildspalten in der Tabelle 'employee':

```
#include <dmbimage.h>

rc = DBiReorgMetadata("employee");
```

DBvAdminGetInaccessibleFiles

Image	Audio	Video
		X

Gibt die Namen der nicht zugänglichen Dateien zurück, auf die in Videospalten von Benutzertabellen verwiesen wird. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvAdminGetInaccessibleFiles(
      char *qualifier,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank durchsucht.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von nicht zugänglichen Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten aller nicht zugänglichen Dateien, auf die in Videospalten von Tabellen verwiesen wird, deren Eigner der Benutzer mit der Benutzer-ID rsmith ist:

```
#include <dmbvideo.h>

long idx;

rc = DBvAdminGetInaccessibleFiles
       ("rsmith", &count,
       &filelist);
```

DBvAdminGetReferencedFiles

Image	Audio	Video
		X

Gibt die Namen der Dateien zurück, auf die in Videospalten von Benutzertabellen verwiesen wird. Wenn eine Datei nicht zugänglich ist (beispielsweise, wenn der Dateiname unter Verwendung der Angaben für die Umgebungsvariable nicht aufgelöst werden kann), wird dem Dateinamen ein Stern vorangestellt. Diese API verwendet das Feld FILENAME der Datenstruktur FILEREF nicht und setzt es daher auf NULL. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur freigeben.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvAdminGetReferencedFiles(
      char *qualifier,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank durchsucht.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten aller Dateien, auf die in Videospalten in Tabellen verwiesen wird, deren Eigner der Benutzer mit der Benutzer-ID ajones ist:

```
#include <dmbvideo.h>

long idx;

rc = DBvAdminGetReferencedFiles
       ("ajones", &count,
       &filelist);
```

DBvAdminIsFileReferenced

Image	Audio	Video
		X

Gibt eine Liste mit Einträgen zu Videospalten in Benutzertabellen zurück, die auf eine angegebene Datei verweisen. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvAdminIsFileReferenced(
      char *qualifier,
      char *fileName,
      long *count,
      FILEREF *(*tableList)
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank durchsucht.

fileName (Eingabe)
Der Name der Datei, auf die verwiesen wird.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

tableList (Ausgabe)
Eine Liste mit Tabelleneinträgen, die auf die angegebene Datei verweisen.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten der Einträge in Videospalten in allen Tabellen in der aktuellen Datenbank, die auf die Datei /videos/asmith.mpg verweisen:

```
#include <dmbvideo.h>

long idx;

rc = DBvAdminIsFileReferenced(NULL,
       "/videos/asmith.mpg",
       &count, &tableList);
```

DBvAdminReorgMetadata

Image	Audio	Video
		X

„Bereinigt" videobezogene Metadatentabellen, z. B.:

- Fordert Speicherbereich zurück, der nicht länger in Videometadatentabellen verwendet wird.
- Löscht in Videometadatentabellen Verweise auf Videodateien, die nicht mehr existieren.

Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvAdminReorgMetadata(
      char *qualifier
      );
```

Parameter

qualifier (Eingabe)
Eine gültige Benutzer-ID oder ein Nullwert. Wird eine Benutzer-ID angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal bereinigt. Wird ein Nullwert angegeben, werden alle Tabellen in der aktuellen Datenbank bereinigt.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_NO_AUTH
Benutzer verfügt nicht über die korrekte Berechtigung für den Aufruf dieser API.

Beispiele

Bereinigen von Metadatentabellen für Videospalten in Tabellen, deren Eigner der Benutzer mit der Benutzer-ID rsmith ist:

```
#include <dmbvideo.h>


rc = DBvAdminReorgMetadata("rsmith");
```

DBvBuildStoryboardFile

Image	Audio	Video
		X

Erstellt eine Aufnahmekatalogdatei mit Einträgen für alle Aufnahmen in einem Video. Das Quellenvideo kann in einer Datenbank oder einer Datei sein. Für jede Aufnahme speichert die API die Aufnahmennummer, die Anfangs- und Endvollbildnummer und die Informationen für mindestens ein repräsentatives Vollbild. Durch die Werte in der Datenstruktur von DBvStoryboardCtrl ist festgelegt, wie viele repräsentative Vollbilder für eine Aufnahme identifiziert werden. Bei Aufnahmen, deren Länge unterhalb eines Schwellenwerts in DBvStoryboardCtrl liegt, identifiziert die API ein repräsentatives Vollbild. Bei Aufnahmen, deren Länge zwischen dem unteren und oberen Schwellenwert in DBvStoryboardCtrl liegt, identifiziert die API zwei repräsentative Vollbilder. Bei Aufnahmen, deren Länge oberhalb des oberen Schwellenwerts in DBvStoryboardCtrl liegt, identifiziert die API drei repräsentative Vollbilder. Zu den Informationen zum repräsentativen Vollbild gehört die Vollbildnummer und der Name der Datei, die den Vollbildinhalt enthält. Diese Informationen können zum Anzeigen eines Storyboards, d. h. einer visuellen Zusammenfassung eines Videos, verwendet werden.

Autorisierung

INSERT, CONTROL

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvBuildStoryboardFile(
      char *fileName,
      DBvIOType *video,
      DBvShotControl *shotCtrl,
      DBvStoryBoardCtrl *sbCtrl
            );
```

Parameter

catalogName (Eingabe)
Der Zeiger auf den Namen der Aufnahmekatalogdatei.

video (Eingabe)
Der Zeiger auf die Videostruktur.

shotCtrl (Eingabe)
Der Zeiger auf die Aufnahmesteuerstruktur.

sbCtrl (Eingabe)
Der Zeiger auf die Storyboard-Steuerstruktur.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_INVALID_CATALOG
Katalog ist ungültig oder existiert nicht.

Beispiele

Erstellen einer Aufnahmekatalogdatei mit dem Namen 'hotshots' und Füllen der Datei mit Daten für alle Aufnahmen in einem Video:

```
#include <dmbshot.h>


rc = DBvBuildStoryboardFile("hotshots",
       video, &shotCtrl, &sbCtrl);
```

DBvBuildStoryboardTable

Image	Audio	Video
		X

Erstellt Einträge in einem Aufnahmekatalog für alle Aufnahmen in einem Video. Das Quellenvideo kann in einer Datenbank oder einer Datei sein. Der Aufnahmekatalog befindet sich in einer Datenbank. Für jede Aufnahme speichert die API die Kennung oder Dateiinformationen für das Quellenvideo. Außerdem speichert sie die Aufnahmennummer, die Anfangs- und Endvollbildnummer und die Informationen für mindestens ein repräsentatives Vollbild. Durch die Werte in der Datenstruktur von DBvStoryboardCtrl ist festgelegt, wie viele repräsentative Vollbilder für eine Aufnahme identifiziert werden. Bei Aufnahmen, deren Länge unterhalb eines Schwellenwerts in DBvStoryboardCtrl liegt, identifiziert die API ein repräsentatives Vollbild. Bei Aufnahmen, deren Länge zwischen dem unteren und oberen Schwellenwert in DBvStoryboardCtrl liegt, identifiziert die API zwei repräsentative Vollbilder. Bei Aufnahmen, deren Länge oberhalb des oberen Schwellenwerts in DBvStoryboardCtrl liegt, identifiziert die API drei repräsentative Vollbilder. Zu den Informationen zum repräsentativen Vollbild gehören die Vollbilddaten. Diese Informationen zum repräsentativen Vollbild, die im Aufnahmekatalog gespeichert sind, können zum Anzeigen eines Storyboards, d. h. einer visuellen Zusammenfassung eines Videos, verwendet werden.

Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

INSERT, CONTROL

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvBuildStoryboardTable(
      char *catalogName ,
      DBvIOType *video,
      DBvShotControl *shotCtrl,
      DBvStoryBoardCtrl *sbCtrl,
      SQLHDBC hdbc
      );
```

Parameter

catalogName (Eingabe)
Der Zeiger auf den Namen des Aufnahmekatalogs.

video (Eingabe)
Der Zeiger auf die Videostruktur.

shotCtrl (Eingabe)
Der Zeiger auf die Aufnahmesteuerstruktur.

sbCtrl (Eingabe)
Der Zeiger auf die Storyboard-Steuerstruktur.

hdbc (Eingabe)
Die Datenbankkennung von SQLConnect.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_INVALID_CATALOG
Katalog ist ungültig oder existiert nicht.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Erstellen von Einträgen für ein Video in einem Aufnahmekatalog 'hotshots':

```
#include <dmbshot.h>


rc = DBvBuildStoryboardTable("hotshots",
       video, &shotCtrl, &sbCtrl, hdbc);
```

DBvClose

Image	Audio	Video
		X

Schließt eine Videodatei, die zum Ermitteln von Szenenwechseln geöffnet wurde.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvClose(
      DB2vIOType *video
      );
```

Parameter

video (Eingabe)
Der Zeiger auf die Videostruktur.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_CANNOT_CLOSE
Videodatei konnte nicht geschlossen werden.

Beispiele

Schließen einer Videodatei, die zuvor zum Ermitteln von Videoszenenwechseln geöffnet wurde:

```
#include <dmbshot.h>

rc=DBvClose (video);
```

DBvCreateIndex

Image	Audio	Video
		X

Erstellt einen Index für ein Video, das in einer Datei gespeichert ist. Der Index wird vom Video Extender verwendet, um auf Aufnahmen und Vollbilder in einem Video zuzugreifen. Der Index wird in einer Flachdatei in demselben Verzeichnis gespeichert wie die Quellenvideodatei.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvCreateIndex(
      char *fileName
      );
```

Parameter

fileName (Eingabe)
Der Zeiger auf einen Videodateinamen.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_OPEN_VIDEO
Videodatei konnte nicht zur Verarbeitung geöffnet werden.

MMDB_RC_INDEX_FAIL
Index konnte nicht erstellt werden.

Beispiele

Erstellen eines Indexes für das Video in der Datei \videos\ajones.mpg:

```
#include <dmbshot.h>

rc = DBvCreateIndex("\videos\ajones.mpg");
```

DBvCreateIndexFromVideo

Image	Audio	Video
		X

Erstellt einen Index für ein Video. Das Video muss zunächst zum Ermitteln der Aufnahmen geöffnet werden. Der Index wird vom Video Extender verwendet, um auf Aufnahmen und Vollbilder in einem Video zuzugreifen. Der Index wird in einer Flachdatei gespeichert. Der Dateiname wird in der Datenstruktur DBvIOType gespeichert.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvCreateIndexFromVideo(
      DBvIOType *video
      );
```

Parameter

video (Aktualisierung)
Der Zeiger auf eine Videostruktur.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_OPEN_VIDEO
Videodatei konnte nicht zur Verarbeitung geöffnet werden.

MMDB_RC_INDEX_FAIL
Index konnte nicht erstellt werden.

Beispiele

Erstellen eines Indexes für ein Video:

```
#include <dmbshot.h>


rc = DBvCreateIndexFromVideo(video);
```

DBvCreateShotCatalog

Image	Audio	Video
		X

Erstellt einen Aufnahmekatalog, bei dem es sich um eine Gruppe von Tabellen handelt, die Informationen zu Aufnahmen, wie z. B. Vollbildnummern, enthalten.

Die Anwendung muss mit einer Datenbank verbunden sein, die sowohl für db2video als auch für db2image aktiviert ist.

Autorisierung

CREATE, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvCreateShotCatalog(
      char *catalogName ,
      SQLHDBC hdbc
      );
```

Parameter

catalogName (Eingabe)
Der Name des zu erstellenden Aufnahmekatalogs.

hdbc (Eingabe)
Die Datenbankkennung von SQLConnect.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Erstellen eines Aufnahmekatalog mit dem Namen 'hotshots':

```
#include <dmbshot.h>

rc = DBvCreateShotCatalog("hotshots", hdbc);
```

DBvDeleteShot

Image	Audio	Video
		X

Löscht eine Aufnahme aus einem Katalog.

Autorisierung

INSERT, CONTROL

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvDeleteShot(
      char *catalogName ,
      char *shotHandle,
      SQLHDBC hdbc
      );
```

Parameter

catalogName (Eingabe)
Der Name des Katalogs.

shotHandle (Eingabe)
Die Aufnahmekennung.

hdbc (Eingabe)
Die Datenbankkennung von SQLConnect.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_ACCESS
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_INVALID_CATALOG
Katalog ist ungültig oder existiert nicht.

Beispiele

Löschen einer Aufnahme aus dem Katalog 'hotshots' unter Verwendung der Aufnahmekennung:

```
#include <dmbshot.h>

rc = DBvDeleteShot("hotshots", shot,
       hdbc);
```

DBvDeleteShotCatalog

Image	Audio	Video
		X

Löscht einen Aufnahmekatalog.

Autorisierung

CONTROL, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvDeleteShotCatalog(
      char *catalogName ,
      SQLHDBC hdbc
      );
```

Parameter

catalogName (Eingabe)
Der Name des zu löschenden Aufnahmekatalogs.

hdbc (Eingabe)
Die Datenbankkennung von SQLConnect.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_ACCESS
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_INVALID_CATALOG
Katalog ist ungültig oder existiert nicht.

Beispiele

```
#include <dmbshot.h>


rc = DBvDeleteShotCatalog("hotshots",
       hdbc);
```

DBvDetectShot

Image	Audio	Video
		X

Sucht nach der nächsten Aufnahme in einer Videodatei. Wenn eine Aufnahme ermittelt wurde, zeichnet diese API die Vollbildnummer und die Vollbilddaten im ersten Vollbild in der ermittelten Aufnahme auf. Sie müssen den Parameter 'shot-Detected' überprüfen, um festzustellen, ob eine Aufnahme ermittelt wurde.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvDetectShot(
      DBvIOType *video,
      unsigned long *start_frame,
      char *shotDetected,
      DBvShotControl *shotCtrl,
      DBvShotType *shot,
            );
```

Parameter

video (Aktualisierung)
Der Zeiger auf die Videostruktur.

start_frame (Ein-/Ausgabe)
Die Vollbildnummer, die als Anfangspunkt für die Suche verwendet wird. Bei der Rückgabe wird der Parameter mit der entsprechenden Position aktualisiert, so dass ab dieser Position nach der nächsten Aufnahme gesucht wird.

shotDetected (Ausgabe)
Parameter für eine gefundene Aufnahme: 1= Vollbild gefunden, 0= kein Vollbild gefunden.

shotCtrl (Eingabe)
Der Zeiger auf die Aufnahmesteuerdaten.

shot (Ausgabe)
Der Zeiger auf die gefundene Aufnahme und die Aufnahmedaten.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_EOF
Dateiende erreicht.

MMDB_NO_INDEX
Videoindex existiert nicht.

Beispiele

Suchen nach der nächsten Aufnahme in einer Videodatei, wobei bei Vollbild 1 begonnen wird:

```
#include <dmbshot.h>


long start_frame=1;

rc = DBvDetectShot(video, start_frame&Detected,
       &shotCtrl, &shot);
```

DBvDisableColumn

Image	Audio	Video
		X

Inaktiviert eine Spalte für Videodaten (DB2Video-Daten), so dass sie keine Videodaten speichern kann. Der Inhalt von Spalteneinträgen wird auf NULL gesetzt und die dieser Spalte zugeordneten Metadaten werden gelöscht. Alle Auslöser, die durch den Video Extender für diese Spalte definiert wurden, werden auch gelöscht. Neue Zeilen können in der Tabelle eingefügt werden, die die inaktivierte Zeile enthält, und die neuen Zeilen können Daten vom Typ DB2Video enthalten, es gibt allerdings keine Metadaten (in den Tabellen zur Verwaltungsunterstützung), die den neuen Zeilen zugeordnet sind. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvDisableColumn(
     char *tableName,
      char *colName,
      );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die die Videospalte enthält.

colName (Eingabe)
Der Name der Videospalte.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Inaktivieren der Spalte 'tv_ads' in der Tabelle 'employee' für Videodaten (DB2Video-Daten):

```
#include <dmbvideo.h>

rc = DBvDisableColumn("employee",
       "tv_ads");
```

DBvDisableDatabase

Image	Audio	Video
		X

Inaktiviert eine Datenbank für Videodaten (DB2Video-Daten), so dass sie keine Videodaten speichern kann. Alle Tabellen in der Datenbank, die für DB2Video definiert sind, werden ebenfalls inaktiviert. Die Metadaten und UDFs, die durch den Video Extender für die Datenbank definiert wurden, werden gelöscht. Neue Zeilen können in den Tabellen in der Datenbank eingefügt werden, die mit dem Typ DB2Video definiert sind, es gibt allerdings keine Metadaten (in den Tabellen zur Verwaltungsunterstützung), die den neuen Zeilen zugeordnet sind.

Autorisierung

DBADM, SYSADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvDisableDatabase(
      );
```

Parameter

DBvDisableDatabase hat keine Parameter.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Inaktivieren der aktuellen Datenbank für Videodaten (DB2Video-Daten):

```
#include <dmbvideo.h>


rc = DBvDisableDatabase();
```

DBvDisableTable

Image	Audio	Video
		X

Inaktiviert eine Tabelle für Videodaten (DB2Video-Daten), so dass sie keine Videodaten speichern kann. Alle Spalten in der Tabelle, die für DB2Video definiert sind, werden ebenfalls inaktiviert. Einige der Metadaten, die durch den Video Extender für die Tabelle definiert wurden, werden gelöscht. Neue Zeilen können in den Tabellen eingefügt werden, die mit dem Typ DB2Video definiert sind, es gibt allerdings keine Metadaten (in den Tabellen zur Verwaltungsunterstützung), die den neuen Zeilen zugeordnet sind. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvDisableTable(
      char *tableName
      );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die die Videospalte enthält.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Inaktivieren der Tabelle 'employee' für Videodaten (DB2Video-Daten):

```
#include <dmbvideo.h>


rc = DBvDisableTable("employee");
```

DBvEnableColumn

Image	Audio	Video
		X

Aktiviert eine Spalte für Videodaten (DB2Video-Daten). Die API definiert und verwaltet die Abhängigkeiten zwischen dieser Spalte und den Metadatentabellen. Bevor diese API aufgerufen werden kann, muss die Anwendung mit einer Datenbank verbunden und die Benutzertabelle muss festgeschrieben worden sein.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Verwendungsberechtigung ist außerdem für Tabellenbereiche und Puffer-Pools erforderlich, die in den API-Parametern angegeben werden.

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvEnableColumn(
     char *tableName,
      char *colName,
      );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die die Videospalte enthält.

colName (Eingabe)
Der Name der Videospalte.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_WARN_ALREADY_ENABLED
Spalte bereits aktiviert.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_WRONG_SIGNATURE
Datentyp für die angegebene Spalte nicht korrekt. Benutzerdefinierter Datentyp MMDBSYS.DB2VIDEO wird erwartet.

MMDB_RC_COLUMN_DOESNOT_EXIST
Spalte ist in der angegebenen Tabelle nicht definiert.

MMDB_RC_NOT_ENABLED
Datenbank oder Tabelle nicht aktiviert.

Beispiele

Aktivieren der Spalte 'video' in der Tabelle 'employee' für Videodaten:

```
#include <dmbvideo.h>


rc = DBvEnableColumn("employee",
       "video");
```

DBvEnableDatabase

Image	Audio	Video
		X

Aktiviert eine Datenbank für Videodaten (DB2Video-Daten). Diese API wird einmal pro Datenbank aufgerufen. Sie definiert einen benutzerdefinierten DB2-Typ, DB2Video, für den Datenbankmanager. Sie erstellt außerdem alle UDFs, die DB2Video-Daten bearbeiten.

Autorisierung

DBADM, SYSADM, SYSCTRL

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvEnableDatabase(
     char *tableSpace
      );
```

Parameter

tableSpace (Eingabe)

Der Name des Tabellenbereichs, bei dem es sich um eine Gruppe von Behältern handelt, in denen Verwaltungstabellen gespeichert werden. Die Angabe zum Tabellenbereich besteht aus den folgenden drei Teilen: *datats, indexts, longts*. Dabei ist *datats* der Tabellenbereich, in dem Metadatentabellen erstellt werden, *indexts* ist der Tabellenbereich, in dem Indizes für die Metadatentabellen erstellt werden, und *longts* ist der Tabellenbereich, in dem die Werte von langen Spalten in Metadatentabellen (z. B. Spalten, die die Datentypen LONG VARCHAR und LOB enthalten) gespeichert werden. Wenn Sie für einen Teil der Angabe zum Tabellenbereich einen Nullwert angeben, wird der Standardtabellenbereich für diesen Teil verwendet.

Nur EEE: Die Tabellenbereiche, die beim Aktivieren einer Datenbank für einen Extender angegeben werden, sollten in einer Knotengruppe definiert sein, die alle Knoten im partitionierten Datenbanksystem umfasst.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_WARN_ALREADY_ENABLED
Die Datenbank ist bereits aktiviert.

MMDB_RC_API_NOT_SUPPORTED_FOR_SERVER
Der Server, zu dem die Verbindung besteht, unterstützt diesen Befehl nicht.

MMDB_WARN_NOT_ALL_NODES
Der angegebene Tabellenbereich enthält nicht alle Knoten für den Extender. **(Nur EEE)**

MMDB_RC_NOT_SAME_NODEGROUP
Die angegebenen Tabellenbereiche befinden sich nicht in derselben Knotengruppe. **(Nur EEE)**

Beispiele

Aktivieren der aktuellen Datenbank für Videodaten (DB2Video-Daten) im Tabellenbereich MYTS. Verwenden der Standardwerte für die Tabellenbereiche für den Index und lange Spalten:

```
#include <dmbvideo.h>


rc = DBvEnableDatabase("myts,,");
```

Aktivieren der aktuellen Datenbank für Videodaten (DB2Video-Daten). Verwenden der Standardtabellenbereiche:

```
#include <dmbvideo.h>


rc = DBvEnableDatabase(NULL);
```

DBvEnableTable

Image	Audio	Video
		X

Aktiviert eine Tabelle für Videodaten (DB2Video-Daten). Diese API wird einmal pro Tabelle aufgerufen. Sie erstellt Metadatentabellen, um Attribute für Videospalten in einer Tabelle zu speichern und zu verwalten. Um die Möglichkeit des Sperrens auszuschließen, sollte die Anwendung die Transaktionen festschreiben, bevor diese API aufgerufen wird. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

CONTROL, ALTER, SYSADM, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvEnableTable(
      char *tableSpace,
      char *tableName
      );
```

Parameter

tableSpace (Eingabe)
Der Name des Tabellenbereichs, bei dem es sich um eine Gruppe von Behältern handelt, in denen Verwaltungstabellen gespeichert werden.Die Angabe zum Tabellenbereich besteht aus den folgenden drei Teilen: *datats, indexts, longts*. Dabei ist *datats* der Tabellenbereich, in dem Metadatentabellen erstellt werden, *indexts* ist der Tabellenbereich, in dem Indizes für die Metadatentabellen erstellt werden, und *longts* ist der Tabellenbereich, in dem die Werte von langen Spalten in Metadatentabellen (z. B. Spalten, die die Datentypen LONG VARCHAR und LOB enthalten) gespeichert werden. Wenn Sie für einen Teil der Angabe zum Tabellenbereich einen Nullwert angeben, wird der Standardtabellenbereich für diesen Teil verwendet.

Wenn Sie für einen Teil der Angabe zum Tabellenbereich einen Nullwert angeben, wird der Standardtabellenbereich für diesen Teil verwendet.

Nur EEE: Der angegebene Tabellenbereich sollte sich in derselben Knotengruppe befinden wie die Benutzertabelle.

tableName (Eingabe)
Der Name der Tabelle, die eine Videospalte enthalten soll.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_WARN_ALREADY_ENABLED
Tabelle ist bereits aktiviert.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_TABLE_DOESNOT_EXIST
Tabelle existiert nicht.

MMDB_RC_TABLESPACE_NOT_SAME_NODEGROUP
Der angegebene Tabellenbereich befindet sich nicht in derselben Knotengruppe wie die Benutzertabelle. **(Nur EEE)**

Beispiele

Aktivieren der Tabelle 'employee' für Videodaten ((DB2Video-Daten) im Tabellenbereich MYTS. Verwenden der Standardwerte für den Index und lange Tabellenbereiche:

```
#include <dmbvideo.h>


rc = DBvEnableTable("myts,,",
       "employee");
```

Aktivieren der Tabelle 'employee' für Videodaten (DB2Video-Daten). Verwenden der Standardtabellenbereiche:

```
#include <dmbvideo.h>


rc = DBvEnableTable(NULL,
       "employee");
```

DBvFrameDataTo24BitRGB

Image	Audio	Video
		X

Setzt ein Videovollbild von einem YUV-Farbwertformat, wie z. B. MPEG, in ein 24-Bit-RGB-Format um. Der Benutzer muss einen Zielpuffer zuordnen, bevor er den API-Aufruf absetzen kann.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvFrameDataTo24BitRGB(
      unsigned char *RGB,
      DBvFrameData *fd,
      unsigned long dx,
      unsigned long dy
      );
```

Parameter

RGB (Ausgabe)
Der Zeiger auf den Ziel-RGB-Puffer.

fd (Eingabe)
Der Zeiger auf die umzusetzenden Vollbilddaten.

dx (Eingabe)
Vollbildbreite.

dy (Eingabe)
Vollbildhöhe.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

Beispiele

Umsetzen eines Videovollbilds von MPEG in 24-Bit-RGB:

```
#include <dmbshot.h>


rc = DBvFrameDataTo24BitRGB(RGB, &video->fd,
       video->dx, video->dy);
```

DBvGetError

Image	Audio	Video
		X

Gibt eine Beschreibung des letzten Fehlers zurück. Rufen Sie diese API auf, wenn eine beliebige andere API einen Fehlercode zurückgibt.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvGetError(
      SQLINTEGER *sqlcode,
      char *errorMsgText
      );
```

Parameter

sqlcode (Ausgabe)
Der generische SQL-Fehlercode.

errorMsgText (Ausgabe)
Der SQL-Fehlernachrichtentext.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

Beispiele

Abrufen des letzten Fehlers und Speichern des SQL-Fehlercodes in `errCode` und des Nachrichtentextes in `errMsg`:

```
#include <dmbvideo.h>


rc = DBvGetError(&errCode, &errMsg);
```

DBvGetFrame

Image	Audio	Video
		X

Ruft das aktuelle Vollbild in einer Videodatei ab. Die Vollbilddaten werden in der Videostruktur DBvFrameData zurückgegeben.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvGetFrame(
      DBvIOType *video
      );
```

Parameter

video (Aktualisierung)
Der Zeiger auf die Videostruktur.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_EOF
Dateiende erreicht.

Beispiele

Abrufen des aktuellen Vollbilds in einer Videodatei:

```
#include <dmbshot.h>


rc = DBvGetFrame(video);
```

DBvGetInaccessibleFiles

Image	Audio	Video
		X

Gibt die Namen der nicht zugänglichen Dateien zurück, auf die in Videospalten von Benutzertabellen verwiesen wird. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

Berechtigung SELECT für aktivierte Videospalten in allen durchsuchten Benutzertabellen und zugehörigen Tabellen zur Verwaltungsunterstützung

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvGetInaccessibleFiles(
     char *tableName,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, wird diese Tabelle nach Verweisen auf nicht zugängliche Dateien durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen mit dem angegebenen Qualifikationsmerkmal durchsucht.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von nicht zugänglichen Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten aller nicht zugänglichen Dateien, auf die in Videospalten in der Tabelle 'employee' verwiesen wird:

```
#include <dmbvideo.h>

long idx;

rc = DBvGetInaccessibleFiles("employee",
       &count, &filelist);
```

DBvGetReferencedFiles

Image	Audio	Video
		X

Gibt die Namen der Dateien zurück, auf die in Videospalten von Benutzertabellen verwiesen wird. Wenn eine Datei nicht zugänglich ist (beispielsweise, wenn der Dateiname unter Verwendung der Angaben für die Umgebungsvariable nicht aufgelöst werden kann), wird dem Dateinamen ein Stern vorangestellt. Diese API verwendet das Feld FILENAME der Datenstruktur FILEREF nicht und setzt es daher auf NULL. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur freigeben.

Autorisierung

Berechtigung SELECT für aktivierte Videospalten in allen durchsuchten Benutzertabellen und zugehörigen Tabellen zur Verwaltungsunterstützung

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvGetReferencedFiles(
     char *tableName,
      long *count,
      FILEREF *(*fileList)
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, wird diese Tabelle nach Verweisen auf Dateien durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen durchsucht, deren Eigner der Benutzer mit der aktuellen Benutzer-ID ist.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

fileList (Ausgabe)
Eine Liste von Dateien, auf die in der Tabelle verwiesen wird.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten aller Dateien, auf die in Videospalten in der Tabelle 'employee' verwiesen wird:

```
#include <dmbvideo.h>
long idx;

rc = DBvGetReferencedFiles("employee",
       &count, &filelist);
```

DBvInitShotControl

Image	Audio	Video
		X

Initialisiert die Werte in der Datenstruktur für die Aufnahmensteuerung.

Autorisierung

INSERT, CONTROL

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvInitShotControl(
      DBvShotControl *shotCtrl,
      );
```

Parameter

shotCtrl (Eingabe)
Der Zeiger auf die Datenstruktur für die Aufnahmensteuerung.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_ACCESS
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Initialisieren der Werte in der Datenstruktur für die Aufnahmensteuerung:

```
#include <dmbshot.h>


rc = DBvInitShotControl(shotCtrl);
```

DBvInitStoryboardCtrl

Image	Audio	Video
		X

Initialisiert die Werte in der Datenstruktur für die Storyboard-Steuerung.

Autorisierung

INSERT, CONTROL

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvInitStoryboardCtrl(
      DBvStoryboardCtrl *sbCtrl,
      );
```

Parameter

shotCtrl (Eingabe)
Der Zeiger auf die Datenstruktur für die Aufnahmensteuerung.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_ACCESS
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Initialisieren der Werte in der Datenstruktur für die Storyboard-Steuerung:

```
#include <dmbshot.h>


rc = DBvInitStoryboardCtrl(shotCtrl);
```

DBvInsertShot

Image	Audio	Video
		X

Fügt eine Aufnahme in einen Aufnahmekatalog ein.

Autorisierung

INSERT, CONTROL

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvInsertShot(
      char *catalogName ,
      DBvShotType *shot,
      DBvIOType *video,
      char *shotHandle,
      SQLHDBC hdbc
      );
```

Parameter

catalogName (Eingabe)
Der Name des Katalogs.

shot (Eingabe)
Der Zeiger auf die erweiterte Aufnahme, die in den Katalog eingefügt werden soll.

shotHandle (Eingabe)
Die Aufnahmekennung.

hdbc (Eingabe)
Die Datenbankkennung von SQLConnect.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_ACCESS
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_INVALID_CATALOG
Katalog ist ungültig oder existiert nicht.

Beispiele

Einfügen einer Aufnahme in einen Aufnahmekatalog 'hotshots':

```
rc = DBvInsertShot(
"hotshots",                    /* shot catalog name */
shot,                          /* pointer to shot structure */
video,                         /* pointer to video structure */
shotHandle,                    /* pointer to shot handle */
hdbc);                         /* database connection handle */
```

DBvIsColumnEnabled

Image	Audio	Video
		X

Stellt fest, ob eine Spalte für Videodaten (DB2Video-Daten) aktiviert wurde. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

SYSADM, DBADM, Tabelleneigner oder SELECT für die Benutzertabelle

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvIsColumnEnabled(
     char *tableName,
      char *colName,
short  *status
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname.

colName (Eingabe)
Der Name einer Spalte.

status (Ausgabe)
Gibt an, ob die Spalte aktiviert ist. Dieser Parameter gibt einen numerischen Wert zurück. Der Extender gibt außerdem eine Konstante zurück, die den Status angibt. Folgende Werte und Konstanten sind möglich:

1 MMDB_IS_ENABLED

0 MMDB_IS_NOT_ENABLED

-1 MMDB_INVALID_DATATYPE

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Feststellen, ob die Spalte 'video' in der Tabelle 'employee' für Videodaten aktiviert ist:

```
#include <dmbvideo.h>

rc = DBvIsColumnEnabled("employee",
       "video", &status);
```

DBvIsDatabaseEnabled

Image	Audio	Video
		X

Stellt fest, ob eine Datenbank für Videodaten (DB2Video-Daten) aktiviert wurde. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvIsDatabaseEnabled(
short  *status
      );
```

Parameter

status (Ausgabe)
Gibt an, ob die Datenbank aktiviert ist. Dieser Parameter gibt einen numerischen Wert zurück. Der Extender gibt außerdem eine Konstante zurück, die den Status angibt. Folgende Werte und Konstanten sind möglich:

1 MMDB_IS_ENABLED

0 MMDB_IS_NOT_ENABLED

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Feststellen, ob die Datenbank 'personnl' für Videodaten aktiviert ist:

```
#include <dmbvideo.h>


rc = DBvIsDatabaseEnabled(&status);
```

DBvIsFileReferenced

Image	Audio	Video
		X

Gibt eine Liste von Tabelleneinträgen in Videospalten zurück, die auf eine angegebene Datei verweisen. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Es ist wichtig, dass Sie die Ressourcen, die durch diese API nach dem Aufruf zugeordnet werden, freigeben. Insbesondere müssen Sie die Dateilistendatenstruktur sowie das Feld für den Dateinamen in jedem Eintrag in der Dateiliste freigeben.

Autorisierung

Berechtigung SELECT für aktivierte Videospalten in allen durchsuchten Benutzertabellen und zugehörigen Tabellen zur Verwaltungsunterstützung

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvIsFileReferenced(
     char *tableName,
      char *fileName,
      long *count,
      FILEREF *(*tableList)
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, wird diese Tabelle nach Verweisen auf die angegebene Datei durchsucht. Wird ein Nullwert angegeben, werden alle Tabellen durchsucht, deren Eigner der Benutzer mit der aktuellen Benutzer-ID ist.

fileName (Eingabe)
Der Name der Datei, auf die verwiesen wird.

count (Ausgabe)
Die Anzahl an Einträgen in der Ausgabeliste.

tableList (Ausgabe)
Eine Liste mit Tabelleneinträgen, die auf die angegebene Datei verweisen.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_MALLOC
System kann keinen Speicher für die Rückgabe der Ergebnisse zuordnen.

Beispiele

Auflisten der Einträge in Videospalten der Tabelle 'employee', die auf die Datei /videos/ajones.mpg verweisen:

```
#include <dmbvideo.h>

long idx;

rc = DBvIsFileReferenced(NULL,
       "/videos/ajones.mpg",
       &count, &tableList);
```

DBvIsIndex

Image	Audio	Video
		X

Prüft, ob ein Videoindex existiert. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvIsIndex(
      char *fileName,
short  *status
      );
```

Parameter

fileName (Eingabe)
Der Name der Datei, auf die verwiesen wird.

status (Ausgabe)
Gibt an, ob der Index existiert. Der Wert 1 bedeutet, dass der Index existiert, der Wert 0 bedeutet, dass er nicht existiert.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_ERROR
Status ungültig.

Beispiele

Überprüfen, ob ein Index für die Videodatei \videos\ajones.mpg existiert:

```
#include <dmbshot.h>

rc = DBvIsIndex("\videos\ajones.mpg", &status);
```

DBvIsTableEnabled

Image	Audio	Video
		X

Stellt fest, ob eine Tabelle für Videodaten (DB2Video-Daten) aktiviert wurde. Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvIsTableEnabled(
     char *tableName,
short  *status
      );
```

Parameter

tableName (Eingabe)
Ein Tabellenname.

status (Ausgabe)
Gibt an, ob die Tabelle aktiviert ist. Dieser Parameter gibt einen numerischen Wert zurück. Der Extender gibt außerdem eine Konstante zurück, die den Status angibt. Folgende Werte und Konstanten sind möglich:

1 MMDB_IS_ENABLED

0 MMDB_IS_NOT_ENABLED

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Feststellen, ob die Tabelle 'employee' für Videodaten aktiviert ist:

```
#include <dmbvideo.h>


rc = DBvIsTableEnabled("employee",
       &status);
```

DBvMergeShots

Image	Audio	Video
		X

Mischt zwei Aufnahmen in eine Aufnahme. Die resultierende Aufnahme verwendet die Aufnahmekennung und das Anfangsvollbild der ersten Aufnahme. Das größere Endvollbild der zwei Aufnahmen wird in der resultierenden Aufnahme verwendet. Die Zeile, auf die die zweite Aufnahme zeigt, wird gelöscht.

Autorisierung

CONTROL, SELECT, DELETE, UPDATE

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvMergeShots(
      char *catalogName ,
      char *shotHandle1,
      char *shotHandle2,
      SQLHDBC hdbc
      );
```

Parameter

catalogName (Eingabe)
Der Name des Aufnahmekatalogs.

shotHandle1 (Eingabe)
Die Kennung der ersten Aufnahme.

shotHandle2 (Eingabe)
Die Kennung der zweiten Aufnahme.

hdbc (Eingabe)
Die Datenbankkennung von SQLConnect.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_CANNOT_MERGE
Aufnahmen können nicht gemischt werden.

MMDB_RC_INVALID_CATALOG
Katalog ist ungültig oder existiert nicht.

Beispiele

Mischen der Aufnahmen mit den Kennungen 'shotHandle1' und 'shotHandle2' im Katalog 'hotshots':

```
#include <dmbshot.h>


rc = DBvMergeShots("hotshots", shotHandle1,
       shotHandle2, hdbc);
```

DBvOpenFile

Image	Audio	Video
		X

Ordnet Speicherbereich für eine Struktur DBvIOType zu und öffnet die Videodatei für den Pixelzugriff. Wurde das Video erfolgreich geöffnet, zeigt es auf die erste Vollbildnummer (Vollbild 0).

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvOpenFile(
      DBvIOType **video,
      char *fileName,
      );
```

Parameter

video (Ausgabe)
Der Zeiger auf den Videostrukturzeiger.

fileName (Eingabe)
Der Name der zu öffnenden Videodatei,

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_CANNOT_OPEN
Videodatei kann nicht geöffnet werden.

MMDB_RC_NO_MEMORY
Nicht genügend Speicher.

MMDB_RC_NO_INDEX
Kein Videodirektzugriffsindex.

Beispiele

Öffnen der Videodatei \videos\ajones.mpg:

```
#include <dmbshot.h>


rc = DBvOpenFile(&videoa,
       "\videos\ajones.mpg");
```

DBvOpenHandle

Image	Audio	Video
		X

Ordnet Speicherbereich für eine Struktur DBvIOType zu und öffnet die Videokennung für den Pixelzugriff. Die Struktur zeigt auf die erste Vollbildnummer (Vollbild 0). Das Video kann ein BLOB (großes binäres Objekt) sein. Das Video wird in die temporäre Datei in dem Verzeichnis kopiert, das durch die Umgebungsvariable DB2VIDEOTEMP angegeben ist. Abhängig von der Existenz des Direktzugriffsindexes wird die Markierung 'isIdx' gesetzt.

Autorisierung

SELECT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvOpenHandle(
      DBvIOType **video,
      DB2Video *videoHandle
      SQLHDBC hdbc
      );
```

Parameter

video (Ausgabe)
Der Zeiger auf die Videostruktur.

videoHandle (Eingabe)
Die Videokennung.

hdbc (Eingabe)
Die Datenbankkennung von SQLConnect.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_CANNOT_OPEN
Videodatei kann nicht geöffnet werden.

MMDB_RC_NO_MEMORY
Nicht genügend Speicher.

MMDB_RC_NO_INDEX
Kein Videodirektzugriffsindex.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_INVALID_HANDLE
Videokennung ungültig.

Beispiele

Öffnen der Videokennung (videoHandle) unter Verwendung des Videozeigers:

```
#include <dmbshot.h>


rc = DBvOpenHandle(&oa, videoHandle, hdbc);
```

DBvPlay

Image	Audio	Video
		X

Öffnet die Wiedergabeeinheit für Videodaten auf dem Client und gibt ein Video wieder. Das Video kann in einer Videospalte oder einer externen Datei gespeichert sein:

- Ist das Video in einer externen Datei gespeichert, können Sie entweder den Namen der Datei oder die Videokennung an diese API übergeben. Die API verwendet die Clientumgebungsvariable DB2VIDEOPATH, um die Dateiadresse aufzulösen. Auf die Datei muss von der Client-Workstation aus zugegriffen werden können.
- Ist das Video in einer Spalte gespeichert, müssen Sie die Videokennung an die API übergeben. Die Anwendung muss mit der Datenbank verbunden sein und über Lesezugriff auf die Tabelle verfügen, in der das Video gespeichert ist.

Wenn das Video in einer Spalte gespeichert ist, erstellt der Extender eine temporäre Datei und kopiert den Inhalt des Objekts aus der Spalte in die Datei. Der Extender erstellt möglicherweise auch eine temporäre Datei, wenn das Video in einer externen Datei gespeichert ist und wenn sein relativer Dateinamen nicht unter Verwendung der Werte in Umgebungsvariablen aufgelöst werden kann oder wenn die Datei auf der Clientmaschine nicht zugänglich ist. Die temporäre Datei wird in dem Verzeichnis erstellt, das in der Umgebungsvariablen DB2VIDEOTEMP angegeben ist. Der Extender gibt dann das Video aus der temporären Datei wieder.

Autorisierung

Auswahlberechtigung (SELECT) für die Benutzertabelle, wenn ein Video in einer Spalte wiedergegeben wird.

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

Wiedergabe eines Videos, das in einer Spalte gespeichert ist

```
long DBvPlay(
      char *playerName,
      MMDB_PLAY_HANDLE,
      DB2Video *videoHandle,
      waitFlag
      );
```

Syntax

Wiedergabe eines Videos, das als Datei gespeichert ist

```
long DBvPlay(
      char *playerName,
     MMDB_PLAY_FILE,
      char *fileName,
      waitFlag
      );
```

Parameter

playerName (Eingabe)
Der Name der Videowiedergabeeinheit. Ist dieser Wert auf NULL gesetzt, wird die Standardeinheit für die Videowiedergabe verwendet, die in der Umgebungsvariablen DB2VIDEOPLAYER angegeben ist.

MMDB_PLAY_HANDLE (Eingabe)
Eine Konstante, die angibt, dass das Video in einer Spalte gespeichert ist.

MMDB_PLAY_FILE (Eingabe)
Eine Konstante, die angibt, dass das Video als Datei gespeichert ist, auf die vom Client aus zugegriffen werden kann.

videoHandle (Eingabe)
Die Kennung für das Video. Dieser Parameter muss übergeben werden, wenn Sie ein Video in einer Spalte wiedergeben. Stellt die Videokennung eine externe Datei dar, wird die Clientumgebungsvariable DB2VIDEOPATH verwendet, um die Dateiadresse aufzulösen.

fileName (Eingabe)
Der Name der Datei, die das Video enthält. Die API verwendet die Clientumgebungsvariable DB2VIDEOPATH, um die Dateiadresse aufzulösen. Auf die Datei muss von der Client-Workstation aus zugegriffen werden können.

waitFlag (Eingabe)
Eine Konstante, die angibt, ob das Programm vor dem Fortfahren wartet, bis der Benutzer die Wiedergabeeinheit schließt. MMDB_PLAY_WAIT führt die Wiedergabeeinheit auf demselben Thread aus wie Ihre Anwendung. MMDB_PLAY_NO_WAIT führt die Wiedergabeeinheit auf einem separaten Thread aus.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Wiedergeben des Videos, das durch 'videoHandle' identifiziert ist. Ausführen der Standardwiedergabeeinheit auf demselben Thread wie die Anwendung:

```
#include <dmbvideo.h>


rc = DBvPlay(NULL, MMDB_PLAY_HANDLE, videoHandle,
                          MMDB_PLAY_WAIT);
```

DBvPrepareAttrs

Image	Audio	Video
		X

Bereitet die vom Benutzer angegebenen Attributattribute vor. Diese API wird verwendet, wenn ein Videoobjekt mit vom Benutzer angegebenen Attributen gespeichert oder aktualisiert wird. Der UDF-Code, der auf dem Server aktiv ist, erwartet Daten immer im „Big-Endian"-Format. Hierbei handelt es sich um ein Format, das von den meisten UNIX-Plattformen verwendet wird. Wenn ein Videoobjekt in einem „Little-Endian"-Format, das heißt von einem Nicht-UNIX-Client, gespeichert oder aktualisiert wird, muss die API DBvPrepare verwendet werden, bevor die Speicher- bzw. Aktualisierungsanforderung abgesetzt wird.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
void DBvPrepareAttrs(
      MMDBVideoAttrs *vidAttr
      );
```

Parameter

vidAttr (Eingabe)
Die vom Benutzer angegebenen Attribute des Videos.

Beispiele

Vorbereiten der vom Benutzer angegebenen Videoattribute:

```
#include <dmbvideo.h>


DBvPrepareAttrs(&vidattr);
```

DBvReorgMetadata

Image	Audio	Video
		X

„Bereinigt" videobezogene Metadatentabellen, z. B.:

- Fordert Speicherbereich zurück, der nicht länger in Videometadatentabellen verwendet wird.
- Löscht in Videometadatentabellen Verweise auf Videodateien, die nicht mehr existieren.

Die Anwendung muss mit einer Datenbank verbunden sein, bevor diese API aufgerufen wird.

Autorisierung

ALTER, CONTROL, SYSADM, SYSCTRL, SYSMAINT, DBADM

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

Kopfdatei

dmbvideo.h

Syntax

```
long DBvReorgMetadata(
     char *tableName,
      );
```

Parameter

tableName (Eingabe)
Ein qualifizierter oder nicht qualifizierter Tabellenname bzw. ein Nullwert. Wird ein Tabellenname angegeben, werden die Videometadatentabellen bereinigt, die der angegebenen Benutzertabelle zugeordnet sind. Wird ein Nullwert angegeben, werden Metadatentabellen für Videospalten in allen Tabellen bereinigt, deren Eigner der Benutzer mit der aktuellen Benutzer-ID ist.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NO_AUTH
Aufrufender verfügt nicht über die korrekte Zugriffsberechtigung.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

Beispiele

Bereinigen von Metadatentabellen für Videospalten in der Tabelle 'employee':

```
#include <dmbvideo.h>

rc = DBvReorgMetadata("employee");
```

DBvSetFrameNumber

Image	Audio	Video
		X

Setzt das aktuelle Vollbild auf eine angegebene Vollbildnummer.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvSetFrameNumber(
      DBvIOType *video
      unsigned long frameNumber
      );
```

Parameter

video (Eingabe)
Der Zeiger auf die Videostruktur.

frameNumber (Eingabe)
Nummer des angeforderten Vollbilds.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_FRAME_NOT_FOUND
Angefordertes Vollbild konnte nicht gefunden werden.

MMDB_NO_INDEX
Videoindex existiert nicht.

Beispiele

Setzen des aktuellen Vollbilds in einer Videodatei auf die Vollbildnummer 85:

```
#include <dmbshot.h>


rc = DBvSetFrameNumber(video, 85);
```

DBvSetShotComment

Image	Audio	Video
		X

Aktualisiert den Anzeigekommentar innerhalb der Aufnahme.

Autorisierung

CONTROL, UPDATE

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvSetShotComment(
      char *catalogName ,
      char *shotHandle,
      char *comment,
      SQLHDBC hdbc
      );
```

Parameter

catalogName (Eingabe)
Der Name des Katalogs.

shotHandle (Eingabe)
Die Kennung der Aufnahme, die aktualisiert werden soll.

comment (Eingabe)
Der neue Kommentar für die Aufnahme.

hdbc (Eingabe)
Die Datenbankkennung von SQLConnect.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_CANNOT_UPDATE
API kann die Aufnahme nicht aktualisieren.

MMDB_RC_INVALID_CATALOG
Katalog ist ungültig oder existiert nicht.

Beispiele

Ändern des Kommentars, der die Aufnahme mit der Kennung 'shotHandle' im Katalog 'hotshots' beschreibt:

```
#include <dmbshot.h>


rc = DBvSetShotComment("hotshot", shotHandle,
       "This is a hot shot.", hdbc);
```

DBvUpdateShot

Image	Audio	Video
		X

Ersetzt die Attribute einer Videoaufnahme im Katalog. Alle Attribute, mit Ausnahme des Kommentars, werden durch die Attribute in der Struktur DBvShotType ersetzt. Ist der Kommentarzeiger NULL, bleibt der bestehende Kommentar unverändert.

Autorisierung

CONTROL, UPDATE

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

Kopfdatei

dmbshot.h

Syntax

```
long DBvUpdateShot(
      char *catalogName ,
      DBvShotType *shot,
      char *shotHandle,
      SQLHDBC hdbc
      );
```

Parameter

catalogName (Eingabe)
Der Name des Katalogs.

shot (Eingabe)
Der Zeiger auf die Aufnahmeinformationsstruktur, die Attribute für die Aufnahme enthält.

shotHandle (Eingabe)
Die Aufnahmekennung.

hdbc (Eingabe)
Die Datenbankkennung von SQLConnect.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RC_NOT_CONNECTED
Keine gültige Verbindung zwischen Anwendung und Datenbank.

MMDB_RC_CANNOT_UPDATE
API kann die Aufnahme nicht aktualisieren.

MMDB_RC_NO_SHOT
Aufnahme existiert nicht.

MMDB_RC_INVALID_CATALOG
Katalog ist ungültig oder existiert nicht.

Beispiele

Aktualisieren der Attribute einer Aufnahme im Katalog 'hotshots':

```
#include <dmbshot.h>


rc = DBvUpdateShot("hotshots", shot,
       shothandle, hdbc);
```

DMBRedistribute (nur EEE)

Image	Audio	Video
X		

Verteilt die QBIC-Merkmaldaten neu, wenn ein Knoten zu einer Knotengruppe hinzugefügt oder aus einer Knotengruppe gelöscht wird, oder wenn eine neue Partitionszuordnung für eine Knotengruppe erstellt wird.

Autorisierung

Die API muss von der ID des Exemplareigners aus ausgeführt werden.

Bibliotheksdatei

Windows	AIX und Solaris
dmbrd.lib	libdmbrd.a (AIX)
	libdmbrd.so (Solaris)

Kopfdatei

dmbrdst.h

Syntax

```
long DMBRedistribute (
       char  *pNodeGroupName,
       char  DataRedistOption    /* ""continue"" use CONTINUE parameter */
       );
                                 /* blank:start redistribution */
```

Parameter

pNodeGroupName (Eingabe)
Der Name der Knotengruppe für die Neuverteilung.

Fehlercodes

MMDB_SUCCESS
API-Aufruf erfolgreich verarbeitet.

MMDB_RD_NO_CONTINUE
Ohne Parameter CONTINUE erneut übergeben.

MMDB_RD_CONTINUE
Mit Parameter CONTINUE erneut übergeben.

Beispiele

Neuverteilen der QBIC-Extender-Daten in der Knotengruppe 'groupone':

```
#include <dmbrdst.h>

rc = DMBRedistribute(groupone,"continue");
```

QbAddFeature

Image	Audio	Video
X		

Fügt ein Merkmal zum momentan geöffneten Katalog hinzu. QbAddFeature erstellt eine Merkmaltabelle für das angegebene Merkmal in der Datenbank. Verwenden Sie nach dem Hinzufügen von Abbildern in der Abbildspalte Ihrer Benutzertabelle die API QbReCatalogColumn, mit der ein Eintrag für jedes Abbild in der Merkmaltabelle hinzugefügt wird und mit der die Abbilder analysiert werden.

Autorisierung

ALTER

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbAddFeature(
      QbCatalogHandle cHdl,
      char *featureName
      );
```

Parameter

cHdl (Eingabe)
Der Zeiger auf die Kennung des Katalogs.

featureName (Eingabe)
Der Name des Merkmals. Die folgenden Merkmale werden mit dem Image Extender zur Verfügung gestellt:

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Fehlercodes

qbicECIvalidHandle
Katalogkennung ungültig.

qbicECCatalogReadOnly
Katalog nur im Lesezugriff geöffnet.

qbicECDupFeature
Merkmal bereits im Katalog.

qbiECinvalidFeatureClass
Angegebenes Merkmal ist kein gültiges Namensformat.

Beispiele

Hinzufügen des Merkmals QbColorFeatureClass zum Katalog, der durch die Kennung 'CatHdl' identifiziert wird:

```
#include <dmbqbapi.h>


rc = QbAddFeature(CatHdl,
       QbColorFeatureClass);
```

QbCatalogColumn

Image	Audio	Video
X		

Katalogisiert die Abbilder in der Abbildspalte Ihrer Benutzertabelle, die noch nicht katalogisiert sind. Die API fügt einen Eintrag für jedes Abbild in der Merkmaltabelle hinzu und analysiert danach die Abbilder. Wenn die API das Abbild analysiert, erstellt sie Abbilddaten und speichert sie im Eintrag für das Abbild in der Merkmaltabelle. Die Standardparameter für die Merkmale werden verwendet. Der Katalog muss geöffnet sein.

Autorisierung

INSERT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbCatalogColumn(
      QbCatalogHandle cHdl
      );
```

Parameter

cHdl (Eingabe)
Der Zeiger auf die Kennung des Katalogs.

Fehlercodes

qbicECIvalidHandle
Katalogkennung ungültig.

qbicECInvalidCatalog
Die angegebene Kennung oder Tabellenspalte ist für den Katalog nicht gültig.

qbicECCatalog Errors
Fehler aufgetreten während des Katalogisierens einzelnder Abbilder. Diese Fehler wurden protokolliert. Keine ROLLBACK-Operation.

qbicECImageNotFound
Abbild nicht gefunden oder Zugriff nicht möglich.

qbicECCatalogRO
Katalog nur im Lesezugriff.

qbicECSQLError
SQL-Fehler aufgetreten.

Beispiele

```
#include <dmbqbapi.h>


rc = QbCatalogColumn(CatHdl);
```

QbCatalogImage

Image	Audio	Video
X		

Katalogisiert ein gesamtes Abbild. Die API fügt einen Eintrag für das Abbild in der Merkmaltabelle hinzu und analysiert danach das Abbild. Wenn die API das Abbild analysiert, erstellt sie Abbilddaten und speichert sie im Eintrag für das Abbild in der Merkmaltabelle. Die Abbildkennung muss aus der Abbildspalte stammen, die dem aktuellen QBIC-Katalog zugeordnet ist. Das Abbild wird entsprechend den momentan definierten Merkmalklassen katalogisiert. Die Standardparameter für die Merkmale im Katalog werden verwendet.

Autorisierung

INSERT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbCatalogImage(
      QbCatalogHandle cHdl,
      char *imgHandle
      );
```

Parameter

cHdl (Eingabe)
Der Zeiger auf die Kennung des Katalogs.

imgHandle (Eingabe)
Die Kennung für das Abbild.

Fehlercodes

qbicECIvalidHandle
Katalogkennung ungültig.

qbicECImageNotFound
Abbild nicht gefunden oder Zugriff nicht möglich.

qbicECCatalogRO
Katalog nur im Lesezugriff.

Beispiele

Katalogisieren eines Abbilds, das durch die Kennung 'Img_hdl' identifiziert wird:

```
#include <dmbqbapi.h>


rc = QbCatalogColumn(CatHdl, Img_hdl);
```

QbCloseCatalog

Image	Audio	Video
X		

Schließt den Katalog. Die API gibt die geöffnete Katalogkennung und die zugeordneten Ressourcen frei.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbCloseCatalog(
      QbCatalogHandle cHdl
      );
```

Parameter

cHdl (Eingabe)
Der Zeiger auf die Kennung des Katalogs.

Fehlercodes

qbicECIvalidHandle
Katalogkennung ungültig.

Beispiele

Schließen des Katalogs, der durch die Kennung 'CatHdl' identifiziert wird:

```
#include <dmbqbapi.h>


rc = QbCloseCatalog(CatHdl);
```

QbCreateCatalog

Image	Audio	Video
X		

Erstellt einen Katalog in der momentan verbundenen Datenbank für die angegebene Abbildspalte. Die Spalte muss für Abbilddaten aktiviert sein. Die API erstellt einen Namen für den Katalog, der als Qualifikationsmerkmal verwendet wird.

Autorisierung

ALTER

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbCreateCatalog(
     char *tableName,
      char *columnName,
     SQLINTEGER autoCatalog,
     char *reserved
     );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die eine Abbildspalte enthält.

columnName (Eingabe)
Der Name der Abbildspalte, für die Sie einen Katalog erstellen.

autoCatalog (Eingabe)
Gibt an, ob Abbilder, die zur Abbildspalte hinzugefügt werden, automatisch katalogisiert, d. h., zu den Merkmaltabellen hinzugefügt und analysiert werden.Geben Sie 1 an, um das automatische Katalogisieren einzuschalten (ON), und geben Sie 0 an, um es auszuschalten (OFF). Wenn Sie das automatische Katalogisieren nicht einschalten, müssen Sie die API QbCatalogColumn oder QbCatalogImage verwenden, um Abbilder, die Sie zur Abbildspalte hinzufügen, zu katalogisieren.

reserved (Eingabe)
Momentan nicht verwendet.

Fehlercodes

qbicECSqlError
SQL-Fehler aufgetreten.

qbicECNotEnabled
Datenbank, Tabelle oder Spalte nicht für den Datentyp DB2Image aktiviert.

qbicECDupCatalog
Katalog existiert bereits.

qbicECunsupportedOption
Nicht unterstützte Option angegeben.

qbicECerrorParameterTooLong
Parameter zu lang für die Verarbeitung.

qbicECqerr
QBIC-Fehler aufgetreten, eine Nachricht wurde erstellt.

qbicECqerrUnknown
Interner QBIC-Fehler aufgetreten, eine generische Fehlernachricht wurde erstellt.

Beispiele

Erstellen eines Katalogs für die Abbilder in der Spalte 'picture' der Tabelle 'employee'. Einschalten des automatischen Katalogisierens:

```
#include <dmbqbapi.h>


rc = QbCreateCatalog("employee",
        "picture", 1);
```

QbDeleteCatalog

Image	Audio	Video
X		

Löscht den angegebenen Katalog aus der aktuellen Datenbank.

Autorisierung

ALTER

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbDeleteCatalog(
     char *tableName,
      char *columnName
      );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die die Abbildspalte enthält.

columnName (Eingabe)
Der Name der Abbildspalte, die dem Katalog zugeordnet ist.

Fehlercodes

qbicECIvalidHandle
Katalogkennung ungültig.

qbicECCatalogInUse
Katalog wurde von einem anderen Benutzer verwendet.

qbicECCatalogRO
Katalog nur im Lesezugriff.

qbicECSystem
Systemfehler aufgetreten.

qbicECSqlError
SQL-Fehler aufgetreten.

Beispiele

Löschen des QBIC-Katalogs, der der Spalte 'picture' in der Tabelle 'employee' zugeordnet ist:

```
#include <dmbqbapi.h>


rc=QbDeleteCatalog("employee", "picture");
```

QbGetCatalogInfo

Image	Audio	Video
X		

Gibt eine Struktur QbCatalogInfo zurück, die die folgenden Informationen enthält:

- Der Name der Benutzertabelle und der Abbildspalte, zu denen der Katalog gehört.
- Die Anzahl an Merkmalen, die im Katalog enthalten sind.
- Der Status des automatischen Katalogisierens.

Autorisierung

SELECT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbGetCatalogInfo(
      QbCatalogHandle cHdl,
      QbCatalogInfo *catInfo
      );
```

Parameter

cHdl (Eingabe)
Der Zeiger auf die Kennung des Katalogs.

catInfo (Ausgabe)
Die Kataloginformationsstruktur.

Fehlercodes

qbicECIvalidHandle
Katalogkennung ungültig.

Beispiele

Abrufen von Informationen zum Katalog, der durch die Kennung 'CatHdl' identifiziert wird und Rückgabe dieser Informationen in einer Struktur mit dem Namen 'catInfo':

```
#include <dmbqbapi.h>


rc = QbGetCatalogInfo(CatHdl, &catInfo);
```

QbListFeatures

Image	Audio	Video
X		

Gibt eine Liste der aktiven Merkmale zurück, die momentan in einem Katalog enthalten sind. Die Liste wird auf einem Puffer, den Sie zuordnen, zurückgegeben.

Autorisierung

SELECT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbListFeatures(
      QbCatalogHandle cHdl,
      SQLINTEGER bufSize,
      SQLINTEGER *count,
      char *featureNames
      );
```

Parameter

cHdl (Eingabe)
Der Zeiger auf die Kennung des Katalogs.

bufSize (Eingabe)
Die Größe Ihres Puffers. Um die benötigte Puffergröße zu schätzen, können Sie die Merkmalanzahl verwenden, die von der API QbGetCatalogInfo zurückgegeben wird, und sie mit der Länge des längsten Merkmalnamens multiplizieren. Merkmalnamen, die im Puffer gespeichert werden, werden durch Leerzeichen voneinander getrennt.

count (Ausgabe)
Die Anzahl der zurückgegebenen Merkmalnamen.

featureNames (Ausgabe)
Der Bereich von Merkmalnamen in Ihrem Puffer.

Fehlercodes

qbicECIvalidHandle
Katalogkennung ungültig.

qbicECTruncateData
Zurückgegebene Daten abgeschnitten, da der Rückgabepuffer zu klein war.

Beispiele

Abrufen einer Liste der aktiven Merkmale in dem Katalog, der durch die Kennung 'CatHdl' identifiziert wird. Speichern der Informationen im Bereich 'featureNames'.

Zunächst muss 'bufSize' berechnet werden, d. h. die Puffergröße, die für die Liste benötigt wird. Verwenden Sie die API QbGetCatalogInfo, um die Anzahl an Merkmalen in der Struktur 'catInfo' zurückzugeben. Multiplizieren Sie dann diese Zahl mit der Konstanten qbiMaxFeatureName, d. h. mit der Größe des längsten Merkmalnamens:

```
#include <dmbqbapi.h>


rc = QbGetCatalogInfo(CatHdl, &catInfo);

bufSize =
   catInfo.featureCount*qbiMaxFeatureName;

rc = QbListFeatures(CatHdl, bufSize,
       count, featureNames);
```

QbOpenCatalog

Image	Audio	Video
X		

Öffnet den QBIC-Katalog für eine bestimmte Abbildspalte. Sie können den Katalog im Lesemodus oder Aktualisierungsmodus öffnen. Die API gibt die Kennung für den geöffneten Katalog zurück. Sie können dann die Kennung in anderen APIs verwenden, um den Katalog zu verwalten und zu füllen.

Stellen Sie sicher, dass Sie den Katalog schließen, wenn Sie die Arbeit damit beendet haben.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbOpenCatalog(
     char *tableName,
      char *columnName,
      SQLINTEGER mode,
      QbCatalogHandle *cHdl
      );
```

Parameter

tableName (Eingabe)
Der Name der Tabelle, die die Abbildspalte enthält.

columnName (Eingabe)
Der Name der Abbildspalte.

mode (Eingabe)
Der Modus, in dem der Katalog geöffnet wird. Gültige Werte sind qbiRead und qbiUpdate.

cHdl (Ausgabe)
Der Zeiger auf die Kennung des Katalogs.

Fehlercodes

qbicECCatalogNotFound
Katalog konnte nicht gefunden werden.

qbicECCatalogInUse
Katalog wurde von einem anderen Benutzer verwendet.

qbicECOpenFailed
Katalog konnte nicht geöffnet werden.

qbicECNotEnabled
Katalog nicht aktiviert.

qbicECNoCatalogFound
Kein Katalog gefunden.

qbicECSqlError
SQL-Fehler aufgetreten.

qbicECSystem
Systemfehler aufgetreten.

Beispiele

Öffnen des Katalogs für die Spalte 'picture' in der Tabelle 'employee' im Lesemodus:

```
#include <dmbqbapi.h>

rc=QbOpenCatalog("employee", "picture",
     qbiread, &CatHdl);
```

QbQueryAddFeature

Image	Audio	Video
X		

Fügt das angegebene Merkmal zu einem QBIC-Katalog hinzu.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQueryAddFeature(
      QbQueryHandle qObj,
      char *featureName
      );
```

Parameter

qObj (Eingabe)
Die Kennung des Abfrageobjekts.

featureName (Eingabe)
Der Name des hinzuzufügenden Abfragemerkmals. Die folgenden Merkmale werden mit dem Image Extender zur Verfügung gestellt:

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

qbiECunknownFeatureClass
Angegebenes Merkmal ist kein gültiger Merkmalklassenname.

qbiECinvalidFeatureClass
Angegebenes Merkmal ist kein gültiges Namensformat.

qbiECfeaturePresent
Angegebenes Merkmal ist bereits ein Member des Abfrageobjekts.

qbiECallocation
System kann nicht genügend Speicher zuordnen.

Beispiele

Hinzufügen des Merkmals QbColorFeatureClass zum Abfrageobjekt, das durch die Kennung 'qoHandle' identifiziert wird:

```
#include <dmbqbapi.h>


rc = QbQueryAddFeature(qoHandle,
        "QbColorFeatureClass");
```

QbQueryCreate

Image	Audio	Video
X		

Erstellt ein Abfrageobjekt und gibt eine Kennung zurück. Sie können die Kennung mit anderen APIs verwenden, um das Abfrageobjekt zu bearbeiten.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQueryCreate(
      QbQueryHandle *qObj
      );
```

Parameter

qObj (Ausgabe)
Ein Zeiger auf die Abfragekennung. Ist die API nicht erfolgreich, wird diese Kennung auf 0 gesetzt.

Fehlercodes

qbiECallocation
System kann nicht genügend Speicher zuordnen.

Beispiele

Erstellen eines Abfrageobjekts und Rückgabe der Kennung in 'qoHandle':

```
#include <dmbqbapi.h>

rc = QbQueryCreate(&qoHandle);
```

QbQueryDelete

Image	Audio	Video
X		

Löscht ein nicht benanntes Abfrageobjekt. Die API gibt den gesamten Speicher, der durch das Abfrageobjekt und alle hinzugefügten Merkmale belegt wurde, frei.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQueryDelete(
      QbQueryHandle qObj
      );
```

Parameter

qObj (Eingabe)
Die Kennung des Abfrageobjekts.

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

Beispiele

Löschen des Abfrageobjekts, das durch die Kennung 'qoHandle' identifiziert wird:

```
#include <dmbqbapi.h>
rc = QbQueryDelete(qoHandle);
```

QbQueryGetFeatureCount

Image	Audio	Video
X		

Gibt die Anzahl an Merkmalen zurück, die zum Abfrageobjekt hinzugefügt wurden. Die folgenden Merkmale werden mit dem Image Extender zur Verfügung gestellt:

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQueryGetFeatureCount(
      QbQueryHandle qObj,
      SQLINTEGER* count
      );
```

Parameter

qObj (Eingabe)
Die Kennung des Abfrageobjekts.

count (Ausgabe)
Der Zeiger auf die Variable, der auf die Anzahl von vorhandenen Merkmalen gesetzt werden soll.

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

Beispiele

Rückgabe der Anzahl von Merkmalen für das Abfrageobjekt, das durch die Kennung 'qoHandle' identifiziert wird:

```
#include <dmbqbapi.h>


rc = QbQueryGetFeatureCount(qoHandle,
       &count);
```

QbQueryGetString

Image	Audio	Video
X		

Gibt die Abfragezeichenfolge aus einer Abfrage zurück. Sie können die Abfragezeichenfolge als Eingabe für UDFs in Ihrer Anwendung verwenden, z. B. für die UDF QbScoreFromStr oder die API QbQueryStringSearch.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQueryGetString(
      QbQueryHandle qObj,
      (char*)*  queryString
      );
```

Parameter

qObj (Eingabe)
Die Kennung des Abfrageobjekts.

queryString (Ausgabe)
Der Zeiger auf die Abfragezeichenfolge für das Abfrageobjekt.

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

Beispiele

Rückgabe der Abfragezeichenfolge für das Abfrageobjekt, das durch die Kennung 'qrHandle' identifiziert wird:

```
#include <dmbqbapi.h>


SQLRETURN  rc;
char *queryString;
QbQueryHandle qrHandle

rc = QbQueryGetString(qrHandle, &queryString);
if (rc == 0)  {
   ...                  /* use the returned queryString for input to UDFs */
   free((void*)queryString);  /* you must free queryString  */
   queryString=(char*)0;
}
```

QbQueryListFeatures

Image	Audio	Video
X		

Gibt eine aktuelle Liste von Merkmalen im Abfrageobjekt zurück. Die API gibt die Liste auf einem Puffer, den Sie zuordnen, zurück. Die folgenden Merkmale werden mit dem Image Extender zur Verfügung gestellt:

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQueryListFeatures(
      QbQueryHandle qObj,
      SQLINTEGER bufSize,
      SQLINTEGER* count,
      char *featureNames
      );
```

Parameter

qObj (Eingabe)
Die Kennung des Abfrageobjekts.

bufSize (Eingabe)
Die Größe des Puffers 'featureNames'. Verwenden Sie die Konstante 'qbiMaxFeatureName' als Puffergröße. Merkmale eines Abfrageobjekts werden durch einen Zeichenfolgenamen identifiziert.

count (Ausgabe)
Die Anzahl der zurückgegebenen Merkmalnamen.

featureNames (Ausgabe)
Der Zeiger auf den Bereich von Merkmalnamen für das Abfrageobjekt. Der Bereich wird auf dem Puffer, den Sie zuordnen, gespeichert.

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

Beispiele

Rückgabe der Anzahl von Merkmalen im Abfrageobjekt, das durch die Kennung 'qoHandle' identifiziert wird. Verwendung der Konstanten 'qbiMaxFeatureName', um die Größe des benötigten Puffers zu bestimmen. Rückgabe des Merkmalnamens auf den Puffer 'feats' und Rückgabe der Anzahl von Merkmalen in der Variablen 'retCount':

```
#include <dmbqbapi.h>


bufSize = qbiMaxFeatureName;

rc = QbQueryListFeatures(qoHandle, bufSize,
        &retCount, feats);
```

QbQueryNameCreate

Image	Audio	Video
X		

Speichert und benennt ein Abfrageobjekt, so dass Sie es in einer UDF verwenden können. Sie stellen den Namen zur Verfügung und können die Beschreibung des Abfrageobjekts zur Verfügung stellen.

Anmerkungen:

1. **Nur EEE:** QbQueryNameCreate wird nicht in einer Umgebung für partitionierte Datenbanken unterstützt.
2. QbQueryNameCreate wird in zukünftigen Releases des Produkts in Umgebungen für nicht partitionierte Datenbanken nicht mehr unterstützt. Um eine Abfrage zu sichern, sollten Sie die API QbQueryGetString verwenden, um die Abfragezeichenfolge abzurufen, und diese Zeichenfolge für die spätere Verwendung in Ihrer Anwendung sichern.

Autorisierung

Keine.

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQueryNameCreate(
      QbQueryHandle qObj,
      char *name,
      char *description
      );
```

Parameter

qObj (Eingabe)
Die Kennung des Abfrageobjekts.

name (Eingabe)
Der Name des Abfrageobjekts. Der Name kann bis zu 18 Zeichen lang sein.

description (Eingabe)
Eine Kurzbeschreibung des Abfrageobjekts, die bis zu 250 Zeichen lang sein kann.

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

Beispiele

Angeben eines Namens und einer Beschreibung für das Abfrageobjekt, das mit der API QbQueryCreate erstellt wurde:

```
#include <dmbqbapi.h>


rc = QbQueryNameCreate(qHandle,
       "fshavgcol",
       "average color query, 10/15/96");
```

QbQueryNameDelete

Image	Audio	Video
X		

Löscht ein Abfrageobjekt. Das Abfrageobjekt muss unter Verwendung der API QbQueryNameCreate erstellt und gespeichert worden sein.

Anmerkungen:

1. **Nur EEE:** QbQueryNameDelete wird nicht in einer Umgebung für partitionierte Datenbanken unterstützt.
2. QbQueryNameDelete wird in zukünftigen Releases des Produkts in Umgebungen für nicht partitionierte Datenbanken nicht mehr unterstützt.

Autorisierung

Keine.

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQueryNameDelete(
      char *name
      );
```

Parameter

name (Eingabe)
Der Name des zu löschenden Abfrageobjekts.

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

Beispiele

Löschen des Abfrageobjekts 'fshavgcol':

```
#include <dmbqbapi.h>


rc = QbQueryNameDelete("fshavgcol",);
```

QbQueryNameSearch

Image	Audio	Video
X		

Durchsucht den QBIC-Katalog nach Abbildern, die mit dem Suchkriterium im Abfrageobjekt übereinstimmen. Das Abfrageobjekt wird nach seinem Namen identifiziert. Die Ergebnisse, die die Abbildkennungen und das Ähnlichkeitsergebnis der QBIC-Suche beinhalten, werden in einem Ergebnisbereich im Clientspeicher gespeichert. Die Ergebnisse werden anhand ihrer Ähnlichkeitsergebnisse sortiert.

Anmerkungen:

1. **Nur EEE:** QbQueryNameSearch wird nicht in einer Umgebung für partitionierte Datenbanken unterstützt.
2. QbQueryNameSearch wird in zukünftigen Releases des Produkts in Umgebungen für nicht partitionierte Datenbanken nicht mehr unterstützt. Um eine Abfrage zu sichern, sollten Sie die API QbQueryGetString verwenden, um die Abfragezeichenfolge abzurufen, und diese Zeichenfolge für die spätere Verwendung in Ihrer Anwendung sichern.

Autorisierung

SELECT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQueryNameSearch(
      char *qName,
     char *tableName,
      char *columnName,
      SQLINTEGER maxReturns,
      QbQueryScope* scope,
      SQLINTEGER resultType,
      SQLINTEGER* count,
      QbResult* returns
      );
```

Parameter

qName (Eingabe)
Der Name des Abfrageobjekts.

tableName (Eingabe)
Der Name der Tabelle, die die Spalte mit Abbildern enthält, die durchsucht werden soll.

columnName (Eingabe)
Der Name der Abbildspalte. Die Spalte muss für Abbilddaten aktiviert sein.

maxReturns (Eingabe)
Die maximale Anzahl von Abbildern, die zurückgegeben werden sollen.

scope (Eingabe) (Reserviert)
Muss auf 0 (NULL) gesetzt sein.

resultType (Eingabe) (Reserviert)
Muss auf 'qbiArray' gesetzt sein.

count (Ausgabe)
Der Zeiger auf die Anzahl der zurückgegebenen Abbilder. Wird Null zurückgegeben, stellen Sie sicher, dass die Abbildspalte für alle Merkmale im Abfrageobjekt katalogisiert ist.

returns (Ausgabe)
Der Zeiger auf den Bereich von QbResult-Strukturen, die die zurückgegebenen Ergebnisse enthalten. Stellen Sie sicher, dass der Puffer, den Sie zuordnen, groß genug ist, um alle zu erwartenden Ergebnisse aufzunehmen.

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

Beispiele

Ausführen der Abfrage FSHAVGCOL für die katalogisierten Abbilder in der Spalte 'picture' der Tabelle 'employee'. Sicherstellen, dass nicht mehr als sechs Abbilder zurückgegeben werden:

```
#include <dmbqbapi.h>

rc = QbQueryNameSearch("fshavgcol",
       "employee", "picture",
       6, 0, qbiArray, &count, &returns);
```

QbQueryRemoveFeature

Image	Audio	Video
X		

Löscht ein Abfragemerkmal aus dem Abfrageobjekt und gibt den zugeordneten Speicher frei. Die folgenden Merkmale werden mit dem Image Extender zur Verfügung gestellt:

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQueryRemoveFeature(
      QbQueryHandle qObj,
      char *featureName
      );
```

Parameter

qObj (Eingabe)
Die Kennung des Abfrageobjekts.

featureName (Eingabe)
Der Name des zu löschenden Merkmals.

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

qbiECinvalidFeatureClass
Angegebenes Merkmal ist kein gültiges Namensformat.

qbiECfeatureNotPresent
Angegebenes Merkmal ist kein Member des Abfrageobjekts.

Beispiele

Löschen des Merkmals QbColorFeatureClass aus dem Abfrageobjekt, das durch die Kennung 'qoHandle' identifiziert wird:

```
#include <dmbqbapi.h>


rc = QbQueryRemoveFeature(qoHandle,
        "QbColorFeatureClass");
```

QbQuerySearch

Image	Audio	Video
X		

Durchsucht den QBIC-Katalog nach Abbildern, die mit dem Suchkriterium im Abfrageobjekt übereinstimmen. Das Abfrageobjekt ist durch eine Abfrageobjektkennung identifiziert. Die Ergebnisse, die die Abbildkennungen und deren Ähnlichkeitsergebnisse der QBIC-Suche beinhalten, werden in einem Ergebnisbereich im Clientspeicher gespeichert. Sie werden anhand ihrer Ähnlichkeitsergebnisse sortiert.

Autorisierung

SELECT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQuerySearch(
      QbQueryHandle qObj,
     char *tableName,
      char *columnName,
      SQLINTEGER maxReturns,
      QbQueryScope* scope,
      SQLINTEGER resultType,
      SQLINTEGER* count,
      QbResult* returns
      );
```

Parameter

qObj (Eingabe)
Die Kennung des Abfrageobjekts.

tableName (Eingabe)
Der Name der Tabelle, die die Spalte mit Abbildern enthält, die durchsucht werden soll.

columnName (Eingabe)
Der Name der Abbildspalte. Die Spalte muss für Abbilddaten aktiviert sein.

maxReturns (Eingabe)
Die maximale Anzahl von Abbildern, die zurückgegeben werden sollen.

scope (Eingabe) (Reserviert)
Muss auf 0 (NULL) gesetzt sein.

resultType (Eingabe) (Reserviert)
Muss auf 'qbiArray' gesetzt sein.

count (Ausgabe)
Der Zeiger auf die Anzahl der zurückgegebenen Abbilder. Wird Null zurückgegeben, stellen Sie sicher, dass die Abbildspalte für alle Merkmale im Abfrageobjekt katalogisiert ist.

returns (Ausgabe)
Der Zeiger auf den Bereich von QbResult-Strukturen, die die zurückgegebenen Ergebnisse enthalten. Stellen Sie sicher, dass der Puffer, den Sie zuordnen, groß genug ist, um alle zu erwartenden Ergebnisse aufzunehmen.

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

Beispiele

Abfragen der katalogisierten Abbilder in der Spalte 'picture' der Tabelle 'employee'. Sicherstellen, dass nicht mehr als sechs Abbilder zurückgegeben werden:

```
#include <dmbqbapi.h>


rc = QbQuerySearch(qHandle, "employee",
       "picture", 6, 0, qbiArray,
       &count, &returns);
```

QbQuerySetFeatureData

Image	Audio	Video
X		

Definiert die Quelle von Abbilddaten für ein Merkmal in einem Abfrageobjekt. Sie können die Datenquelle erst definieren, nachdem Sie ein Merkmal zu einem Abfrageobjekt hinzugefügt haben. Die Datenquelle kann ein Abbild in einer Benutzertabelle, Datei oder in einem Workstationpuffer sein. Sie können eine Clientdatei oder einen Workstationpuffer nur in einer Umgebung für nicht partitionierte Datenbanken als Datenquelle verwenden. Darüber hinaus können Sie explizit Daten für das Merkmal 'Durchschnittsfarbe' oder 'Histogrammfarbe' angeben.

Verwenden Sie die API QbQueryStringSearch zum Definieren der Quelle für Abbilddaten in einer Serverdatei mit Hilfe von QbQuerySetFeatureData. QbQuerySearch verwendet nicht die Quelle für Abbilddaten aus einer Serverdatei, die mit QbQuerySetFeatureData definiert wurde.

Die folgenden Merkmale werden mit dem Image Extender zur Verfügung gestellt:

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQuerySetFeatureData(
      QbQueryHandle qObj,
      char *featureName,
      QbImageSource* imgSource
      );
```

Parameter

qObj (Eingabe)
Die Kennung des Abfrageobjekts.

featureName (Eingabe)
Der Name des zu definierenden Merkmals.

imgSource (Eingabe)
Der Zeiger auf die Abbildquellenstruktur. Wenn Sie 0 (NULL) für 'imgSource' angeben, bedeutet dies, dass die Informationen im Merkmal nicht geändert werden sollen. Weitere Informationen befinden sich im Abschnitt „Strukturen für Datenquellen verwenden" auf Seite 173.

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

qbiECunknownFeatureClass
Angegebenes Merkmal ist kein gültiger Merkmalklassenname.

qbiECinvalidFeatureClass
Angegebenes Merkmal ist kein gültiges Namensformat.

qbiECfeatureNotPresent
Angegebenes Merkmal ist kein Member des Abfrageobjekts.

qbiECfileUnreadable
Abbildquellendatei kann nicht gefunden oder gelesen werden.

Beispiele

Setzen der Datenquelle für das Merkmal 'Histogrammfarbe' in einem Abfrageobjekt. Die Datenquelle für das Merkmal ist eine Datei auf der Client-Workstation:

```
#include <dmbqbapi.h>

QbQueryHandle qoHandle;
QbImageSource imgSource;

imgSource.sourceType = qbiSource_ClientFile;
strcpy(featureName, "QbColorHistogramFeatureClass");
strcpy(imgSource.clientFile, "/tmp/image.gif");

rc = QbQuerySetFeatureData(qoHandle, featureName, &imgSource);
```

QbQuerySetFeatureWeight

Image	Audio	Video
X		

Definiert die Wertigkeit des angegebenen Merkmals in einem Abfrageobjekt.

Autorisierung

Keine

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQuerySetFeatureWeight(
      QbQueryHandle qObj,
      sqldouble* weight
      );
```

Parameter

qObj (Eingabe)
Die Kennung des Abfrageobjekts.

weight (Ausgabe)
Der Zeiger auf die Variable, die auf die Merkmalwertigkeit gesetzt werden soll.

Fehlercodes

qbiECinvalidQueryHandle
Die angegebene Kennung für das Abfrageobjekt verweist nicht auf ein gültiges Abfrageobjekt.

Beispiele

Definieren der Wertigkeit für das Merkmal 'Durchschnittfarbe' in einem Abfrageobjekt, das durch die Kennung 'qoHandle' identifiziert wird:

```
#include <dmbqbapi.h>


weight=2.0
rc = QbQuerySetFeatureWeight(qoHandle, "QbColorFeatureClass", &weight);
```

QbQueryStringSearch

Image	Audio	Video
X		

Durchsucht den QBIC-Katalog nach Abbildern, die mit dem Suchkriterium in der Abfragezeichenfolge übereinstimmen. Die Ergebnisse, die die Abbildkennungen und deren Ähnlichkeitsergebnisse der QBIC-Suche beinhalten, werden in einem Ergebnisbereich im Clientspeicher gespeichert. Sie werden anhand ihrer Ähnlichkeitsergebnisse sortiert.

Autorisierung

SELECT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbQueryStringSearch(
      char *queryString,
     char *tableName,
      char *columnName,
      SQLINTEGER maxReturns,
      QbQueryScope* scope,
      SQLINTEGER resultType,
      SQLINTEGER* count,
      QbResult* returns
      );
```

Parameter

queryString (Eingabe)
Die Abfragezeichenfolge.

tableName (Eingabe)
Der Name der Tabelle, die die Spalte mit Abbildern enthält, die durchsucht werden soll.

columnName (Eingabe)
Der Name der Abbildspalte. Die Spalte muss für Abbilddaten aktiviert sein.

maxReturns (Eingabe)
Die maximale Anzahl von Abbildern, die zurückgegeben werden sollen.

scope (Eingabe) (Reserviert)
Muss auf 0 (NULL) gesetzt sein.

resultType (Eingabe) (Reserviert)
Muss auf 'qbiArray' gesetzt sein.

count (Ausgabe)
Der Zeiger auf die Anzahl der zurückgegebenen Abbilder. Wird Null zurückgegeben, stellen Sie sicher, dass die Abbildspalte für alle Merkmale in der Abfragezeichenfolge katalogisiert ist.

returns (Ausgabe)
Der Zeiger auf den Bereich von QbResult-Strukturen, die die zurückgegebenen Ergebnisse enthalten. Stellen Sie sicher, dass der Puffer, den Sie zuordnen, groß genug ist, um alle zu erwartenden Ergebnisse aufzunehmen.

Fehlercodes

qbiECinvalidQueryString
Die angegebene Abfragezeichenfolge ist ungültig.

Beispiele

Abfragen der katalogisierten Abbilder in der Spalte 'picture' der Tabelle 'employee'. Sicherstellen, dass nicht mehr als sechs Abbilder zurückgegeben werden:

```
#include <dmbqbapi.h>


rc = QbQueryStringSearch("QbColorFeatureClass color=<255, 0, 0>"
       "employee",
       "picture", 6, 0, qbiArray,
       &count, &returns);
```

QbReCatalogColumn

Image	Audio	Video
X		

Analysiert alle vorhandenen Abbilder im geöffneten QBIC-Katalog für ein neues Merkmal erneut. Die Standardparameter für die Merkmale werden verwendet. Verwenden Sie diese API, um ein neues Merkmal zu einem Katalog hinzuzufügen, das bereits Abbilder enthält.

Autorisierung

UPDATE, INSERT

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbReCatalogColumn (
      QbCatalogHandle cHdl
      );
```

Parameter

cHdl (Eingabe)
Der Zeiger auf die Kennung des Katalogs.

Fehlercodes

qbicECIvalidHandle
Katalogkennung ungültig.

qbicECInvalidCatalog
Die angegebene Kennung oder Tabellenspalte ist für den Katalog nicht gültig.

qbicECCatalog Errors
Fehler aufgetreten während des Katalogisierens einzelnder Abbilder. Diese Fehler wurden protokolliert. Keine ROLLBACK-Operation.

qbicECImageNotFound
Abbild nicht gefunden oder Zugriff nicht möglich.

qbicECCatalogRO
Katalog nur im Lesezugriff.

qbicECSQLError
SQL-Fehler aufgetreten.

Beispiele

Erneutes Analysieren aller vorhandenen Abbilder im geöffneten QBIC-Katalog für ein neues Merkmal:

```
#include <dmbqbapi.h>

rc = QbReCatalogColumn(CatHdl);
```

QbRemoveFeature

Image	Audio	Video
X		

Löscht das angegebene Merkmal aus dem geöffneten Katalog.

Autorisierung

ALTER

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbRemoveFeature(
      QbCatalogHandle cHdl,
      char *featureName
      );
```

Parameter

cHdl (Eingabe)
Der Zeiger auf die Kennung des Katalogs.

featureName (Eingabe)
Der Name des Merkmals.

Fehlercodes

qbicECIvalidHandle
Katalogkennung ungültig.

qbicECCatalogReadOnly
Katalog nur im Lesezugriff geöffnet.

qbicECFeatureNotFound
Merkmal nicht im Katalog vorhanden.

qbiECinvalidFeatureClass
Angegebenes Merkmal ist kein gültiges Namensformat.

Beispiele

Löschen des Merkmals QbColorHistogramFeatureClass aus dem Katalog, der durch die Kennung 'CatHdl' identifiziert wird:

```
#include <dmbqbapi.h>


rc=QbRemoveFeature(CatHdl,
     "QbColorHistogramFeatureClass");
```

QbSetAutoCatalog

Image	Audio	Video
X		

Katalogisiert automatisch Abbilder, die in eine Abbildspalte importiert werden. Die API fügt einen Eintrag für jedes Abbild in der Merkmaltabelle hinzu und analysiert danach die Abbilder. Wenn die API das Abbild analysiert, erstellt sie Abbilddaten und speichert sie im Eintrag für das Abbild in der Merkmaltabelle.

Wenn Sie das automatische Katalogisieren nicht einschalten, müssen Sie die API QbCatalogColumn oder QbCatalogImage verwenden, um Abbilder nach dem Hinzufügen zur Abbildspalte zu katalogisieren.

Autorisierung

ALTER

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbSetAutoCatalog(
      QbCatalogHandle cHdl
      SQLINTEGER autoCatalog
      );
```

Parameter

cHdl (Eingabe)
Der Zeiger auf die Kennung des Katalogs.

autoCatalog (Eingabe)
Gibt an, ob Abbilder, die zur Abbildspalte hinzugefügt werden, automatisch zu den Merkmaltabellen hinzugefügt und analysiert werden. Geben Sie 1 an, um das automatische Katalogisieren einzuschalten (ON), und geben Sie 0 an, um es auszuschalten (OFF).

Fehlercodes

qbicECIvalidHandle
Katalogkennung ungültig.

Beispiele

Einschalten des automatischen Katalogisierens für den Katalog, der durch die Kennung 'CatHdl' identifiziert wird:

```
#include <dmbqbapi.h>

rc=QbSetAutoCatalog(CatHdl, 1);
```

QbUncatalogImage

Image	Audio	Video
X		

Löscht ein Abbild aus einem Katalog. Die Abbildkennung muss aus der Abbildspalte stammen, die dem geöffneten QBIC-Katalog zugeordnet ist. Das Abbild wird aus dem geöffneten Katalog gelöscht. Die entsprechende Zeile in der Abbildattributtabelle zeigt an, dass das Abbild nicht katalogisiert ist.

Autorisierung

DELETE

Bibliotheksdatei

OS/2 und Windows	AIX, HP-UX und Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

Kopfdatei

dmbqbapi.h

Syntax

```
SQLRETURN QbUncatalogImage(
      QbCatalogHandle cHdl,
      char *imgHandle
      );
```

Parameter

cHdl (Eingabe)
Der Zeiger auf die Kennung des Katalogs.

imgHandle (Eingabe)
Die Kennung für das Abbild. Sie können diese Kennung aus der Benutzertabelle abrufen.

Fehlercodes

qbicECIvalidHandle
Katalogkennung ungültig.

qbicECImageNotFound
Abbild nicht gefunden oder Zugriff nicht möglich.

qbicECCatalogRO
Katalog nur im Lesezugriff.

Beispiele

Löschen des Abbilds, das durch die Kennung 'Img_hdl' identifizert wird, aus dem Katalog, der durch die Kennung 'CatHdl' identifiziert wird:

```
#include <dmbqbapi.h>


rc=QbUncatalogImage(CatHdl, Img_hdl);
```

QbUncatalogImage

Kapitel 15. Verwaltungsbefehle für den Client

In diesem Kapitel wird die Eingabe von DB2 Extender-Verwaltungsbefehlen für den Client beschrieben. Außerdem werden Referenzinformationen zu jedem DB2 Extender-Verwaltungsbefehl für den Client bereitgestellt.

DB2 Extender-Verwaltungsbefehle eingeben

Sie können DB2 Extender-Verwaltungsbefehle an den db2ext-Befehlszeilenprozessor übergeben. Diese Übergabe kann im interaktiven Modus oder im Befehlsmodus erfolgen. Im interaktiven Modus wird die db2ext-Eingabeaufforderung angezeigt. In diesem Modus können Sie ausschließlich DB2 Extender-Verwaltungsbefehle eingeben. Im Befehlsmodus können Sie Befehle über die Eingabeaufforderung des Betriebssystems eingeben. Es können DB2 Extender-Befehle, DB2-Befehle und Betriebssystembefehle eingegeben werden.

Geben Sie an der DB2-Eingabeaufforderung keine DB2 Extender-Befehle ein.

Um den db2ext-Befehlszeilenprozessor im interaktiven Modus zu starten, gehen Sie wie folgt vor:

Client	Aktion
AIX, HP-UX und Solaris	Geben Sie den Befehl DB2EXT von der Eingabeaufforderung des Betriebssystems aus ein.
Windows	Klicken Sie doppelt auf das Symbol für den DB2EXT-Befehlszeilenprozessor im DB2 Extender-Ordner oder geben Sie den Befehl DB2EXT im DB2-Befehlsfenster ein.

Geben Sie den Befehl QUIT oder TERMINATE ein, um den interaktiven Modus zu beenden. Mit dem Befehl QUIT wird der interaktive Modus beendet, aber die aktuelle Verbindung zu DB2 beibehalten. Der Befehl TERMINATE beendet den interaktiven Modus und unterbricht die aktuelle Verbindung zu DB2.

Um DB2 Extender-Befehle im Befehlsmodus zu übergeben, müssen Sie diese über die Befehlszeile des Betriebssystems eingeben. Jedem DB2 Extender-Befehl muss hierbei das Präfix 'db2ext' vorangestellt werden. Beispiel:

```
db2ext enable database for db2image using mydataspace, myindxspace, mylongspace
```

Die Onlinehilfefunktion für DB2 Extender-Befehle aufrufen

Um die Onlinehilfefunktion für alle DB2 Extender-Befehle aufzurufen, geben Sie Folgendes ein:

```
db2ext ?
```

ADD QBIC FEATURE

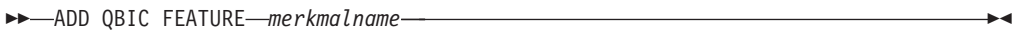
Image	Audio	Video
X		

Erstellt eine Merkmaltabelle für das angegebene Merkmal im aktuellen Katalog. Vorhandene Abbilder im Katalog werden nicht automatisch durch den Image Extender erneut analysiert.

Autorisierung

ALTER, CONTROL, SYSADM, DBADM

Befehlssyntax



Befehlsparameter

merkmalname
Der Name des Merkmals, das zum QBIC-Katalog hinzugfügt wird. Die folgenden Merkmale werden mit dem Image Extender zur Verfügung gestellt:

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Beispiele

Hinzufügen des Merkmals QbColorFeatureClass zum momentan geöffneten Katalog:

```
add qbic feature qbcolorfeatureclass
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

Der Katalog muss geöffnet sein.

CATALOG QBIC COLUMN

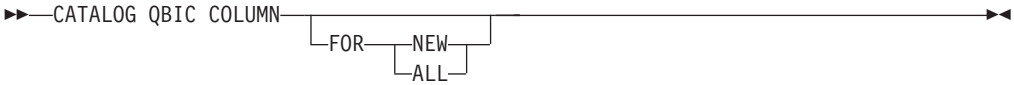
Image	Audio	Video
X		

Katalogisiert die Abbilder in der Abbildspalte und aktualisiert den momentan geöffneten QBIC-Katalog mit Merkmaldaten. Sie können den Katalog für alle Abbilder in der Abbildspalte aktualisieren oder nur für die neuen Abbilder, die seit der letzten Analyse des Katalogs zur Abbildspalte hinzugefügt wurden.

Autorisierung

INSERT, CONTROL, SYSADM, DBADM

Befehlssyntax



Befehlsparameter

Keine

Beispiele

Katalogisieren der neuen Abbilder im aktuellen Katalog, d. h. Abbilder, die noch nicht katalogisiert wurden:

```
catalog qbic column for new
```

Hinweise

Wird NEW angegeben, aktualisiert der Image Extender den Katalog nur mit den Abbilder, die noch nicht katalogisiert sind. Wird ALL angegeben, analysiert der Image Extender jedes Abbild in der Abbildspalte für den aktuellen Katalog. NEW ist der Standardwert.

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

Der Katalog muss geöffnet sein.

CLOSE QBIC CATALOG

Image	Audio	Video
X		

Schließt einen QBIC-Katalog.

Autorisierung

Keine

Befehlssyntax

Befehlsparameter

Keine

Beispiele

Schließen des aktuellen Katalogs:

```
close qbic catalog
```

Hinweise

Der QBIC-Katalog muss geöffnet sein.

CONNECT

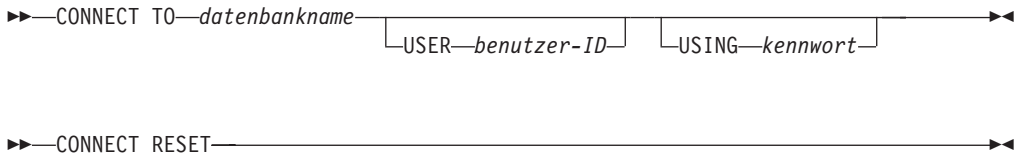
Image	Audio	Video
X	X	X

Stellt eine Verbindung zu einer Datenbank her. Für die Extender ist eine unabhängige Verbindung, getrennt von der DB2-Verbindung, zur Datenbank erforderlich.

Autorisierung

CONNECT

Befehlssyntax



Befehlsparameter

datenbankname
Der Name der Datenbank.

benutzer-ID
Die Benutzer-ID, die berechtigt ist, eine Verbindung zur Datenbank herzustellen.

kennwort
Das Kennwort für die Benutzer-ID.

RESET
Unterbricht die Verbindung zur Datenbank, nachdem anstehende Änderungen festgeschrieben wurden.

Beispiele

Herstellen einer Verbindung zur Datenbank PERSONNL. Die Benutzer-ID ist 'anita' und das Kennwort ist 'anitapas':

```
connect to personnl user anita
   using anitapas
```

Hinweise

Die Verbindung zur Datenbank wird im Zugriffsmodus (SHARE) hergestellt.

Führen Sie diesen Befehl aus, bevor Sie andere Extender-Befehle ausführen.

CREATE QBIC CATALOG

Image	Audio	Video
X		

Erstellt einen QBIC-Katalog in der aktuellen Datenbank für die angegebene DB2IMAGE-Spalte. Der Extender generiert den Katalognamen automatisch.

Autorisierung

ALTER, CONTROL, SYSADM, DBADM

Befehlssyntax

Befehlsparameter

tabellenname
Der Name der für DB2IMAGE aktivierten Tabelle.

spaltenname
Der Name der für DB2IMAGE aktivierten Spalte.

OFF Abbilder werden manuell katalogisiert.

ON Abbilder werden automatisch katalogisiert.

bereichsname
Die Tabellenbereichsangabe und die Indexoptionen für den QBIC-Katalog. Die Angabe besteht aus vier Teilen:

- Der Name des Tabellenbereichs für die Katalogtabellen, die Merkmaldaten enthalten. Der Tabellenbereich muss angegeben werden. Der Tabellenbereich sollte ein segmentierter Tabellenbereich sein.
- Für den Index, der für die Katalogtabellen erstellt wurde, eine beliebige Kombination aus Using-Block, freiem Block, Gbpcache-Block oder Indexoptionen für nicht partitionierte Indizes vom Typ 2. Diese Angabe ist wahlfrei. Die Standardwerte werden verwendet, wenn Sie diesen Teil nicht angeben.
- Der Name des Tabellenbereichs für die Katalogprotokolltabelle. Der Tabellenbereich kann ein einfacher Tabellenbereich oder ein segmentierter Tabellenbereich sein. Diese Angabe ist wahlfrei. Wenn Sie keinen Tabellenbereich für die Protokolltabelle angeben, wird der für die Merkmaldatentabellen angegebene Tabellenbereich verwendet.
- Für den Index, der für die Protokolldatentabelle erstellt wurde, eine beliebige Kombination aus Using-Block, freiem Block, Gbpcache-Block oder Indexoptionen für nicht partitionierte Indizes vom Typ 2. Diese Angabe ist wahlfrei. Die Standardwerte werden verwendet, wenn Sie diesen Teil nicht angeben.

Beispiele

Erstellen eines QBIC-Katalogs für die Spalte 'picture' in der Tabelle 'employee', wobei das automatische Katalogisieren eingeschaltet (ON) wird:

```
create qbic catalog employee picture on
```

Hinweise

Wird ON angegeben, werden die Abbilder, die in die Spalte importiert werden, automatisch im zugeordneten QBIC-Katalog katalogisiert. Der Standardwert ist OFF.

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

DELETE QBIC CATALOG

Image	Audio	Video
X		

Löscht einen QBIC-Katalog, einschließlich aller Unterstützungsdaten für die QBIC-Suche.

Autorisierung

ALTER, CONTROL, SYSADM, DBADM

Befehlssyntax

Befehlsparameter

tabellenname
Der Name der für DB2IMAGE aktivierten Tabelle.

spaltenname
Der Name der für DB2IMAGE aktivierten Spalte.

Beispiele

Löschen des Katalogs, der der Spalte 'picture' in der Tabelle 'employee' zugeordnet ist:

```
delete qbic catalog employee picture
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

DISABLE COLUMN

Image	Audio	Video
X	X	X

Inaktiviert die angegebene Spalte, so dass sie keine angegebenen Multimediadaten mehr speichern kann.

Autorisierung

SYSADM, DBADM, CONTROL, ALTER

Befehlssyntax

Befehlsparameter

tabellenname
Der Name der Tabelle in der aktuellen Datenbank.

spaltenname
Der Name der Spalte, die inaktiviert werden soll.

extender-name
Der Name des Extenders, für den die Spalte inaktiviert werden soll. Gültige Extender-Namen sind db2image, db2audio und db2video.

Beispiele

Inaktivieren der Spalte 'photo' in der Tabelle 'employee', so dass sie keine Abbilddaten speichern kann:

```
disable column employee photo for db2image
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

Wenn Sie eine Spalte inaktivieren,

- kann die Spalte keine Daten für den angegebenen Extender speichern. Dies wirkt sich nicht darauf aus, ob andere Spalten in der Tabelle für Multimediadatentypen aktiviert oder inaktiviert sind.
- wird der Inhalt von Spalteneinträge auf NULL gesetzt, und die entsprechenden Zeilen in den Verwaltungstabellen werden gelöscht.
- werden die Auslöser, die der Spalte zugeordnet sind, gelöscht.

DISABLE DATABASE

Image	Audio	Video
X	X	X

Inaktiviert die aktuelle Datenbank, so dass sie keine Multimediadaten mehr speichern kann.

Autorisierung

SYSADM, DBADM

Befehlssyntax

Befehlsparameter

extender-name
Der Name des Extenders, für den die aktuelle Datenbank inaktiviert werden soll. Gültige Extender-Namen sind db2image, db2audio und db2video.

Beispiele

Inaktivieren der aktuellen Datenbank, so dass sie keine Abbilddaten speichern kann:

```
disable database for db2image
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

Wenn Sie eine Datenbank inaktivieren,

- inaktiviert das System alle Tabellen, die nur für den angegebenen Extender aktiviert sind.
- löscht das System die Tabellen zur Verwaltungsunterstützung für UDFs für den angegebenen Extender.

DISABLE TABLE

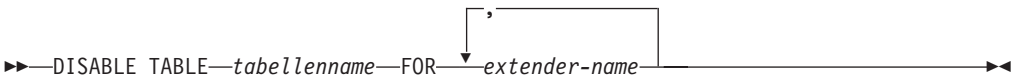
Image	Audio	Video
X	X	X

Inaktiviert die angegebene Tabellen, so dass sie keine Multimediadaten mehr speichern kann.

Autorisierung

SYSADM, DBADM, CONTROL, ALTER

Befehlssyntax



Befehlsparameter

tabellenname
Der Name der Tabelle in der aktuellen Datenbank, die inaktiviert werden soll.

extender-name
Der Name des Extenders, für den die Tabelle inaktiviert werden soll. Gültige Extender-Namen sind db2image, db2audio und db2video.

Beispiele

Inaktivieren der Tabelle 'employee', so dass sie keine Abbilddaten speichern kann:

```
disable table employee for db2image
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor Sie diesen Befehl verwenden.

Wenn Sie eine Tabelle inaktivieren,

- inaktiviert das System alle Spalten in der Tabelle, die für den angegebenen Extender aktiviert sind.
- Löschen Sie die Tabellen zur Verwaltungsunterstützung, zu der Tabelle zugeordnet sind.

DISCONNECT SERVER AT NODENUM (nur EEE)

Image	Audio	Video
X	X	X

Unterbricht die Verbindung vom Server zum angegebenen Knoten auf allen Datenbanken.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT, DBADM

Befehlssyntax

Befehlsparameter

knotennummer
Der Knoten, dessen Verbindung zum Server unterbrochen werden soll.

Beispiele

Unterbrechen der Verbindung des Servers zu allen Datenbanken am Knoten mit der Nummer 2:

```
disconnect server at nodenum 2
```

Hinweise

Um die Verbindung zwischen dem Server und allen Datenbanken auf allen Knoten zu unterbrechen, verwenden Sie den Befehl DMBSTOP.

DISCONNECT SERVER FOR DATABASE (nur EEE)

Image	Audio	Video
X	X	X

Unterbricht die Verbindung vom Server zu allen Knoten der angegebenen Datenbank.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT, DBADM

Befehlssyntax

Befehlsparameter

datenbankname
Die Datenbank, deren Verbindung zum Server unterbrochen werden soll.

Beispiele

Unterbrechen der Verbindung des Servers zur Datenbank MY_DATABASE:

```
disconnect server for database my_database
```

Hinweise

Um die Verbindung zwischen dem Server und allen Datenbanken auf allen Knoten zu unterbrechen, verwenden Sie den Befehl DMBSTOP.

DISCONNECT SERVER FOR DATABASE AT NODENUM (nur EEE)

Image	Audio	Video
X	X	X

Unterbricht die Verbindung vom Server zur angegebenen Datenbank auf dem angegebenen Knoten.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT, DBADM

Befehlssyntax

Befehlsparameter

datenbankname
Die Datenbank, deren Verbindung zum Server unterbrochen werden soll.

knotennummer
Der Knoten, dessen Verbindung zum Server unterbrochen werden soll.

Beispiele

Unterbrechen der Verbindung des Servers zur Datenbank MY_DATABASE am Knoten mit der Nummer 2:

```
disconnect server for database my_database at nodenum 2
```

Hinweise

Um die Verbindung zwischen dem Server und allen Datenbanken auf allen Knoten zu unterbrechen, verwenden Sie den Befehl DMBSTOP.

ENABLE COLUMN

Image	Audio	Video
X	X	X

Aktiviert die angegebene Spalte, so dass sie Multimediadaten speichern kann.

Autorisierung

SYSADM, DBADM, CONTROL, ALTER

Befehlssyntax

Befehlsparameter

tabellenname
Der Name der Tabelle in der aktuellen Datenbank.

spaltenname
Der Name der Spalte, die aktiviert werden soll.

extender-name
Der Name des Extenders, für den die Tabelle aktiviert werden soll. Gültige Extender-Namen sind db2image, db2audio und db2video.

Beispiele

Aktivieren der Spalte 'photo' in der Tabelle 'employee', so dass sie Abbilddaten speichern kann:

```
enable column employee photo for db2image
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor Sie diesen Befehl verwenden.

ENABLE DATABASE

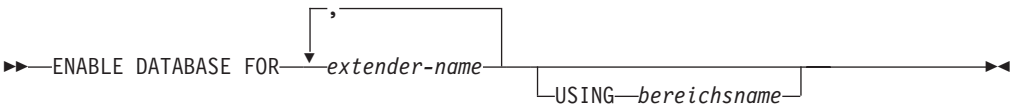
Image	Audio	Video
X	X	X

Aktiviert die aktuelle Datenbank, so dass sie Multimediadaten unter Verwendung des angegebenen Tabellenbereichs speichern kann.

Autorisierung

SYSADM, SYSCTRL, DBADM

Befehlssyntax



Befehlsparameter

extender-name
Der Name des Extenders, für den die aktuelle Datenbank aktiviert werden soll. Gültige Extender-Namen sind db2image, db2audio und db2video.

bereichsname
Der Name des Tabellenbereichs, bei dem es sich um eine Gruppe von Behältern handelt, in denen Verwaltungstabellen gespeichert werden. Die Angabe zum Tabellenbereichsnamen besteht aus den folgenden drei Teilen: *datats, indexts, longts*. Hierbei ist *datats* der Name des Tabellenbereichs, in dem Metadatentabellen erstellt werden. Die Variable *indexts* steht für den Namen des Tabellenbereichs, in dem Indizes für die Metadatentabellen erstellt werden und *longts* gibt den Namen des Tabellenbereichs an, in dem Werte für lange Spalten in der Metadatentabelle gespeichert werden. (Hierzu gehören z. B. die Spalten für die Datentypen LONG VARCHAR und LOB.) Wenn Sie für einen Teil des Namens des Tabellenbereichs einen Nullwert angeben, wird der Name des Standardtabellenbereichs für diesen Teil verwendet. Der angegebene Tabellenbereich sollte in einer Knotengruppe definiert sein, die alle Knoten im partitionierten Datenbanksystem umfasst.

Beispiele

Aktivieren der aktuellen Datenbank, so dass sie Abbilddaten speichern kann:

```
enable database for db2image using mydataspace, myindxspace, mylongspace
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

Ist der Tabellenbereich nicht angegeben, verwendet das System den Tabellenbereich USERSPACE1 für die Verwaltungstabellen.

ENABLE TABLE

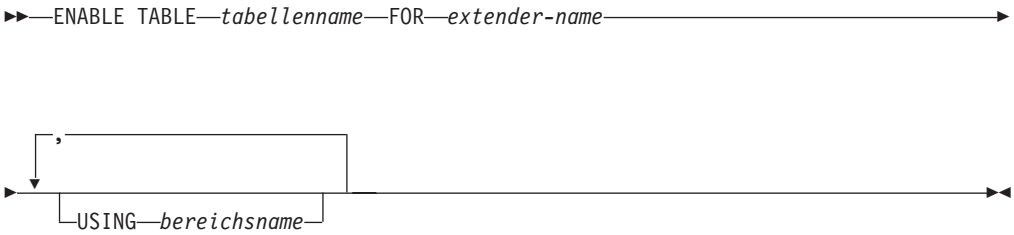
Image	Audio	Video
X	X	X

Aktiviert die angegebene Tabelle, so dass sie Multimediadaten unter Verwendung des angegebenen Tabellenbereichs speichern kann.

Autorisierung

SYSADM, DBADM, CONTROL, ALTER

Befehlssyntax



Befehlsparameter

tabellenname
Der Name der Tabelle, die in der aktuellen Datenbank aktiviert werden soll.

extender-name
Der Name des Extenders, für den die Tabelle aktiviert werden soll. Gültige Extender-Namen sind db2image, db2audio und db2video.

bereichsname
Der Name des Tabellenbereichs, bei dem es sich um eine Gruppe von Behältern handelt, in denen Verwaltungstabellen gespeichert werden.Die Angabe zum Tabellenbereich besteht aus den folgenden drei Teilen: *datats, indexts, longts*. Dabei ist *datats* der Tabellenbereich, in dem Metadatentabellen erstellt werden, *indexts* ist der Tabellenbereich, in dem Indizes für die Metadatentabellen erstellt werden, und *longts* ist der Tabellenbereich, in dem die Werte von langen Spalten in Metadatentabellen (z. B. Spalten, die die Datentypen LONG VARCHAR und LOB enthalten) gespeichert werden. Wenn Sie für einen Teil der Angabe zum Tabellenbereich einen Nullwert angeben, wird der Standardtabellenbereich für diesen Teil verwendet.

Wenn Sie für einen Teil der Angabe zum Tabellenbereich einen Nullwert angeben, wird der Standardtabellenbereich für diesen Teil verwendet.

Nur EEE: Der angegebene Tabellenbereich sollte sich in derselben Knotengruppe befinden wie die Benutzertabelle.

Beispiele

Aktivieren der Tabelle 'employee', so dass sie Abbilddaten speichern kann:

```
enable table employee for db2image
   using mydataspace, myindxspace, mylongspace
```

Aktivieren der Tabelle 'employee', so dass sie Abbilddaten speichern kann. Verwenden der Verwenden der Standardtabellenbereiche:

```
enable table employee for db2image
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor Sie diesen Befehl verwenden.

Ist der Tabellenbereich nicht angegeben, verwendet das System den Tabellenbereich, der bei der Aktivierung der aktuellen Datenbank definiert wurde.

GET EXTENDER STATUS

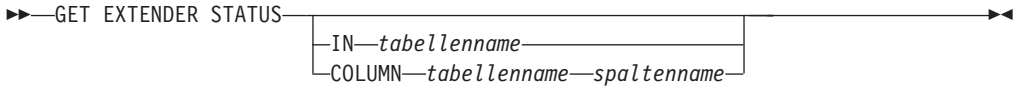
Image	Audio	Video
X	X	X

Zeigt die Namen der Extender (falls vorhanden) an, für die eine Spalte, Tabelle oder die aktuelle Datenbank aktiviert ist.

Autorisierung

Keine

Befehlssyntax



Befehlsparameter

tabellenname
Der Name der Tabelle in der aktuellen Datenbank.

spaltenname
Der Name der Spalte.

Beispiele

Anzeigen der Namen von aktivierten Extendern in der Datenbank:

```
get extender status
```

Anzeigen des Status der Tabelle 'employee':

```
get extender status in employee
```

Anzeigen des Status der Spalte 'address' in der Tabelle 'employee':

```
get extender status column employee address
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor Sie diesen Befehl verwenden.

GET INACCESSIBLE FILES

Image	Audio	Video
X	X	X

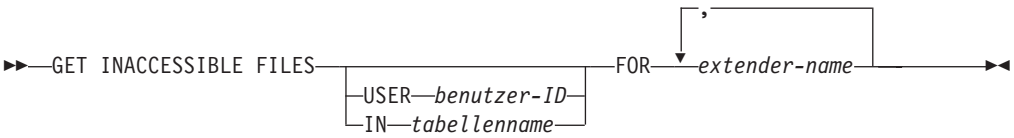
Listet alle Multimediadateien auf, die nicht zugänglich sind und auf die durch eine Tabelle, durch Tabellen mit einem angegebenen Qualifikationsmerkmal oder durch alle Tabellen in der aktuellen Datenbank verwiesen wird.

Autorisierung

Für alle Tabellen in der aktuellen Datenbank, d. h. wenn nicht USER oder IN angegeben ist: SYSADM, SYSCTRL, SYSMAINT, DBADM

Für eine bestimmte Tabelle (wenn IN angegeben ist) oder für Tabellen, die zu einem Qualifikationsmerkmal gehören (wenn USER angegeben ist): SELECT

Befehlssyntax



Befehlsparameter

benutzer-ID
Das Qualifikationsmerkmal der Tabellen in der aktuellen Datenbank, deren nicht zugängliche Dateien aufgelistet werden sollen.

tabellenname
Der Name der Tabelle in der aktuellen Datenbank, deren nicht zugängliche Dateien aufgelistet werden sollen.

extender-name
Der Name des Extenders. Gültige Extender-Namen sind db2image, db2audio und db2video.

Beispiele

Auflisten aller Abbilddateien, auf die durch Tabellen in der Datenbank verwiesen wird, die aber nicht zugänglich sind:

```
get inaccessible files
   for db2image
```

Auflisten aller Abbilddateien, auf die in Tabellen mit dem Qualifikationsmerkmal anita verwiesen wird, die aber nicht zugänglich sind:

```
get inaccessible files
   user anita for db2image
```

Auflisten aller Abbilddateien, auf die durch Einträge in der Tabelle 'employee' verwiesen wird, die aber nicht zugänglich sind:

```
get inaccessible files
   in employee FOR db2image
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor Sie diesen Befehl verwenden.

Wenn Sie eine Tabelle angeben, listet der Befehl die nicht zugänglichen Datei für diese Tabelle auf. Wenn Sie ein Qualifikationsmerkmal angeben, listet der Befehl die nicht zugänglichen Dateien nur für die Tabellen mit diesem Qualifikationsmerkmal auf. Wenn Sie keine Tabelle angeben, listet der Befehl die nicht zugänglichen Dateien für alle Tabellen in der aktuellen Datenbank auf.

GET QBIC CATALOG INFO

Image	Audio	Video
X		

Gibt die folgenden Informationen zum momentan geöffneten Katalog zurück:

- Den Namen der Benutzertabelle und Abbildspalte, denen der Katalog zugeordnet ist.
- Die Namen der Merkmale im Katalog.
- Die Anzahl an Merkmalen im Katalog.
- Den Status der automatischen Analyse.

Autorisierung

SELECT, CONTROL, SYSADM, DBADM

Befehlssyntax

Befehlsparameter

Keine

Beispiele

Abrufen von Informationen zum momentan geöffneten QBIC-Katalog:

```
get qbic catalog info
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

Der Katalog muss geöffnet sein.

GET REFERENCED FILES

Image	Audio	Video
X	X	X

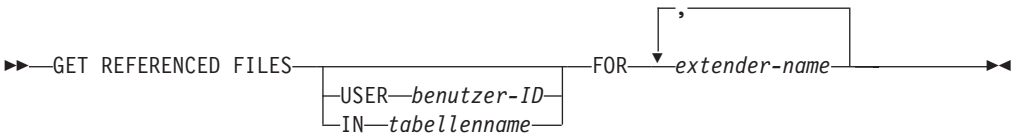
Listet alle Multimediadateien auf und die Spaltennamen, die in einer Tabelle, in Tabellen mit einem angegebenen Qualifikationsmerkmal oder in allen Tabellen in der aktuellen Datenbank auf diese Dateien verweisen.

Autorisierung

Für alle Tabellen in der aktuellen Datenbank, d. h. wenn nicht USER oder IN angegeben ist: SYSADM, SYSCTRL, SYSMAINT, DBADM

Für eine bestimmte Tabelle (wenn IN angegeben ist) oder für Tabellen, die zu einem Qualifikationsmerkmal gehören (wenn USER angegeben ist): SELECT

Befehlssyntax



Befehlsparameter

benutzer-ID
Das Qualifikationsmerkmal der Tabellen in der Datenbank, in denen sich Verweise auf die aufzulistenden Dateien befinden. Der Befehl durchsucht nur Tabellen mit diesem Qualifikationsmerkmal.

tabellenname
Der Name der Tabelle in der aktuellen Datenbank, in der sich Verweise auf die aufzulistenden Dateien befinden. Der Befehl durchsucht nur diese Tabelle.

extender-name
Der Name des Extenders. Gültige Extender-Namen sind db2image, db2audio und db2video.

Beispiele

Auflisten aller Abbilddateien, auf die durch Tabelleneinträge in allen Tabellen in der Datenbank verwiesen wird:

```
get referenced files
   for db2image
```

Auflisten aller Abbilddateien, auf die durch Einträge in Tabellen mit dem Qualifikationsmerkmal 'anita' verwiesen wird:

```
get referenced files
   user anita for db2image
```

Auflisten aller Abbilddateien, auf die durch Einträge in der Tabelle 'employee' verwiesen wird:

```
get referenced files
   in employee for db2image
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor Sie diesen Befehl verwenden.

Wenn Sie keinen Parameter angeben, durchsucht der Befehl alle Tabellen in der Datenbank.

GET SERVER STATUS

Image	Audio	Video
X	X	X

Zeigt den Status des Extender-Servers für die aktuelle Datenbank bzw. für alle Datenbanken an.

Nur EEE: Wenn ein Knoten angegeben ist, zeigt der Befehl den Status des Extender-Servers - für die aktuelle Datenbank bzw. für alle Datenbanken - nur an diesem Knoten an.

Autorisierung

Keine

Befehlssyntax

Befehlsparameter

ALL Zeigt den Status aller Datenbanken an.

knotennummer
Die Nummer des Knotens. Der Befehl zeigt nur den Status dieses Knotens an. **(Nur EEE)**

Beispiele

Anzeigen des Status des Extender-Servers für die aktuelle Datenbank:

```
get server status
```

Anzeigen des Status des Extender-Servers für alle Datenbanken:

```
get server status all
```

Anzeigen des Status des Extender-Servers für den Knoten mit der Nummer 2 für alle Datenbanken:

```
get server status all nodenum 2
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

Wenn Sie keinen Parameter angeben, zeigt der Befehl den Status aller Knoten, die in der Datei db2nodes.cfg aufgelistet sind, für die aktuelle Datenbank an.

OPEN QBIC CATALOG

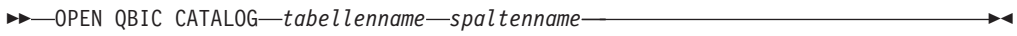
Image	Audio	Video
X		

Öffnet den Katalog für die angegebene DB2IMAGE-Spalte. Die Datenbank versucht immer, den Katalog im Aktualisierungsmodus zu öffnen. Ist der Katalog bereits im Aktualisierungsmodus geöffnet, wird er im Lesemodus geöffnet.

Autorisierung

CONNECT

Befehlssyntax



Befehlsparameter

tabellenname
Der Name der für DB2IMAGE aktivierten Tabelle.

spaltenname
Der Name der für DB2IMAGE aktivierten Spalte.

Beispiele

Öffnen des QBIC-Katalog für die Spalte 'picture' in der Tabelle 'employee':

```
open qbic catalog employee picture
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

Dieser Befehl führt dazu, dass jeder geöffnete Katalog geschlossen wird.

QUIT

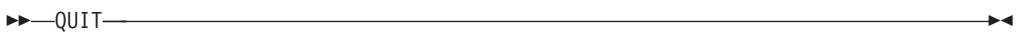
Image	Audio	Video
X	X	X

Schaltet den db2ext-Befehlszeilenprozessor für die Befehlseingabe im interaktiven Modus ab. Die Verbindung zu DB2 bleibt bestehen, so dass Sie noch immer Befehle im db2ext-Befehlszeilenprozessor im Befehlsmodus eingeben können.

Autorisierung

Keine

Befehlssyntax



Befehlsparameter

Keine

Beispiele

Abschalten der Befehlszeilenschnittstelle für den interaktiven Modus:

```
quit
```

Hinweise

Bei dem Befehl QUIT bleibt die Verbindung zur Datenbank bestehen.

RECONNECT SERVER AT NODENUM (nur EEE)

Image	Audio	Video
X	X	X

Stellt die Verbindung vom Server zum angegebenen Knoten auf allen Datenbanken wieder her.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT, DBADM

Befehlssyntax

Befehlsparameter

knotennummer
Der Knoten, dessen Verbindung zum Server wiederhergestellt werden soll.

Beispiele

Wiederherstellen der Verbindung des Servers mit allen Datenbanken am Knoten mit der Nummer 2:

```
reconnect server at nodenum 2
```

Hinweise

Um die Verbindung zwischen dem Server und allen Datenbanken auf allen Knoten wiederherzustellen, verwenden Sie den Befehl DMBSTART.

RECONNECT SERVER FOR DATABASE (nur EEE)

Image	Audio	Video
X	X	X

Stellt die Verbindung vom Server zu allen Knoten der angegebenen Datenbank wieder her.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT, DBADM

Befehlssyntax

Befehlsparameter

datenbankname
Die Datenbank, deren Verbindung zum Server wiederhergestellt werden soll.

Beispiele

Wiederherstellen der Verbindung des Servers zur Datenbank MY_DATABASE:

```
disconnect server for database my_database
```

Hinweise

Um die Verbindung zwischen dem Server und allen Datenbanken auf allen Knoten wiederherzustellen, verwenden Sie den Befehl DMBSTART.

RECONNECT SERVER FOR DATABASE AT NODENUM (nur EEE)

Image	Audio	Video
X	X	X

Stellt die Verbindung vom Server zur angegebenen Datenbank auf dem angegebenen Knoten wieder her.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT, DBADM

Befehlssyntax

Befehlsparameter

datenbankname
Die Datenbank, deren Verbindung zum Server wiederhergestellt werden soll.

knotennummer
Der Knoten, dessen Verbindung zum Server wiederhergestellt werden soll.

Beispiele

Wiederherstellen der Verbindung des Servers zur Datenbank MY_DATABASE am Knoten mit der Nummer 2:

```
reconnect server for database my_database at nodenum 2
```

Hinweise

Um die Verbindung zwischen dem Server und allen Datenbanken auf allen Knoten wiederherzustellen, verwenden Sie den Befehl DMBSTART.

REDISTRIBUTE NODEGROUP (nur EEE)

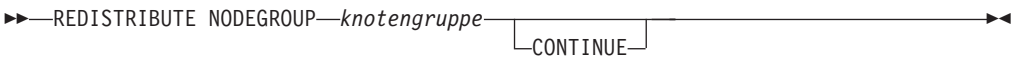
Image	Audio	Video
X		

Verteilt die Extender-Daten neu, wenn ein Knoten zu einer Knotengruppe hinzugefügt oder aus einer Knotengruppe gelöscht wird, oder wenn eine neue Partitionszuordnung für eine Knotengruppe erstellt wird.

Autorisierung

SYSADM, DBADM

Befehlssyntax



Befehlsparameter

Knotengruppe
Der Name der Knotengruppe, die neu verteilt werden soll.

CONTINUE
Wenn der Neuverteilungsprozess einen Fehler zurückgibt, können Sie den Befehl mit oder ohne den Parameter CONTINUE erneut ausführen, je nachdem, welche Anweisungen in der Antwort auf den Befehl angegeben werden. Mit dieser Option wird das System angewiesen, an der Stelle fortzufahren, an der es gestoppt wurde, und nicht am Anfang erneut zu starten.

Beispiele

Neuverteilen der Knotengruppe 'my_nodegroup':

```
redistribute nodegroup my_nodegroup
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

Der Parameter CONTINUE sollte nicht verwendet werden, wenn der Befehl REDISTRIBUTE NODEGROUP das erste Mal nach dem DB2-Befehl REDISTRIBUTE verwendet wird. Wird der Parameter verwendet, wird ein Fehler protokolliert und die Neuverteilung beginnt am Anfang.

Um die Datenintegrität zu gewähren, müssen die einzelnen Knotengruppen nacheinander neu verteilt werden. Warten Sie, bis die Neuverteilung für eine Knotengruppe beendet ist, bevor Sie die nächste starten.

Wenn der Befehl REDISTRIBUTE NODEGROUP fehlschlägt, finden Sie eine genaue Erläuterung in der Datei "redist.log" in einem der folgenden Verzeichnisse:

- **UNIX:** /<home-exemplar>/dmb/redist
- **Windows:** \\<instance_owning_machine>\DB2<instance_name>\<exemplarname>\dmb\redist

REMOVE QBIC FEATURE

Image	Audio	Video
X		

Löscht die Merkmaltabelle des angegebenen Merkmals aus dem geöffneten Katalog.

Autorisierung

ALTER, CONTROL, SYSADM, DBADM

Befehlssyntax

Befehlsparameter

merkmalname

Der Name des Merkmals, das aus dem QBIC-Katalog gelöscht wird. Die folgenden Merkmale werden mit dem Image Extender zur Verfügung gestellt:

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Beispiele

Löschen des Merkmals QbColorFeatureClass aus dem momentan geöffneten Katalog:

```
remove qbic feature qbcolorfeatureclass
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

Der Katalog muss geöffnet sein.

REORG

Image	Audio	Video
X	X	X

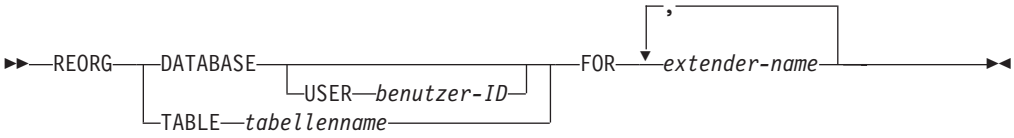
Bereinigt die Verwaltungstabellen (Verwaltungstabelle und Attributtabelle), die einer bestimmten Tabellen, Tabellen mit einem bestimmten Qualifikationsmerkmal oder allen Tabellen in der aktuellen Datenbank zugeordnet sind.

Autorisierung

Für eine bestimmte Tabelle (wenn REORG TABLE ausgeführt wird) oder für Tabellen mit einem bestimmten Qualifikationsmerkmal (wenn REORG DATABASE ausgeführt wird): SYSADM, SYSCTRL, SYSMAINT, DBADM, CONTROL

Für alle Tabellen in der Datenbank (wenn REORG DATABASE ausgeführt wird): SYSADM, SYSCTRL, SYSMAINT, DBADM

Befehlssyntax



Befehlsparameter

benutzer-ID
Das Qualifikationsmerkmal der Tabellen.

tabellenname
Der Name der Tabelle in der aktuellen Datenbank, deren Verwaltungstabellen bereinigt werden sollen.

extender-name
Der Name des Extenders. Gültige Extender-Namen sind db2image, db2audio und db2video.

Beispiele

Reorganisieren und Bereinigen der Verwaltungstabellen für Abbilder für die aktuelle Datenbank:

```
reorg database for db2image
```

Reorganisieren und Bereinigen der Verwaltungstabellen für Abbilder in allen Tabellen mit dem Qualifikationsmerkmal 'anita':

```
reorg database user anita for db2image
```

Reorganisieren und Bereinigen der Verwaltungstabellen für Abbilder für die Tabelle 'employee':

```
reorg table employee for db2image
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

SET QBIC AUTOCATALOG

Image	Audio	Video
X		

Katalogisiert automatisch Abbilder, wenn sie in eine Spalte importiert werden. Die Abbilder werden zum QBIC-Katalog hinzugefügt, der der Spalte zugeordnet ist.

Autorisierung

ALTER, CONTROL, SYSADM, DBADM

Befehlssyntax

Befehlsparameter

Keine

Beispiele

Einschalten des automatischen Katalogisierens:

```
set qbic autocatalog on
```

Hinweise

Der QBIC-Katalog muss geöffnet sein.

START SERVER (nur Nicht-EEE)

Image	Audio	Video
X	X	X

Startet den Extender-Server für die aktuelle Datenbank.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT, DBADM

Befehlssyntax

Befehlsparameter

Keine

Beispiele

Starten des Extender-Servers für die aktuelle Datenbank:

```
start server
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

STOP SERVER (nur Nicht-EEE)

Image	Audio	Video
X	X	X

Stoppt den Extender-Server für die aktuelle Datenbank.

Autorisierung

SYSADM, SYSCTRL, SYSMAINT, DBADM

Befehlssyntax

Befehlsparameter

Keine

Beispiele

Stoppen des Extender-Servers für die aktuelle Datenbank:

```
stop server
```

Hinweise

Stellen Sie eine Verbindung zur Datenbank her, bevor dieser Befehl verwendet wird.

TERMINATE

Image	Audio	Video
X	X	X

Schaltet den db2ext-Befehlszeilenprozessor ab und unterbricht die Verbindung zu DB2.

Autorisierung

Keine

Befehlssyntax

Befehlsparameter

Keine

Beispiele

Abschalten des db2ext-Befehlszeilenprozessors:

```
quit
```

Hinweise

Mit dem Befehl TERMINATE wird die Verbindung zur Datenbank unterbrochen.

Kapitel 16. Diagnoseinformationen

Alle eingebetteten SQL-Anweisungen und DB2 CLI-Aufrufe in Ihrem Programm, einschließlich derer, die benutzerdefinierte DB2 Extender-Funktionen aufrufen, generieren Codes, die angeben, ob die eingebettete SQL-Anweisung oder der DB2 CLI-Aufruf erfolgreich ausgeführt wurde. Andere DB2 Extender-APIs, wie z. B. Verwaltungs-APIs, geben auch Code zurück, der angibt, ob die API erfolgreich oder nicht erfolgreich ausgeführt wurde. Ihr Programm sollte diese Rückkehrcodes überprüfen und beantworten.

Ihr Programm sollte außerdem Informationen zur Ergänzung dieser Codes abrufen. Dazu gehören SQLSTATE-Informationen und Fehlernachrichten. Sie können diese Diagnoseinformationen verwenden, um Fehler in Ihrem Programmeditor zu isolieren und zu berichtigen.

In manchen Fällen kann die Ursache eines Fehlers nicht sofort gefunden werden. In diesen Fällen ist es möglicherweise erforderlich, dass Sie dem Kundendienst Informationen zur Verfügung stellen, so dass dieser den Fehler isoliert und berichtigt. Die DB2 Extender umfassen eine Tracefunktion, mit der die Extender-Aktivitäten aufgezeichnet werden können. Die Trace-Informationen können wertvoll für den Kundendienst sein. Sie sollten die Tracefunktion nur unter Anleitung des IBM Kundendienstes verwenden.

In diesem Kapitel wird beschrieben, wie Sie auf diese Diagnoseinformationen zugreifen können. Folgende Punkte werden erläutert:

- Bearbeitung der UDF-Rückkehrcodes und der API-Rückkehrcodes der DB2 Extender
- Steuerung des Trace

Außerdem werden die SQLSTATEs und Fehlernachrichten aufgelistet und beschrieben, die von den Extendern zurückgegeben werden können.

UDF-Rückkehrcodes bearbeiten

Eingebettete SQL-Anweisungen geben Rückkehrcodes in den Feldern SQLCODE, SQLWARN und SQLSTATE der SQLCA-Struktur zurück. Diese Struktur ist in einer SQLCA-Kopfdatei definiert. (Weitere Informationen zur SQLCA-Struktur und zur SQLCA-Kopfdatei befinden sich im Handbuch *DB2 Application Development Guide*.)

DB2 CLI-Aufrufe geben SQLCODE- und SQLSTATE-Werte zurück, die Sie mit Hilfe der Funktion SQLError abrufen können. (Weitere Informationen zum Abrufen von Fehlerinformationen mit Hilfe der Funktion SQLError befinden sich im Handbuch *CLI Guide and Reference*.)

Ein SQLCODE-Wert 0 bedeutet, dass die Anweisung erfolgreich ausgeführt wurde (mit möglichen Warnungsbedingungen). Ein positiver SQLCODE-Wert bedeutet, dass die Anweisung erfolgreich, aber mit einer Warnung ausgeführt wurde. (Eingebettete SQL-Anweisungen geben die Warnung, die einem SQLCODE-Wert 0 oder einem positiven SQLCODE-Wert zugeordnet ist, im Feld SQLWARN zurück.) Ein negativer SQLCODE-Wert bedeutet, dass eine Fehlerbedingung aufgetreten ist.

UDF-Rückkehrcodes bearbeiten

DB2 ordnet jedem SQLCODE-Wert eine Nachricht zu. Wenn eine DB2 Extender-UDF auf eine Warnung oder Fehlerbedingung trifft, übergibt sie die zugehörigen Informationen zum Einfügen in die SQLCODE-Nachricht an DB2.

Die Werte SQLSTATE enthalten Code, der die SQLCODE-Nachrichten ergänzt. Eine Beschreibung der einzelnen SQLSTATE-Codes, die durch die DB2 Extender zurückgegeben werden, befindet sich im Abschnitt „SQLSTATE-Codes" auf Seite 489.

Eingebettete SQL-Anweisungen und DB2 CLI-Aufrufe, die DB2 Extender-UDFs aufrufen, geben möglicherweise SQLCODE-Nachrichten und SQLSTATE-Werte zurück, die nur für diese UDFs gelten. DB2 gibt diese Werte auf die gleiche Weise zurück wie für andere eingebettete SQL-Anweisungen oder andere DB2 CLI-Aufrufe. Somit können Sie auf diese Werte genauso zugreifen wie auf Werte für eingebettete SQL-Anweisungen oder DB2 CLI-Aufrufe, die keine DB2 Extender-UDFs starten.

Informationen zu den SQLSTATE-Werten und der Nachrichtennummer der zugeordneten Nachricht, die von den Extendern zurückgegeben werden können, befinden sich im Abschnitt „SQLSTATE-Codes" auf Seite 489. Informationen zu den einzelnen Nachrichten befinden sich im Abschnitt „Nachrichten" auf Seite 493.

API-Rückkehrcodes bearbeiten

Jeder Aufruf einer DB2 Extender-API gibt einen Code zurück. Der Rückkehrcode 0 gibt an, dass der API-Aufruf erfolgreich verarbeitet wurde. Ein Rückkehrcode ungleich 0 gibt an, dass der API-Aufruf erfolgreich verarbeitet wurde, aber eine Warnungsbedingung aufgetreten ist, oder dass der Aufruf auf Grund einer Fehlerbedingung nicht erfolgreich verarbeitet werden konnte.

Im Kapitel 14, „Anwendungsprogrammierschnittstellen", auf Seite 253 werden die symbolischen Werte für die einzelnen Codes, die von den DB2 Extender-APIs zurückgegeben werden können, aufgelistet und beschrieben.

Sie können zusätzliche Informationen zu den Fehlern, auf die eine API trifft, abrufen. Verwenden Sie die API DB*x*GetError, um diese zusätzlichen Informationen abzurufen, wobei Sie für *x* ein 'a' für den Audio Extender, ein 'i' für den Image Extender und ein 'v' für den Video Extender einsetzen. Die API DB*x*GetError gibt den SQL-Fehlercode und die zugeordnete Nachricht für die letzte DB2 Extender-API zurück, bei der ein Fehler aufgetreten ist. Informationen zu SQL-Fehlercodes befinden sich im Handbuch *DB2 Fehlernachrichten*. Informationen zu den einzelnen Nachrichten, die von der API DB*x*GetError zurückgegeben werden können, befinden sich im Abschnitt „Nachrichten" auf Seite 493.

Beispielsweise aktivieren die folgenden Anweisungen in einem C-Anwendungsprogramm eine Tabelle für den Audio Extender und suchen danach nach Fehlern:

```
rc=DBaEnableTable((char *)NULL, "employee");

rc=DBaGetError(&errCode, &errMsg);
```

SQLSTATE-Codes

Tabelle 16 werden die SQLSTATE-Werte, die von den DB2 Extendern zurückgegeben werden können, aufgelistet und beschrieben. Die Beschreibung eines SQLSTATE-Wertes umfasst auch dessen symbolische Darstellung. In der Tabelle wird außerdem die Nachrichtennummer aufgelistet, die den einzelnen SQLSTATE-Werten zugeordnet ist. Informationen zu den einzelnen Nachrichten befinden sich im Abschnitt „Nachrichten" auf Seite 493.

Tabelle 16. SQLSTATE-Codes und zugehörige Nachrichtennummern

SQLSTATE	Nachrichtennummer	Beschreibung
00000		MMDB_SQLSTATE_OK Erfolgreich
01H01	DMB0211W	MMDB_SQLSTATE_WARN_NO_OVERWRITE Die Dateiüberschreibung findet nicht statt.
38A00	DMB0101E	MMDB_SQLSTATE_AUDIO_NULL_PARM Eingabeparameter für die UDF kann nicht Null sein.
38A02	DMB0209E	MMDB_SQLSTATE_AUDIO_OPEN_HDR_ERROR Fehler aufgetreten beim Öffnen der Kopfzeile der Audiodatei.
38A03	DMB0209E	MMDB_SQLSTATE_AUDIO_BAD_WAVE_HDR Ungültige Audiodatei zur Verfügung gestellt.
38V00	DMB0101E	MMDB_SQLSTATE_VIDEO_NULL_PARM Eingabeparameter für die UDF kann nicht Null sein.
38V02	DMB0051E	MMDB_SQLSTATE_VIDEO_OPEN_HDR_ERROR Fehler aufgetreten beim Öffnen der Kopfzeile der Videodatei.
38V03	DMB0105E	MMDB_SQLSTATE_VIDEO_BAD_MPEG1_HDR Ungültige MPEG1-Datei zur Verfügung gestellt.
38V04	DMB0104E	MMDB_SQLSTATE_VIDEO_BLOB_TOO_SHORT Zur Verfügung gestellter Videopuffer ist zu klein.
38V05	DMB0106E	MMDB_SQLSTATE_VIDEO_BAD_AVI_HDR Ungültige AVI-Datei zur Verfügung gestellt.
38V07	DMB0106E	MMDB_SQLSTATE_VIDEO_BAD_QT_HDR Ungültige Quicktime-Datei zur Verfügung gestellt.
38600	DMB0075E DMB0101E DMB0102E DMB0103E DMB0210E	MMDB_SQLSTATE_INVALID_INPUT Eingabeparameter für die UDF ist ungültig.
38601	DMB0009E	MMDB_SQLSTATE_MALLOC_FAIL Speicherzuordnung fehlgeschlagen.
38602	DMB0386E	MMDB_SQLSTATE_CANNOT_COLLOCATE Benutzerdaten können nicht zusammengestellt werden.
38603	DMB0077E	MMDB_SQLSTATE_READ_FILE_FAIL Aus der Datei kann nicht gelesen werden.
38604	DMB0080E	MMDB_SQLSTATE_WRITE_FILE_FAIL In die Datei kann nicht geschrieben werden.
38610	DMB0070E	MMDB_SQLSTATE_INVALID_HANDLE Multimediaspalte enthält ungültige Daten.
38611	DMB0073E	MMDB_SQLSTATE_INVALID_SESSION_HANDLE Ungültige UDF-Sitzungskennung.

Tabelle 16. SQLSTATE-Codes und zugehörige Nachrichtennummern (Forts.)

SQLSTATE	Nachrichtennummer	Beschreibung
38612	DMB0074E	MMDB_SQLSTATE_INVALID_STATEMENT_HANDLE Ungültige UDF-Anweisungskennung.
38613	DMB0083E	MMDB_SQLSTATE_INVALID_IMPORT_REQUEST Die Importanforderung ist ungültig.
38615	DMB0071E	MMDB_SQLSTATE_CONNECT_FAIL Fehler aufgetreten beim Herstellen einer Verbindung zur Datenbank.
38617	DMB0071E	MMDB_SQLSTATE_ALLOC_STMT_FAIL Fehler aufgetreten beim Zuordnen einer neuen Anweisungskennung.
38618	DMB0208E DMB0138E	MMDB_SQLSTATE_FREE_STMT_FAIL Fehler aufgetreten beim Freigeben der Anweisung.
38619	DMB0208E DMB0132E	MMDB_SQLSTATE_BIND_FAIL Fehler aufgetreten beim Binden.
38620	DMB0208E	MMDB_SQLSTATE_BIND_COLUMN_FAIL Fehler aufgetreten beim Binden einer Spalte.
38621	DMB0208E	MMDB_SQLSTATE_BIND_FILE_FAIL Fehler aufgetreten beim Binden einer Datei.
38622	DMB0208E DMB0132E	MMDB_SQLSTATE_SET_PARAM_FAIL Fehler aufgetreten beim Setzen von Parametern.
38623	DMB0208E DMB0131E	MMDB_SQLSTATE_PREPARE_FAIL Fehler aufgetreten beim Vorbereiten einer SQL-Anweisung.
38624	DMB0208E DMB0133E DMB0172E	MMDB_SQLSTATE_EXECUTE_FAIL Fehler aufgetreten beim Ausführen einer Anweisung.
38625	DMB0208E DMB0133E	MMDB_SQLSTATE_EXEC_DIRECT_FAIL Fehler aufgetreten bei der direkten Ausführung der SQL-Anweisung in UDF.
38626	DMB0208E DMB0133E	MMDB_SQLSTATE_FETCH_FAIL Fehler aufgetreten beim Abrufen der nächsten Zeile mit Daten.
38627	DMB0208E	MMDB_SQLSTATE_COMMIT_FAIL Fehler aufgetreten beim Festschreiben der Transaktion.
38628	DMB0208E	MMDB_SQLSTATE_GET_LENGTH_FAIL Fehler aufgetreten beim Abrufen der Länge eines Zeichenfolgewertes.
38629	DMB0208E	MMDB_SQLSTATE_GET_SUBSTRING_FAIL Fehler aufgetreten beim Abrufen eines Teils eines Zeichenfolgewerts.
38650	DMB0077E DMB0079E	MMDB_SQLSTATE_COPY_BLOB_2_FILE_FAIL Fehler aufgetreten beim Kopieren eines BLOB in eine Datei.
38651	DMB0086E	MMDB_SQLSTATE_BLOB_BUFFER_TOO_SMALL Zur Verfügung gestellter Puffer ist zu klein.
38652	DMB0082E	MMDB_SQLSTATE_BUILD_HANDLE Fehler aufgetreten beim Erstellen von Multimediaspaltendaten.

Tabelle 16. SQLSTATE-Codes und zugehörige Nachrichtennummern (Forts.)

SQLSTATE	Nachrichtennummer	Beschreibung
38653	DMB0083E	MMDB_SQLSTATE_INVALID_INSERT_VIA_SELECT Die Anforderung zum Einfügen über SELECT ist ungültig.
38654	DMB0081E	MMDB_SQLSTATE_INVALID_OFFSET_SIZE Die Größe der relativen Position ist ungültig.
38655	DMB0068E	MMDB_SQLSTATE_METATABLE_DOESNOT_EXIST Die erforderliche Metadatentabelle existiert nicht.
38670	DMB0134E DMB0103E	MMDB_SQLSTATE_UNKNOWN_FORMAT Die gespeicherten Multimediadaten haben ein unbekanntes Format.
38671	DMB0135E	MMDB_SQLSTATE_CREATE_THUMBNAIL_FAIL Fehler aufgetreten beim Erstellen des Piktogramms.
38672	DMB0114E	MMDB_SQLSTATE_FORMAT_CONVERSION_FAIL Fehler aufgetreten beim Umsetzen des Dateiformats.
38673	DMB0363E	MMDB_SQLSTATE_INVALID_UPDATE Fehler aufgetreten, als eine Aktualisierungs-UDF ohne Verweis auf eine Tabelle aufgerufen wurde.
38674	DMB0361E	MMDB_SQLSTATE_NOT_ENABLED Fehler aufgetreten, als eine Import-UDF auf eine Spalte angewendet wurde, die nicht für den Extender aktiviert war.
38675	DMB0129E	MMDB_SQLSTATE_VIDEO_INTERNAL Interner Fehler in Video Extender-UDFs.
38676	DMB0129E	MMDB_SQLSTATE_AUDIO_INTERNAL Interner Fehler in Audio Extender-UDFs.
38677	DMB0129E	MMDB_SQLSTATE_IMAGE_INTERNAL
38678	DMB0089E DMB0208E	MMDB_SQLSTATE_BASE_INTERNAL_ERROR Interner Fehler in der allgemeinen Schicht.
38681	DMB0108E	MMDB_SQLSTATE_IMPORT_ENV_NOT_SETUP Umgebungsvariable für den Import ist nicht korrekt definiert.
38682	DMB0111E	MMDB_SQLSTATE_STORE_ENV_NOT_SETUP Umgebungsvariable für Speicheroperation ist nicht korrekt definiert.
38683	DMB0107E	MMDB_SQLSTATE_EXPORT_ENV_NOT_SETUP Umgebungsvariable für Exportoperation ist nicht korrekt definiert.
38684	DMB0117E	MMDB_SQLSTATE_TEMP_ENV_NOT_SETUP Umgebungsvariable für das Erstellen von temporären Dateien ist nicht korrekt definiert.
38686	DMB0109E	MMDB_SQLSTATE_CANT_RESOLVE_IMPORT_FILE Fehler aufgetreten beim Auflösen von Importdateinamen.
38687	DMB0112E	MMDB_SQLSTATE_CANT_RESOLVE_STORE_FILE Fehler aufgetreten beim Auflösen von Speicherdateinamen.

Tabelle 16. SQLSTATE-Codes und zugehörige Nachrichtennummern (Forts.)

SQLSTATE	Nachrichtennummer	Beschreibung
38688	DMB0110E	MMDB_SQLSTATE_CANT_RESOLVE_EXPORT_FILE Fehler aufgetreten beim Auflösen von Exportdateinamen.
38689	DMB0116E	MMDB_SQLSTATE_CANT_CREATE_TMP_FILE Fehler aufgetreten beim Erstellen der temporären Datei.
38690	DMB0076E	MMDB_SQLSTATE_OPEN_IMPORT_FILE_FAIL Die Importdatei kann nicht geöffnet werden.
38691	DMB0115E	MMDB_SQLSTATE_OPEN_STORE_FILE_FAIL Die Importdatei kann nicht geöffnet werden.
38692	DMB0114E	MMDB_SQLSTATE_OPEN_EXPORT_FILE_FAIL Die Exportdatei kann nicht geöffnet werden.
38693	DMB0118E	MMDB_SQLSTATE_OPEN_TEMP_FILE_FAIL Die temporäre Datei kann nicht geöffnet werden.
38694	DMB0117E	MMDB_SQLSTATE_OPEN_CONTENT_FILE_FAIL Die Nachrichtenkomponente kann nicht geöffnet werden.
38695	DMB0135E	MMDB_SQLSTATE_OPEN_THUMBNAIL_FILE_FAIL Die Piktogrammdatei kann nicht geöffnet werden.
38696	DMB0135E	MMDB_SQLSTATE_READ_THUMBNAIL_FILE_FAIL Die Piktogrammdatei kann nicht gelesen werden.
38697	DMB0207E	MMDB_SQLSTATE_OVERWRITE_NOT_ALLOWED Die Überschreibungsoperation kann nicht ausgeführt werden.
38699	DMB0171E	MMDB_SQLSTATE_QUERY_NAME_NOT_FOUND Abfrage mit diesem Namen nicht gefunden.
38700		MMDB_SQLSTATE_NO_MANAGEBLOB
38701		MMDB_SQLSTATE_UDFLOCATOR_FAIL
38702		MMDB_SQLSTATE_SQL_FAIL
38703		MMDB_SQLSTATE_INVALID_UPDATE
38704		MMDB_SQLSTATE_NOT_ENABLED
38705	DMB0366E DMB0382E	MMDB_SQLSTATE_QBIC_QUERY_FAIL_TO_BUILD Fehler aufgetreten beim Erstellen der Abfrage.
38706	DMB0205E	MMDB_SQLSTATE_QBIC_TABLE_COLUMN_PAIR_NOT_VALID Fehler aufgetreten beim Zugriff auf einen QBIC-Katalog. Entweder wurde eine Abbildkennung nicht im Katalog gefunden oder die Kombination aus Tabellenname und Spaltenname hat keinen Katalog.
38707	DMB0383E	MMDB_SQLSTATE_QBIC_QUERY_EXECUTE_FAILED Fehler aufgetreten beim Ausführen der Abfrage.
38708		MMDB_SQLSTATE_QBIC_UKNOWN_ERROR Unbekannter Fehler in QBIC
38709	DMB0208E	MMDB_COPY_FILE_TO_LOCATOR_FAILURE Fehler aufgetreten beim Kopieren einer Datei in einen LOB-Zeiger.
38710	DMB0534E	MMDB_SQLSTATE_QBIC_UNSUPPORTED_UDF UDF nicht unterstützt.

Nachrichten

DMB0001E Der DB2 Extender-Server wurde nicht verbunden. Grund: ″<code>″.

Ursache: Für die versuchte Operation ist es erforderlich, dass die DB2 Extender-Services aktiv sind.

Aktion: Führen Sie auf dem Server den Befehl DMBSTART von der Befehlszeile des Betriebssystems aus aus.

DMB0003W Für diese Sitzung wird die Tracefunktion der DB2 Extender ausgeführt.

Ursache: Die Tracefunktion verbraucht Systemressourcen.

Aktion: Wenn die Leistung des Systems beeinträchtigt ist, sollten Sie den Trace möglicherweise ausschalten.

DMB0004I Dieses Programm kann nur von Exemplareigner ″<name>″ ausgeführt werden.

Ursache: Die DB2 Extender-Server müssen von der Benutzer-ID aus gestartet werden, unter der das Exemplar erstellt wurde.

Aktion: Führen Sie den Befehl DMBSTART von der Benutzer-ID aus aus, unter der das Exemplar erstellt wurde.

DMB0005E Die aktuelle Datenbank ist nicht für den ″<extender-name>″ Extender aktiviert.

Ursache: Eine Operation wurde versucht, für die es erforderlich ist, dass die Datenbank für einen bestimmten DB2 Extender aktiviert ist. Wenn Sie beispielsweise eine Tabelle für DB2IMAGE-Daten aktivieren wollen, müssen Sie zunächst die Datenbank, in der die Tabelle gespeichert ist, für DB2IMAGE-Daten aktivieren.

Aktion: Aktivieren Sie die Datenbank für den gewünschten Extender-Datentyp und versuchen Sie die Operation erneut.

DMB0006E Der Benutzer ″<name>″ ist zum Aufruf dieser API nicht berechtigt.

Ursache: Der Aufruf an eine Anwendungsprogrammierschnittstelle wurde von einer Benutzer-ID aus versucht, die nicht über die Berechtigungsstufe verfügt, die für diese API erforderlich ist.

Aktion: Führen Sie die Anwendung über eine andere Benutzer-ID aus, oder fordern Sie den Datenbankadministrator auf, die Berechtigungsstufe für die aktuelle Benutzer-ID zu ändern.

DMB0007E Die Benutzertabelle ″<tabellenname>″ ist nicht für den ″<extender-name>″ Extender aktiviert.

Ursache: Die Tabelle, für die die Operation versucht wurde, ist nicht für den entsprechenden DB2 Extender aktiviert. Eine Tabelle muss beispielsweise zur Speicherung von Audiodaten aktiviert sein, bevor eine Spalte in der Tabelle für Audiodaten aktiviert werden kann.

Aktion: Stellen Sie zunächst sicher, dass die Tabelle für den Extender aktiviert ist. Aktivieren Sie dann die Spalte.

DMB0008E Beim Ausführen der gespeicherten Prozedur ″<name>″ trat ein Fehler auf.

Ursache: Entweder ist ein Fehler in der gespeicherten Prozedur, die in der Nachricht angegeben wird, oder ein Problem mit der Umgebung liegt vor.

Aktion: Überprüfen Sie die Anwendung, und wiederholen Sie die Operation.

DMB0009E Speicherzuordnungsfehler.

Ursache: Das System konnte den Speicher, der zur Unterstützung der versuchten Operation erforderlich ist, nicht zuordnen.

Aktion: Überprüfen Sie, dass in Ihrem System genügend Speicher zum Beenden der Operation zur Verfügung steht.

DMB0010E Der ″<extender-name>″ Extender wurde zuvor für den UDT ″<name>″ definiert.

Ursache: Der Name des benutzerdefinierten Typs (UDT) wurde bereits für einen UDT verwendet, der für einen anderen DB2 Extender definiert war.

Aktion: Wählen Sie einen anderen Namen für den UDT.

DMB0011E Die Benutzerspalte ″<spaltenname>″ kann für den ″<extender-name>″ Extender nicht aktiviert werden. Die Definition der Benutzerspalte ist mit dem einzigartigen Typ ″MMDBSYS.<name>″, der dem Extender zugeordnet ist, nicht kompatibel.

Ursache: Die angegebene Spalte ist nicht für den Datentyp definiert, der in der Nachricht gezeigt wird, so dass sie nicht für diesen Extender aktiviert werden kann.

Aktion: Stellen Sie sicher, dass die Spalte, die Sie aktivieren wollen, unter Verwendung des dem Extender entsprechenden Datentyps definiert wurde.

DMB0012E Die angegebene Benutzertabelle ″<tabellenname>″ existiert nicht.

Ursache: Eine Tabelle mit dem angegebenen Namen existiert nicht.

Aktion: Überprüfen Sie den Namen der Tabelle und ob die Tabelle existiert.

DMB0013E Spalte ″<spaltenname>″ ist für die Tabelle ″<tabellenname>″ nicht definiert.

Ursache: Die versuchte Operation verwies auf einen Spaltennamen, der in der angegebenen Tabelle nicht existiert.

Aktion: Überprüfen Sie den Namen der Tabelle und der Spalte.

DMB0014W Spalte ″<spaltenname>″ in der Benutzertabelle ″<tabellenname>″ ist bereits für den ″<extender-name>″ Extender aktiviert.

Ursache: Der Versuch wurde unternommen, die Spalte für einen Extender zu aktivieren, für den sie bereits aktiviert war.

Aktion: Keine Aktion erforderlich.

DMB0015W Die Datenbank ist bereits für den ″<extender-name>″ Extender aktiviert.

Ursache: Der Versuch wurde unternommen, die Datenbank für einen Extender zu aktivieren, für den sie bereits aktiviert war.

Aktion: Keine Aktion erforderlich.

DMB0016W Die Benutzertabelle ″<tabellenname>″ ist bereits für den ″<extender-name>″ Extender aktiviert.

Ursache: Der Versuch wurde unternommen, die Tabelle für einen Extender zu aktivieren, für den sie bereits aktiviert war.

Aktion: Keine Aktion erforderlich.

DMB0017E Die Benutzertabelle ″<tabellenname>″ ist bereits für den ″<extender-name>″ Extender aktiviert. Mindestens eine der zugeordneten Metadatentabellen ″<tabellenname>″ oder ″<tabellenname>″ existiert jedoch nicht.

Ursache: Eine oder mehrere Tabellen zur Verwaltungsunterstützung (Metadatentabellen), die der Tabelle zugeordnet sind, wurden beschädigt oder zerstört. Ohne diese Metadatentabellen kann die Benutzertabelle nicht für Daten vom Typ dieses Extenders verwendet werden.

Aktion: Inaktivieren Sie die Benutzertabelle und aktivieren Sie sie für den entsprechenden Extender.

DMB0018E Das System kann keinen eindeutigen Auslösernamen für Spalte ″<spaltenname>″ in Tabelle ″<tabellenname>″ erstellen.

Ursache: Als das System versuchte, die Spalte in der Benutzertabelle zu aktivieren, trat beim Erstellen der Auslöser, die von den DB2 Extendern verwendet werden, ein Fehler auf.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie Ihren Datenbankadministrator und danach den IBM Kundendienst.

DMB0019I In Tabelle ″<tabellenname>″ für den ″<extender-name>″ Extender wird auf ″<anzahl>″ Dateien verwiesen.

Ursache: Die Nachricht zeigt die Anzahl der externen Multimediadateien an, auf die durch eine Benutzertabelle für einen bestimmten Extender verwiesen wird.

Aktion: Keine Aktion erforderlich.

DMB0020I In Tabellen mit dem Tabellenschema ″<name>″ für den ″<extender-name>″ Extender wird auf ″<anzahl>″ Dateien verwiesen.

Ursache: Die Nachricht zeigt die Anzahl der externen Multimediadateien an, auf die durch Benutzertabellen mit einem bestimmten Schemanamen verwiesen wird.

Aktion: Keine Aktion erforderlich.

DMB0021I Vom ″<extender-name>″ Extender wird auf ″<anzahl>″ Dateien verwiesen, auf die nicht zugegriffen werden kann.

Ursache: Die Nachricht zeigt die Anzahl der externen Multimediadateien an, auf die durch eine Benutzertabelle für einen bestimmten Extender verwiesen wird, die aber nicht zugänglich sind. Die Dateien sind möglicherweise gelöscht.

Aktion: Keine Aktion erforderlich.

DMB0022I Vom ″<extender-name>″ Extender wird auf ″<anzahl>″ Dateien verwiesen, auf die nicht zugegriffen werden kann.

Ursache: Die Nachricht zeigt die Anzahl der externen Multimediadateien an,

- auf die durch Benutzertabellen in der aktuellen Datenbank verwiesen wird.
- die zu einem bestimmten Extender-Multimediatyp (z. B. Video) gehören.
- die nicht zugänglich sind. Beispielsweise sind die Dateien möglicherweise gelöscht.

Aktion: Keine Aktion erforderlich.

DMB0023I In Tabellen mit dem Tabellenschema ″<name>″ für den ″<extender-name>″ Extender wird auf ″<anzahl>″ Dateien verwiesen, auf die nicht zugegriffen werden kann.

Ursache: Die Nachricht zeigt die Anzahl der externen Multimediadateien an, auf die durch Benutzertabellen mit einem bestimmten Schemanamen verwiesen wird, die aber nicht zugänglich sind. Die Dateien sind möglicherweise gelöscht. Die Nachricht gibt außerdem die Anzahl von Extendern an, für die die Tabellen aktiviert sind.

Aktion: Keine Aktion erforderlich.

DMB0024I Die aktuelle Datenbank ist für ″<anzahl>″ Extender aktiviert.

Ursache: Die Nachricht listet die Anzahl von DB2 Extendern auf, für die die aktuelle Datenbank aktiviert ist.

Aktion: Keine Aktion erforderlich.

DMB0025I Die Tabelle ″<tabellenname>″ ist für ″<anzahl>″ Extender aktiviert.

Ursache: Die Nachricht listet die Anzahl von DB2 Extendern auf, für die die angegebene Tabelle aktiviert ist.

Aktion: Keine Aktion erforderlich.

DMB0026I Spalte ″<spaltenname>″ in Tabelle ″<tabellenname>″ ist für ″<anzahl>″ Extender aktiviert.

Ursache: Die Nachricht listet die Anzahl von DB2 Extendern auf, für die die angegebene Spalte aktiviert ist.

Aktion: Keine Aktion erforderlich.

DMB0027I Die aktuelle Datenbank ist für den ″<extender-name>″ Extender aktiviert.

Ursache: Die Nachricht gibt den DB2 Extender an, für den die aktuelle Datenbank aktiviert ist.

Aktion: Keine Aktion erforderlich.

DMB0028I Die Tabelle ″<tabellenname>″ ist für den ″<extender-name>″ Extender aktiviert.

Ursache: Die Nachricht gibt den Multimediadatentyp an, für den die Benutzertabelle aktiviert ist.

Aktion: Keine Aktion erforderlich.

DMB0029I Spalte ″<spaltenname>″ in Tabelle ″<tabellenname>″ ist für den ″<extender-name>″ Extender aktiviert.

Ursache: Die Nachricht gibt den Multimediadatentyp an, für den die Benutzerspalte aktiviert ist.

Aktion: Keine Aktion erforderlich.

DMB0030E Die aktuelle Datenbank kann für den ″<extender-name>″ Extender nicht aktiviert werden. RC = ″<code>.″

Ursache: Entweder existiert die Datenbank nicht oder Sie sind nicht berechtigt, die Datenbank zu aktivieren.

Aktion: Stellen Sie sicher, dass die Datenbank existiert und dass Sie berechtigt sind, die Datenbank zu aktivieren.

DMB0031E Die Tabelle kann für den ″<extender-name>″ Extender nicht aktiviert werden. RC = ″<code>.″

Ursache: Die Datenbank existiert nicht, die Tabelle ist nicht aktiviert oder Sie sind nicht berechtigt, die Tabelle zu aktivieren.

Aktion: Stellen Sie sicher, dass die Datenbank existiert und dass sowohl die Datenbank als auch die Tabelle für den Extender aktiviert sind. Stellen Sie sicher, dass Sie berechtigt sind, die Tabelle zu aktivieren.

DMB0032E Die Spalte kann für den ″<extender-name>″ Extender nicht aktiviert werden. RC = ″<code>.″

Ursache: Die Spalte wurde nicht unter Verwendung des Datentyps für diesen Extender definiert, die Spalte existiert nicht, die Tabelle ist nicht aktiviert oder Sie sind nicht berechtigt, die Spalte zu aktivieren.

Aktion: Stellen Sie sicher, dass die Spalte unter Verwendung des korrekten Datentyps definiert wurde. Stellen Sie sicher, dass die Tabelle aktiviert ist und Sie berechtigt sind, die Spalte zu aktivieren.

DMB0033E Sie sind zum Ausführen dieses Befehls nicht berechtigt.

Ursache: Ihre Benutzer-ID hat nicht die Berechtigungsstufe, die zum Ausführen des Befehls erforderlich ist.

Aktion: Führen Sie den Befehl über eine andere Benutzer-ID aus, oder fordern Sie den Datenbankadministrator auf, die Berechtigungsstufe für die aktuelle Benutzer-ID zu ändern.

DMB0034I Der DB2 Extender-Server für Datenbank ″<datenbankname>″ wurde erfolgreich gestartet.

Ursache: Der Server wurde erfolgreich für die aktuelle Datenbank gestartet.

Aktion: Keine Aktion erforderlich.

DMB0035I Der DB2 Extender-Server für Datenbank ″<datenbankname>″ wurde gestoppt.

Ursache: Der Server wurde erfolgreich für die aktuelle Datenbank gestoppt.

Aktion: Keine Aktion erforderlich.

DMB0036E Der DB2 Extender-Server kann nicht gestartet oder gestoppt werden. Wahrscheinlich arbeitet der DB2 Extender-Serverdämon nicht. Wenden Sie sich an den Datenbankadministrator.

Ursache: Beim Starten oder Stoppen des DB2 Extender-Servers ist ein Fehler aufgetreten. Wahrscheinlich ist der DB2 Extender-Serverdämon nicht aktiv.

Aktion: Wenden Sie sich an den Datenbankadministrator.

DMB0037E Die USE-Sitzungskennung ist ungültig.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0038E Die USE-Anweisungskennung ist ungültig.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0039E USE-Fehler: ″<fehler>.″

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Befolgen Sie die Anweisungen, die in der zugehörigen Fehlernachricht enthalten sind, und wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0040E SQL-Fehler: ″<fehler>″

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Befolgen Sie die Anweisungen, die in der zugehörigen Fehlernachricht enthalten sind, und wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0041W Die aktuelle Datenbank wird mit dem neu angegebenen Tabellenbereich für den ″<extender-name>″ Extender erneut aktiviert.

Ursache: Als die aktuelle Datenbank zuvor aktiviert wurde, verwendete sie einen anderen Tabellenbereich. Die Datenbank ist jetzt mit einem neuen Tabellenbereich für die Tabellen zur Verwaltungsunterstützung aktiviert.

Aktion: Keine Aktion erforderlich.

DMB0042E Spalte ″<spaltenname>″ in Tabelle ″<tabellenname>″ ist für den ″<extender-name>″ Extender nicht aktiviert.

Ursache: Die angegebene Spalte ist nicht für den Extender aktiviert, für den die Operation versucht wurde. Sie haben beispielsweise versucht, eine Spalte zu inaktivieren, die zu dem Zeitpunkt nicht für den angegebenen Extender aktiviert war.

Aktion: Stellen Sie sicher, dass die Spalte für den Extender aktiviert ist, der in der Nachricht angegeben ist.

DMB0043I Die aktuelle Datenbank ist für den ″<extender-name>″ Extender inaktiviert.

Ursache: Die Operation zum Inaktivieren war erfolgreich.

Aktion: Keine Aktion erforderlich.

DMB0044I Tabelle ″<tabellenname>″ ist für den ″<extender-name>″ Extender inaktiviert.

Ursache: Die Operation zum Inaktivieren war erfolgreich.

Aktion: Keine Aktion erforderlich.

DMB0045I Spalte ″<spaltenname>″ in Tabelle ″<tabellenname>″ ist für den ″<extender-name>″ Extender inaktiviert.

Ursache: Die Operation zum Inaktivieren war erfolgreich.

Aktion: Keine Aktion erforderlich.

DMB0046E Die aktuelle Datenbank kann für den ″<extender-name>″ Extender nicht inaktiviert werden. RC = ″<code>.″

Ursache: Die Datenbank existiert nicht, sie ist nicht für den Extender aktiviert oder Sie sind nicht berechtigt, die Datenbank zu inaktivieren.

Aktion: Stellen Sie sicher, dass die Datenbank existiert und für den Extender aktiviert ist. Stellen Sie sicher, dass Sie berechtigt sind, die Datenbank zu inaktivieren.

DMB0047E Die Tabelle kann für den ″<extender-name>″ Extender nicht inaktiviert werden. RC = ″<code>.″

Ursache: Die Tabelle existiert nicht, sie ist nicht für den Extender aktiviert oder Sie sind nicht berechtigt, die Tabelle zu inaktivieren.

Aktion: Stellen Sie sicher, dass die Tabelle für den Extender aktiviert ist. Stellen Sie sicher, dass Sie berechtigt sind, die Tabelle zu inaktivieren.

DMB0048E Die Spalte kann für den ″<extender-name>″ Extender nicht inaktiviert werden. RC = ″<code>″

Ursache: Die Spalte ist nicht für den Extender, der in der Nachricht angegeben ist, aktiviert, so dass sie nicht für diesen Extender inaktiviert werden kann.

Aktion: Prüfen Sie den Namen des Extenders und ob die Benutzerspalte die Spalte ist, die Sie inaktivieren wollen.

DMB0049E Sie sind zum Ausführen dieses Befehls nicht berechtigt.

Ursache: Ihre Benutzer-ID hat nicht die Berechtigungsstufe, die zum Ausführen des Befehls erforderlich ist.

Aktion: Führen Sie die Anwendung über eine andere Benutzer-ID aus, oder fordern Sie den Datenbankadministrator auf, die Berechtigungsstufe für die aktuelle Benutzer-ID zu ändern.

DMB0050E Sie haben keine ″<berechtigungsstufe>″-Berechtigung für Tabelle ″<tabellenname>″.

Ursache: Für die Operation ist eine höhere Berechtigungsstufe erforderlich als die Stufe der Benutzer-ID, die die Operation versuchte.

Aktion: Führen Sie die Operation über die Benutzer-ID mit der korrekten Berechtigung aus, oder fordern Sie den Datenbankadministrator auf, die Berechtigungsstufe für die aktuelle Benutzer-ID zu ändern.

DMB0051E Der Kennsatz der Multimediadatei ist defekt.

Ursache: Das System kann die Kopfzeile der Multimediadatei nicht lesen bzw. öffnen. Entweder ist die Datei beschädigt oder es handelt sich nicht um eine Multimediadatei.

Aktion: Überprüfen Sie, dass die Datei eine Multimediadatei ist und dass sie nicht beschädigt ist.

DMB0052I Der DB2 Extender-Server für ″<datenbankname>″ wurde erfolgreich gestartet.

Ursache: Der Server wurde erfolgreich gestartet.

Aktion: Keine Aktion erforderlich.

DMB0053I Der DB2 Extender-Server für ″<datenbankname>″ wurde erfolgreich gestoppt.

Ursache: Der Server wurde erfolgreich gestoppt.

Aktion: Keine Aktion erforderlich.

DMB0054E Der DB2 Extender-Server kann keine Verbindung zur Datenbank herstellen oder keine DB2-Anweisungskennung zuordnen. Wahrscheinlich arbeitet der DB2 Extender-Server für die Datenbank ″<datenbankname>″ nicht.

Ursache: Der DB2 Extender-Server kann keine Verbindung zur Datenbank herstellen oder keine DB2-Anweisungskennung zuordnen. Wahrscheinlich ist der DB2 Extender-Server für die Datenbank nicht aktiv.

Aktion: Stellen Sie sicher, dass der DB2 Extender-Server für die Datenbank aktiv ist. Ist dies nicht der Fall, starten Sie den spezifischen Extender-Server für die Datenbank oder fordern Sie Ihren Systemadministrator auf, die Extender-Services erneut zu starten.

DMB0055I Der Befehl ″<befehlsname>″ wurde erfolgreich ausgeführt.

Ursache: Der Befehl wurde erfolgreich beendet.

Aktion: Keine Aktion erforderlich.

DMB0056E Ein unerwartetes Token ″<token>″ wurde nach ″<schlüsselwort>″ gefunden. Zu den erwarteten Token gehört: <extender-name>.

Ursache: Der Befehl erwartete den Namen eines DB2 Extenders an Stelle des Tokens, das in der Nachricht angegeben ist.

Aktion: Befolgen Sie die Syntax des Befehls, und wiederholen Sie die Operation.

DMB0057E Der Tabellenbereich ″<tabellenbereichsname>″ ist ungültig.

Ursache: Der in der Nachricht angegebene Tabellenbereich existiert nicht.

Aktion: Überprüfen Sie den Namen des Tabellenbereichs und ob der Tabellenbereich existiert.

DMB0058I Vom ″<extender-name>″ Extender wird auf ″<anzahl>″ Dateien verwiesen.

Ursache: Die Nachricht zeigt die Anzahl der externen Multimediadateien an, auf die durch einen bestimmten Extender verwiesen wird.

Aktion: Keine Aktion erforderlich.

DMB0059E ″<name>″ ist kein gültiger Name für einen DB2 Extender. Gültige Extender-Namen sind ″<extender-name>″, DB2VIDEO, DB2AUDIO und DB2IMAGE.

Ursache: Der Extender-Name ist nicht korrekt geschrieben.

Aktion: Überprüfen Sie den Extender-Namen.

DMB0060E Die korrekte Syntax für ″<funktion>″ ist: ″<syntax>.″

Ursache: Die Syntax des eingegebenen Befehls ist falsch.

Aktion: Befolgen Sie die Syntax, wie sie in der Nachricht beschrieben ist.

DMB0061E Der Tabellenname ″<tabellenname>″, der nach ″<schlüsselwort>″ folgt, ist ungültig.

Ursache: Der Befehl erwartete den Namen einer Tabelle.

Aktion: Befolgen Sie die Syntax des Befehls, und wiederholen Sie die Operation.

DMB0062E Der Spaltenname ″<name>″, der nach ″<schlüsselwort>″ folgt, ist ungültig.

Ursache: Der Befehl erwartete den Namen einer Spalte.

Aktion: Befolgen Sie die Syntax des Befehls, und wiederholen Sie die Operation.

DMB0064E Das System kann das Token ″<token>″, das nach ″<schlüsselwort>″ folgt, nicht erkennen.

Ursache: Der Befehl erwartete eine andere Angabe als das Token, das in der Nachricht angegeben ist.

Aktion: Befolgen Sie die Syntax des Befehls, und wiederholen Sie die Operation.

DMB0065E Die Benutzer-ID ″<benutzer-id>″, die nach ″<schlüsselwort>″ folgt, ist ungültig.

Ursache: Der Befehl erwartete eine gültige Benutzer-ID.

Aktion: Überprüfen Sie die gewünschte Benutzer-ID, und wiederholen Sie den Befehl.

DMB0066E Das Kennwort ″<kennwort>″, das nach ″<schlüsselwort>″ folgt, ist ungültig.

Ursache: Der Befehl erwartete ein gültiges Kennwort an Stelle des Tokens, das in der Nachricht angegeben ist.

Aktion: Überprüfen Sie das Kennwort, und wiederholen Sie den Befehl.

DMB0067E Der eingegebene Befehl ist ungültig.

Ursache: Der Name des Befehls wurde nicht korrekt geschrieben oder die Syntax war falsch.

Aktion: Befolgen Sie die Syntax des Befehls, und wiederholen Sie die Operation.

DMB0068E Die Metadatentabelle existiert nicht.

Ursache: Die Funktion versuchte, eine Tabelle zur Verwaltungsunterstützung (Metadatentabelle) zu verwenden, die für das Datenobjekt existieren sollte. Die Metadatentabelle ist möglicherweise beschädigt oder gelöscht.

Aktion: Überprüfen Sie den Namen und ob die Metadatentabelle existiert. Wenn die Metadatentabellen versehentlich gelöscht oder beschädigt wurden, inaktivieren Sie das Datenobjekt und aktivieren Sie es erneut.

DMB0069E Der Datenbankname ″<name>″ ist ungültig.

Ursache: Eine Datenbank mit dem angegebenen Namen existiert nicht.

Aktion: Überprüfen Sie den Namen und ob die Datenbank existiert.

DMB0070E Die Kennung ist ungültig.

Ursache: Der Kennungswert, den Sie an Ihre Anwendung übergeben haben, ist möglicherweise beschädigt.

Aktion: Überprüfen Sie Ihre Anwendung, um sicherzustellen, dass die Kennungswerte des Extenders nicht geändert sind.

DMB0071E Verbindung zu ″<datenbankname>″ kann nicht hergestellt werden.

Ursache: Der DB2 Extender-Server für die Datenbank wurde möglicherweise nicht gestartet.

Aktion: Überprüfen Sie den Status des Servers. Wenn der Server nicht aktiv ist, starten Sie ihn erneut unter Verwendung des Befehls START SERVER von der DMB-Befehlszeile aus.

DMB0072E Der UDF SQL-Server kann die DB nicht trennen.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0073E Die USE-Sitzungskennung ist ungültig.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0074E Die USE-Anweisungskennung ist ungültig.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0075E Geben Sie einen Dateinamen an.

Ursache: Für die Operation ist der Name einer Multimediadatei erforderlich.

Aktion: Geben Sie den Namen einer Multimediadatei ein.

DMB0076E Die Importdatei kann nicht geöffnet werden.

Ursache: Die Importdatei fehlt oder ist beschädigt.

Aktion: Überprüfen Sie den Namen der Importdatei und ob die Datei existiert.

DMB0077E Nachrichtenkomponente kann nicht geöffnet/gelesen werden.

Ursache: Die Extender-Kennung zeigt auf eine Datei, die nicht existiert oder beschädigt ist. Die Datei ist für den Extender nicht zugänglich.

Aktion: Verwenden Sie die UDF FILENAME, um den Namen der Datei zu suchen, oder prüfen Sie, ob die Nachrichtenkomponente existiert.

DMB0078E Exportdatei kann nicht erstellt werden.

Ursache: Die Exportdatei fehlt oder ist beschädigt.

Aktion: Überprüfen Sie den Namen der Exportdatei und ob die Datei existiert.

DMB0079E BLOB kann nicht in eine Datei kopiert werden.

Ursache: Die Datei kann das BLOB nicht akzeptieren. Möglicherweise ist nicht genügend Speicherplatz vorhanden, um das BLOB zu speichern.

Aktion: Vergleichen Sie die Größe des BLOB mit der Größe des verfügbaren Speichers und erhöhen Sie gegebenenfalls die Speichergröße.

DMB0080E In die Datei kann nicht geschrieben werden.

Ursache: Die Datei ist beschädigt bzw. existiert nicht, oder der Name ist nicht korrekt geschrieben.

Aktion: Überprüfen Sie den Namen und ob die Datei existiert.

DMB0081E Relative Position oder Größe ist ungültig.

Ursache: Die Operation konnte die erwarteten Daten in der Datenstruktur nicht finden. Entweder ist die Größe des Feldes oder die relative Position falsch.

Aktion: Überprüfen Sie die relative Position und die Größe des Feldes.

DMB0082E Die Kennung kann nicht erstellt werden.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0083E ″<extender-name>″ und ″<extender-name>″ sind nicht kompatibel.

Ursache: Die zwei in der Nachricht angegebenen Extender sind bei dieser Verwendung nicht kompatibel. Die Einfügeoperation (vollständig oder eine Unterauswahl) ist ungültig.

Aktion: Stellen Sie sicher, dass das Zielobjekt für den gleichen Extender aktiviert ist wie das Quellenobjekt.

DMB0084E Die Importanforderung ist ungültig: Dateiname, Inhalt, Speichertyp.

Ursache: Die Importoperation ist fehlgeschlagen. Der Dateiname, der Inhalt oder der Speichertyp war ungültig.

Aktion: Überprüfen Sie die Daten, und wiederholen Sie die Operation.

DMB0085E Die Aktualisierungsanforderung ist ungültig: Dateiname, Inhalt, Speichertyp.

Ursache: Die Aktualisierungsoperation ist fehlgeschlagen. Der Dateiname, der Inhalt oder der Speichertyp war ungültig.

Aktion: Überprüfen Sie die Daten, und wiederholen Sie die Operation.

DMB0086E Die angeforderte Größe ist zu groß.

Ursache: Die Größe, die Sie angefordert haben, ist größer als die maximale BLOB-Größe für die UDF.

Aktion: Fordern Sie eine kleinere Größe an.

DMB0087E Der Dateiname ist ungültig.

Ursache: Eine Datei mit dem angegebenen Namen ist nicht vorhanden.

Aktion: Überprüfen Sie den Namen und ob die Datei existiert.

DMB0088E Die Kennung ist NULL.

Ursache: Die UDF erwartete eine Kennung, die ungleich Null ist.

Aktion: Stellen Sie sicher, dass die Anwendung eine gültige Kennung abruft und an die UDF übergibt.

DMB0089E Der Kennungswert existiert nicht.

Ursache: Die an die UDF übergebene Kennung ist ungültig.

Aktion: Stellen Sie sicher, dass die Anwendung eine gültige Kennung übergibt.

DMB0090E Daten sind abgeschnitten.

Ursache: Die Datei oder der Puffer ist zu klein, um die Daten aufzunehmen.

Aktion: Vergrößern Sie die Datei oder den Puffer.

DMB0091W Der Inhalt ist bereits in der Datei.

Ursache: Die Datei verfügt bereits über Inhalt. Dieser Inhalt wird überschrieben.

Aktion: Keine Aktion erforderlich.

DMB0092E Die für Spalte ″<spaltenname>″ versuchte Einfügeoperation ist ungültig. Die Spalte ist für den ″<extender-name>″ Extender aktiviert.

Ursache: Der Datentyp der einzufügenden Daten unterscheidet sich von dem Extender, für den die Spalte aktiviert ist.

Aktion: Stellen Sie sicher, dass das Zielobjekt für den gleichen Extender aktiviert ist wie das Quellenobjekt.

DMB0093E Die für Spalte ″<spaltenname>″ versuchte Aktualisierungsoperation ist ungültig. Die Spalte ist für den ″<extender-name>″ Extender aktiviert.

Ursache: Der Datentyp der zu aktualisierenden Daten unterscheidet sich von dem Extender, für den die Spalte aktiviert ist.

Aktion: Stellen Sie sicher, dass das Zielobjekt für den gleichen Extender aktiviert ist wie das Quellenobjekt.

DMB0094I Die Tabelle ″<tabellenname>″ existiert nicht.

Ursache: Das System kann keine Tabelle mit dem angegebenen Namen finden. Die Tabelle existiert möglicherweise in einer anderen Datenbank.

Aktion: Keine Aktion erforderlich.

DMB0095W Die Tabelle ″<tabellenname>″ ist nicht für den ″<extender-name>″ Extender aktiviert.

Ursache: Die Tabelle ist nicht für den Extender aktiviert.

Aktion: Keine Aktion erforderlich.

DMB0096W Spalte ″<spaltenname>″ in Tabelle ″<tabellenname>″ wurde für den ″<extender-name>″ Extender nicht aktiviert.

Ursache: Das System erwartete, dass die Spalte aktiviert ist.

Aktion: Keine Aktion erforderlich.

DMB0097W Die aktuelle Datenbank ist nicht für den ″<extender-name>″ Extender aktiviert.

Ursache: Das System erwartete, dass die Datenbank aktiviert ist.

Aktion: Aktivieren Sie die Datenbank für den Extender, der in der Nachricht angegeben ist.

DMB0098E Der Benutzer hat keine ″<berechtigungsstufe>″-Berechtigung für Tabelle ″<tabellenname>″.

Ursache: Für die Operation ist eine höhere Berechtigungsstufe erforderlich als die Stufe der Benutzer-ID, die die Operation versuchte.

Aktion: Führen Sie die Operation von der Benutzer-ID aus aus, die der Eigner der Tabelle ist, oder fordern Sie den Datenbankadministrator auf, die Berechtigungsstufe für die aktuelle Benutzer-ID zu ändern.

DMB0099E Die Transaktion kann nicht festgeschrieben werden.

Ursache: Der Extender-Server für die aktuelle Datenbank wurde möglicherweise gestoppt.

Aktion: Überprüfen Sie den Status des Servers. Wenn der Server nicht aktiv ist, starten Sie ihn erneut unter Verwendung des Befehls START SERVER von der db2ext-Befehlszeile aus.

DMB0100E ″<name>″ ist kein gültiger Tabellenname.

Ursache: Eine Tabelle mit dem angegebenen Namen existiert nicht.

Aktion: Überprüfen Sie den Namen und ob die Datei existiert, und wiederholen Sie die Operation.

DMB0101E Ungültiger Parameter NULL.

Ursache: Der Befehl erwartete einen Parameter, der ungleich Null ist.

Aktion: Überprüfen Sie die Syntax, und wiederholen Sie den Befehl.

DMB0102E Ungültiger Speichertyp.

Ursache: Für die DB2 Extender gibt der Speichertyp an, wo die Multimediadaten gespeichert sind.

Aktion: Geben Sie 0 für 'intern' (Datei) oder 1 für 'extern' (Datenbank) an.

DMB0103E Format nicht unterstützt.

Ursache: Die DB2 Extender unterstützen das Format dieses Objekts nicht.

Aktion: Setzen Sie das Objekt in ein unterstütztes Format um.

DMB0104E Der Videoinhaltsspeicher ist zu klein.

Ursache: Der Videoclip ist für den Puffer, der dafür zugeordnet ist, zu groß.

Aktion: Ordnen Sie einen größeren Puffer zu.

DMB0105E Der MPEG1-Kennsatz ist ungültig.

Ursache: Die Kopfzeile der MPEG1-Datei fehlt oder ist beschädigt.

Aktion: Überprüfen Sie, dass die Datei eine MPEG1-Datei ist.

DMB0106E Der AVI-Kennsatz ist ungültig.

Ursache: Die Kopfzeile der AVI-Datei fehlt oder ist beschädigt.

Aktion: Überprüfen Sie, dass die Datei eine AVI-Datei ist.

DMB0107E Die Exportumgebung ist nicht definiert.

Ursache: Bei den DB2 Extendern sind die Umgebungsvariablen für die Exportumgebung nicht korrekt definiert.

Aktion: Stellen Sie sicher, dass die Umgebungsvariablen korrekt definiert sind, wie in Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521 beschrieben.

DMB0108E Die Importumgebung ist nicht definiert.

Ursache: Bei den DB2 Extendern sind die Umgebungsvariablen für die Importumgebung nicht korrekt definiert.

Aktion: Stellen Sie sicher, dass die Umgebungsvariablen korrekt definiert sind, wie in Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521 beschrieben.

DMB0109E Die Importdatei kann nicht formatiert werden.

Ursache: Eine Importdatei mit dem angegebenen Namen ist nicht vorhanden.

Aktion: Überprüfen Sie den Namen und ob die Datei existiert, und stellen Sie sicher, dass die Umgebungsvariablen korrekt definiert sind, wie im Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521 beschrieben.

DMB0110E Die Exportdatei kann nicht formatiert werden.

Ursache: Eine Exportdatei mit dem angegebenen Namen ist nicht vorhanden.

Aktion: Überprüfen Sie den Namen und ob die Datei existiert, und stellen Sie sicher, dass die Umgebungsvariablen korrekt definiert sind, wie im Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521 beschrieben.

DMB0111E Die Speicherumgebung ist nicht definiert.

Ursache: Die Umgebungsvariablen für die Speicherumgebung sind nicht korrekt definiert.

Aktion: Stellen Sie sicher, dass die Umgebungsvariablen korrekt definiert sind, wie im Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521 beschrieben.

DMB0112E Die Speicherdatei kann nicht formatiert werden.

Ursache: Eine Speicherdatei mit dem angegebenen Namen ist nicht vorhanden.

Aktion: Überprüfen Sie den Namen und ob die Datei existiert, und stellen Sie sicher, dass die Umgebungsvariablen korrekt definiert sind, wie im Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521 beschrieben.

DMB0113E Die Importdatei kann nicht geöffnet werden.

Ursache: Die Datei ist möglicherweise von einem anderen Benutzer gesperrt, oder die Datei fehlt oder ist beschädigt.

Aktion: Überprüfen Sie den Namen, die Existenz und den Status der Datei und Ihre Berechtigungsstufe.

DMB0114E Die Exportdatei kann nicht geöffnet werden.

Ursache: Die Datei ist möglicherweise von einem anderen Benutzer gesperrt, oder die Datei fehlt oder ist beschädigt.

Aktion: Überprüfen Sie den Namen, die Existenz und den Status der Datei und Ihre Berechtigungsstufe.

DMB0115E Die Speicherdatei kann nicht geöffnet werden.

Ursache: Das System versucht, eine Datei zu schreiben, aber die Datei existiert bereits. Der Server hat keine Berechtigung, die Datei zu überschreiben.

Aktion: Überprüfen Sie den Namen, die Existenz und den Status der Datei und Ihre Berechtigungsstufe.

DMB0116E Temporäre Datei kann nicht erstellt werden.

Ursache: Möglicherweise ist nicht genügend Speicherplatz vorhanden, um die temporäre Datei zu erstellen.

Aktion: Stellen Sie sicher, dass die Umgebungsvariablen für die temporäre Datei für den Extender korrekt definiert sind. Vergrößern Sie gegebenenfalls den Speicher.

DMB0117E Die temporäre Umgebung wurde nicht eingerichtet.

Ursache: Die Umgebungsvariablen für die temporäre Umgebung sind nicht korrekt definiert.

Aktion: Stellen Sie sicher, dass die Umgebungsvariablen korrekt definiert sind, wie in Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521 beschrieben.

DMB0118E Die temporäre Datei kann nicht geöffnet werden.

Ursache: Die temporäre Datei wurde möglicherweise überschrieben oder beschädigt.

Aktion: Stellen Sie sicher, dass die Umgebungsvariablen korrekt definiert sind, wie in Anhang A, „Umgebungsvariablen für DB2 Extender einstellen", auf Seite 521 beschrieben.

DMB0119I Der dmbsrv-Server wird für ″<name>″ mit ″<anzahl>″ Verbindungen gestartet.

Ursache: Die Nachricht gibt an, wie viele Verbindungen aufgebaut werden, wenn der Server gestartet wird.

Aktion: Keine Aktion erforderlich.

DMB0120E Der dmbsrv-Server konnte nicht für ″<name>″ mit ″<anzahl>″ Verbindungen gestartet werden.

Ursache: DB2 wurde möglicherweise noch nicht gestartet, die Datenbank existiert möglicherweise nicht, oder auf dem System stehen keine weiteren lizenzierten Verbindungen zur Verfügung.

Aktion: Stellen Sie sicher, dass DB2 gestartet wurde und dass die Datenbank existiert. Bleibt das Problem weiterhin bestehen, benachrichtigen Sie IBM, um weitere Lizenzen zu erhalten.

DMB0121I Der dmbsrv-Server wurde für ″<name>″ mit ″<anzahl>″ Verbindungen gestartet.

Ursache: Die Nachricht gibt an, wie viele Verbindungen aufgebaut werden, wenn der Server gestartet wird.

Aktion: Keine Aktion erforderlich.

DMB0122I Der dmbssd-Server ist bereit.

Ursache: Der Server ist zur Ausführung Ihrer Anwendung bereit.

Aktion: Keine Aktion erforderlich.

DMB0129E Ungültige Operation: ″<operationsname>″.

Ursache: Ein Befehl oder eine API mit dem angegebenen Namen ist nicht vorhanden.

Aktion: Überprüfen Sie den Befehl oder die API, und wiederholen Sie die Operation.

DMB0130E Für Spalte ″<spaltenname>″ konnte keine Bindeoperation mit der SQL-Anweisung durchgeführt werden.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0131E SQL-Anweisung Prepare schlug fehl.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0132E SQL-Parameter Set schlug fehl.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0133E SQL-Anweisung Execute schlug fehl.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0134E Die Dateiformatumsetzung schlug fehl.

Ursache: Das Format der gespeicherten Multimediadaten wird bei der Formatumsetzung nicht unterstützt.

Aktion: Sie können das Format dieser Datei nicht umsetzen.

DMB0135E Das Piktogramm kann nicht geöffnet/gelesen werden.

Ursache: Die Piktogrammdatei fehlt oder ist beschädigt.

Aktion: Überprüfen Sie den Namen, die Existenz und die Integrität der Piktogrammdatei.

DMB0136E Die Bindedatei kann nicht gefunden werden.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0137E Die Verbindung zu DB ″<datenbankname>″ kann nicht hergestellt werden.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0138E Eine SQL-Anweisung kann nicht freigegeben werden.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0139E Der Merkmalname ″<name>″, der nach ″<schlüsselwort>″ folgt, ist ungültig.

Ursache: Der Image Extender erwartete einen gültigen Merkmalnamen in diesem Befehl.

Aktion: Wiederholen Sie den Befehl mit einem gültigen Merkmalnamen. Gültige Merkmalnamen sind:
- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

DMB0141E Das Qualifikationsmerkmal ″<kennung>″, das nach ″<schlüsselwort>″ folgt, ist nicht gültig.

Ursache: Das System kann das Qualifikationsmerkmal in dem Befehl nicht identifizieren.

Aktion: Überprüfen Sie das Qualifikationsmerkmal, und wiederholen Sie den Befehl.

DMB0142E Es wurde kein Katalog geöffnet.

Ursache: Bei den DB2 Extendern muss für den aktuellen Befehl ein QBIC-Katalog geöffnet werden.

Aktion: Öffnen Sie den QBIC-Katalog mit dem Befehl OPEN QBIC CATALOG.

DMB0143I Im QBIC-Katalog für Spalte ″<spaltenname>″ in Tabelle ″<tabellenname>″ wurde die automatische Katalogisierung auf ″<status>″ gesetzt. ″<anzahl>″ Merkmale sind vorhanden:

Ursache: Die Nachricht gibt die Anzahl von Merkmalen an, die im QBIC-Katalog für eine bestimmte Abbildspalte definiert sind, und ob das automatische Katalogisieren eingeschaltet ist.

Aktion: Keine Aktion erforderlich.

DMB0145E Die Abfragekennung ist ungültig.

Ursache: Die Abfragekennung, die Sie im API-Aufruf verwendet haben, ist ungültig.

Aktion: Überprüfen Sie die Anwendung, ob Sie die korrekte Abfragekennung abrufen können.

DMB0146E Der Merkmalklassenname ″<funktionsklasse>″ ist ungültig.

Ursache: Eine Merkmalklasse mit dem angegebenen Namen ist nicht vorhanden. Gültige Merkmalnamen sind:
- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Aktion: Korrigieren Sie den Namen des Merkmals, und wiederholen Sie die Operation.

DMB0147E Der Merkmalklassenname ″<funktionsklasse>″ fehlt oder ist ungültig.

Ursache: Gültige Merkmalnamen sind:
- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Aktion: Korrigieren Sie den Namen des Merkmals, und wiederholen Sie die Operation.

DMB0148E Das Merkmal ″<merkmalname>″ ist bereits ein Member der Abfrage.

Ursache: Die Abfrage unterstützt bereits das in der Nachricht angegebene Merkmal.

Aktion: Keine Aktion erforderlich.

DMB0149E Das Merkmal ″<merkmalname>″ ist kein Member der Abfrage.

Ursache: Die Abfrage enthält den angegebenen Merkmalnamen nicht.

Aktion: Um das Merkmal zur Abfrage hinzuzufügen, bevor Sie andere APIs aufrufen, die auf das Merkmal zugreifen, verwenden Sie die API QbQueryAddFeature.

DMB0150E Das System kann keinen Speicher zuordnen.

Ursache: Das System konnte den Speicher, der zur Unterstützung der versuchten Operation erforderlich ist, nicht zuordnen.

Aktion: Überprüfen Sie, dass in Ihrem System genügend Speicher zum Beenden der Operation zur Verfügung steht.

DMB0151E Der Zeiger auf den Rückgabewert ist NULL.

Ursache: Der API-Aufruf wurde nicht erfolgreich ausgeführt, da der Zeiger auf einen Rückgabewert nicht NULL sein darf.

Aktion: Stellen Sie sicher, dass gültige Parameter für den API-Aufruf zur Verfügung gestellt werden und dass die Syntax befolgt wird.

DMB0152E Der Zeiger auf den Listenrückgabewert ist NULL.

Ursache: Der API-Aufruf wurde nicht erfolgreich ausgeführt, da der Zeiger auf einen Rückgabewert nicht NULL sein darf.

Aktion: Stellen Sie sicher, dass gültige Parameter für den API-Aufruf zur Verfügung gestellt werden und dass die Syntax befolgt wird.

DMB0153E Der Bereichsparameter ist reserviert und muss 0 sein.

Ursache: Der Parameter ist für die zukünftige Verwendung reserviert.

Aktion: Setzen Sie den Bereich auf 0.

DMB0154E Der Zeiger auf den Merkmalklassennamen ist ungültig.

Ursache: Der API-Aufruf erwartete einen gültigen Zeiger auf den Klassennamen des Eingabemerkmals.

Aktion: Stellen Sie sicher, dass gültige Parameter für den API-Aufruf zur Verfügung gestellt werden und dass die Syntax befolgt wird.

DMB0155I An die Funktion ″<funktionsname>″ wurde eine Puffergröße von Null übergeben.

Ursache: Für den API-Aufruf ist es erforderlich, dass der Puffer Informationen zurückgibt.

Aktion: Keine Aktion erforderlich.

DMB0156E Der Zeiger für QbImageSource ist NULL.

Ursache: Der Wert NULL gibt an, dass die Struktur nicht geändert werden soll.

Aktion: Keine Aktion erforderlich.

DMB0157E Der Typ "<typ>" für QbImageSource ist ungültig.

Ursache: Die Datenstruktur, auf die durch diese DB2 Extender-API verwiesen wird, hat den falschen Datentyp.

Aktion: Der Datentyp der Struktur sollte QbImageSource sein.

DMB0159E Der Zeiger auf den Abbildpuffer für QbImageSource ist NULL.

Ursache: Der API-Aufruf erwartete, dass ein Zeiger zurückgegeben wird.

Aktion: Überprüfen Sie die Anwendung, ob der API-Aufruf und der Puffer korrekt angegeben sind.

DMB0160I Die Länge des Abbildpuffers oder der Abbilddatei ist Null.

Ursache: Die Länge ist Null.

Aktion: Keine Aktion erforderlich.

DMB0161E Der Zeiger auf den Tabellen- und/oder Spaltennamen ist NULL.

Ursache: Der API-Aufruf erwartete, dass ein Zeiger zur Verfügung gestellt wird.

Aktion: Überprüfen Sie die Anwendung, ob die Eingabe für den API-Aufruf korrekt angegeben ist.

DMB0162I Sie haben den Wert für requestedHits auf Null gesetzt.

Ursache: Wenn Sie 'requestedHits' auf Null setzen, erhalten Sie keinen Rückgabewert.

Aktion: Keine Aktion erforderlich.

DMB0163I Diese Funktion wird noch nicht unterstützt.

Ursache: Diese Funktion wird noch nicht unterstützt.

Aktion: Keine Aktion erforderlich.

DMB0164E Das System kann die Abfrage "<abfragename>" nicht verarbeiten.

Ursache: Beim Erstellen der Abfrage ist ein Fehler aufgetreten.

Aktion: Überprüfen Sie die Eingabe für den Befehl oder die API, und wiederholen Sie die Operation.

DMB0165E Das System kann die Abfrage "<abfragename>" nicht ausführen.

Ursache: Beim Erstellen der Abfrage ist ein Fehler aufgetreten.

Aktion: Überprüfen Sie die Eingabe für den Befehl oder die API, und wiederholen Sie die Operation.

DMB0166E In "<name>" trat während der Ausführung von "<name>" ein Anweisungsfehler auf: "<fehler>"

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wenden Sie sich an den Datenbankadministrator.

DMB0167E Beim Lesen von QbGenericImageDataClass trat ein Fehler auf (<fehler>).

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wenden Sie sich an den Datenbankadministrator.

DMB0168E Das Merkmal "<merkmalname>" einer Abfrage wurde vor der Suche nicht gesetzt.

Ursache: Die Abfrage wurde nicht ausgeführt, da ihr kein Merkmal zugeordnet wurde.

Aktion: Fügen Sie ein Merkmal zur Abfrage hinzu, indem Sie entweder die API QbAddFeature oder den Befehl ADD QBIC FEATURE verwenden.

DMB0169E In der DMB-Befehlszeile trat folgender Fehler auf: "<fehler>".

Ursache: CLI-Fehler.

Aktion: Befolgen Sie die Anweisungen im Nachrichtentext.

DMB0170E Der Abfragename "<abfragename>" ist bereits im Gebrauch.

Ursache: Eine andere Abfrage mit dem angegebenen Namen existiert bereits.

Aktion: Wählen Sie einen anderen Namen.

DMB0171E Der Abfragename "<abfragename>" wurde nicht gesichert.

Ursache: Nach dem Erstellen der Abfrage konnte das System sie nicht speichern.

Aktion: Stellen Sie sicher, dass Sie über Schreibberechtigung verfügen und genügend Speicherplatz

vorhanden ist, um die Abfrage zu speichern.

DMB0172E SQL-Fehler: ″<fehler>″.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Befolgen Sie die Anweisungen, die in der zugehörigen Fehlernachricht enthalten sind, und wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0173E Der Katalog ist geöffnet, aber nur im Lesezugriff: ″<katalogname>″.

Ursache: Sie können den Katalog nicht aktualisieren, weil ein anderer Benutzer den Katalog bereits im Schreibzugriff geöffnet hat oder weil Sie nicht über Schreibberechtigung für den Katalog verfügen.

Aktion: Warten Sie, bis der andere Benutzer die Arbeit mit dem Katalog beendet hat, führen Sie die Anwendung über eine andere Benutzer-ID aus, oder fordern Sie den Datenbankadministrator auf, die Berechtigungsstufe für die aktuelle Benutzer-ID zu ändern.

DMB0174E Ein Systemfehler ist aufgetreten: ″<fehler>″.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Befolgen Sie die Anweisungen, die in der zugehörigen Fehlernachricht enthalten sind, und wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0175I Abbilder nicht gefunden: ″<information>″.

Ursache: Es wurden keine Abbilder gefunden, die mit der Abfrage übereinstimmen. Die Datenbank ist möglicherweise leer.

Aktion: Keine Aktion erforderlich.

DMB0176I Die Spalte hat bereits einen QBIC-Katalog: ″<tabellenname spaltenname>″.

Ursache: Ein anderer Katalog mit dem angegebenen Namen existiert bereits.

Aktion: Keine Aktion erforderlich.

DMB0177E Das System kann den Katalog nicht öffnen; die Fehlermeldung ist: ″<fehler>″.

Ursache: Der Katalog wurde beschädigt.

Aktion: Befolgen Sie die Anweisungen im Nachrichtentext.

DMB0178E Das System kann den Katalog nicht löschen; die Fehlermeldung ist: ″<fehler>″.

Ursache: Entweder existiert der Katalog nicht oder er wurde beschädigt.

Aktion: Überprüfen Sie den Namen und ob der Katalog existiert, und wiederholen Sie die Operation.

DMB0179E Die Katalogkennung ist ungültig: ″<fehler>″.

Ursache: Die Katalogkennung, die Sie im API-Aufruf verwendet haben, ist ungültig.

Aktion: Überprüfen Sie die Anwendung, ob Sie die korrekte Katalogkennung abrufen können.

DMB0180I Der Zugriff auf den Katalog wurde verweigert: ″<fehler>″.

Ursache: Der Zugriff wurde verweigert.

Aktion: Keine Aktion erforderlich.

DMB0181I Der Katalog ist im Gebrauch: ″<fehler>″.

Ursache: Eine andere Operation verwendet diesen Katalog.

Aktion: Keine Aktion erforderlich.

DMB0184I Der Trace wurde bereits gestartet:

Ursache: Der Trace wurde bereits gestartet.

Aktion: Keine Aktion erforderlich.

DMB0185I Der Trace wurde noch nicht gestartet.

Ursache: Der Trace wurde noch nicht gestartet.

Aktion: Keine Aktion erforderlich.

DMB0186I Der Trace wurde um ″<zeit>″ von Verzeichnis ″<verzeichnisname>″ aus gestartet. Die Tracedatei ist ″<dateiname>″. ″<anzahl>″ Byte an Tracedaten wurden bis jetzt geschrieben.

Ursache: Der Trace ist eingeschaltet.

Aktion: Keine Aktion erforderlich.

DMB0187E Die Kommunikation kann nicht gestartet werden, da das System die Datei ″<dateiname>″ nicht zum Schreiben öffnen kann.

Ursache: Entweder sind Sie nicht der Eigner des aktuellen Exemplars, das durch die Umgebungsvariable DB2INSTANCE beschrieben wird, oder die

Umgebungsvariablen, wie etwa DB2MMTOP, sind nicht korrekt definiert.

Aktion: Melden Sie sich mit der Benutzer-ID an, die der Eigner des Exemplars ist. Überprüfen Sie, dass die Umgebungsvariablen korrekt definiert sind.

DMB0188I Beim Erstellen des Tracedämonen trat ein Fehler auf: <fehler>″

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0189I Der Trace wurde erfolgreich gestartet:

Ursache: Der Trace wurde bereits gestartet.

Aktion: Keine Aktion erforderlich.

DMB0190E Der Trace kann nicht gestartet werden.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0191E Die Umgebungsvariable ″<name>″ muss gesetzt werden.

Ursache: Die Systemkonfiguration ist nicht korrekt.

Aktion: Definieren Sie die Variable, und wiederholen Sie die Operation.

DMB0192I Der Trace wurde erfolgreich gestoppt.

Ursache: Der Trace ist ausgeschaltet.

Aktion: Keine Aktion erforderlich.

DMB0193E Das System kann nicht in die Datei ″<dateiname>″ schreiben.

Ursache: Sie verfügen nicht über Schreibberechtigung für das Verzeichnis der angegebenen Datei.

Aktion: Wenden Sie sich an den Datenbankadministrator, um die Berechtigung zu erhalten.

DMB0194E Das System kann nicht aus der Datei ″<dateiname>″ lesen.

Ursache: Entweder existiert die Datei nicht oder Sie verfügen nicht über Leseberechtigung für die Datei.

Aktion: Stellen Sie sicher, dass die Datei existiert und dass Sie über Leseberechtigung für die Datei verfügen.

DMB0198E Der Trace-Code ″<code>″ in der Eingabedatei ist unbekannt. Die Eingabedatei ist möglicherweise beschädigt.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0199E Sie haben für keine der Tabellen, auf die verwiesen wird, die ″<berechtigungsstufe>″-Berechtigung.

Ursache: Ihre Benutzer-ID hat nicht die Berechtigungsstufe, die für die Operation erforderlich ist.

Aktion: Führen Sie die Operation über eine andere Benutzer-ID aus, oder fordern Sie den Datenbankadministrator auf, die Berechtigungsstufe für die aktuelle Benutzer-ID zu ändern.

DMB0200W Sie haben nicht die ″<berechtigungsstufe>″-Berechtigung für mindestens eine der Tabellen, auf die verwiesen wird.

Ursache: Ihre Benutzer-ID hat nicht die Berechtigungsstufe, die für einige Tabellen erforderlich ist.

Wenn Sie Dateien auflisten, auf die verwiesen wird, werden nur die Dateien aufgelistet, auf die in Tabellen verwiesen wird, für die Sie die Berechtigung zum Auswählen (SELECT) haben. Wenn auf dem System Tabellen vorhanden sind, für die Sie keine Berechtigung zum Auswählen haben, werden Dateien, auf die in diesen Tabellen verwiesen wird, nicht aufgelistet.

Wenn Sie Metadaten reorganisieren, reorganisiert das System nur Metadaten für Tabellen, für die Sie über die Berechtigung zum Steuern (CONTROL) verfügen.

Aktion: Um alle Dateien abzurufen, führen Sie die Operation über eine andere Benutzer-ID aus, oder fordern Sie den Datenbankadministrator auf, die Berechtigungsstufe für die aktuelle Benutzer-ID zu ändern.

DMB0201I Ein Merkmal mit diesem Namen existiert bereits: ″<merkmalname>″.

Ursache: Ein Merkmal mit dem angegebenen Namen ist bereits im QBIC-Katalog enthalten.

Aktion: Keine Aktion erforderlich.

DMB0202E Der Merkmalname ist ungültig: ″<merkmalname>″.

Ursache: Eine Merkmalklasse mit dem angegebenen Namen ist nicht vorhanden. Gültige Merkmalnamen sind:

- QbColorFeatureClass

- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Aktion: Korrigieren Sie den Namen des Merkmals, und wiederholen Sie die Operation.

DMB0203E Das Merkmal konnte nicht gefunden werden: ″<merkmalname>″.

Ursache: Eine Merkmalklasse mit dem angegebenen Namen ist nicht vorhanden oder die Merkmalklasse ist nicht im QBIC-Katalog enthalten. Gültige Merkmalnamen sind:

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

Aktion: Korrigieren Sie den Namen des Merkmals, und wiederholen Sie die Operation.

DMB0204E Die Spalte ist für DB2IMAGE nicht aktiviert: ″<spaltenname>″.

Ursache: Die Spalte ist nicht für den Image Extender aktiviert.

Aktion: Stellen Sie sicher, dass die Spalte für den DB2 Image Extender aktiviert ist.

DMB0205E Kein QBIC-Katalog für ″<tabellenname spaltenname>″ gefunden.

Ursache: Der angegebenen Spalte ist kein QBIC-Katalog zugeordnet.

Aktion: Erstellen Sie einen QBIC-Katalog für die Spalte, bevor Sie andere QBIC-Operationen ausführen.

DMB0206W Die angegebene Spalte ist für keinen Extender aktiviert.

Ursache: Die Spalte existiert möglicherweise nicht, oder ihr Datentyp ist möglicherweise nicht mit den Extendern kompatibel.

Aktion: Überprüfen Sie, dass die Spalte unter Verwendung des korrekten Datentyps definiert wurde.

DMB0207E Die Datei kann nicht überschrieben werden.

Ursache: Die Datei existiert bereits, aber die UDF EXPORT kann sie nicht überschreiben.

Aktion: Exportieren Sie die Datei in eine Datei mit einem anderen Namen oder ermöglichen Sie der UDF EXPORT, die Datei zu überschreiben.

DMB0208E Sqlcode=<code> Clistate=<code>.

Ursache: Ein interner Fehler ist aufgetreten.

Aktion: Wiederholen Sie die Operation. Tritt der Fehler erneut auf, benachrichtigen Sie den IBM Kundendienst.

DMB0209E Ungültiger Audiokennsatz.

Ursache: Die Kopfzeile der Audiodatei fehlt oder ist beschädigt.

Aktion: Überprüfen Sie, ob das Format der Audiodatei von den DB2 Extendern unterstützt wird.

DMB0211W Die Datei existiert ohne Überschreiben.

Ursache: Die angegebene Zieldatei existiert bereits und wird nicht überschrieben.

Aktion: Keine Aktion erforderlich.

DMB0212E Der Parameter resultType ist reserviert und muss 0 sein.

Ursache: Der Parameter ist für die zukünftige Verwendung reserviert.

Aktion: Setzen Sie 'resultType' auf 0.

DMB0214E Der Zeiger auf den Abfragenamen ist ungültig.

Ursache: Der API-Aufruf erwartete einen gültigen Zeiger auf den Eingabeabfragenamen.

Aktion: Stellen Sie sicher, dass gültige Parameter für den API-Aufruf zur Verfügung gestellt werden und dass die Syntax befolgt wird.

DMB0352E Umgebung für Befehlszeile nicht initialisiert.

Ursache: Die Befehlszeilenumgebung ist nicht dazu initialisiert, den db2ext-Befehlszeilenprozessor auszuführen. (Diese Nachricht gilt nur für Windows NT- und Windows 95-Umgebungen.)

Aktion: Geben Sie den Befehl db2cmd ein, um ein DB2CLP-Fenster zu öffnen, und geben Sie anschließend den Befehl db2ext ein, um den db2 -Befehlszeilenprozessor in diesem Fenster auszuführen.

DMB0353E Kommunikation mit Hintergrundprozess für db2ext-Befehlszeilenprozessor nicht möglich.

Ursache: Der Hintergrundprozess für den db2ext-Befehlszeilenprozessor ist aktiv, aber der db2ext-Befehlszeilenprozessor kann nicht mit dem Prozess kommunizieren.

Aktion: Versuchen Sie, den Befehl db2ext in einem

anderen Fenster einzugeben.

DMB0354E Hintergrundprozess für db2ext-Befehlszeilenprozessor kann nicht gestartet werden.

Ursache: Der Hintergrundprozess für den db2ext-Befehlszeilenprozessor ist aktiv, aber der db2ext-Befehlszeilenprozessor kann nicht mit dem Prozess kommunizieren.

Aktion: Überprüfen Sie, ob das ausführbare Modul für den Hintergrundprozess (db2extb oder db2extb.exe) existiert und ob das entsprechende Verzeichnis in der Umgebungsvariable PATH angegeben ist.

DMB0355E Zeitlimitüberschreitung für Hintergrundprozess für db2ext-Befehlszeilenprozessor.

Ursache: Der Hintergrundprozess für den db2ext-Befehlszeilenprozessor wurde erfolgreich gestartet, aber der db2ext-Befehlszeilenprozessor kann innerhalb des zulässigen Zeitlimits nicht mit dem Prozess kommunizieren.

Aktion: Versuchen Sie, den Befehl db2ext in einem anderen Fenster einzugeben.

DMB0356E Kommunikation mit Hintergrundprozess für db2ext-Befehlszeilenprozessor nicht möglich.

Ursache: Der db2ext-Befehlszeilenprozessor hat eine Anforderung an seinen Hintergrundprozess gesendet, aber die Anforderung wurde nicht empfangen.

Aktion: Stellen Sie sicher, dass der Hintergrundprozess für den db2ext-Befehlszeilenprozessor noch immer aktiv ist.

DMB0357E Hintergrundprozess für db2ext-Befehlszeilenprozessor antwortet nicht.

Ursache: Der db2ext-Befehlszeilenprozessor hat eine Anforderung an seinen Hintergrundprozess gesendet, aber der Hintergrundprozess hat nicht innerhalb des zulässigen Zeitlimits geantwortet.

Aktion: Stellen Sie sicher, dass der Hintergrundprozess für den db2ext-Befehlszeilenprozessor noch immer aktiv ist.

DMB0359E Anforderungswarteschlange für Hintergrundprozess für db2ext-Befehlszeilenprozessor oder Eingabewarteschlange wurde nicht innerhalb des Zeitlimits erstellt.

Ursache: Der Hintergrundprozess für den db2ext-Befehlszeilenprozessor hat während des zulässigen Zeitlimits keine Nachrichtenwarteschlangen erstellt. (Diese Nachricht gilt nur für UNIX-Umgebungen.)

Aktion: Stellen Sie sicher, dass die Platte, auf der sich das Benutzerverzeichnis des DB2-Exemplars befindet, nicht voll ist (der Hintergrundprozess benötigt dieses Verzeichnis, um eine Datei für Nachrichtenwarteschlangen zu erstellen). Wenn die Platte nicht voll ist, überprüfen Sie, ob Sie zu viele db2ext-Prozesse gestartet haben. Dies ist möglich, wenn der db2ext-Befehlszeilenprozessor in vielen Fenstern aktiv ist. Ein Hintergrundprozess wird in einem Fenster gestartet, wenn Sie das erste Mal eine Anforderung für einen db2ext-Befehlszeilenprozessor im Befehlsmodus eingeben. Stellen Sie sicher, dass Sie den Befehl `db2ext terminate` eingeben, um den db2ext-Befehlszeilenprozessor, wenn Sie ihn nicht länger benötigen, zu beenden. Nachrichtenwarteschlangen für den Backend-Prozess werden nur gelöscht, wenn Sie den Befehl `terminate` eingeben.

DMB0361E Spalte oder Tabelle nicht aktiviert.

Ursache: Eine Import-UDF wurde angegeben, aber die angegebene Tabellenspalte ist nicht für den Extender aktiviert.

Aktion: Aktivieren Sie die Tabellenspalte, und wiederholen Sie die Operation.

DMB0363E Fehlender Tabellen- oder Spaltenname.

Ursache: Eine Aktualisierungs-UDF wurde aufgerufen, aber es wurde keine Tabelle angegeben.

Aktion: Geben Sie eine Tabelle an, und wiederholen Sie die Operation.

DMB0364E Der ″<extender-name>″ Extender wurde zuvor für den Tabellenbereich ″<tabellenbereichsname>″ aktiviert.

Ursache: Die angegebene Datenbank, Tabelle oder Spalte wurde bereits für den Extender aktiviert, aber unter Verwendung eines anderen Tabellenbereichs als der angegebene.

Aktion: Überprüfen Sie, ob die Angabe zum Tabellenbereich korrekt ist.

DMB0365E Sie haben keine Berechtigung CONTROL für ″<metadatentabellenname>″ und ″<metadatentabellenname>″, die Metadatentabellen für die Benutzertabelle ″<schemaname>″.″<tabellenname>″ sind.

Ursache: Ihre Anforderung wurde zurückgewiesen, da Sie nicht über die erforderliche Berechtigung CONTROL für die Metadatentabellen zur angegebenen Benutzertabelle verfügen.

Aktion: Lassen Sie sich vom zuständigen Datenbank-

administrator die Berechtigung CONTROL für die Metadatentabellen erteilen.

DMB0366E Merkmalname erwartet.

Ursache: In der Abfragezeichenfolge wird ein Merkmalname erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0367E Farbe | Histogramm | Datei erwartet.

Ursache: In der Abfragezeichenfolge wird entweder „Farbe", „Histogramm" oder „Datei" erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0368E ',' erwartet.

Ursache: In der Abfragezeichenfolge wird ',' erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0369E Datei ist nicht gültig.

Ursache: Die in der Abfragezeichenfolge angegebene Datei ist nicht gültig.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0370E Dateiname erwartet.

Ursache: In der Abfragezeichenfolge wird ein Dateiname erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0371E Server | Client erwartet.

Ursache: In der Abfragezeichenfolge wird entweder „Server" oder „Client" erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0372E '(' erwartet.

Ursache: In der Abfragezeichenfolge wird '(' erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0373E ')' erwartet.

Ursache: In der Abfragezeichenfolge wird ')' erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0374E Prozentsatz erwartet.

Ursache: In der Abfragezeichenfolge wird ein Prozentwert erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0375E Farbe erwartet.

Ursache: In der Abfragezeichenfolge werden Werte für rot, grün und blau erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0376E '=' erwartet.

Ursache: In der Abfragezeichenfolge wird '=' erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0377E '<' erwartet.

Ursache: In der Abfragezeichenfolge wird '<' erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0378E '>' erwartet.

Ursache: In der Abfragezeichenfolge wird '>' erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0379E 'and' erwartet.

Ursache: In der Abfragezeichenfolge wird ein 'and' erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0380E Wertigkeit erwartet.

Ursache: In der Abfragezeichenfolge wird eine Wertigkeit erwartet.

Aktion: Korrigieren Sie die Abfragezeichenfolge, und wiederholen Sie die Operation.

DMB0381E Merkmal nicht definiert.

Ursache: Das Merkmal wurde nicht zum QBIC-Katalog hinzugefügt.

Aktion: Fügen Sie das Merkmal zum QBIC-Katalog hinzu und katalogisieren Sie die Abbilder erneut.

DMB0382E Abfrage konnte nicht erstellt werden.

Ursache: Der Extender-Server für die aktuelle Datenbank wurde möglicherweise gestoppt.

Aktion: Überprüfen Sie den Status des Servers. Wenn der Server nicht aktiv ist, starten Sie ihn erneut unter Verwendung des Befehls START SERVER von der db2ext-Befehlszeile aus.

DMB0383E Abfrage konnte nicht ausgeführt werden.

Ursache: Der Extender-Server für die aktuelle Datenbank wurde möglicherweise gestoppt.

Aktion: Überprüfen Sie den Status des Servers. Wenn der Server nicht aktiv ist, starten Sie ihn erneut unter Verwendung des Befehls START SERVER von der db2ext-Befehlszeile aus.

DMB0384E Nächstes Element konnte nicht abgerufen werden.

Ursache: Das Ende der Liste wurde erreicht.

Aktion: Stellen Sie sicher, dass die Anwendung nicht versucht, ein Element nach dem Ende der Liste abzurufen.

DMB0386E Benutzerdaten können nicht zusammengestellt werden.

Ursache: Die SQL-API sqluihsh() hat einen Rückkehrcode ungleich Null zurückgegeben.

Aktion: Wiederholen Sie die Operation. Bleibt das Problem bestehen, benachrichtigen Sie die IBM Unterstützungsfunktion.

DMB0387E Die Knotengruppe für die angegebenen Tabellenbereiche weicht von der Knotengruppe der Benutzertabelle ab.

Ursache: Ein oder mehrere Tabellenbereiche (d. h. für die Metadatentabelle, den Index oder das BLOB), die als Eingabe zum Aktivieren einer Tabelle übergeben wurden, sind in einer Knotengruppe definiert, die von der Knotengruppe abweicht, in der die Benutzertabelle definiert ist.

Aktion: Verwenden Sie Tabellenbereiche, die in derselben Knotengruppe definiert sind wie die zu aktivierende Benutzertabelle.

DMB0388E Die regulären, langen oder Indextabellenbereiche sind nicht in derselben Knotengruppe definiert.

Ursache: Ein oder mehrere Tabellenbereiche (d. h. für die Metadatentabelle, den Index oder das BLOB), die als Eingabe zum Aktivieren einer Datenbank übergeben wurden, sind nicht in derselben Knotengruppe definiert wie die übrigen Tabellenbereiche.

Aktion: Verwenden Sie Tabellenbereiche, die in derselben Knotengruppe definiert sind.

DMB0389W Die Knotengruppe für die angegebenen Tabellenbereiche enthält nicht alle Partitionsserver.

Ursache: Die Tabellenbereiche, die als Eingabe übergeben wurden, sind in einer Knotengruppe definiert, die nicht alle Partitionsserver einschließt.

Aktion: Keine Aktion erforderlich. Die Import- und Aktualisierungs-UDFs arbeiten allerdings effizienter, wenn die Tabellenbereiche in einer Knotengruppe definiert sind, die alle Partitionsserver umfasst. Dies gilt besonders, wenn die Extender-Anwendungen den Inhalt von Mediadaten im BLOB-Format speichern.

DMB0391I Dieser Befehl ist nur für einen DB2 UDB-Client gültig, der auf einen DB2 UDB-Server zugreift.

Ursache: Entweder ist der db2ext-Befehlszeilenprozessor nicht mit einem DB2 UDB-Server verbunden oder der db2ext-Befehlszeilenprozessor wurde nicht von einem DB2 UDB-Client gestartet. Beispielsweise ist der Befehl START SERVER nur gültig, wenn der db2ext-Befehlszeilenprozessor mit einem DB2-Server verbunden ist, der kein Extended Enterprise Edition-Server ist.

Aktion: Geben Sie diesen Befehl in der aktuellen Client/Server-Konfiguration nicht ein.

DMB0392I Dieser Befehl ist nur für einen DB2 UDB-Client gültig, der auf einen DB2 UDB Extended Enterprise Edition-Server zugreift. Beispielsweise ist der Befehl DISCONNECT SERVER nur gültig, wenn der db2ext-Befehlszeilenprozessor mit einem DB2 Extended Enterprise Edition-Server verbunden ist.

Ursache: Entweder ist der db2ext-Befehlszeilenprozessor nicht mit einem DB2 UDB Extended Enterprise Edition-Server verbunden oder der db2ext-Befehlszeilenprozessor wurde nicht von einem DB2 UDB-Client gestartet.

Aktion: Geben Sie diesen Befehl in der aktuellen Client/Server-Konfiguration nicht ein.

DMB0402E Die Option ″<optionsname>″ für den Befehl ″<befehlsname>″ ist nur gültig, wenn die Anwendung mit einem DB2 ″<servertyp>″-Server verbunden ist.

Ursache: Der angegebene Parameter ist nicht gültig, da der db2ext-Befehlszeilenprozessor nicht mit dem Typ von Server verbunden ist, der diese Option unterstützt. Beispielsweise kann der Befehl GET SERVER

STATUS nur mit dem Parameter NODENUM <knotennummer> angegeben werden, wenn der db2ext-Befehlszeilenprozessor mit einem DB2 Extended Enterprise Edition-Server verbunden ist.

Aktion: Geben Sie diese Kombination aus Befehl und Parameter in der aktuellen Client/Server-Konfiguration nicht ein.

DMB0411E Ungültiger Basisanschluss

Ursache: Während der Exemplarerstellung wurde eine ungültige TCP/IP-Anschlussnummer als Basisanschluss angegeben.

Aktion: Die korrekte Syntax lautet dmbicrt -r:basisanschluss,endanschluss -t:basisanschluss,endanschluss. Korrigieren Sie die Parameter, und wiederholen Sie den Befehl.

DMB0412E Ungültiger Endanschluss

Ursache: Während der Exemplarerstellung wurde eine falsche TCP/IP-Anschlussnummer als Endanschluss angegeben.

Aktion: Die korrekte Syntax lautet dmbicrt -r:basisanschluss,endanschluss -t:basisanschluss,endanschluss. Korrigieren Sie die Parameter, und wiederholen Sie den Befehl.

DMB0413E Der DB2 Extender-Installationspfad kann nicht erkannt/aufgelöst werden.

Ursache: Das Programm zur Exemplarerstellung konnte keinen Wert für die Umgebungsvariable "DMBPATH" finden.

Aktion: Legen Sie die Variable "DMBPATH" fest, und wiederholen Sie die Operation.

DMB0414E Der Computer-Host-Name kann nicht erkannt/aufgelöst werden.

Ursache: Ein interner Fehler ist aufgetreten, als versucht wurde, den Namen des Computers aufzulösen.

Aktion: Benachrichtigen Sie die IBM Unterstützungsfunktion.

DMB0415E Die Knotennummer für diese Maschine kann nicht erkannt/aufgelöst werden.

Ursache: Die Maschine, auf der die Exemplarerstellung ausgeführt wird, ist nicht in der Datei "db2nodes.cfg" aufgelistet.

Aktion: Fügen Sie die Maschine zur Datei "db2nodes.cfg" hinzu, und wiederholen Sie die Operation.

DMB0416E Dieses Programm muss vom Root-Benutzer gestartet werden. Die Operation kann nicht fortgeführt werden.

Ursache: Die Benutzer-ID, unter der dieses Programm ausgeführt wird, hat keine Root-Berechtigung.

Aktion: Melden Sie sich unter der Benutzer-ID 'Root' an, und wiederholen Sie die Operation.

DMB0417E Dieses Programm muss von einem Benutzer mit Administratorberechtigung ausgeführt werden. Die Operation kann nicht fortgeführt werden.

Ursache: Die Benutzer-ID, unter der dieses Programm ausgeführt wird, hat keine Administratorberechtigung.

Aktion: Melden Sie sich unter einer Benutzer-ID mit Administratorberechtigung an, und wiederholen Sie die Operation.

DMB0418E Informationen zum Benutzer "<benutzer-id>" können nicht abgerufen werden.

Ursache: Ein interner Fehler ist aufgetreten, als versucht wurde, Benutzerinformationen abzurufen, die dem zu erstellenden Exemplar zugeordnet sind.

Aktion: Stellen Sie sicher, dass eine gültige Benutzer-ID mit dem gleichen Namen wie das zu erstellende Exemplar vorhanden ist, und wiederholen Sie die Operation.

DMB0419E Das AIV Extender-Verzeichnis "<verzeichnisname>" kann nicht erstellt werden. Rückkehrcode = <code>

Ursache: Ein Fehler ist aufgetreten, als versucht wurde, das angegebene Verzeichnis zu erstellen. Der Rückkehrcode gibt den Fehler an, der vom Betriebssystem zurückgegeben wurde.

Aktion: Stellen Sie sicher, dass das angegebene Dateisystem/Laufwerk im entsprechenden Verzeichnis existiert und dass die Berechtigung vorliegt, ein Verzeichnis zu erstellen.

DMB0420E Die Verbindung für das AIV Extender-Verzeichnis "<verzeichnisname>" kann nicht hergestellt werden. Rückkehrcode = <code>

Ursache: Ein Fehler ist aufgetreten, als versucht wurde, die angegebene symbolische Verbindung zu erstellen. Der Rückkehrcode gibt den Fehler an, der vom Betriebssystem zurückgegeben wurde.

Aktion: Stellen Sie sicher, dass das angegebene Dateisystem/Laufwerk im entsprechenden Verzeichnis existiert und dass die Berechtigung vorliegt, eine Verbindung zu erstellen.

DMB0421E Die Datei ″<dateiname>″ kann nicht geöffnet werden. Rückkehrcode = <code>

Ursache: Ein Fehler ist aufgetreten, als versucht wurde, die angegebene Datei zu öffnen. Der Rückkehrcode gibt den Fehler an, der vom Betriebssystem zurückgegeben wurde.

Aktion: Stellen Sie sicher, dass die Datei existiert und dass die Berechtigung vorliegt, die Datei zu öffnen.

DMB0422E Die Datei ″<dateiname>″ kann nicht geschrieben werden. Rückkehrcode = <code>

Ursache: Ein Fehler ist aufgetreten, als versucht wurde, in die angegebene Datei zu schreiben. Der Rückkehrcode gibt den Fehler an, der vom Betriebssystem zurückgegeben wurde.

Aktion: Stellen Sie sicher, dass die Datei existiert und dass die Berechtigung vorliegt, in die Datei zu schreiben.

DMB0424E ″db2nodes.cfg″ kann nicht gefunden werden.

Ursache: Die DB2-Datei ″db2nodes.cfg″ konnte nicht gefunden werden.

Aktion: Stellen Sie sicher, dass die korrekte Version von DB2 UDB Extended Enterprise Edition installiert wurde, und wiederholen Sie die Operation.

DMB0426E Fehler: ″<fehlercode>″ beim Öffnen von Schlüssel ″<registrierungsschlüssel>″.

Ursache: Ein Fehler ist aufgetreten, als versucht wurde, den angegebenen Registrierungsschlüssel zu öffnen.

Aktion: Notieren Sie den Rückkehrcode und benachrichtigen Sie die IBM Unterstützungsfunktion.

DMB0427E Die Variable ″<variable>″ wurde in der Profilregistrierung nicht definiert.

Ursache: Der angegebene Wert wurde in der Windows NT-Registrierung nicht gefunden.

Aktion: Stellen Sie sicher, dass der Name einer gültigen DB2 Extender-Variablen angegeben wurde.

DMB0430E DB2-Registrierungswerte konnten nicht gefunden werden.

Ursache: Die Registrierungswerte, die von DB2 verwendet wurden, konnten nicht gefunden werden.

Aktion: Stellen Sie sicher, dass die korrekte Version von DB2 UDB Extended Enterprise Edition installiert ist, und wiederholen Sie die Operation.

DMB0431E Der Extender-Registrierungsschlüssel ″<registrierungsschlüssel>″ konnte nicht erstellt werden.

Ursache: Ein interner Fehler ist aufgetreten, als versucht wurde, einen Extender-Registrierungsschlüssel zu erstellen.

Aktion: Benachrichtigen Sie die IBM Unterstützungsfunktion.

DMB0432E Der Wert für den Extender-Registrierungsschlüssel ″<registrierungsschlüssel>″ konnte nicht definiert werden.

Ursache: Ein interner Fehler ist aufgetreten, als versucht wurde, einen Wert für den Extender-Registrierungsschlüssel festzulegen.

Aktion: Benachrichtigen Sie die IBM Unterstützungsfunktion.

DMB0435E Auf die Steuerdatei ″<steuerdatei>″ kann nicht zugegriffen werden.

Ursache: Die angegebene Steuerdatei konnte nicht gefunden werden.

Aktion: Benachrichtigen Sie die IBM Unterstützungsfunktion.

DMB0443E Das Verzeichnis ″<verzeichnisname>″ kann nicht geöffnet werden. Rückkehrcode = <code>

Ursache: Ein Fehler ist aufgetreten, als versucht wurde, das angegebene Verzeichnis zu öffnen. Der Rückkehrcode gibt den Fehler an, der vom Betriebssystem zurückgegeben wurde.

Aktion: Stellen Sie sicher, dass das angegebene Dateisystem/Laufwerk im entsprechenden Verzeichnis existiert und dass die Berechtigung vorliegt, ein Verzeichnis zu öffnen.

DMB0449W -q:datenpfad ist für die Erstellung des DB2 Extender-Exemplars erforderlich.

Ursache: Der Parameter -q wurde nicht angegeben, als versucht wurde, ein DB2 Extender-Exemplar zu erstellen.

Aktion: Geben Sie den Parameter an, und wiederholen Sie die Operation.

DMB0450W Einer oder mehrere der angegebenen ″<anschluss>″-Anschlüsse ist/sind bereits im Gebrauch.

Ursache: Ein Anschluss wurde für die Verwendung durch die DB2 Extender angegeben, der in der Servicedatei bereits als 'im Gebrauch' aufgelistet ist.

Aktion: Geben Sie Anschlüsse an, die nicht im Gebrauch sind, und wiederholen Sie die Operation.

DMB0452E Die Knotennummer "<knotennummer>" konnte in db2nodes.cfg nicht gefunden werden.

Ursache: Die Knotennummer dieser Maschine konnte in der Datei `db2nodes.cfg` nicht gefunden werden.

Aktion: Fügen Sie die Knotennummer zur Datei `db2nodes.cfg` hinzu, und wiederholen Sie die Operation.

DMB0460W Ob TCP/IP-Anschlüsse verfügbar sind, lässt sich nicht feststellen.

Ursache: Ein Fehler ist aufgetreten, als versucht wurde zu prüfen, ob die angegebenen TCP/IP-Anschlüsse bereits im Gebrauch sind.

Aktion: Stellen Sie sicher, dass die angegebenen Anschlüsse nicht in der Servicedatei als bereits im Gebrauch durch eine andere Anwendung aufgelistet sind.

DMB0462E Dieser Knoten kann nicht initialisiert werden. Rückkehrcode = <code>

Ursache: Der Extender-Systemstart ist auf einen Fehler gestoßen, als versucht wurde, den aktuellen Knoten zu initialisieren.

Aktion: Benachrichtigen Sie die IBM Unterstützungsfunktion.

DMB0495E Diese Version der AIV Extender unterstützt keine langen Namen.

Ursache: Sie haben beim Aufrufen einer Extender-Verwaltungs-API oder beim Ausgeben eines db2ext-Befehlszeilenbefehls eine lange Kennung angegeben. Die maximale Länge von Kennungen, die in dieser Version der AIV Extender unterstützt wird, lautet:

- Lokale Berechtigungs-ID (AUTHID) - 8 Zeichen
- Tabellenschema (TABSCHEMA) - 8 Zeichen
- Tabellennamen (TABNAME) - 18 Zeichen
- Spaltennamen - 18 Zeichen

Prüfen Sie den API-Aufruf bzw. den Befehl, um sicherzustellen, dass Sie kurze Kennungen verwenden.

DMB0496E Ungültiger Tabellenname oder Spaltenname angegeben.

Ursache: Sie haben beim Aufrufen einer Extender-Verwaltungs-API oder beim Ausgeben eines db2ext-Befehlszeilenbefehls eine ungültige Kennung angegeben. Wahrscheinlich ist der Kennungsname zu lang. Im Buch *Einstieg* finden Sie Informationen zur Länge von Namen bei DB2 UDB.

Prüfen Sie den API-Aufruf bzw. den Befehl, um sicherzustellen, dass Sie kurze Kennungen verwenden.

DMB497E Zugriff verweigert auf DB2MMDATAPATH.

Ursache: (Nur EEE) Sie haben ein Verzeichnis oder einen gemeinsam benutzten Namen angegeben, auf das/den nicht auf allen Knoten zugegriffen werden kann. Das Verzeichnis oder der gemeinsam benutzte Name, das/der beim Erstellen eines Exemplars der DB2 Extender angegeben wird, muss existieren und muss auf allen Knoten im Zugriff sein. Prüfen Sie, ob das Verzeichnis oder der gemeinsam benutzte Name, das/den Sie beim Erstellen des Exemplars angegeben haben, auf allen Knoten existiert und im Zugriff ist.

DMB498E Mindestens ein Teil von DB2MMDATAPATH ist kein Verzeichnis

Ursache: (Nur EEE) Sie haben ein Verzeichnis oder einen gemeinsam benutzten Namen angegeben, das/der kein Verzeichnis auf einem Knoten ist. Das Verzeichnis oder der gemeinsam benutzte Name, das/der beim Erstellen eines Exemplars der DB2 Extender angegeben wird, muss existieren und muss auf allen Knoten im Zugriff sein. Prüfen Sie, ob das Verzeichnis oder der gemeinsam benutzte Name, das/den Sie beim Erstellen des Exemplars angegeben haben, auf allen Knoten existiert und im Zugriff ist.

DMB499E DB2MMDATAPATH-Pfadzeichenfolge zu lang.

Ursache: (Nur EEE) Sie haben ein Verzeichnis oder einen gemeinsam benutzten Namen angegeben, das/der dazu führt, dass die Variable DB2MMDATAPATH zu lang ist. Das Verzeichnis oder der gemeinsam benutzte Name, das/der beim Erstellen eines Exemplars der DB2 Extender angegeben wird, muss existieren und muss auf allen Knoten im Zugriff sein. Prüfen Sie, ob das Verzeichnis oder der gemeinsam benutzte Name, das/den Sie beim Erstellen des Exemplars angegeben haben, korrekt ist, auf allen Knoten existiert und im Zugriff ist.

DMB500E Das DB2MMDATAPATH-Verzeichnis existiert nicht.

Ursache: (Nur EEE) Sie haben ein Verzeichnis oder einen gemeinsam benutzten Namen angegeben, das/der nicht auf einem Knoten existiert. Das Verzeichnis oder der gemeinsam benutzte Name, das/der beim Erstellen eines Exemplars der DB2 Extender angegeben wird, muss existieren und muss auf allen Knoten im Zugriff sein. Prüfen Sie, ob das Verzeichnis oder der gemeinsam benutzte Name, das/den Sie beim Erstellen des Exemplars angegeben haben, auf allen Knoten existiert und im Zugriff ist.

DMB501E Unbekannter stat()-Fehler bei DB2MMDATAPATH.

Ursache: (Nur EEE) Ein Problem ist aufgetreten beim Versuch, auf ein Verzeichnis oder einen gemeinsam benutzten Namen in dieser Umgebungsvariablen zuzugreifen. Das Verzeichnis oder der gemeinsam benutzte Name, das/der beim Erstellen eines Exemplars der DB2 Extender angegeben wird, muss existieren und muss auf allen Knoten im Zugriff sein. Prüfen Sie, ob das Verzeichnis oder der gemeinsam benutzte Name, das/den Sie beim Erstellen des Exemplars angegeben haben, auf allen Knoten existiert und im Zugriff ist.

DMB502E DB2MMDATAPATH existiert, ist aber kein Verzeichnis.

Ursache: (Nur EEE) Sie haben den Namen für ein Verzeichnis oder einen gemeinsam benutzten Namen angegeben, aber der Name ist nicht der Name eines Verzeichnisses oder gemeinsam benutzten Namens auf allen Knoten. Das Verzeichnis oder der gemeinsam benutzte Name, das/der beim Erstellen eines Exemplars der DB2 Extender angegeben wird, muss existieren und muss auf allen Knoten im Zugriff sein. Prüfen Sie, ob das Verzeichnis oder der gemeinsam benutzte Name, das/den Sie beim Erstellen des Exemplars angegeben haben, auf allen Knoten existiert und im Zugriff ist.

DMB503E DB2MMDATAPATH existiert, kann aber nicht gelesen werden.

Ursache: (Nur EEE) Sie haben ein Verzeichnis oder einen gemeinsam benutzten Namen angegeben, das/der nicht auf allen Knoten gelesen werden kann. Das Verzeichnis oder der gemeinsam benutzte Name, das/der beim Erstellen eines Exemplars der DB2 Extender angegeben wird, muss existieren und muss auf allen Knoten im Zugriff sein. Prüfen Sie, ob das Verzeichnis oder der gemeinsam benutzte Name, das/den Sie beim Erstellen des Exemplars angegeben haben, auf allen Knoten existiert und im Zugriff ist.

DMB504E DB2MMDATAPATH existiert, ist aber nicht beschreibbar.

Ursache: (Nur EEE) Sie haben ein Verzeichnis oder einen gemeinsam benutzten Namen angegeben, auf das/den nicht auf allen Knoten geschrieben werden kann. Das Verzeichnis oder der gemeinsam benutzte Name, das/der beim Erstellen eines Exemplars der DB2 Extender angegeben wird, muss existieren und muss auf allen Knoten im Schreib-/Lesezugriff sein. Prüfen Sie, ob das Verzeichnis oder der gemeinsam benutzte Name, das/den Sie beim Erstellen des Exemplars angegeben haben, auf allen Knoten existiert und im Zugriff ist.

DMB504E DB2MMDATAPATH ist nicht gesetzt.

Ursache: (Nur EEE) Die Umgebungsvariable DB2MMDATAPATH wurde bei der Erstellung eines DB2 Extender-Exemplars nicht gesetzt. Wenn es sich um ein neues DB2 Extender-Exemplar handelt, löschen Sie das Exemplar mit Hilfe von DMBIDROP und erstellen Sie es erneut, wobei Sie die Option `-q` korrekt angeben.

Wenn es sich nicht um ein neues DB2 Extender-Exemplar handelt, führen Sie folgende Schritte aus:

- In der UNIX-Umgebung:
  1. Prüfen Sie, ob das Verzeichnis korrekt ist, existiert und auf allen Knoten im Zugriff ist.
  2. Ändern Sie *$INSTHOME*`/dmb/dmbprofile`, um DB2MMDATAPATH als Verzeichnis zu exportieren.
- In der Windows-Umgebung:
  1. Prüfen Sie, ob der gemeinsam benutzte Name für das Verzeichnis korrekt ist und ob das Verzeichnis existiert und auf allen Knoten im Zugriff ist.
  2. Fügen Sie einen Registrierungsdatenbankeintrag DB2MMDATAPATH mit dem entsprechenden gemeinsam benutzten Namen als Wert im AIV Extender-Exemplarregister hinzu. Der Schlüssel lautet

     \\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2 Extenders

     \PROFILE*exemplarname*\DB2MMDATAPATH.

DMB506E Exemplarname ist nicht gesetzt.

Ursache: Die Umgebungsvariable DB2INSTANCE wurde bei der Ausführung von DMBSTART nicht gesetzt. Stellen Sie sicher, dass DB2START korrekt arbeitet, bevor Sie die DB2 Extender-Services mit DMBSTART starten.

DMB507E dmbssd *name* **node** *argumente*

Ursache: Interner Fehler. Kontaktieren Sie Ihren IBM Ansprechpartner.

DMB508E Die Knotennummer muss größer-gleich 0 sein.

Ursache: Interner Fehler. Kontaktieren Sie Ihren IBM Ansprechpartner.

DMB509E Dieses Programm darf nicht manuell gestartet werden.

Ursache: Interner Fehler. Kontaktieren Sie Ihren IBM Ansprechpartner.

DMB512E **Verwendung:** *argumente* **dmbExemplarname.**

Ursache: Interner Fehler. Kontaktieren Sie Ihren IBM Ansprechpartner.

DMB513E *Name* **ist kein gültiges Exemplar.**

Ursache: Der Name, den Sie beim Versuch, ein DB2 Extender-Exemplar zu löschen, angegeben haben, wurde nicht als Name eines Exemplars erkannt. Prüfen Sie, ob Sie den korrekten Exemplarnamen angegeben haben und dass das Verzeichnis *$INSTHOME*/dmb mit diesem Namen existiert.

DMB514I **Weder Server noch Client wurden auf diesem Exemplar installiert**

Ursache: Sie haben versucht, ein DB2 Extender-Exemplar zu löschen, aber die Extender wurden nicht installiert. Stellen Sie sicher, dass Ihre Installation korrekt ist und dass die Installationsverzeichnisse nicht umbenannt wurden.

DMB515I **Das DB2-Exemplar wurde NICHT gelöscht. Das DB2-Exemplar kann durch den Aufruf von DB2IDROP gelöscht werden.**

Ursache: Wenn Sie ein DB2 Extender-Exemplar löschen, wird das zugehörige Exemplar von DB2 nicht gelöscht. Verwenden Sie DB2IDROP, um das DB2-Exemplar zu löschen.

DMB518E **Unerwarteter Fehler. Funktion =** *funktionsname***, Rückkehrcode =** *rückkehrcode***.**

Ursache: Ein unerwarteter Fehler ist aufgetreten, als Sie versucht haben, ein DB2 Extender-Exemplar zu erstellen oder zu löschen. Prüfen Sie, ob Ihre Installation und Konfiguration korrekt sind.

DMB520E **Sie können dieses Programm nicht als Root ausführen.**

Ursache: Prüfen Sie, ob Sie die korrekte Berechtigung haben, um diese Aktion ausführen zu können.

DMB521E **Der Versuch, die Berechtigungen für** *name* **zu ändern, ist fehlgeschlagen.**

Ursache: Stellen Sie sicher, dass Sie die korrekte Berechtigung haben, um Berechtigungen zu ändern.

DMB522E **Der Versuch, das Eigentumsrecht für** *name* **zu ändern, ist fehlgeschlagen.**

Ursache: Stellen Sie sicher, dass Sie die korrekte Berechtigung haben, um das Eigentumsrecht zu ändern.

DMB523E **Der Versuch, das Gruppeneigentumsrecht für** *name* **zu ändern, ist fehlgeschlagen.**

Ursache: Stellen Sie sicher, dass Sie die korrekte Berechtigung haben, um das Gruppeneigentumsrecht zu ändern.

DMB524E **Datei/Verzeichnis** *name* **existiert bereits.**

Ursache: Eine Datei oder ein Verzeichnis mit dem angegebenen Namen existiert bereits. Wählen Sie einen anderen Namen, und wiederholen Sie den Befehl.

DMB525E **Der Versuch,** *name* **zu erstellen, ist fehlgeschlagen.**

Ursache: Prüfen Sie, ob Sie die korrekte Berechtigung haben, um diese Aktion ausführen zu können.

DMB526E **Datei/Verzeichnis** *name* **fehlt.**

Ursache: Die angegebene Datei oder das angegebene Verzeichnis konnte nicht gefunden werden. Prüfen Sie, ob Sie einen gültigen Datei- oder Verzeichnisnamen angegeben haben.

DMB527E **Der Versuch, die Datei oder das Verzeichnis** *name* **in** *name* **zu kopieren, ist fehlgeschlagen.**

Ursache: Prüfen Sie, ob Sie die korrekte Berechtigung haben, die Datei oder das Verzeichnis zu kopieren. Prüfen Sie, ob ausreichend Speicherbereich für den Kopiervorgang vorhanden ist.

DMB528E **Die Benutzer-ID** *benutzer-ID* **ist ungültig.**

Ursache: Sie haben eine ungültige Benutzer-ID angegeben. Prüfen Sie die Benutzer-ID, und wiederholen Sie den Befehl.

DMB529E **Die Primärgruppe** *gruppe* **der Benutzer-ID** *benutzer-ID* **ist ungültig.**

Ursache: Sie haben eine ungültige Primärgruppe für die Benutzer-ID angegeben. Prüfen Sie, ob die Primärgruppe korrekt ist, und wiederholen Sie den Befehl.

DMB530E **Der Exemplarname** *name* **ist ungültig.**

Ursache: Sie haben einen ungültigen Namen angegeben, als Sie versucht haben, ein Exemplar zu erstellen oder damit zu arbeiten. Prüfen Sie, ob der Exemplarname korrekt ist, und wiederholen Sie den Befehl.

DMB531E Nicht unterstütztes Betriebssystem *name*, Version *versionsnummer*.

Ursache: Sie haben versucht, diesen Befehl unter einer nicht unterstützten Version des Betriebssystems auszuführen. Prüfen Sie die Voraussetzungen für die Ausführung dieser Operation.

DMB535E Auf die angegebene Datei kann nicht zugegriffen werden.

Ursache: Prüfen Sie, ob Sie Zugriff auf die Datei haben, bevor Sie diesen Befehl ausführen.

DMB0533E Die API <*API-name*> wird in einer Umgebung für partitionierten Datenbankserver nicht unterstützt.

Ursache: Sie können die angegebene API nicht in einer Umgebung für partitionierte Datenbanken verwenden.

Aktion: Informationen zur Ausführung dieser Funktion in Ihrer Anwendung finden Sie im Abschnitt zur angegebenen API.

DMB0534E UDF nicht unterstützt.

Ursache: Sie können die benutzerdefinierte Funktion nicht in einer Umgebung für partitionierte Datenbanken verwenden.

Aktion: Prüfen Sie die Nachricht SQL0443N, um festzustellen, bei welcher UDF ein Problem aufgetreten ist. Informationen zur Ausführung dieser Funktion in Ihrer Anwendung finden Sie im Abschnitt zur entsprechenden UDF.

Diagnosetrace

Die DB2 Extender umfassen eine Tracefunktion, mit der Extender-Serveraktivitäten aufgezeichnet werden können. Sie sollten die Tracefunktion nur unter Anleitung des IBM Kundendienstes verwenden.

Die Tracefunktion zeichnet Informationen in einer Serverdatei zu einer Vielzahl von Ereignissen auf. Dazu gehören Start oder Ende einer DB2 Extender-Komponente oder die Rückgabe eines Fehlercodes durch eine DB2 Extender-Komponente. Da sie Informationen zu vielen Ereignissen aufzeichnet, sollte die Tracefunktion nur verwendet werden, wenn es erforderlich ist, beispielsweise bei der Suche nach Fehlerbedingungen. Darüber hinaus sollten Sie die Anzahl der aktiven Anwendungen begrenzen, wenn Sie die Tracefunktion verwenden. Durch die Begrenzung der Anzahl aktiver Anwendungen ist es möglicherweise einfacher, den Grund für ein Problem zu isolieren.

Verwenden Sie den Befehl DMBTRC, um den Trace zu steuern. Sie können den Befehl über eine Befehlszeile auf einem AIX-Server oder Windows NT-Server (oder später) eingeben. Hierzu müssen Sie über die Berechtigung SYSADM, SYSCTRL oder SYSMINT verfügen.

Verwenden Sie den Befehl DMBTRC zu folgenden Zwecken:

- Starten des Trace
- Stoppen des Trace
- Formatieren von Trace-Informationen, um sie lesbarer zu machen
- Anzeigen des Tracestatus

Trace starten

Sie können den Trace durch die Eingabe des folgenden Befehls starten:

dmbtrc on *pfad*

Dabei gibt *pfad* den Pfad einer Serverdatei an, die die Trace-Informationen enthalten soll.

Beispielsweise startet der folgende Befehl den Trace:

```
dmbtrc on /tmp/trace.txt
```

Trace stoppen

Sie können den Trace durch die Eingabe des folgenden Befehls stoppen:

dmbtrc off

Trace-Informationen neu formatieren

Trace-Informationen werden im binären Format aufgezeichnet. Sie können die Informationen neu formatieren und sie lesbarer machen, indem Sie folgenden Befehl eingeben:

`dmbtrc format` *eingabedatei ausgabedatei*

Dabei gibt *eingabedatei* die Datei an, die die Trace-Informationen im binären Format enthält, und *ausgabedatei* gibt die Datei an, die die neu formatierten Informationen enthalten soll. Der Parameter *ausgabedatei* ist wahlfrei. Wenn Sie ihn nicht angeben, erscheinen die neu formatierten Informationen auf der Anzeige.

Beispielsweise formatiert der folgende Befehl Trace-Informationen neu:

```
dmbtrc format /tmp/trace.txt /tmp/fmttrace.txt
```

Tracestatus anzeigen

Verwenden Sie den Befehl

```
dmbtrc info
```

zur Anzeige der folgenden Tracestatusinformationen:

- Tracefunktion ein- oder ausgeschaltet
- Pfad der Datei, die die Trace-Informationen enthält

Anhang A. Umgebungsvariablen für DB2 Extender einstellen

Mit den DB2 Extendern sind Sie flexibel, was die Angabe von Dateinamen beim Speichern, Abrufen oder Aktualisieren von Abbild-, Audio- oder Videoobjekten betrifft. Sie sind auch flexibel, was die Angabe von Programmen zum Anzeigen oder Wiedergeben von Abbild-, Audio- und Videoobjekten betrifft, die von einer Datenbanktabelle abgerufen werden.

Umgebungsvariablen zum Auflösen von Dateinamen verwenden

Sie können einen vollständig qualifizierten Dateinamen (d. h. einen vollständigen Pfad, gefolgt vom Dateinamen) für Speicher-, Abruf- und Aktualisierungsoperationen angeben; die Angabe eines relativen Pfadnamens ist jedoch vorteilhafter. Unter AIX, HP-UX oder Solaris ist ein relativer Dateiname jeder Dateiname, der nicht mit einem Schrägstrich beginnt. Unter Windows ist ein relativer Dateiname jeder Dateiname, der nicht mit einem Laufwerkbuchstaben, gefolgt von einem Doppelpunkt und einem umgekehrten Schrägstrich beginnt.

Wenn Sie einen relativen Dateinamen angeben, verwenden die Extender die Verzeichnisangaben in den verschiedenen Client- und Serverumgebungsvariablen, um den Dateinamen aufzulösen. Hierdurch können Dateien in einer Client/Server-Umgebung verschoben werden, ohne dass der Dateiname geändert werden muss. Ein vollständig qualifizierter Dateiname müsste bei jedem Verschieben einer Datei geändert werden.

In Tabelle 17 werden die Umgebungsvariablen aufgelistet und beschrieben, die Sie für die Auflösung von Dateinamen durch die Image, Audio und Video Extender definieren können.

Tabelle 17. Umgebungsvariablen für die DB2 Extender

Image Extender	Audio Extender	Video Extender	Beschreibung
		Serverumgebungsvariablen	
DB2IMAGEPATH	DB2AUDIOPATH	DB2VIDEOPATH	Zum Auflösen von Quellendateinamen für Speicher-, Abruf- und Aktualisierungsoperationen aus einer Serverdatei.
DB2IMAGESTORE	DB2AUDIOSTORE	DB2VIDEOSTORE	Zum Auflösen von Zieldateinamen für Speicher- und Aktualisierungsoperationen in eine Serverdatei.
DB2IMAGEEXPORT	DB2AUDIOEXPORT	DB2VIDEOEXPORT	Zum Auflösen von Zieldateinamen für Abrufoperationen in eine Serverdatei.
DB2IMAGETEMP			Zum Auflösen von Zieldateinamen für Operationen, die temporäre Serverdateien erstellen. Wenn jedoch die Umgebungsvariable TMP angegeben ist, wird das Verzeichnis TMP verwendet, um Dateinamen aufzulösen.

Tabelle 17. Umgebungsvariablen für die DB2 Extender (Forts.)

Image Extender	Audio Extender	Video Extender	Beschreibung
		Clientumgebungsvariablen	
DB2IMAGEPATH	DB2AUDIOPATH	DB2VIDEOPATH	Zum Auflösen von Quellendateinamen für Anzeige- und Wiedergabeoperationen für eine Clientdatei.
DB2IMAGETEMP	DB2AUDIOTEMP	DB2VIDEOTEMP	Zum Auflösen von Zieldateinamen für Operationen, die temporäre Clientdateien erstellen. Wenn jedoch die Umgebungsvariable TMP angegeben ist, wird das Verzeichnis TMP verwendet, um Dateinamen aufzulösen.

Wenn Sie nicht die passende Umgebungsvariable für den entsprechenden Extender setzen, verwendet der Extender die folgenden Umgebungsvariablen zum Auflösen von Dateinamen:

Umgebungsvariable	Beschreibung
DB2MMPATH	Zum Auflösen von Quellendateinamen für Speicher-, Abruf- und Aktualisierungsoperationen.
DB2MMSTORE	Zum Auflösen von Zieldateinamen für Speicher- und Aktualisierungsoperationen.
DB2MMEXPORT	Zum Auflösen von Zieldateinamen für Abrufoperationen.
DB2MMTEMP	Zum Auflösen von Dateinamen für Operationen, die temporäre Dateien erstellen.

Umgebungsvariablen zum Identifizieren von Anzeige- oder Wiedergabeprogrammen

Umgebungsvariablen werden nicht nur zum Auflösen von Dateinamen verwendet, sondern auch, um Programme zu identifizieren, mit denen Abbildobjekte angezeigt werden, die vom Image Extender abgerufen werden, oder mit denen Audio- oder Videoobjekte wiedergegeben werden, die vom Audio bzw. Video Extender abgerufen werden. Verwenden Sie die APIs DBiBrowse, DBaPlay und DBvPlay, um die jeweiligen Objekte anzuzeigen oder wiederzugeben. Wenn Sie die einzelnen APIs verwenden, können Sie ein Anzeige- oder Wiedergabeprogramm angeben oder das Standardprogramm für die Anzeige oder Wiedergabe des Objekts verwenden.

Die DB2 Extender verwenden die folgenden Umgebungsvariablen auf dem Client, um das Standardanzeige- oder -wiedergabeprogramm zu identifizieren:

Umgebungsvariable	Beschreibung
DB2IMAGEBROWSER	Zum Identifizieren des Standardprogramms für die Abbildanzeige.
DB2AUDIOPLAYER	Zum Identifizieren des Standardprogramms für die Audiowiedergabe.
DB2VIDEOPLAYER	Zum Identifizieren des Standardprogramms für die Videowiedergabe.

Umgebungsvariable DB2MMDATAPATH verwenden (nur EEE)

Die DB2 Extender verwenden die Umgebungsvariable DB2MMDATAPATH, um Standorte für verschiedene Operationen in einer Umgebung für partitionierte Datenbanken aufzulösen. Beispielsweise verwendet der DB2 Image Extender den Wert von DB2MMDATAPATH, um QBIC-Daten in einer Umgebung für partitionierte Datenbanken zu speichern.

Sie definieren DB2MMDATAPATH bei der Erstellung eines DB2 Extender-Exemplars. Dieser Arbeitsschritt wird in „DMBICRT" auf Seite 10 und in den folgenden Readme-Dateien für die Installation beschrieben:

- install.txt im Verzeichnis aixeee (Installation der DB2 Extender zur Verwendung mit DB2 Extended Enterprise Edition unter AIX)
- install.txt im Verzeichnis soleee (Installation der DB2 Extender zur Verwendung mit DB2 Extended Enterprise Edition in der Solaris Operating Environment)

DB2MMDATAPATH kann beispielsweise verwendet werden, um QBIC-Merkmal- und -Indexdaten zu speichern. Unter UNIX speichert der DB2 Image Extender diese QBIC-Daten im folgenden Verzeichnis:

db2mmdatapath /NODE *knotennummer*/QBIC/*datenbankname*

Dabei ist *db2mmdatapath* der Wert der Umgebungsvariable DB2MMDATAPATH, *knotennummer* die Nummer des Knotens und *datenbankname* der Name der Datenbank.

Nehmen Sie das folgende AIX-Beispiel. Darin ist DB2MMDATAPATH auf /localfs/dmbdata gesetzt. Außerdem ist eine Datenbank mit dem Namen 'sample' auf den Knoten 0, 2 und 5 partitioniert.

QBIC-Daten für die Datenbank 'sample' werden in den folgenden Verzeichnissen gespeichert:

Knoten 0: /localfs/dmbdata/NODE0000/QBIC/sample

Knoten 2: /localfs/dmbdata/NODE0002/QBIC/sample

Knoten 5: /localfs/dmbdata/NODE0005/QBIC/sample

Umgebungsvariablen setzen

Sie können Umgebungsvariablen unter AIX, HP-UX, Solaris, OS/2 und Windows setzen.

Umgebungsvariablen auf AIX-, HP-UX- und Solaris-Servern und -Clients setzen

Unter AIX, HP-UX und Solaris werden die Umgebungsvariablen in C-Shell-, Korn-Shell- und Bourne-Shell-Prozeduren angegeben. Wenn die DB2 Extender installiert sind, sind die Umgebungsvariablen für den Server wie folgt gesetzt:

C-Shell

```
setenv DB2MMPATH /usr/lpp/db2ext/samples:/tmp
setenv DB2MMTEMP /tmp
setenv DB2MMSTORE /tmp
setenv DB2MMEXPORT /tmp
```

Korn- und Bourne-Shell

```
DB2MMPATH=/usr/lpp/db2ext/samples:/tmp
export DB2MMPATH

DB2MMSTORE=/tmp
export DB2MMSTORE

DB2MMEXPORT=/tmp
export DB2MMEXPORT

DB2MMTEMP=/tmp
export DB2MMTEMP
```

Die Umgebungsvariablen für den Server sind anfänglich auf solche Werte gesetzt, dass es möglich ist, auf die Multimediadateien zuzugreifen, die in den Beispielprogrammen verwendet werden, die mit den DB2 Extendern geliefert werden. (Informationen zu den Beispielprogrammen und Multimediadateien befinden sich im Anhang B, „Beispielprogramme und Multimediadateien", auf Seite 529.)

Die Umgebungsvariablen für den Client sind wie folgt gesetzt, wenn Sie die DB2 Extender auf einem AIX-, HP-UX- oder Solaris-Client installieren:

C-Shell

```
setenv DB2MMPATH /tmp
setenv DB2MMTEMP /tmp
```

Korn- und Bourne-Shell

```
DB2MMPATH=/tmp
export DB2MMPATH

DB2MMTEMP=/tmp
export DB2MMTEMP
```

Setzen Sie die Server- und Clientumgebungsvariablen, die zum Auflösen von Dateinamen verwendet werden. Geben Sie Werte an, die für Ihre Umgebung passen. Sie können für die Umgebungsvariablen, die mit PATH enden, mehrere Verzeichnisse angeben, die durch einen Begrenzer voneinander getrennt sein müssen. Die Umgebungsvariablen, die mit STORE, EXPORT und TEMP enden, können nur mit einem Verzeichnis angegeben werden.

Geben Sie die Namen der entsprechenden Programme für die Abbildanzeige, Audio- und Videowiedergabe in den Clientumgebungsvariablen DB2IMAGEBROWSER, DB2AUDIOPLAYER und DB2VIDEOPLAYER an.

Sie können die Anfangswerte der Umgebungsvariablen wie folgt ändern:

C-Shell

Verwenden Sie den Befehl SETENV, um Umgebungsvariablen zu setzen:

```
setenv umgeb_var verzeichnis
```

Beispiel:

```
setenv DB2MMPATH /usr/lpp/db2ext/samples:/media
setenv DB2IMAGEPATH /employee/pictures:/images
setenv DB2AUDIOSTORE /employee/sounds
setenv DB2IMAGEBROWSER 'xv %s'
```

Bourne-Shell

Verwenden Sie den Befehl EXPORT, um Umgebungsvariablen zu setzen:

```
umgeb_var=verzeichnis
export umgeb_var
```

Beispiel:

```
DB2MMPATH=/usr/lpp/db2ext/samples:/media
export DB2MMPATH

DB2IMAGEPATH=/employee/pictures:/images
export DB2IMAGEPATH

DB2AUDIOSTORE=/employee/sounds
export DB2AUDIOSTORE
```

Korn-Shell

Verwenden Sie den Befehl EXPORT, um Umgebungsvariablen zu setzen:

```
export umgeb_var=verzeichnis
```

Beispiel:

```
export DB2MMPATH=/usr/lpp/db2ext/samples:/media
export DB2IMAGEPATH=/employee/pictures:/images
export DB2AUDIOSTORE=/employee/sounds
```

Umgebungsvariablen auf Windows-Servern und -Clients setzen

Das Setzen der Umgebungsvariablen unter Windows hängt davon ab, ob Sie die DB2 Extender in einer Umgebung für nicht partitionierte Datenbanken oder einer Umgebung für partitionierte Datenbanken (d. h. mit DB2 Extended Enterprise Edition für Windows) verwenden.

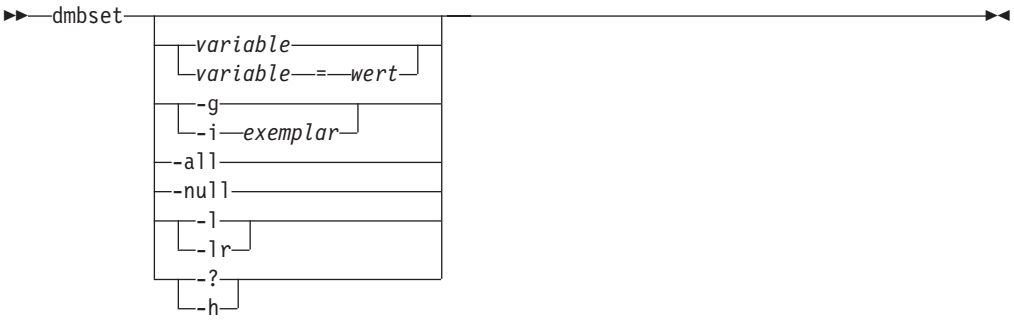
Umgebungsvariablen in Umgebungen für nicht partitionierte Datenbanken unter Windows setzen (nur Nicht-EEE)

Unter Windows werden Umgebungsvariablen im Systemregister gespeichert. Die Variablen können gesetzt werden, wenn Sie die Windows-Systemsteuerung öffnen und das Systemsymbol auswählen. Wählen Sie im Dialog 'Systemeigenschaften' die Registerkarte 'Umgebung' aus. Zwei Fenster mit Umgebungsvariablen und deren Werten werden angezeigt. Das obere Fenster zeigt die Variablen an, die für alle Benutzer gelten. Das untere Fenster zeigt die Variablen an, die nur für den aktuellen Benutzer gelten.

Umgebungsvariablen in Umgebungen für partitionierte Datenbanken unter Windows setzen (nur EEE)

In einer partitionierten Windows-Umgebung werden alle Variablen, die von den DB2 Extendern verwendet werden, in einem privaten Bereich des Systemregisters gespeichert. Mit dem mitgelieferten Programm DMBSET können die Extender-Variablen geprüft und geändert werden.

Die Syntax des Programms lautet:



Um den Wert einer Variablen abzufragen, geben Sie dmbset *variablenname* ein. Beispiel:

```
dmbset DB2MMPATH
```

Um den Wert einer Variablen zu setzen, geben Sie dmbset *variablenname=wert* ein. Beispiel:

```
dmbset DB2MMPATH=C:\DMB\SAMPLES
```

Um die Werte aller Variablen für ein definiertes Exemplar anzuzeigen, geben Sie dmbset -i *exemplarname* ein. Beispiel:

```
dmbset -i dmbinst1
```

Um einen Wert auf Null zu setzen, geben Sie dmbset *variablenname -null* ein. Beispiel:

```
dmbset DB2MMPATH -null
```

Um die Werte der Variablen anzuzeigen, die von allen Exemplaren verwendet werden, geben Sie `dmbset -g` ein.

Um die Namen aller Variablen, die von den DB2 Extendern verwendet werden, aufzulisten, geben Sie `dmbset -lr` ein.

Um die Namen aller Exemplarprofile, die im Register definiert sind, aufzulisten, geben Sie `dmbset -l` ein.

Beim Setzen der Umgebungsvariablen für die DB2 Extender in einer Umgebung für partitionierte Datenbanken sind Sie sehr flexibel. Beispielsweise können Sie Werte für eine beliebige Umgebungsvariable, mit Ausnahme von DB2MMDATAPATH, in einem der folgenden Formate angeben:

- UNC-Name (Universal Naming Convention): *einheitenname**gemeinsamer_name*. Beispiel:

  ```
  \\harmony\JimsShr
  ```

- Laufwerk:Pfad. Beispiel:

  ```
  f:\media
  ```

- Alles übrige: *gemeinsamer_name**verzeichnisname*. Beispiel:

  ```
  JimsShr\images
  ```

Anhang B. Beispielprogramme und Multimediadateien

Zum Lieferumfang der DB2 Extender gehören verschiedene Beispielprogramme. Die Beispielprogramme verwenden Abbild-, Audio- und Videodateien, die ebenfalls mit den Extendern geliefert werden. Die meisten Beispielprogramme sind in der Programmiersprache C geschrieben. Alle C-Beispielprogramme liegen im CLI-Format (CLI = Call Level Interface) vor. Einige Java-Beispielprogramme und ein Net.Data-Beispielmakro werden ebenfalls zur Verfügung gestellt.

Die Beispielprogramme werden im Unterverzeichnis `SAMPLES` des Zielverzeichnisses installiert, wenn Sie die DB2 Extender installieren. Die Abbild-, Audio- und Videodateien werden auch im Unterverzeichnis SAMPLES des Zielverzeichnisses installiert, wenn Sie die DB2 Extender installieren. Während der Installation werden die Extender-Umgebungsvariablen so definiert, dass sie auf das Unterverzeichnis 'samples' im Zielverzeichnis zeigen.

Beispielprogramme

Die Beispielprogramme für die DB2 Extender bestehen aus einer Reihe von Dateien. Diese lauten wie folgt:

Datei	Beschreibung
`enable.c`	Aktiviert eine Datenbank für die Audio, Image und Video Extender, erstellt eine Tabelle und aktiviert die Tabelle und ihre Spalten.
`populate.c`	Importiert Daten in die Tabelle (das Programm ist im C-Format).
`Populate.java`	Importiert Daten in die Tabelle (das Programm ist im Java-Format).
`query.c`	Fragt Daten aus der Tabelle ab (das Programm ist im C-Format).
`Query.java`	Fragt Daten aus der Tabelle ab (das Programm ist im Java-Format).
`api.c`	Verwendet Extender-APIs zum Abfragen der Datenbank.
`handle.c`	Demonstriert die Verwendung von Kennungen in UDFs und die Durchführung von Vergleichen in der Klausel WHERE in SELECT-Anweisungen.
`qbcatdmo.c`	Erstellt einen QBIC-Katalog und katalogisiert eine Spalte mit Abbildern im Katalog.
`qbicdemo.c`	Fragt einen QBIC-Katalog ab.
`color.c`	Erstellt Farbtabellendeklarationen für die Datei `qbicdemo.c`.
`QbicQry.java`	Stellt die Selektoren für die Durchschnitts- und die Histogrammfarbe für eine QBIC-Abfrage dar.
`makesf.c`	Erstellt eine Aufnahmekatalogdatei zur Verwendung mit `makehtml.exe`.
`makehtml.c`	Greift auf einen Aufnahmekatalog zu und erstellt HTML-Seiten, die mit einem Web Browser angezeigt werden können.
`storybrd.java`	Applet für die Anzeige von Aufnahmen; wird von den HTML-Seiten aufgerufen, die von `makehtml.c` generiert werden.
`utility.c`	Dienstprogrammroutinen

utility.h	Kopfdatei für Dienstprogrammroutinen
makefile.aix	Make-Datei zum Erstellen der Programme unter AIX
makefile.os2	Make-Datei zum Erstellen der Programme unter OS/2
makefile.iva	Make-Datei zum Erstellen der Programme unter Windows NT (oder später) unter Verwendung von IBM VisualAge C++
makefile.mvc	Make-Datei zum Erstellen der Programme unter Windows unter Verwendung von Microsoft Visual C++
makefile.sun	Make-Datei zum Erstellen der Programme unter Solaris
makefile.hp	Make-Datei zum Erstellen der Programme unter HP-UX

Ausführbare Dateien werden für die im Folgenden aufgeführten Beispielprogramme zur Verfügung gestellt. Die Beispielprogramme sollten in der gezeigten Reihenfolge ausgeführt werden.

1. Enable
2. Populate
3. Query
4. API
5. Handle
6. Qbcatdmo
7. Qbicdemo
8. QbicQry
9. Makesf
10. Makehtml

Ausführbare Klassendateien (`Populate.class`, `Query.class`, `QbicQry.class` und `storybrd.class`) werden mit den Java-Beispielprogrammen geliefert.

Bevor Sie die Beispielprogramme ausführen, müssen Sie eine Datenbank auf Ihrem Server erstellen. Die Extender-Services müssen auch auf dem Server gestartet sein. Um ein Beispielprogramm auszuführen, geben Sie den Namen des Programms ein (dadurch wird die ausführbare Datei des Programms gestartet). Sie werden zur Eingabe des Datenbanknamens, der Benutzer-ID und des Kennworts aufgefordert. Verwenden Sie die Benutzer-ID und das Kennwort des Benutzers, der die Datenbank erstellt hat.

Sie können auch eigene ausführbare Dateien für die Beispielprogramme erstellen. Dazu müssen Sie folgende Schritte ausführen:

1. Kopieren Sie die Beispielprogrammdateien in ein Verzeichnis mit Schreibzugriff.
2. Editieren Sie die Make-Datei, um die Standorte auf dem System anzugeben, an denen DB2, die Extender und der Compiler installiert sind.
3. Verwenden Sie 'make' oder 'nmake', um die Dateien in ausführbare Programme zu kompilieren.

Weitere Informationen zur Installation und Verwendung der Beispielprogramme befinden sich der Datei `README.CNT` im Verzeichnis der Beispielprogramme.

Beispielabbild-, -audio- und -videodateien

Für die DB2 Extender werden die folgenden Abbild-, Audio- und Videobeispieldateien bereitgestellt:

- Abbilddateien
  - `lizzi.bmp`
  - `sws_stri.bmp`
  - `nitecry.bmp`
  - `ranger_r.bmp`
  - `fuzzblue.bmp`
- Audiodateien
  - `lizzi.wav`
  - `sws_stri.wav`
  - `nitecry.wav`
  - `ranger_r.wav`
  - `fuzzblue.wav`
- Videodateien
  - `nitecry.avi`
  - `sample.mpg`

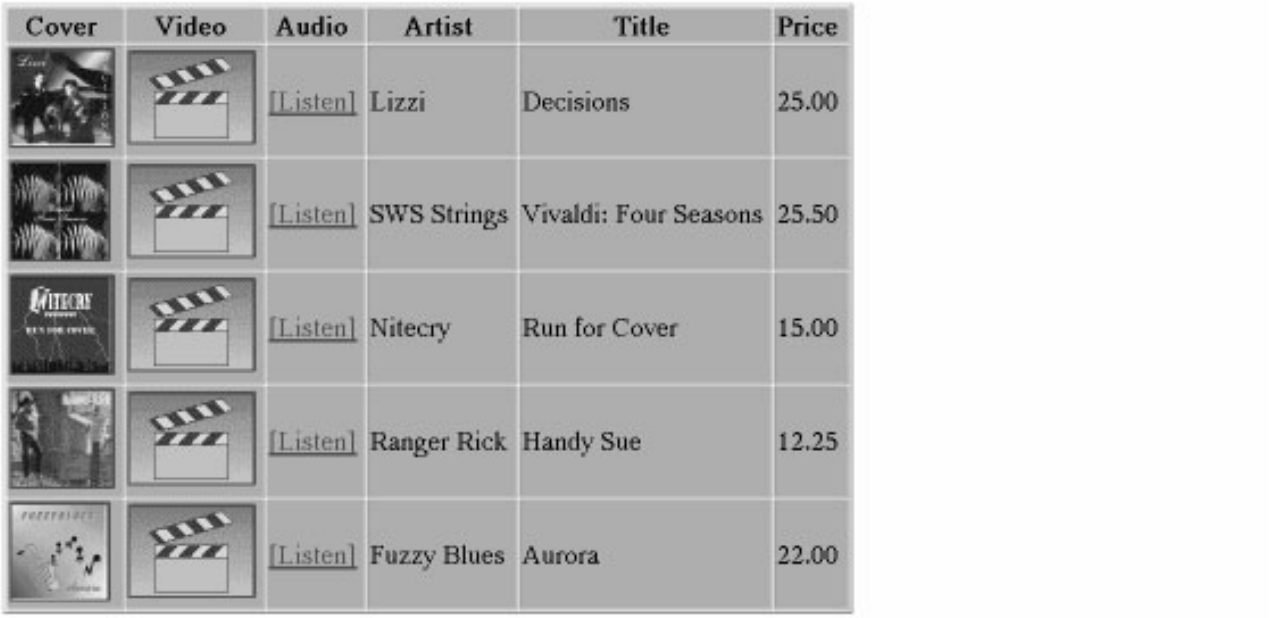
Net.Data-Beispiel-Makrodatei

In den DB2 Extendern ist eine Net.Data-Makrodatei mit dem Namen `extender.d2w` enthalten. Wenn diese über einen Webserver ausgeführt wird, werden von der Makrodatei SQL-Anweisungen ausgeführt, mit denen DB2 Extender-UDFs aufgerufen werden. Die Makrodatei gibt Ergebnisse zurück, die durch einen Webbrowser angezeigt werden. Wie die Abb. 30 auf Seite 532 zeigt, zeigt jede Ergebnisseite außerdem die SQL-Anweisung an, die zum Erzeugen des Ergebnisses ausgeführt wurde. Abb. 31 auf Seite 533 zeigt den Inhalt der Net.Data-Beispiel-Makrodatei.

Um die Net.Data-Beispiel-Makrodatei auszuführen, geben Sie folgende URL von einem Webbrowser aus ein: `http://`*`ihr_server`*`/cgi-bin/db2www/extender.d2w/startHere`

Dabei ist *ihr_server* der Name Ihres Webservers.

select cast(mmdbsys.thumbnail(covers) as blob(10000), cast(mmdbsys.thumbnail(video) blob(3000)), mmdbsys.comment(music), artist, title, price, stock_no from sobay_catalog



Cover	Video	Audio	Artist	Title	Price
		[Listen]	Lizzi	Decisions	25.00
		[Listen]	SWS Strings	Vivaldi: Four Seasons	25.50
		[Listen]	Nitecry	Run for Cover	15.00
		[Listen]	Ranger Rick	Handy Sue	12.25
		[Listen]	Fuzzy Blues	Aurora	22.00

Abbildung 30. Webanwendung zur Ausführung der Net.Data-Beispiel-Makrodatei. Jede Ergebnisseite zeigt die SQL-Anweisung an, die zum Erzeugen des Ergebnisses ausgeführt wurde.

```
%{ ------------------------------------------------------------------------ %}
%{ Copyright International Business Machines Corporation, 1998.            %}
%{ All rights reserved.                                                     %}
%{                                                                          %}
%{ Sample Net.Data macro which shows how to call image, audio, and video   %}
%{ extender UDFs.                                                          %}
%{                                                                          %}
%{ To run, put this macro in your MACRO_PATH root, make sure the tmplobs   %}
%{ directory exists under your web server's document root, and create      %}
%{ the database to be used when running the extender sample programs       %}
%{ 'enable' and 'populate'.  Run 'enable' and 'populate'.  If you name your %}
%{ database something other than 'testdb2', you'll need to change the     %}
%{ definition of DATABASE below.  The extender environment variable        %}
%{ DB2MMEXPORT needs to be set for the instance used by Net.Data to point  %}
%{ to the webserver's <document root>/tmplobs directory.  Then restart DB2 %}
%{ and the extenders to have the variable take effect.                     %}
%{ If you are not running Net.Data's Connection Manager, you'll need to    %}
%{ provide the LOGIN and PASSWORD to the database.  If these instructions  %}
%{ seem unfamiliar to you, you should read the Net.Data documentation at   %}
%{ http://www.software.ibm.com/data/netdata/docs (or the extender documen- %}
%{ tation on the extender sample programs).                                %}
%{                                                                          %}
%{ To disable the showing of SQL statements, change the value of SHOWSQL   %}
%{ below to "no".                                                          %}
%{ ------------------------------------------------------------------------ %}

%{ ------------------------------------------------------------------------ %}
%{ Definitions section                                                     %}
%{ ------------------------------------------------------------------------ %}
%define{
  DATABASE="testdb2"
  SHOWSQL="yes"
%}
```

Abbildung 31. Net.Data-Beispiel-Makrodatei (Teil 1 von 5)

```
%{ ------------------------------------------------------------------------ %}
%{ SQL functions                                                            %}
%{ ------------------------------------------------------------------------ %}
%function (DTW_SQL) startHereSQL(){
  select artist, title, stock_no, price from sobay_catalog

  %REPORT{
   <table border="2" bgcolor="#b1b1b1">
           <tr><th>Artist <th> Title <th> Stock<th> Number <th> Price </tr>
    %ROW{ <tr><td> $(V_artist) <td> $(V_title) <td>  $(V_stock_no) <td> $(V_price) <tr>
    %}
    </table>
   %}
%}

%function (DTW_SQL) addThumbsSQL(){
  select cast(mmdbsys.thumbnail(covers) as blob(10000)),
         cast(mmdbsys.thumbnail(video) as blob(3000)),
         mmdbsys.comment(music), artist, title, price, stock_no
  from sobay_catalog

  %REPORT{
   <table border="2" bgcolor="#b1b1b1">
                 <tr><th>Cover <th>Video <th>Audio <th>Artist <th>Title <th>Price </tr>
    %ROW{  <tr><td>< a href="showCover?stock_no=$(V_stock_no)"><img src="$V1)"></a>
               <td><  a href="getVideo?stock_no=$(V_stock_no)"><img src="$V2)"></a>
               <td><  a href="getAudio?stock_no=$(V_stock_no)&filename=$V3")>[Listen]</a>
               <td> $(V_artist) <td> $(V_title) <td> $(V_price) </tr>
    %}
    </table>
  %}
%}

%function (DTW_SQL) showCoverSQL(){
  select cast(mmdbsys.content(covers, 'GIF') as blob(150000)), mmdbsys.format(covers)
  from sobay_catalog
   where stock_no = '$(stock_no)'

  %REPORT{
    %ROW{  <img src="$(V1) "> <br><br><b>Original image format: $(V2)</b>%}
  %}
%}
```

Abbildung 31. Net.Data-Beispiel-Makrodatei (Teil 2 von 5)

```
%{ The following Content call depends on DB2MMEXPORT being set properly to
   point to the tmplobs directory under the web server's document root.  %}

%function (DTW_SQL) showVideoSQL(){
  select mmdbsys.comment(video), mmdbsys.content(video, mmdbsys.comment(video), 1),
         mmdbsys.format(video)
  from sobay_catalog
   where stock_no = '$(stock_no)'

  %REPORT{
    %ROW{ <a href="/tmplobs/$(V1)"><i><b> Play Video Clip></b></i></a>
       <br><br><b>Format: $(V3) <br>(Note: NT/Win95 may not come with
          a decompressor<br>for this video format.)</br>
    %}
  %}
%}

%{ The following Content call depends on DB2MMEXPORT being set properly to
   point to the tmplobs directory under the web server's document root.  %}

%function (DTW_SQL) showAudioSQL(){
  select mmdbsys.comment(music), mmdbsys.content(music, mmdbsys.comment(music), 1),
         mmdbsys.format(music)
  from sobay_catalog
   where stock_no = '$(stock_no)'

  %REPORT{
      %ROW{ < a href="/tmplobs/$(V1) "<i><b>Play Audio Clip</b></i></a>
         <br><br><b>Format: $(V3)</b>
    %}
  %}
%}
```

Abbildung 31. Net.Data-Beispiel-Makrodatei (Teil 3 von 5)

```
%{ ------------------------------------------------------------------------ %}
%{ HTML sections                                                            %}
%{   E.g., http://<your server>/cgi-bin/db2www/extender.d2w/startHere       %}
%{ ------------------------------------------------------------------------%}
%{   E.g., http://
%{   E.g., http://
%HTML(startHere){
<html>
   <head><title>UDB Extenders Macro Sample: Simple Row Listing</title></head>
   <body bgcolor="#ffffff">
   <font color="#3300ff" size="3"><b>If no data appears below, you might need
    to run the UDB Extender sample programs <i>enable</i> and <i>populate</i>.
    This first HTML section of the extender.d2w macro simply retrieves all the
    traditional data for all the rows in the UDB Extenders' sample database.
    %if ( "$(SHOWSQL)" == "yes" || "$(SHOWSQL)" == "YES" )
    <br><br> By default, every page generated by this macro shows the SQL used
      to generate that page.  Here is the SQL statement for this page:
    %else
      <br>
    %endif
    </b></font>
     <br>@startHereSQL()
    <br><b>Click < a href="addThumbs"><i>here</i></a> to display thumbnails
    and links to image/audio/video data.</b>
 </body>
</html>
%}

%HTML(addThumbs){
<html>
   <head><title>UDB Extenders Macro Sample: Add Thumbnails</title></head>
   <body bgcolor="#ffffff">
     <font color="#3300ff" size="3"><b>This page adds album cover thumbnails
    and links to display the multimedia content of the database.  To access
    the multimedia content:
    <ul>
      <li> Click on a thumbnail of a CD cover to view a full-size image
      <li> Click on a "video thumbnail" to view a video
      <li> Click on a "[Listen]" link to listen to an audio clip
    </ul>
    </b></font> @addThumbsSQL()
    <br><b>Click < a href="startHere"><i>here</i></a> to go back to the first page.</b>
 </body>
</html>
%}
```

Abbildung 31. Net.Data-Beispiel-Makrodatei (Teil 4 von 5)

```
%HTML(showCover){
<html>
   <head><title>UDB Extenders Macro Sample: Cover for item $(stock_no)</title></head>
   <body bgcolor="#ffffff">
     <font color="#3300ff" size="3"><b>For this page, the macro gets a full-size cover
image, converting the image format to GIF so that a browser can show it:
    </b></font><br><br>
    <table width="400" border="2" bgcolor="#b1b1b1" cellpadding="5">
      <tr><td align=center> @showCoverSQL()
      <tr><td align=center> <b>Stock Number: $(stock_no)</b>
    </table>
    <br><b>Go <a href="addThumbs"><i>back</i></a>.</b>
 </body>
</html>
%}

%HTML(getVideo){
<html>
   <head><title>UDB Extenders Macro Sample: Video clip for item $(stock_no)</title></head>
   <body bgcolor="#ffffff">
     <font color="#3300ff" size="3"><b>From this page, you can view a video clip:
    </b></font><br><br>
    <table width="400" border="2" bgcolor="#b1b1b1" cellpadding="5">
      <tr><td align=center> @showVideoSQL()
      <tr><td align=center> <b>Stock Number: $(stock_no)</b>
    </table>
    <br><b>Go <a href="addThumbs"><i>back</i></a>.</b>
 </body>
</html>
%}

%HTML(getAudio){
<html>
   <head><title>UDB Extenders Macro Sample: Audio clip for item $(stock_no)</title></head>
   <body bgcolor="#ffffff">
     <font color="#3300ff" size="3"><b>From this page, you can listen to an audio clip:
    </b></font><br><br>
    <table width="400" border="2" bgcolor="#b1b1b1" cellpadding="5">
      <tr><td align=center> @showAudioSQL()
      <tr><td align=center> <b>Stock Number: $(stock_no)</b>
    </table>
    <br><b>Go <a href="addThumbs"><i>back</i></a>.</b>
 <body>
</html>
%}
```

Abbildung 31. Net.Data-Beispiel-Makrodatei (Teil 5 von 5)

Beispielmultimediadateien

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. An Stelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb der Produkte, Programme oder Dienstleistungen in Verbindung mit Fremdprodukten und Fremddienstleistungen liegt beim Kunden, soweit nicht ausdrücklich solche Verbindungen erwähnt sind.

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

IBM Europe,
Director of Licensing,
92066 Paris La Defense Cedex,
France,

zu richten. 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 bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

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
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
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 Programmpakete oder einer äquivalenten Vereinbarung.

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 alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden, Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHT-LIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellensprache 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. Die in diesem Handbuch aufgeführten Beispiele sollen lediglich der Veranschaulichung und zu keinem anderen Zweck dienen. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. _Jahr/Jahre angeben_. Alle Rechte vorbehalten.

Informationen zur Programmierschnittstelle

Diese Veröffentlichung soll Sie bei der Verwaltung der DB2 Extender und beim Entwickeln von Programmen unterstützen, die zusammen mit den DB2 Extendern eingesetzt werden. Diese Veröffentlichung dokumentiert allgemeine Programmierschnittstellen und zugehörige Informationen, die von den DB2 Extendern bereitgestellt werden.

Allgemeine Programmierschnittstellen ermöglichen Ihnen das Schreiben von Programmen, die die Services der DB2 Extender nutzen. Sie dürfen die für die zu entwickelnde Anwendung erforderliche Laufzeitkomponente der DB2 Extender auf die Client- oder Servermaschinen kopieren. Informationen zum Installieren der Laufzeitkomponente finden Sie in den Installationsanweisungen, die in der Datei README.TXT für das verwendete Betriebssystem auf der CD-ROM für die DB2 Extender enthalten sind.

Marken

Folgende Namen sind in gewissen Ländern Marken der International Business Machines Corporation:

AIX	DB2 Universal Database	PS/2
DB2	IBM	QBIC
DB2 Extender	OS/2	VisualAge

Microsoft, Windows, Windows NT und das Windows-Logo sind in gewissen Ländern Marken oder eingetragene Marken der Microsoft Corporation.

UNIX ist in gewissen Ländern eine eingetragene Marke von The Open Group.

Intel ist eine eingetragene Marke von Intel.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken anderer Unternehmen sein.

Glossar

A

Abbild. Eine elektronische Darstellung eines Bildes.

Abfrage anhand des Abbildinhalts (QBIC). Eine Funktion, die durch den Image Extender zur Verfügung gestellt wird, die es ermöglicht, dass Benutzer nach Abbildern anhand ihrer visuellen Merkmale, wie z. B. *Durchschnittsfarbe* und *Textur* suchen.

Abfrageobjekt. Ein Objekt, das die Merkmale, die Merkmalwerte und die Merkmalwertigkeiten für eine QBIC-Abfrage angibt. Das Objekt kann benannt und für die nachfolgende Verwendung in einer QBIC-Abfrage gespeichert werden. Gegensatz zu Abfragezeichenfolge.

Abfragezeichenfolge. Eine Zeichenfolge, die die Merkmale, die Merkmalwerte und die Merkmalwertigkeiten für eine QBIC-Abfrage angibt. Die Abfragezeichenfolge kann in einer Abfrage von der DB2-Befehlszeile aus eingegeben werden. Gegensatz zu Abfrageobjekt.

Analysieren. Berechnen von numerischen Werten für die Merkmale eines Abbilds und Hinzufügen der Werte zu einem QBIC-Katalog.

Anwendungsprogrammierschnittstelle (API).

(1) Eine vom Betriebssystem oder einem separat bestellbaren Lizenzprogramm bereitgestellte Funktionsschnittstelle. Eine API ermöglicht einem Anwendungsprogramm in einer höhere Programmiersprache die Verwendung spezifischer Daten oder Funktionen des Betriebssystems bzw. des Lizenzprogramms.

(2) In DB2 eine Funktion innerhalb der Schnittstelle, z. B. die API zum Abrufen von Fehlernachrichten.

(3) Die DB2 Extender bieten APIs zum Anfordern von benutzerdefinierten Funktionen, Verwaltungsoperationen, Anzeigeoperationen und zur Erkennung der Änderung von Videoszenen.

API. Siehe *Anwendungsprogrammierschnittstelle*.

Audioclip. Ein Abschnitt mit aufgezeichnetem Tonmaterial.

Aufnahme. Die Vollbilder zwischen zwei Szenenwechseln.

Aufnahmekatalog. Eine Datenbanktabelle oder Datei, die zum Speichern von Daten zu den Aufnahmen in einem Videoclip verwendet werden kann, wie z. B. Start- und Endvollbildnummer für eine Aufnahme. Der Benutzer kann über eine SQL-Abfrage auf eine Sicht der Tabelle oder auf die Daten in der Datei zugreifen.

Auslöser. Die Definition einer Gruppe von Aktionen, die ausgeführt werden sollen, wenn eine Tabelle geändert wird. Auslöser können für folgende Aktionen verwendet werden: Prüfen von Eingabedaten, automatisches Generieren eines Wertes für eine neu eingefügte Zeile, Lesen von anderen Tabellen zu Querverweiszwecken oder Schreiben in andere Tabellen zu Prüfzwecken. Auslöser werden oft zur Überprüfung der Integrität oder zum 'Erzwingen' von Geschäftsregeln verwendet.

Ähnlichkeitsergebnis. Ein errechneter Wert, der wiedergibt, wie ähnlich die Merkmalwerte den Werten sind, die in einer Abfrage anhand des Abbildinhalts angegeben sind. Je höher die Zahl, desto größer die Übereinstimmung. Das Ähnlichkeitsergebnis wird verwendet, um die Ergebnisse einer Abfrage anhand des Abbildinhalts (QBIC) zu sortieren.

B

Benutzerdefinierte Funktion (UDF). Eine Funktion, die durch einen Benutzer für DB2 definiert wird. Ist die Funktion definiert, kann sie in SQL-Abfragen und Videoobjekten verwendet werden. Beispielsweise können UDFs erstellt werden, um das Komprimierungsformat eines Videos abzurufen oder die Abtastrate eines Tons zurückzugeben. Hierdurch kann die Funktionsweise von Objekten definiert werden, die zu einem bestimmtem Typ gehören.

Benutzerdefinierter Typ (UDT). Ein Datentyp, der durch einen Benutzer für DB2 definiert wird. UDTs werden verwendet, um die einzelnen LOBs voneinander zu unterscheiden. Beispielsweise kann ein UDT für Abbildobjekte und ein anderes für Audioobjekte erstellt werden. Obwohl sie als BLOBs gespeichert werden, werden Abbild- und Audioobjekte von BLOBs und auch untereinander unterschieden.

D

Dateireferenzvariable. Eine Programmiervariable, die sinnvoll beim Verschieben eines LOB von und zu einer Datei auf einer Client-Workstation ist.

Datenbankpartition. Ein Teil der Datenbank, der aus eigenen Benutzerdaten, Indizes, Konfigurationsdateien und Transaktionsprotokollen besteht. In einigen Fällen auch als Knoten oder Datenbankknoten bezeichnet.

Datenbankpartitionserver. Verwaltet eine *Datenbankpartition*. Ein Datenbankpartitionsserver besteht aus einem Datenbankmanager und aus der Sammlung von

Daten und Systemressourcen, die er verwaltet. Normalerweise ist jeder Maschine ein Datenbankpartitionsserver zugeordnet.

DB2 Extender. Eine Gruppe von Programmen, mit denen Sie Datentypen über die traditionellen numerischen und Zeichendaten hinaus speichern und abrufen können, beispielsweise Abbild-, Audio- und Videodaten sowie komplexe Dokumente.

Direktionalität. Ein Attribut der *Textur*, das beschreibt, ob das Abbild eine bevorzugte Richtung hat (wie z. B. Gras) oder ob es ein Objekt ohne Richtung ist (wie z. B. Glas).

Durchschnittsfarbe. Ein Farbmesswert, der als Durchschnitt der Farbwerte, die in den Pixeln eines Abbilds enthalten sind, berechnet wird.

E

Eindeutiger Typ. Siehe *benutzerdefinierter Typ*.

Exemplar. Eine logische DB2 Extender-Serverumgebung. Sie können mehrere Exemplare des DB2 Extender-Servers auf derselben Workstation implementieren, jedoch nur jeweils ein Exemplar für jedes DB2-Exemplar. Sie können diese Exemplare für folgende Aufgaben verwenden:

Trennen der Entwicklungsumgebung von der Produktionsumgebung

Begrenzen vertraulicher Informationen auf eine bestimmte Personengruppe

Extender. Siehe *DB2 Extender*.

G

Gigabyte (GB). Eine Milliarde (10^9) Byte. Wenn es sich um Speicherkapazität handelt, 1 073 741 824 Byte.

Grobheit. Ein Attribut der *Textur*, das die Skala der Textur misst ('Kieselsteine versus Felsbrocken').

Großes binäres Objekt (BLOB). Eine binäre Zeichenfolge, die bis zu
2 GB lang sein kann. Abbild-, Audio- und Videoobjekte werden in einer DB2-Datenbank als BLOBs gespeichert.

Großes Doppelbytezeichenobjekt (DBCLOB). Eine Zeichenfolge aus Doppelbytezeichen oder eine Kombination aus Einzel- und Doppelbytezeichen, die bis zu 2 GB lang sein kann. DBCLOBs haben eine zugeordnete Zeichenumsetztabelle. Textobjekte, die Doppelbytezeichen enthalten, werden in einer DB2-Datenbank als DBCLOBs gespeichert.

Großes Objekt (LOB). Eine Bytefolge, die bis zu 2 GB lang sein kann. Für ein LOB gibt es drei Typen: *großes binäres Objekt (BLOB)*, *großes Zeichenobjekt (CLOB)* oder *großes Doppelbytezeichenobjekt (DBCLOB)*.

Großes Zeichenobjekt (CLOB). Eine Zeichenfolge aus Einzelbytezeichen, die bis zu 2 GB lang sein kann. CLOBs haben eine zugeordnete Zeichenumsetztabelle. Textobjekte, die Einzelbytezeichen enthalten, werden in einer DB2-Datenbank als CLOBs gespeichert.

H

Histogrammfarbe. Ein Messwert der eindeutigen Farben in einem Abbild. Die Daten für die einzelnen Farben werden separat in einem *QBIC-Katalog* gespeichert.

Hostvariable. Eine Variable in einem Anwendungsprogramm, auf die in eingebetteten SQL-Anweisungen verwiesen werden kann. Hostvariablen sind der Primärmechanismus zur Übertragung von Daten zwischen einer Datenbank und den Arbeitsbereichen des Anwendungsprogramms.

I

Indexdatei. Eine Datei, die Indexierungsinformationen enthält, die vom Video Extender bei der Suche nach einer *Aufnahme* oder einem einzelnen Vollbild in einem Videoclip verwendet werden.

K

Kennung. Eine Zeichenfolge, die durch einen Extender erstellt wird und zur Darstellung eines Abbild-, Audio- oder Videoobjekts in einer Tabelle verwendet wird. Eine Kennung wird für ein Objekt in einer Benutzertabelle und in den *Tabellen zur Verwaltungsunterstützung* gespeichert. Auf diese Weise kann ein Extender eine Verbindung zwischen der Kennung, die in einer Benutzertabelle gespeichert ist, und den Informationen zum Objekt, die in den Tabellen zur Verwaltungsunterstützung gespeichert sind, herstellen.

Kilobyte (KB). Eintausend (10^3) Byte. Wenn es sich um Speicherkapazität handelt, 1024 Byte.

Knoten. Bei der Datenbankpartitionierung gleich bedeutend mit Datenbankpartition.

Knotengruppe. Eine benannte Gruppe aus einer oder mehreren Datenbankpartitionen.

Kontrast. Ein Attribut der *Textur*, das sich auf die Intensität des Musters bezieht. Der Kontrast ist eine Funktion der Varianz eines Graustufenhistogramms.

L

LOB-Zeiger. Ein kleiner (4 Byte), in einer Hostvariable gespeicherter Wert, der in einem Programm verwendet werden kann, um auf ein viel größeres LOB in einer DB2-Datenbank zu verweisen. Unter Verwendung eines LOB-Zeigers kann der Benutzer das LOB bearbeiten, als wäre es in einer regulären Hostvariable gespeichert. Es

ist nicht erforderlich, das LOB zwischen der Anwendung auf der Clientmaschine und dem Datenbankserver zu bewegen.

M

Megabyte (MB). Eine Million (10^6) Byte. In Bezug auf die Speicherkapazität bezeichnet ein Megabyte 1 048 576 Byte.

Mehrpartitionsknotengruppe. Eine *Knotengruppe*, die mehr als einen *Datenbankpartitionsserver* enthält.

Merkmal. Ein visuelles Attribut eines Abbilds, z. B. *Durchschnittsfarbe*.

Metadatentabellen. Siehe *Tabellen zur Verwaltungsunterstützung*.

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

V

W

Z

Kontaktaufnahme mit IBM

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3 313233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0190 7 72243 erreichen Sie die DB2 Helpline, wo Sie Antworten zu DB2-spezifischen Problemen erhalten.

Informationen zur nächsten IBM Niederlassung in Ihrem Land oder Ihrer Region finden Sie im IBM Verzeichnis für weltweite Kontakte, das Sie im Web unter www.ibm.com/planetwide abrufen können.

Produktinformationen

Informationen zu DB2 Universal Database-Produkten erhalten Sie telefonisch oder im World Wide Web unter www.ibm.com/software/data/db2/udb.

Diese Site enthält die neuesten Informationen zur technischen Bibliothek, zum Bestellen von Büchern, zu Client-Downloads, Newsgroups, FixPaks, Neuerungen und Links auf verfügbare Webressourcen.

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3 313233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0180 5 5090 können Sie Handbücher telefonisch bestellen.

Informationen dazu, wie Sie sich mit IBM in Verbindung setzen können, finden Sie auf der globalen IBM Internet-Seite unter folgender Adresse:
www.ibm.com/planetwide

SH12-3037-00