Query Management Facility

QMF for Windows 入门

版本 7

SB84-0253-00

Query Management Facility

QMF for Windows 入门

版本 7

SB84-0253-00

注意!

使用此信息及其支持的产品之前，请先阅读第131页的『附录. 注意事项』中的一般信息。

第五版（2000 年 9 月）

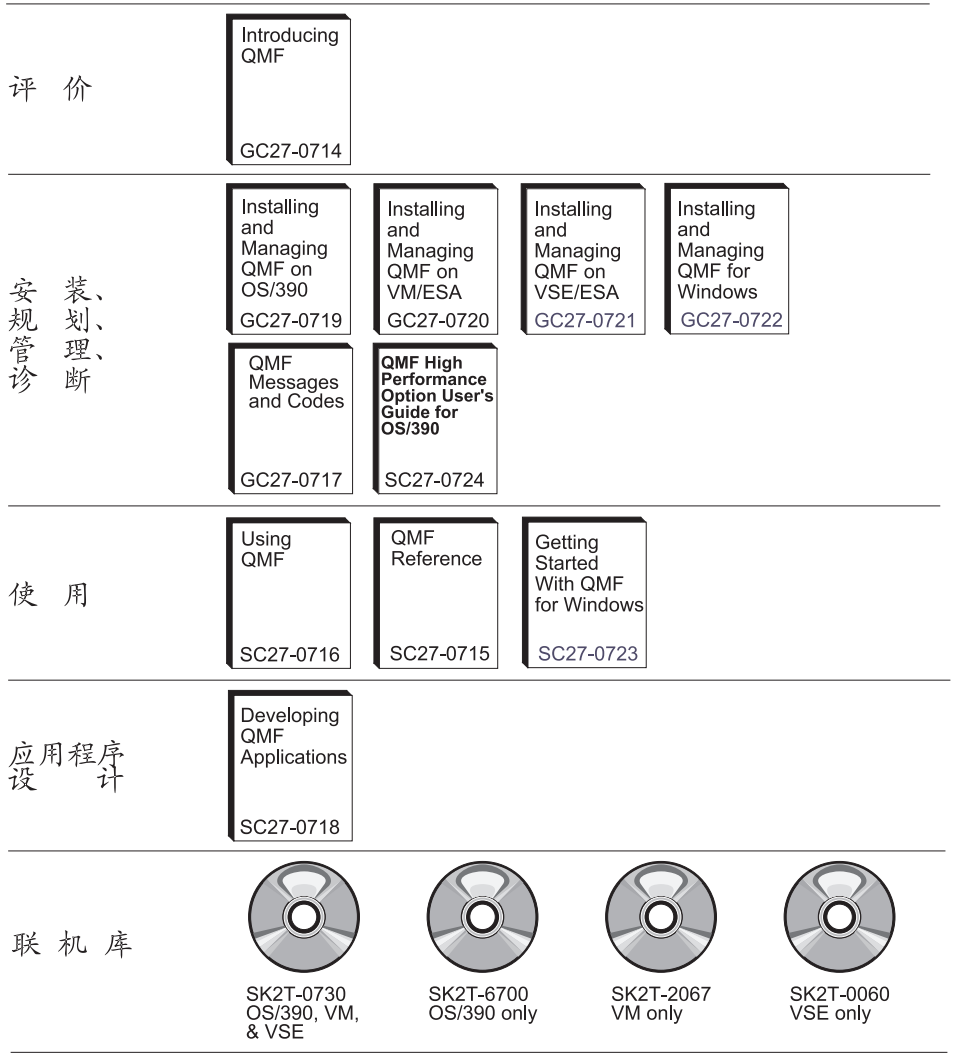
本版本适用于 DB2 Universal Database Server for OS/390 (DB2 UDB for OS/390) 第 7 版的功能组件 Query Management Facility for Windows，5675–DB2、DATABASE 2 Server for VM and VSE（DB2 for VM and VSE）第 7 版的功能组件 Query Management Facility，5697–F42、 Query Management Facility for Windows for AS/400，5697–G24、Query Management Facility for Windows for DB2 Workstation Databases，5697–G22、 DB2 Warehouse Manager，5648–D35、DB2 Warehouse Manager for AS/400，5697–G23，以及所有后继的发行版和修正版，直至在新版本中另行说明。

本版本替换前一版本，SC26-9582-02。在此版本中，更改的左侧有一垂直栏指出其技术更改。不具技术属性的编辑更改不加标注。

目录

QMF 库

可通过 IBM 代表或者在美国拨打电话 1-800-879-2755 订购以下这些手册。



第1章 介绍

本章提供 QMF for Windows 的概述，并说明 QMF for Windows 入门的一些基本任务。

数据库服务器

查询、表单、过程和表是在数据库服务器上运行并保存的。

DB2 系列数据库

QMF for Windows 可以连接到各种各样的 DB2 数据库。

- DB2 UDB for OS/390、DB2 for OS/390 和 DB2 for MVS
- DB2 Server for VSE & VM 和 SQL/DS
- DB2 Universal Database 和 DB2 Common Server
- DB2 Parallel Edition
- DataJoiner

您的 QMF for Windows 许可证决定了您可以使用 QMF for Windows 副本来安装和连接到哪些 DB2 系列产品。

用户名 - 技术性名称

不同版本和类型的 DB2 通过 RDB 名称、位置名称或其它技术性名称引用数据库。

使用 QMF for Windows，管理员可以给数据库指定一个便于记忆的名称，例如，用“购买数据库”来代替 DB2P_01_PURCH。

QMF for Windows 将数据库服务器或 DB2 数据库称作“服务器”。

设置服务器名称

查询数据库之前，QMF for Windows 需要知道该数据库的存储位置。

1. 在**文件**菜单中选择**新建 SQL 查询**。打开新的 SQL 查询文档。

2. 在**查询**菜单中选择**设置服务器**。将打开“设置服务器”对话框。

3. 从可用服务器列表中选择要查询的服务器，并单击**确定**。开始下一个 QMF for Windows 对话之前，QMF for Windows 将自动重新连接到同一个服务器。

数据库安全

连接到服务器之前，您必须提供用户标识和口令。

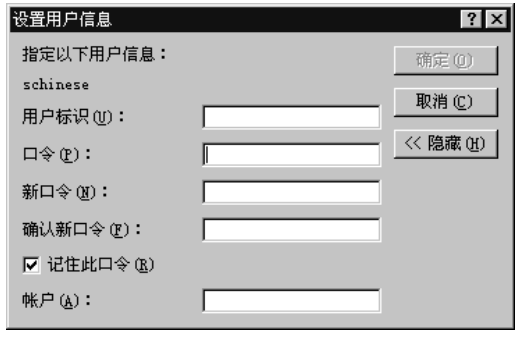
注册

您必须输入想要访问的数据库服务器的有效用户标识和口令。此数据库服务器的用户标识和口令不必与本地或网络用户标识和口令相同。

如果正在运行 Windows，则系统会提供是否要记住 QMF for Windows 会话之间服务器口令的选项。如果当前您已注册到 Windows，那么“设置用户信息”对话框将多显示出一个名为“**记住此口令**”的复选框。如果选中此复选框，那么您为该服务器所输入的口令将被保存到 Windows 口令列表中。无论您什么时候注册到 Windows，QMF for Windows 都可以自动检索到该口令，因此不必再提示您输入口令。如果运行 QMF for Windows 时没有注册，或者以其他用户身份注册，则 QMF for Windows 将提示您输入用户标识和口令。

注: 如果选择保存口令，那么任何人只要能够注册到您的 Windows 帐户，就可以用您的（服务器）用户标识和口令来访问您的数据库服务器。

1. 在**查询**菜单中选择**设置用户信息**。将打开“设置用户信息”对话框。



2. 在相应的字段中输入您的用户标识和口令。

   注: 用户标识和口令是区别大小写的。例如，如果您的用户标识或口令是大写的，则必须以大写输入。有些类型的数据库服务器要求区分大小写，但有些不需要。

3. 如果想要保存用户标识和口令，请选中**记住此口令**复选框。
4. 单击**确定**。QMF for Windows 存储此信息，准备访问服务器。

更正口令

如果您键入了错误的口令，那么可以重新打开“设置用户信息”对话框更正口令。

1. 在**查询**菜单中单击**设置用户信息**。将打开“设置用户信息”对话框。
2. 再次键入口令并单击**确定**。口令被更正。

更改口令

您可以在来自 QMF for Windows 的数据库服务器上更改口令。这个功能当前只能由 DB2 for OS/390 版本 5 及更高版本支持。

1. 在**查询**菜单中选择**设置用户信息**。将打开“设置用户信息”对话框。
2. 单击**更改**。将显示**新口令**和**确认新口令**字段。
3. 在**新口令**和**确认新口令**字段中输入新口令，然后单击**确定**。您的数据库服务器口令将会更改。

指定帐户

数据库服务器使用帐户来跟踪系统使用情况。可咨询系统管理员以确定您的系统是否使用帐户。

1. 在**查询**菜单中选择**设置用户信息**。将打开“设置用户信息”对话框。

2. 在**帐户**字段中输入您想要使用的帐户管理字符串，然后单击**确定**。QMF for Windows 将存储此信息，以备访问服务器之用。

管理

QMF for Windows 管理程序总是在后台运行，它监视数据库和系统资源的使用情况。管理程序还限制了您可以运行的查询的类型和大小。

查看资源限制

从**查看**菜单中选择**资源限制**。将打开“资源限制”对话框。“资源限制”对话框中所有的信息都是只读的。这些限制是由系统管理员设置的。

可以起作用的限制和控制类型有:

- 超时
- 限制
- SQL 动词
- 选项
- 保存数据
- 绑定
- 对象跟踪

设置自己的行限制

您可以指定查询可检索的最大行数。超过这个限制时，QMF for Windows 将取消该查询。您的资源限制组中指定的最大已授权行限制的优先级比此参数高。

输入 0，指定这个字段没有限制。

QMF for Windows 已检索到的，但超过此限制的行将被保留，并且可以查看。

1. 在**查询**菜单中单击**设置行限制**。打开“设置行限制”对话框。

2. 输入您希望查询返回的最大行数并单击**确定**。这个行限制将在下一次运行该查询时有效。

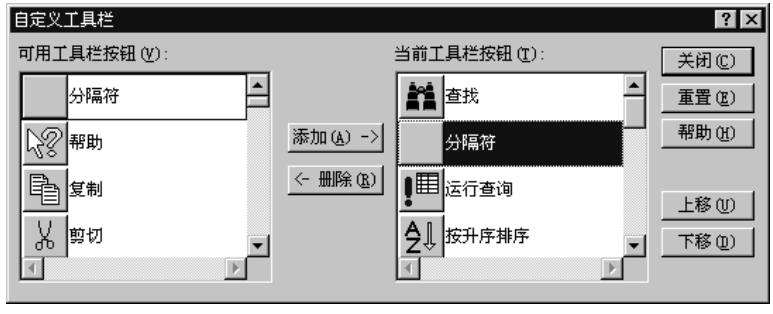
定制工具栏

您可以定制工具栏，让它仅显示希望看见的按钮。

添加按钮到工具栏

您可以在现有的 QMF for Windows 工具栏中添加按钮。这些按钮代表的功能并不是所有用户都需要的，但您可以从工具栏获得这些功能。

1. 双击工具栏周围的灰色区域。将打开“自定义工具栏”对话框。



2. 从**可用工具栏按钮**列选择要添加的按钮并单击**添加**。该按钮将被添加到工具栏中。
3. 添加按钮之后，请单击**关闭**。该对话框将关闭，并且新按钮添加到工具栏中。

在工具栏上移动按钮

您可以选择重新安排 QMF for Windows 工具栏上的按钮。

1. 双击工具栏周围的灰色区域。将打开“自定义工具栏”对话框。
2. 从**可用工具栏按钮**列中选择想要移动的按钮。
3. 使用**上移**和**下移**按钮在工具栏中移动按钮。
4. 移动按钮之后，单击**关闭**。该对话框关闭，按钮出现在新的位置上。

从工具栏中移去按钮

您可以选择移去 QMF for Windows 工具栏上的按钮。

1. 双击工具栏周围的灰色区域。将打开“自定义工具栏”对话框。
2. 从**可用工具栏按钮**列中选择想移去的按钮，然后单击**删除**。该按钮将从工具栏中移去。
3. 移去按钮之后，单击**关闭**。该对话框关闭，按钮从工具栏中移去。

第2章 使用 SQL 查询

结构化查询语言 (SQL) 是用户与数据库之间最基本的接口。查询以 SQL 格式编写，并由数据库进行处理。用户可以用 SQL 来编写 QMF for Windows 查询，也可以使用“指向并单击”的方法来创建查询。

SQL 查询

“结构化查询语言”查询需要 SQL 命令和语法的知识。不熟悉 SQL 的用户应当创建提示查询。

创建新的 SQL 查询

单击工具栏上的**新建 SQL 查询**按钮。

将打开新的查询文档。

在数据库服务器上运行 SQL 查询

1. 打开新的查询文档并在查询中输入，或打开现有查询文件，或者从数据库打开查询。
2. 单击工具栏上的**运行查询**按钮。

该查询运行，并显示结果。

在结果视图和 SQL 视图之间切换

您可以查看查询结果或 SQL 语句本身。

在运行过的查询的 SQL 视图中，单击工具栏上的**查看结果**按钮。

将出现查询结果。

-或者-

在查询的结果视图中，单击**查看 SQL** 按钮。

将显示 SQL 语句。

选择字体

您可更改用于显示查询的字体。根据您计算机上所安装的不同字体，字体的选项也有所不同。关于添加字体的更多信息，请参阅操作系统的帮助程序。

注: 如果在选择新查询显示字体之后保存了查询，则查询将总是以新字体显示。

选择查询显示字体

1. 从“SQL”视图单击**查询**菜单的**设置字体**。将打开“字体”对话框。
2. 选择用来显示查询文本的字体，然后单击**确定**。该查询以新的字体出现。

   注: 单击**设置为缺省值**，将选定的字体用作所有新查询的缺省字体。

多个查询

可以在同一时间打开多个查询文档。也可以在同一时间运行多个查询。您可以使用这个功能来生成多个报表，或将 SQL 文本从一个查询剪切到另一个查询。

同时显示多个查询

1. 打开至少两个查询文档。
2. 从**窗口**菜单选择以下任一命令:

命令	结果
层叠	将查询按照以层错开的方式显示。
水平平铺	将查询窗口以垂直层叠的方式显示。
垂直平铺	将查询窗口以并排方式显示。

查询结果是按照所选择的选项排列的。

制作查询

使用“制作查询”命令来创建新的 SQL 查询文档。可指定一个或多个表名以及期望的 SQL 语句类型，QMF for Windows 将自动地创建 SQL 语句，此语句中引用了表中列的名称和数据类型。

创建新的 SQL 查询

1. 在**文件**菜单上单击**制作查询**。将打开“制作查询”对话框。

2. 选择您想要创建的查询类型:

查询类型	结果
选择	从一个或多个表中检索行。
更新	更改表中的信息。
插入	在表中添加新行。

3. 输入要查询的表的拥有者和名称。

   注: 可使用模式从匹配表的列表中选择表名。

   - 百分号 (%) 用于匹配任意长度任意字符的字符串。例如，要列出名称以字母 A 开头的所有表，可输入 A%。
   - 下划线字符 (_) 用于匹配单个字符。例如，要列出拥有者的第二个字符为字母 A 的所有表时，可输入 _A%。

   输入模式后，请单击**从列表添加**，然后从结果列表中选择表。

4. 为该表输入唯一标识。
5. 单击**添加**。该表被添加到查询中。
6. 添加了要查询的表后，可单击**确定**。选定表的 SQL 查询将创建并显示出来。

SQL 查询中的替换变量

有了替换变量，就可以用同一查询检索不同的信息，只需通过每次运行查询时为变量提供不同的值即可。要检索不同的数据集，您不需要重新编写查询，相反，只要在运行查询时为替换变量指定不同的值就可以了。

替换变量是包含在查询中的一个文本。它必须以 & 字符开头，可包含 18 个字符，这 18 个字符可以是字母、数字或以下字符之一：^ ! $ ~ { } ? @ # % \ 或 _。例如，以下值是有效的替换变量:

```
&VARIABLE1
&DEPARTMENT_NUMBER
```

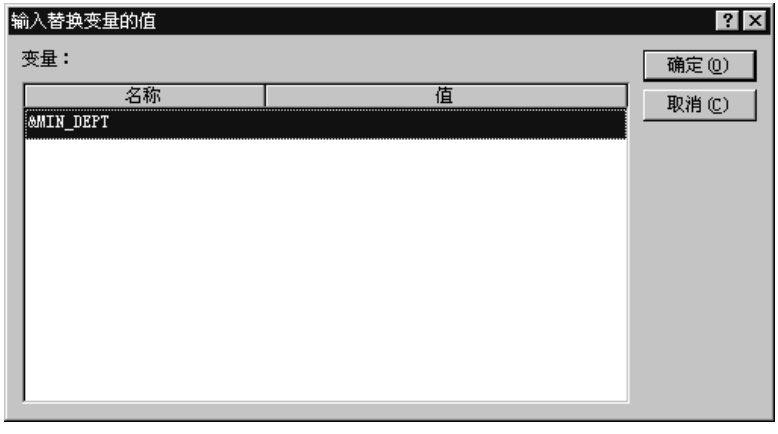
替换变量可以出现在查询的任何地方，它的值可以是写入查询的任何内容（除注释以外）。例如，可使用替换变量来替代列名、搜索条件、子查询或任意特定的值。

用替换变量来运行 SQL 查询

1. 打开新查询文档，输入此 SQL 语句:

   ```
   SELECT * FROM Q.STAFF WHERE DEPT >= &MIN_DEPT
   ```

2. 运行此查询。将打开“输入替换变量的值”对话框。



3. 在**值**字段输入值 50 并单击**确定**。该查询运行，并显示出查询结果。

可尝试用替换变量代替 SELECT 和 FROM 子句中的值。再查看由不同输入所返回的查询结果。

保存和打开 SQL 查询

您可以将查询保存到个人计算机、文件服务器或数据库服务器中。

将 SQL 查询保存到文件

1. 在打开的查询中，单击工具栏上的**保存**按钮。

如果该查询曾经保存过，则再次保存。如果以前没有保存过，则将打开“另存为”对话框。

2. 输入准备用来保存查询的文件名并单击**确定**。该查询被保存。

打开已保存的 SQL 查询

1. 单击工具栏上的**打开**按钮。

将打开“打开”对话框。

2. 选择要打开的文件并单击**确定**。选定的查询在一个新的查询文档中打开。

在数据库服务器上保存 SQL 查询

保存到服务器的查询可以被其他用户访问。如果您希望和其他用户共享查询，则应将它们保存到数据库服务器上。

1.

在打开的查询中，单击工具栏上的**保存到服务器**按钮。

将打开“保存查询”对话框。

2. 输入拥有者、名称，选择是否与其他用户共享保存的查询，并单击**确定**。该查询被保存到服务器上。

   如果这个名称的查询已经存在，系统将提示您是否覆盖已存在的查询。

打开数据库服务器上已保存的 SQL 查询

您可以打开保存在数据库服务器上的查询。

1. 单击工具栏上的**从服务器打开**按钮。

将打开“从服务器打开”对话框。

2. 输入服务器、拥有者和名称，然后单击**确定**。将打开 SQL 查询。

打印 SQL 查询

您可预览并打印 SQL 查询。

预览查询

1. 打开一个查询并激活 SQL 视图。将出现 SQL 语句。
2. 在**文件**菜单上单击**页面设置**。将打开“页面设置”对话框。

3. 对页面布局作适当的更改，并单击**确定**。
4. 单击工具栏上的**打印预览**按钮。

所打印查询的预览将出现。

打印 SQL 查询

1. 打开一个查询并激活 SQL 视图。将出现 SQL 语句。
2. 在**文件**菜单上单击**页面设置**。将打开“页面设置”对话框。
3. 对页面布局作适当的更改，并单击**确定**。
4. 单击工具栏上的**打印**按钮。

该查询被打印。

第3章 使用提示查询

提示查询是一种创建查询简单方法，只需从菜单和列表选择选项即可。一旦创建了提示查询，就可以保存此查询，或将其转换为 SQL 查询。

构建简单查询

可以使用提示查询界面方便地构建简单查询。

打开新的提示查询

- 在**文件**菜单上单击**新建提示查询**。将打开新的提示查询文档。

提示查询操作按钮

使用查询操作按钮来编辑提示查询。将会有一系列按钮将出现在它们所控制的那部分上面。

提示查询操作按钮	外观	结果
添加		通过单击在提示查询中添加项查询中。
编辑		通过单击来编辑查询中突出显示的项。

删除		通过单击来删除选定的项。
上移和下移		通过单击将选定项在提示查询中向上移动或向下移动。

将表添加到提示查询

1. 在提示查询文档的“表”部分单击**添加**按钮。

   将打开“表”对话框。

2. 输入要添加的表的拥有者和名称，并单击**添加**。该表将添加到查询中。

   注: 您可以使用模式从匹配对象列表中选择对象名。

   - 百分号 (%) 用于匹配任意长度任意字符的字符串。例如，要列出名称以字母 A 开头的所有表，可输入 A%。
   - 下划线字符 (_) 用于匹配单个字符。例如，要列出拥有者的第二个字符为字母 A 的所有表时，可输入 _A%。

   输入了模式后，可单击**从列表添加**，并从结果列表中选择表。

3. 将其它表条件也添加到查询中，单击**关闭**。“提示查询”文档出现，其中列出了新表。

运行提示查询

可以用与运行 SQL 查询相同的方式来运行提示查询。单击工具栏上的“运行查询”按钮。

提示查询将运行。

构建复杂查询

还可以使用提示查询界面来构建更为复杂的查询。

向提示查询中添加列

1. 在“提示查询”文档的“列”部分单击**添加**按钮。

将打开“列”对话框。

2. 选择要添加的列，并单击**添加**。该列被添加到提示查询中。
3. 将所有附加列添加到查询中，然后单击**关闭**。“提示查询”文档出现，其中列出了新的列。

注： 您可以在“函数”字段中选择一个汇总函数，从而将其应用到列中。可用的汇总函数包括：AVERAGE、COUNT、MAXIMUM、MINIMUM 和 SUM。

注： 您可以在**新列名**字段中键入新的列名称来重新命名一个列。

使用排序条件

排序条件用于指定查询中行的排序方式。行可以按升序 (A-Z) 排序，也可以按降序 (Z-A) 排序。

如果要对多列数据行进行排序，则第一列将先排序，第二列将在第一列排好的顺序中排序，依此类推。

添加排序条件

1. 在“提示查询”文档的“排序条件”部分单击**添加**按钮。

将打开“排序条件”对话框。

2. 选择要排序的列、排序的方向，并单击**添加**。排序条件被添加到提示查询中。
3. 将其它排序条件也添加到查询中，单击**关闭**。“提示查询”文档出现，其中列出了新的排序条件。

使用行条件

大部分时侯，您可能只需要查看表中的某几行。要查看指定的行，请添加行条件。如果不使用行条件，那么表中的所有行都将被显示出来。

可以使用以下行条件:

- 等于
- 小于
- 小于或等于
- 大于
- 大于或等于
- 在...之间
- 开始于
- 结束于
- 包含

- 空值

行条件是由以下运算符控制的:

- 是
- 不是

添加行条件

1. 在“提示查询”文档的“行条件”部分单击**添加**按钮。

将打开“行条件”对话框。

2. 选择条件语句的部分，并单击**添加**。

行条件的部分	函数
左侧	选择要检查的列。
运算符	决定行的左端和右端的关系。
右侧	输入要检查的条件。

行条件被添加到提示查询中。

3. 将其它行条件也添加到查询中，单击**关闭**。“提示查询”文档出现，其中列出了新的行条件。

使用提示查询中的多个表

可在提示查询中包含多个表中的信息。

必须将两个表关联起来，在每个表中指定一个或多个连接条件。表中只有连接列相同的行才会出现在结果中。连接条件的每一列的数据类型都必须相一致。一旦指定了两列之间的关系，QMF for Windows 将记住这些关系，并在以后的查询中建议使用这些关系，使得后续查询的创建更简单且有效。

创建提示查询连接条件

1. 在“提示查询”窗口的“表”部分单击**添加**按钮，以添加至少两个表。如果从未连接过这些表，则“连接表”对话框将打开。如果连接过，则 QMF for Windows 将建议采用先前所使用的连接条件。

2. 从每个表中选择具有相同数据类型的列，并单击**添加**。新的连接条件出现在提示查询中。

提示查询和 SQL

您可以使用提示查询界面来学习 SQL。

查看提示查询的 SQL

在提示查询视图中单击工具栏上的**查看 SQL** 按钮。

与提示查询等价的 SQL 语句将出现。您不可以从这个视图中修改 SQL 语句。

将提示查询转换为 SQL

可将提示查询转换成新的 SQL 查询文档。新的 SQL 查询可以修改、保存、打印和运行。在**查询**菜单中单击**转换为 SQL**。该查询将转换成新的 SQL 查询文档。

在提示查询中使用替换变量

替换变量在提示查询中的用法与在 SQL 查询中的用法相同。请参阅“查询中的替换变量”。

例如，替换变量可以用在:

- 行条件中

  ```
  DEPT Is Greater Than Or Equal To &MinDept
  ```

- 列说明中

  ```
  &InputNum
  ```

保存提示查询

提示查询可保存到您个人计算机上的文件中、保存到文件服务器上、或保存到数据库服务器上。

将提示查询保存到文件

1. 在一个打开的提示查询中，单击工具栏上的**保存**按钮。

   注: 如果该查询曾经保存过，则再次保存。如果以前没有保存过，则将打开“另存为”对话框。
2. 输入准备用来存放提示查询的文件名并单击**确定**。该查询被保存。

打开已保存的提示查询文件

1. 单击工具栏上的**打开**按钮。

   将打开“打开”对话框。
2. 选择要打开的文件并单击**确定**。选定的提示查询将在新的查询文档中打开。

在数据库服务器上保存提示查询

1. 在打开的提示查询中，单击工具栏上的**保存到服务器**按钮。

将打开“保存查询”对话框。

2. 输入拥有者、名称，选择是否与其他用户共享保存的查询，然后单击**确定**。该查询被保存到服务器上。

   如果有此名称的查询存在，则将提示您覆盖已有的查询。

打开数据库服务器上已保存的提示查询

您可以打开保存在数据库服务器上的提示查询。

1. 单击工具栏上的**从服务器打开**按钮。

将打开“从服务器打开”对话框。

2. 输入服务器、拥有者和名称，然后单击**确定**。将打开该提示查询。

打印提示查询

您可以打印提示查询。还可以打印提示查询的 SQL 文本。请参阅第12页的『打印 SQL 查询』。

预览提示查询

您可以在打印前预览提示查询的结果或文本。

1. 打开查询并激活“提示”视图。该查询出现。
2. 在**文件**菜单上单击**页面设置**。将打开“页面设置”对话框。
3. 对页面布局作适当的更改，并单击**确定**。
4. 单击工具栏上的**打印预览**按钮。

将出现打印查询的预览。

第4章 使用查询结果

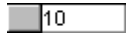
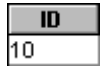
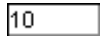
您可直接将格式、分组和聚合应用于查询结果。此格式可与查询一起保存，或导出为表单。

对查询结果排序并缩放大小

用户可对来自查询的数据结果进行选择、重新缩放、重新排序和排序。

选择列和行

一旦运行了查询，就可以使用“结果”视图中的控制来编辑和选择信息。

列和行选择器	外观	函数
行选择器		单击，以选中一行中的所有数据。
列选择器		单击，以选中一列中的所有数据。
单元格		直接在该单元格上单击来选中它。
“滚动到底部”与“滚动到顶部”按钮		单击，以滚动至查询结果集的顶部或底部。

重新缩放列和行

您可以通过重新缩放列与行来更改查询结果集的外观。

1. 用鼠标选择两列或两行间的黑色分割线。
2. 左右拖动或上下拖动分割线来缩放列或行。

注： 如果在重新缩放行或列后保存查询，则该查询今后将按照新的格式显示。

自动调整列和行

您可以使列与行的大小自动与它们所包含的数据相匹配。

用鼠标选择整列或行，然后双击该列（或行）与相邻对象之间的黑色分割线。该列或行将自动重新根据数据宽度调整其大小。

注： 如果在重新缩放行或列后保存查询，则该查询今后将按照新的格式显示。

排序查询结果

一旦运行了查询，就可以根据列对结果进行字母排序。

在查询的“结果”视图中选择列，然后从**结果**菜单中选择**按升序排序**。

查询结果将按升序排列。

-或者-

从查询“结果”视图中选择某列，然后从**结果**菜单中选择**按降序排序**。

查询结果将按降序排列。

注： 要对选定的列应用更复杂的排序，请选择**结果**菜单中的**排序**。

对列重新排序

您可更改查询结果中列的顺序。

从查询的“结果”视图中选择列，然后将它拖动到新的位置。

列将以新的顺序显示。

查询结果格式化

您可以更改用于显示查询和查询结果的字体。根据您计算机上所安装的不同字体，字体的选项也有所不同。关于添加字体的更多信息，请参阅操作系统的帮助程序。

注： 如果在选择了新的查询结果显示字体之后保存查询，查询结果将总是以新字体显示。

选择查询结果显示字体

1. 在“结果”视图中，选择**结果**菜单中的**字体**。将打开“字体”对话框。
2. 选择显示查询结果的字体、字形和大小，然后单击**确定**。查询结果将按照您指定的格式显示。

   注： 单击**设置为缺省值**，将选定的字体用作所有查询结果的缺省字体。

数字查询结果格式化

1. 从“结果”视图中选择包含数字值的列，然后从**结果**菜单中选择**格式**。将打开“格式”对话框。

2. 指定您想应用的格式，然后单击**确定**。值将根据您的选择进行格式化。

   注：单击**设置为缺省值**，将选定的字体用作所有查询结果的缺省字体。

将查询结果格式转换为表单

您可将查询结果格式转换为表单。

1. 在**结果**菜单中选择**显示报表**。
   将打开“选择表单”对话框。
2. 选择**从查询**并单击**确定**。
   查询结果格式转换为表单，并在新建表单窗口中打开。

分组和聚合查询结果

您可将分组、聚合和汇总格式应用到查询结果上。

分组查询结果

您可按照有无汇总信息来对查询结果进行分组。

1. 选择想要分组的列。
2. 从**结果**菜单中选择想要应用的分组类型。
   列将根据您的选择分组。

汇总查询结果

您可按列对查询结果进行汇总。

1. 选择想要分组的列。
2. 从**结果**菜单中选择想要应用的汇总类型。
   列将按照您的选择进行汇总。

保存查询结果和格式

您可保存查询结果并将格式保存为表单。

将查询结果保存为表

您可在数据库服务器上将查询结果保存为表。

1. 从**结果**菜单中选择**保存到数据库**。
   将打开“保存数据”对话框。
2. 输入拥有者和表名称，然后单击**确定**。
   查询结果将在数据库中保存为表。

将查询结果保存到文件

您可将查询结果保存到您个人计算机的文件中，或保存到文件服务器的文件中。

1. 从**结果**菜单中选择**保存到文件**。
   将打开“导出数据”对话框。
2. 指定想要保存文件的位置、所有导出选项，然后单击**确定**。
   查询结果将保存为文件。

打印查询结果

您可预览和打印查询结果。

预览查询结果

1. 打开并运行查询。将出现查询的结果。
2. 从**文件**菜单中选择**页面设置**。将打开“页面设置”对话框。
3. 对页面布局作适当的更改，并单击**确定**。
4. 单击工具栏上的**打印预览**按钮。

将出现打印查询结果的预览。

打印查询结果

1. 打开查询并激活“结果”视图。将出现该查询的结果。
2. 从**文件**菜单中选择**页面设置**。将打开“页面设置”对话框。
3. 对页面布局作适当的更改，并单击**确定**。
4. 单击工具栏上的**打印**按钮。

将打印查询结果。

第5章 使用报表

报表由具有表单格式的组合查询结果创建。

表单

表单是格式说明的集合，用于创建、显示和打印报表。

理解表单

表单是由许多组件组成的。这些组件都可以在表单文档中编辑。

主要 表单的主要组件，包括标题、脚注和断点。

断点 报表中可达 6 个小计行的属性、内容和位置。

计算 报表计算表达式。

注: 要使用表单计算，机器上必须安装 IBM 的 ObjectREXX。

列 报表中列的外观和格式。可定义属性包括: 列顺序、格式、用法、缩进和宽度。

条件 条件的格式化限制。例如，您可以将表单设置成只显示满足某些属性的行。

详细信息

报表详细信息标题和主体正文。在此可用自由格式文本组合或替换表列数据，以创建格式信件或地址标签。

最终 您的报表的最终文本的内容和位置。例如，可选择将最终文本和合计数据放在报表的末尾。

HTML HTML 标记和格式在 HTML 报表中的内容和位置。

选项 报表的多种外观选项。

页面 您的报表上的页面标题和脚注的内容和位置。

利用表单产生报表

报表是通过用表单中包含的格式选项组合查询结果而创建的。重复这一过程，就可以利用一个查询结果集产生多个报表。

1. 从查询结果视图中单击**显示报表**按钮。

将打开“选择表单”对话框。

2. 根据“选择表单”对话框中选择的表单类型，将要求您提供额外信息。根据情况指定文件的位置或拥有者及姓名、文档标题，并单击**确定**。然后，系统将使用选定的表单和当前的查询结果生成报表。

编辑表单

“表单”窗口提供了许多编辑和格式化表单的选项。

在打开的表单中，显示“表单”菜单。“表单”菜单列出了所有编辑和格式化表单的选项。您可以通过单击工具栏上任意一个按钮来编辑相应的组件。

创建表单

这些步骤都采用了表 Q.STAFF 中的样本数据。可尝试用不同的设置来创建您自己的定制表单。

步骤 1: 创建表单

1. 运行如下 SQL 查询以检索报表中要显示的数据:

```
SELECT * FROM Q.STAFF ORDER BY DEPT, NAME
```

将出现查询结果。

2. 单击工具栏上的**显示报表**按钮。将打开“选择表单”对话框。
3. 指定想要使用的缺省表单，然后单击**确定**。QMF for Windows 显示缺省的报表。要改变缺省格式，可单击工具栏上的某个表单组件按钮。表单工具栏上显示了用于各表单组件的按钮。

步骤 2: 更改列顺序

假设，我们希望 NAME 作为报表中的第一列，ID 作为报表中的第二列。列的次序在表单的“列”组件中指定。

1. 单击**表单**菜单上的**编辑...**，以显示“表单”对话框的列标签。
2. 通过在原有的顺序值上输入新的值来更改列的顺序。要将 NAME 放在报表的第一列，可将它的顺序号（列表中标记为 Seq 的列）改为 1。
3. 要使 ID 成为报表中的第二列，可将它的顺序号改成 2，并单击**确定**。QMF for Windows 在“表单”窗口中将按新的列次序显示报表。

步骤 3: 更改列标题

我们希望第一列的列标题为 EMPLOYEE，第二列的列标题为 COMMISSION。列标题文本是在表单的“列”组件中指定的。

1. 单击**表单**菜单上的**编辑...**，以显示“表单”对话框的列标签。
2. 通过在原有列标题文本上键入新的文本来更改列标题。将第一列的列标题改成 EMPLOYEE，最后一列的列标题改成 COMMISSION，然后单击**确定**。QMF for Windows 在“表单”窗口中以新的列标题显示报表。

步骤 4: 更改列格式

我们希望 SALARY 列以适当的货币符号显示。列的格式是由它的编辑代码确定的，该代码在表单的“列”组件中指定。

1. 单击**表单**菜单上的**编辑...**，以显示“表单”对话框的列标签。
2. 通过在原有编辑代码上键入新的值将 SALARY 列编辑代码改成 D2，单击**确定**。QMF for Windows 将使用“表单”窗口中适当的货币符号显示具有 SALARY 列的报表。

步骤 5: 添加汇总信息

我们希望将报表分成几部分，每一部分针对一个部门。另外，我们还希望在每一部分结束时查看各部门的总的 SALARY 和 COMMISSION。要做到这一点，我们需要指定报表中每一列的用法。列的用法是由用法代码确定的，在表单的“列”组件中指定。

1. 单击**表单**菜单上的**编辑...**，以显示“表单”对话框的列标签。
2. 要将报表根据 DEPT 分成几部分，可把 DEPT 的用法代码改成 BREAK1。以 BREAK 开头的用法代码将为指定列产生一个小节断点。单词 BREAK 后面的数字决定了断点级别；一个报表中最多可支持 6 个断点级别。
3. 要对每个 DEPT 包含一个总 SALARY 和 COMMISSION，可将 SALARY 和 COMMISSION 的用法代码改成 SUM。

4. 如果我们还在每个小节断点的后面加上描述性信息，报表就更易于理解了。要达到此目的，请单击**表单**菜单上的**断点...**。
5. 在“表单”对话框的“断点”标签上指定断点脚注文本。将断点脚注的第一行设置成“部门合计”并单击**确定**。QMF for Windows 将显示“表单”窗口。

步骤 6: 添加页标题和页脚注

我们希望为报表添加一个页标题和脚注。页标题和脚注是在表单的“页面”组件中指定的。

1. 单击**表单**菜单上的**编辑...**，以显示“表单”对话框的“页面”标签。
2. 这个对话框的上面一部分可用于指定页标题。将页标题的第一行设置成“部门报表”，第二行设置成“工资和津贴汇总”。选择标题的对齐方式。
3. 这个对话框的下面一部分可用于指定脚注。将脚注的第一行设置成 End Of Page。选择脚注的对齐方式，并单击**确定**。QMF for Windows 将显示“表单”窗口。

保存表单

您可以将表单保存到个人计算机、文件服务器或数据库服务器中。

将表单保存到文件

1. 在一个打开的表单中，单击**保存**按钮。
2. 如果该表单曾经保存过，则选择**保存**。如果以前没有保存过，则将打开“另存为”对话框。
3. 输入准备用来存放表单的文件名并单击**确定**。该表单被保存。

打开已保存的表单文件

1. 单击工具栏上的**打开**按钮。

将打开“打开”对话框。

2. 选择要打开的文件并单击**确定**。选定的表单在新的表单文档中打开。

在数据库服务器上保存表单

保存到服务器的表单可以被其他用户访问。如果您希望和其他用户共享表单，则应将它们保存到数据库服务器上。

1.

在打开的表单中，单击工具栏上的**保存到服务器**按钮。

将打开“保存表单”对话框。

2. 输入拥有者、名称，选择是否与其他用户共享保存的表单，并单击**确定**。该表单被保存到服务器上。

   如果此名称的表单已经存在，则将提示您覆盖已存在的表单。

打开数据库服务器上保存的表单

您可以打开保存在数据库服务器上的表单。

1. 单击工具栏上的**从服务器打开**按钮。

将打开“从服务器打开”对话框。

2. 输入服务器、拥有者和名称并单击**确定**。该表单被打开。

打印报表

可以生成打印的报表。

1. 打开一个表单并单击**页面设置**。
2. 对页面布局作适当的更改，并单击**确定**。
3. 单击**文件**菜单上的**打印报表**。

   该报表被打印。

导出报表

可以将报表导出到文件中。

1. 打开表单并单击**页面设置**。
2. 对页面布局作适当的更改，并单击**确定**。
3. 单击**文件**菜单上的**导出报表**。将打开“导出报表”对话框。

4. 输入准备用来存放报表的文件名并单击**确定**。该报表被导出。

第6章 运行过程

线性过程允许您执行单独的命令来运行查询、生成报表、编辑数据并执行其它函数。关于 QMF for Windows 支持的所有过程命令的完整列表，请参阅联机帮助程序。

带有逻辑的过程或 REXX 过程都与线性过程类似，但是它们包含 IBM 对象 REXX 程序设计语言和过程命令。对象 REXX 必须在本地安装，以便运行带有逻辑的过程。

运行过程

过程用于以一个命令执行多个函数。

创建新的线性过程

在**文件**菜单中选择**新建过程**。

新的过程文档将被打开。

创建带逻辑的新过程

1. 在**文件**菜单中选择**新建过程**。
   新的过程文档将被打开。
2. 输入 REXX 注释行作为过程的第一行。REXX 注释行以 /* 开头，以 */ 结尾。
3. 输入过程中想使用的任何 QMF 过程命令。QMF 必须以大写字母输入，并且括在引号内。
4. 输入过程中想使用的任何 REXX 命令。

注: REXX 命令在本地运行，而不是在数据库服务器上。因此，对象 REXX 必须在本地安装。

在数据库服务器上运行过程

1. 打开新的过程文档并键入一系列命令，或者从文件或数据库服务器打开一个已有的过程。

2. 单击工具栏上的**运行过程**按钮。

此过程运行。

保存过程

您可以将过程保存到个人计算机、文件服务器或数据库服务器中。

将过程保存到文件

1. 在一个打开的过程中，单击工具栏上的**保存**按钮。

如果该过程曾经保存过，则保存该过程。如果以前没有保存过，则将打开“另存为”对话框。

2. 输入准备用来存放过程的文件名并单击**确定**。该过程被保存。

打开已保存的过程文件

1. 单击工具栏上的**打开**按钮。

将出现“打开”对话框。

2. 选择要打开的文件并单击**确定**。选定的过程在新的过程文档中打开。

将过程保存到数据库服务器上

1. 在一个打开的过程中，单击工具栏上的**保存到服务器**按钮。

将打开“保存过程”对话框。

2. 输入拥有者、名称，选择是否与其他用户共享保存的过程，并单击**确定**。该过程被保存到服务器上。

   如果具有此名称的过程已经存在，则系统将提示您覆盖先前存在的过程。

打开数据库服务器上保存的过程

您可以打开保存在数据库服务器上的过程。

1.

   单击工具栏上的**从服务器打开**按钮。

   将打开“从服务器打开”对话框。

2. 输入服务器、拥有者和名称并单击**确定**。该过程将被打开。

打印过程

您可以打印过程的文本。

预览过程

1. 打开一个过程。过程命令将会出现。
2. 在**文件**菜单上单击**页面设置**。将打开“页面设置”对话框。

3. 对页面布局作适当的更改，并单击**确定**。
4. 单击工具栏上的**打印预览**按钮:

打印过程的预览出现。

打印过程

1. 打开一个过程。过程命令将会出现。
2. 在**文件**菜单上单击**页面设置**。将打开“页面设置”对话框。
3. 对页面布局作适当的更改，并单击**确定**。
4. 单击工具栏上的**打印**按钮:

该过程将被打印。

第7章 使用列表

列表为您提供了一种查看 QMF 对象集合的简便方法。

对象

QMF for Windows 识别四种类型的对象：查询、表单、过程以及表。您可以使用“列表”窗口根据对象名称、拥有者和类型来查看对象。

列表对象

1. 在**文件**菜单中选择**新建列表**。

   “列表”窗口将打开。

2. 指定拥有者和名称。

   注： 您可以使用模式从匹配对象列表中选择对象名。

   - 百分号 (%) 用于匹配任意长度任意字符的字符串。例如，要列出名称以字母 A 开头的所有表，可输入 A%。
   - 下划线字符 (_) 用于匹配单个字符。例如，要列出拥有者的第二个字符为字母 A 的所有表时，可输入 _A%。

3. 选择要搜索的对象的类型。

4. 单击**刷新列表**。数据库服务器中保存的匹配对象的列表将显示出来。

列表窗口命令

用鼠标右键单击“列表”窗口中的对象将显示一张命令列表，与“列表”菜单中的相同。

显示 打开选定的对象进行查看。对于查询、表单、过程和表有效。

运行 运行选定的对象。适用于查询和过程。

制作 根据选定的表创建查询。您可以选择制作 SQL SELECT 查询、 SQL UPDATE 查询、 SQL INSERT 查询或提示查询。适用于表。

编辑 打开选定的对象进行编辑。适用于表。

属性 显示所选对象的属性，包括注释、属性和历史使用信息。对于查询、表单、过程和表有效。

创建列表

您可以创建列表作为对象集合。例如，可以创建与查询、表单、过程和表相关的所有目录清单的列表，以便保持在一个地方工作。一旦创建此列表，就可从此列表中添加和移去对象，以及保存列表以备后用。

将对象添加到列表

您可将对象添加到列表中。

在打开的列表中，指定想要添加对象的拥有者信息和名称信息，然后单击工具栏上的**添加到列表**按钮。

与拥有者和名称相匹配的对象将添加到列表中。

从列表中移去对象

您可以从列表中移去不相关的对象。

在一个打开的列表中，单击工具栏上的**移去**按钮。

该对象被从列表中移去，但并未被删除。

将列表保存到文件

1. 在一个打开的列表中，单击工具栏上的**保存**按钮。

如果该列表曾经保存过，则保存该列表。如果以前没有保存过，则将打开“另存为”对话框。

2. 输入准备用来存放列表的文件名并单击**确定**。该列表被保存。

打开已保存的列表文件

1. 单击工具栏上的**打开**按钮。

将打开“打开”对话框。

2. 选择要打开的文件并单击**确定**。选定的列表将在列表窗口中打开。

第8章 使用作业文件

您可使用作业文件调度并运行过程。作业文件按照您预设的时间与日期使用 Windows 调度程序运行过程。

作业文件

您可创建作业文件，并将它们保存在本地或数据库服务器上。

创建作业文件

1. 在**文件**菜单中选择**新建作业**。
   新的作业文档将打开。

运行作业文件

您可运行已经保存在本地的作业文件。

1. 打开作业文件。
2. 单击工具栏上的**运行作业**按钮。

3. 左右拖动或上下拖动分割线来缩放列或行。

   注： 如果在重新缩放行或列后保存查询，则该查询今后将按照新的格式显示。

自动调整列和行

您可以使列与行的大小自动与它们所包含的数据相匹配。

用鼠标选择一整列或一整行，然后双击这些列或行与相邻对象之间的黑色分割线。该列或行将自动重新根据数据宽度调整其大小。

注： 如果在重新缩放行或列后保存查询，则该查询今后将按照新的格式显示。

排序查询结果

一旦运行了查询，就可以根据列对结果进行字母排序。

从查询的“结果”视图中选择某列，然后从**结果**菜单中选择**按升序排序**。

查询结果将按升序排列。

-或者-

从查询“结果”视图中选择某列，然后从**结果**菜单中选择**按降序排序**。

查询结果将按降序排列。

注： 要对选定的列应用更复杂的排序，请选择**结果**菜单中的**排序**。

对列重新排序

您可更改查询结果中列的顺序。

从查询的“结果”视图中选择列，然后将它拖动到新的位置。

列将以新的顺序显示。

查询结果格式化

您可以更改用于显示查询和查询结果的字体。根据您计算机上所安装的不同字体，字体的选项也有所不同。关于添加字体的更多信息，请参阅操作系统的帮助程序。

注： 如果在选择了新的查询结果显示字体之后保存查询，查询结果将总是以新字体显示。

选择查询结果显示字体

1. 在“结果”视图中，选择**结果**菜单中的**字体**。将打开“字体”对话框。
2. 选择显示查询结果的字体、字形和大小，然后单击**确定**。查询结果将按照您指定的格式显示。

   注： 单击**设置为缺省值**，将选定的字体用作所有查询结果的缺省字体。

数字查询结果格式化

1. 从“结果”视图中选择包含数字值的列，然后从**结果**菜单中选择**格式**。将打开“格式”对话框。
2. 指定您想应用的格式，然后单击**确定**。

   注： 单击**设置为缺省值**，将选定的字体用作所有查询结果的缺省字体。

将查询结果格式转换为表单

您可将查询结果格式转换为表单。

1. 在**结果**菜单中选择**显示报表**。
   将打开“选择表单”对话框。
2. 选择“从查询”并单击**确定**。
   查询结果格式转换为表单，并在新建表单窗口中打开。

分组和聚合查询结果

您可将分组、聚合和汇总格式应用到查询结果上。

分组查询结果

您可按照有无汇总信息来对查询结果进行分组。

1. 选择想要分组的列。
2. 从**结果**菜单中选择想要应用的分组类型。
   列将根据您的选择分组。

汇总查询结果

您可按列对查询结果进行汇总。

1. 选择想要分组的列。
2. 从**结果**菜单中选择想要应用的汇总类型。
   列将按照您的选择进行汇总。

保存查询结果和格式

您可保存查询结果并将格式保存为表单。

将查询结果保存为表

您可在数据库服务器上将查询结果保存为表。

1. 从**结果**菜单中选择**保存到数据库**。
   将打开“保存数据”对话框。
2. 输入拥有者和表名称，然后单击**确定**。
   查询结果将在数据库中保存为表。

将查询结果保存到文件

您可将查询结果保存到您个人计算机的文件中，或保存到文件服务器的文件中。

1. 从**结果**菜单中选择**保存到文件**。
   将打开“导出数据”对话框。
2. 指定想要保存文件的位置、所有导出选项，然后单击**确定**。
   查询结果将保存为文件。

打印查询结果

您可预览和打印查询结果。

预览查询结果

1. 打开并运行一个查询。将出现该查询的结果。
2. 从**文件**菜单中选择**页面设置**。将打开“页面设置”对话框。
3. 对页面布局作适当的更改，并单击**确定**。
4. 单击工具栏上的**打印预览**按钮:

将出现打印查询结果的预览。

打印查询结果

1. 打开一个查询并激活“结果”视图。将出现该查询的结果。
2. 从**文件**菜单中选择**页面设置**。将出现“页面设置”对话框。
3. 对页面布局作适当的更改，并单击**确定**。
4. 单击工具栏上的**打印**按钮。

将打印查询结果。

第9章 使用静态查询

静态查询是一个 SQL 查询，它被预先传送到数据库服务器并绑定到包中。运行静态查询时，数据库服务器使用绑定在包中的 SQL 文本，而不是当前出现在查询窗口中的 SQL 文本。静态查询在资源使用上的效率要比动态查询高，但静态查询不可以进行编辑。

静态查询

静态查询是从先前存在的 SQL 和提示查询创建的。

创建静态查询

1. 从**查询**菜单中选择**绑定包**。将打开“绑定静态包”对话框。

2. 选择“包”标签，输入一个集合标识和包名，并改变其它必要的选项。
3. 如果该查询中包含有替换变量，则选择“输入变量”标签。用主变量代替所有替换变量。
4. 单击**确定**。这个静态查询被绑定。

   注： 绑定查询之后，就必须同时将该查询保存到文件或数据库服务器中。

用主变量代替替换变量

绑定软件包时，必须指定主变量来代替 SQL 文本中的每个替换变量。但是，替换变量并不总是可以由一个主变量直接代替。在将文本传送到数据库服务器之前，替换变量提供了查询文本中的直接文本替换。主变量被作为查询的一部分传送到数据库服务器。请参阅您的数据库服务器的文档，来查看有关可以在哪里以及怎样在查询中使用主变量的规则。

指定了替换变量和主变量间的关系之后，QMF for Windows 将记住这些关系，并在以后的查询中建议使用这些关系，使得后续的软件包绑定更简单。

主变量的有效数据类型有:

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

1. 从“绑定静态包”对话框中选择“输入变量”标签。

2. 输入每个主变量的变量类型，并单击**确定**。替换变量被转换成主变量。

运行静态查询

您就可以象运行其它查询一样来运行静态查询。请参阅第7页的『SQL 查询』 上的“SQL 查询”。

第10章 使用表编辑器

可使用表编辑器来搜索、添加、编辑或删除您的表中存放的数据，而不必编写 SQL 语句。

表编辑器

表编辑器赋予您编辑和搜索数据的灵活性。

使用编辑器搜索行

1. 从**文件**菜单选择**表编辑器**。将打开“表编辑器”对话框。

2. 指定表。

   注： 可使用匹配模式从表列表中选择表名。

   - 百分号 (%) 用于匹配任意长度任意字符的字符串。例如，要列出名称以字母 A 开头的所有表，可输入 A%。
   - 下划线字符 (_) 用于匹配单个字符。例如，要列出拥有者的第二个字符为字母 A 的所有表时，可输入 _A%。

   输入了模式后，可单击**列出表**，并从所显示的列表中选择表。

3. 选择一种“保存方式”。
   - 立即保存 - 每次更改之后表将在数据库中立即更新。

- 结束时保存 - 完成所有更改之后表才在数据库中更新。当您在进行更改时，其他用户将无法对该表进行更改。

4. 单击**编辑**。将打开“编辑表”对话框。
5. 在“值”列表中输入要查找的值，或者在“其它搜索条件”字段中键入搜索条件，来指定更复杂的搜索条件。可在“其它搜索条件”字段中输入任何有效 SQL 谓词。
6. 单击**开始搜索**。第一个满足条件的行出现在“值”列中。

添加行

1. 在“编辑表”对话框中输入新记录的有关信息。
2. 单击**插入行**。这个新的行被添加到表中。
3. 单击**确定**。您所作的更改被保存。

更改行

1. 从“编辑表”对话框中搜索要更改的行。
2. 单击**下一行**，直至显示您想要更改的行。
3. 在“值”列中编辑数据并单击**更新行**。该行被更新。
4. 单击**确定**。您所作的更改被保存。

删除行

1. 从“编辑表”对话框中搜索要删除的行。
2. 单击**下一行**，直至显示您想要删除的行。

3. 单击**删除行**。该行被删除。
4. 单击**确定**。您所作的更改被保存。

在查询结果视图中编辑表

您可以直接从查询结果视图中编辑表。

从查询结果视图中删除行

您可以在查询结果视图中删除表中个别的行。

从查询结果视图中选定行，然后选择**编辑**菜单中的**删除**。该行被删除。

在查询结果视图中更新列

您可以在查询结果视图中更新单独几列的内容。

在查询结果视图中，双击单元格，输入新的值，然后按 Enter 键。该表被更新。

DB2 表单

如果您安装了 “DB2 表单用户” 组件，就可将它作为不包含 LOB 数据的表的表编辑器使用。有关 DB2 表单的更多信息，请访问 www.rocketsoftware.com/db2forms 网站上的 Resource Center for DB2 Forms。

第11章 分布数据

您可以将数据导出到其它数据库和应用程序中。

导出数据

您可以用以下方式将数据从 QMF for Windows 导出到其它应用程序:

- 将数据导出到文本、CSV、IXF 或 HTML 文件中
- 将查询结果保存到表中
- 将查询结果直接添加到 Microsoft Excel 电子数据表中

将数据导出到文件

1. 查看查询结果时，请选择**文件**菜单上的**导出数据**。将打开“导出数据”对话框。

2. 选择期望的“输出文件类型”，然后单击**选项**按钮。根据所选输出文件类型的不同，将会打开“导出文本/DEL 选项”对话框、“导出 HTML 选项”对话框、“导出 IXF 选项”对话框或“导出 CSV 选项”对话框。
   - 可以生成有 .TXT 扩展名的文本文件。这是一个标准的 ASCII 文件，有可选字符串和列定界符（在“导出文本/DEL 选项”对话框中指定）。
   - 可以生成有 .HTM 扩展名的 HTML 文件。这是一个 HTML 文件，可用任何 Web 浏览器来查看。所有 HTML 标记都是在文件中自动生成的；我们已经准备好在您的因特网或内部网站上发布它。您在“导出 HTML 选项”对话框中选择的选项控制了要导出数据的外观。

- 可以生成 .IXF 文件。IXF 导出保存所有数据库信息，包括列标题和数据类型。它通常用于从一个数据库到另一个数据库的传送。
- 可以生成 .CSV 文件。CSV 导出很类似于文本导出，它使用逗号作为列定界符。电子表格应用程序常用此格式。

3. 为导出文件的选定类型选择选项，并单击**确定**。“选项”对话框关闭。
4. 在“导出数据”对话框上单击**确定**。数据将被导出。

导入数据

您可以导入保存在 IXF 文件中的数据。一旦数据导入查询窗口，就可保存到数据库服务器、导出为新的文件、或用于报表。支持 PC/IXF 和字符方式 System/370 IXF 文件。

1. 从**文件**菜单中选择**导入数据**。“导入数据”对话框将打开。

2. 选择要导入的文件并单击**确定**。导入的数据将在新查询窗口中显示。

将数据保存到数据库服务器

您可将导入的查询结果保存到数据库表中。

1. 当查看导入的查询结果时，请选择**文件**菜单上的**保存数据**，将打开“保存数据”对话框。

2. 选择数据库服务器，输入表的拥有者和名称，选择所有其它需要的选项，然后单击**确定**。数据被保存。

使用发送到命令

QMF for Windows 包含“发送到”命令和基本电子邮件客户程序。您可将“发送到”命令与作业文件一起使用，以调度查询并分发它们的结果。

1. 从**文件**菜单中选择**发送到**和**因特网邮件收件人**，将打开“消息”对话框。
2. 指定收件人、主题和消息文本，然后单击**下一步**，将打开“附件”对话框。
3. 添加或移去此消息的任何附件，然后单击**下一步**，将打开“发送消息”对话框。
4. 指定您邮件服务器的名称，然后单击**完成**。消息将被发送出去。

使用 Microsoft Excel 加载宏

QMF for Windows 包括 Microsoft Excel 7.0 或更高版本的加载宏。这些加载宏可从 Excel 运行 QMF for Windows，并将查询结果直接返回到电子表格。如果您选择“典型”安装选项，或者选择“定制”安装选项、并选择 Microsoft Excel 加载宏选项，那么将自动安装适当的加载宏。

1. 单击 Excel 工具栏上的 **QMF for Windows** 按钮。

   打开 QMF for Windows。
2. 从 QMF for Windows 选择并运行一个查询。将出现该查询的结果。
3. 选择要返回到 Excel 的数据。
4. 从**文件**菜单中选择**将数据返回 Microsoft Excel**，将打开 Excel 并显示 “QMF for Windows 加载宏” 对话框。
5. 输入数据的目的地范围，并单击**确定**。数据被添加到电子表格。

使用示例应用程序

QMF for Windows 提供了一些示例应用程序和集成方案。要想了解更多信息，请访问 IBM 的 Web 站点: http://www.ibm.com/qmf/。

第12章 使用 QMF 报表中心

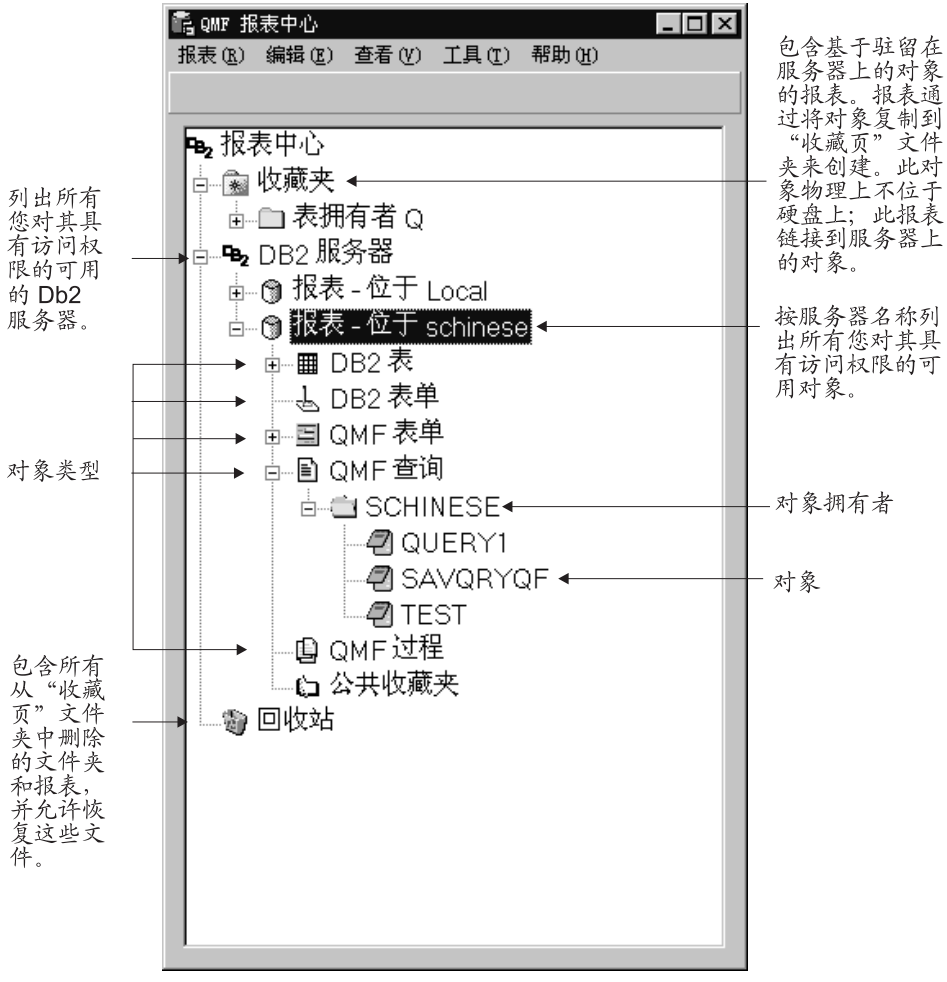
通过使用共享 QMF 查询、表单、过程和表，“QMF 报表中心”使您能够生成定制报表。利用对这些对象的快速访问，可指定数据格式优先选项，生成可在多种应用程序中查看和处理的定制报表。

QMF 报表中心入门

- 用鼠标右键单击任何对象或文件夹，激活与工具条菜单中可获得的选项功能相同的选项。
- 单击任何文件夹旁的加号（+）可打开第一层目录。在单击加号（+）时按住 SHIFT 键可打开此文件夹下所有层次的目录。

QMF 报表中心窗口

“QMF 报表中心”窗口包含由可用的收藏夹、DB2 服务器、公共收藏夹、对象和回收站等组成的树形结构。

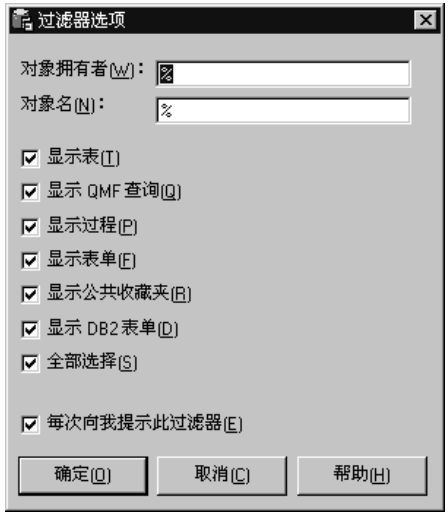


此窗口中显示的对象包括一个图标，它表示对象输出所关联的应用程序的类型。

连接到服务器

1. 如果在“DB2 服务器”下没有服务器名称显示，请单击旁边的加号（+）。

2. 单击服务器旁边的加号（**+**），将打开“过滤器选项”对话框。



3. 选择您想看到的对象类型，然后单击**确定**。服务器上可用的对象将显示，并按照对象类型分组。

使用报表和对象

报表基于 QMF 对象。包含在您个人收藏夹和公共收藏夹文件夹中的所有项都被认为是报表；您可处理这些项目的格式和显示它们的选项。包含在这些“收藏夹”文件夹中的项目链接到驻留在服务器上的 QMF 对象。实际上，您不能修改 QMF 对象，您所修改的只是引用为报表的对象的链接。由于报表是基于对象的，因此对象的属性也应用于报表。

您可从驻留在服务器的对象创建报表；但是，它们不保存到服务器上。此功能允许您快速地创建一次性报表。然而，在从某个服务器上的对象创建报表之后，将向您提供把报表保存到收藏夹文件夹的选项。

运行报表

可从您的收藏夹文件夹中运行报表，或从位于服务器上的对象运行。

1. 选中报表和对象之后，从“报表”菜单中选择*属性*，“报表属性”对话框将打开。

2. 如果需要，可定义属性。
3. 单击**运行**按钮，以处理报表，并且如果选择了“报表属性输出”对话框中的*发布之后查看报表*选项，报表将在您指定的应用程序中显示。

您还可通过下述任意方法迅速运行报表:

- 选择报表，然后从“报表”菜单中选择*运行*。
- 用鼠标右键单击报表，然后选择*运行*。
- 双击报表名称。

使用文件夹和收藏夹

文件夹用于对报表和 QMF 对象分组；它根据对象拥有者的名称命名。您可对文件夹执行与报表相同的操作，例如运行报表和定义报表属性。对文件夹执行这些操作将把操作应用到此文件夹中包含的每一个报表上。例如，如果您想要连续的运行包含在一个文件夹中的报表，请选中文件夹，然后选择“报表”菜单中的**运行**。

“QMF 报表中心”包含两个用于存储报表的顶层文件夹。这两个文件夹包含指向服务器上对象的报表；对象自身不包含在收藏夹文件夹中。个人收藏夹文件夹驻留在本地（在您的个人计算机上），因此您是可访问该文件夹及其内容的唯一用

户。公共收藏夹文件夹驻留在服务器上，可被所有经过授权的用户访问。尽管在每个服务器上不会有多个公共收藏夹文件夹，但您可以访问多个公共收藏夹文件夹，这取决于您的资源限制。

当您将 QMF 对象复制到收藏夹文件夹时，文件夹会自动重命名，以包括对象类型和拥有者名称。从服务器（例如，所有查询）复制整个对象类型（也就是相同类型对象的文件夹）时，服务器名称也要包括在新名称之中。

将报表添加到收藏夹中

您可将任何对象或报表从服务器添加到您的个人收藏夹文件夹中，或者添加到服务器上的公共文件夹中（只要系统管理员已授予您许可权）。

要将报表添加到个人收藏夹中:

选中报表或对象之后，从“报表”菜单中选择*添加到收藏夹*，或将报表或对象拖动到个人收藏夹文件夹中。一个以如下命名约定命名的报表将添加到您个人收藏夹文件夹的顶部：ObjecttypeOWNERNAME.OBJECTNAME。

要将报表添加到公共收藏夹:

将 QMF 对象或报表拖动到服务器上的公共收藏夹文件夹中。可从您的个人收藏夹文件夹或从任何服务器添加报表。

注： 当添加到公共收藏夹或在公共收藏夹中修改报表，并且把更新保存到服务器之前，您必须从“报表”菜单中选择*将更改保存到公共收藏夹*。

有关使用“QMF 报表中心”的更多信息，请参阅联机帮助系统。

第13章 使用 QMF for Windows API

您可创建使用 QMF for Windows 应用程序接口的定制应用程序。

通过 API 控制 QMF for Windows

以下步骤提供了如何利用 API 控制 QMF for Windows 的概述。

1. 创建 QMF for Windows API 对象的实例。如果是在使用 Microsoft Visual Basic，则添加对 QMF for Windows 类型库 qmfwin.tlb 的引用，然后使用 Dim 语句:

   ```
   Dim QMFWin As New QMFWin
   ```

   或 CreateObject 语句:

   ```
   Dim QMFWin As Object
   Set QMFWin = CreateObject("QMFWin.Interface")
   ```

   注: 如果是在使用其它开发环境，请参阅相应的产品文档来确定如何完成此步骤。

2. 选择要使用的 DB2 服务器，并调用 InitializeServer() 来初始化到数据库的连接。

   注: 只有在用户标识和口令经 DB2 验证后方可初始化服务器。可让 QMF for Windows 提示输入用户标识和口令，或者在应用程序中提示输入这些信息，并在 InitializeServer() 函数调用时将它们作为参数传递。

3. 用 InitializeQuery() 选择想要运行的查询。如果查询包含变量，则可使用 SetVariable() 函数来设置这些变量的值。
4. 打开或执行该查询。使用 Open() 函数为 SELECT 语句来打开查询的游标，使用 Execute() 函数执行非 SELECT 语句的 SQL。
5. 如果查询是一 SELECT 语句，则通过重复调用 FetchNextRow() 访存数据行。想一次访存多行，请使用 FetchNextRows()，也可使用 CompleteQuery() 指导 QMF for Windows 访存所有行。
6. 如果查询是一 SELECT 语句，则通过使用 Close() 函数关闭查询。
7. 使用 Commit() 或 Rollback() 函数终止工作单元。

块调用

所有 QMF for Windows API 函数都是同步的。这意味着数据库操作完成之前，它们都是阻塞的，或不返回数据。这样的实现方式正是所期望的，因为它简化了客

户应用程序的编程。但是，如果您的客户应用程序是单线程的，它在等候 QMF for Windows API 返回值时，将不能响应用户输入或执行屏幕刷新。

QMF for Windows API 一次只响应来自一个客户的函数调用。如果您的客户应用程序是多线程的，则必须:

- 在进行另一个函数调用之前等待正在进行的那个函数调用的结束，或者
- 创建 QMF for Windows API 的多个实例（为每个使用 API 的线程创建一个）。

连接到数据库

对于所有受后继回滚或提交影响的数据库活动（包括打开查询、读取数据及执行 SQL 语句），QMF for Windows API 对象的每个实例会创建并使用唯一的数据库连接。

如果通过多次调用 InitializeQuery() 在 QMF for Windows API 对象的一个给定实例中创建了多个查询，则所有的这些查询都共享同一个单独连接。

QMF for Windows API 一次只响应来自一个客户的函数调用。如果您的客户应用程序是多线程的，则必须调用:

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

QMF for Windows 创建并使用第二个数据库连接，以处理管理性的数据库活动（如检索 QMF 信息）。这第二个连接对于支持客户应用程序的一致性回滚和提交机制是必需的。

QMF for Windows API 对象自动地处理这些数据库连接。但是，如果您的系统管理员限制了允许连接个数的，则一定要记住 QMF for Windows API 对象的每个实例都会使用两个连接。

API 参考

此参考列出了所有使用 QMF for Windows API 创建应用程序的有效命令。

AddDecimalHostVariable()

short AddDecimalHostVariable(long *QueryID*, short *Type*, short *Precision*, short *Scale*, const VARIANT& *Value*)

描述

此函数将 Value 中的数据应用到由 *QueryID* 初始化的静态 SQL 语句的变量中。对该语句中的每个变量都调用这个函数。QMF for Windows 不将值与变量相匹配，因此，您必须按 SQL 语句中变量出现的次序来调用这个函数。

参数

名称	描述
QueryID	查询的标识，由 InitializeStaticQuery() 返回。
Type	传递到数据库服务器的值的 SQL 数据类型。这个值会影响 *Value* 从 VARIANT 数据类型到实际传递的值的转换。AddDecimalHostVariable() 的唯一有效值是 484 (RSDT_DECIMAL)。
Precision	小数值的精度。
Scale	小数值的标度。
Value	语句中要替换的数据值。要指定空值，请将其类型设置为 VT_EMPTY。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

AddHostVariable()

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

描述

此函数将 Value 中的数据应用到由 *QueryID* 初始化的静态 SQL 语句的变量中。对该语句中的每个变量都必须调用这个函数。QMF for Windows 不将值与变量相匹配，因此，您必须按 SQL 语句中变量出现的次序来调用这个函数。

参数

名称	描述
QueryID	查询的标识，由 InitializeStaticQuery() 返回。
Type	传递到数据库服务器的值的 SQL 数据类型。这个值会影响 *Value* 从 VARIANT 数据类型到实际传递的值的转换。

Value	语句中要替换的数据值。要指定空值，就要将它的类型设置为 VT_EMPTY。

Type 的有效值有:

值	含义
384 (RSDT_DATE)	日期
388 (RSDT_TIME)	时间
392 (RSDT_TIMESTAMP)	时间戳记
448 (RSDT_VARCHAR)	可变长度字符串
452 (RSDT_CHAR)	字符串
464 (RSDT_VARGRAPHIC)	可变长度图形
468 (RSDT_GRAPHIC)	图形
480 (RSDT_FLOAT)	浮点数
496 (RSDT_INTEGER)	4 位整数
500 (RSDT_SMALLINT)	2 位整数

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

BindDecimalHostVariable()

short BindDecimalHostVariable(BSTR *CollectionName*, BSTR *PackageName*, short *SectionNumber*, short *Number*, BSTR *Name*, short *DataType*, short *Precision*, short *Scale*)

描述

此函数绑定指定段中的一个变量。将文本 ″:H″ 包含在 SQL 文本中，作为主变量的位置标志符。对于 SQL 文本中的每个小数主变量，都必须调用 BindDecimalHostVariable() 来指定有关该变量的信息。

参数

名称	描述
CollectionName	想要绑定的包的集合标识。
PackageName	想要绑定的包的名称。
SectionNumber	想要绑定的集合和包中语句的段号。
Number	想要绑定的变量的标识。SQL 语句中的第一个变量是变量 0，依此类推。

Name	数据库服务器用于诊断的一个参数。QMF for Windows 不验证也不要求这个值。
DataType	变量的 SQL 数据类型。BindDecimalHostVariable() 的唯一有效值是 484 (RSDT_DECIMAL)。
Precision	小数值的精度。
Scale	小数值的标度。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

BindHostVariable()

short BindHostVariable(BSTR *CollectionName*, BSTR *PackageName*, short *SectionNumber*, short *Number*, BSTR *Name*, short *DataType*, short *Length*)

描述

此函数绑定指定段中的一个变量。将文本 ″:H″ 包含在 SQL 文本中，作为主变量的位置标志符。对于 SQL 文本中的每个主变量，都必须调用 BindHostVariable() 来指定有关该变量的信息。

参数

名称	描述
CollectionName	想要绑定的包的集合标识。
PackageName	想要绑定的包的名称。
SectionNumber	想要绑定的集合和包中语句的段号。
Number	想要绑定的变量的标识。SQL 语句中的第一个变量是变量 0，依此类推。
Name	数据库服务器用于诊断的一个参数。QMF for Windows 不验证也不要求这个值。
DataType	变量的 SQL 数据类型。
Length	变量的长度。

DataType 的有效值有:

值	含义
384 (RSDT_DATE)	日期
388 (RSDT_TIME)	时间

392 (RSDT_TIMESTAMP)	时间戳记
448 (RSDT_VARCHAR)	可变长度字符串
452 (RSDT_CHAR)	字符串
464 (RSDT_VARGRAPHIC)	可变长度图形
468 (RSDT_GRAPHIC)	图形
480 (RSDT_FLOAT)	浮点数
484 (RSDT_DECIMAL)	小数
496 (RSDT_INTEGER)	4 位整数
500 (RSDT_SMALLINT)	2 位整数

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

BindSection()

short BindSection(BSTR *CollectionName*, BSTR *PackageName*, short *SectionNumber*, BSTR *SQLText*)

描述

此函数设置在绑定期间集合和包的指定段号中所用的 SQL 文本。

参数

名称	描述
CollectionName	想要绑定的包的集合标识。
PackageName	想要绑定的包的名称。
SectionNumber	想要绑定的集合和包中语句的段号。
SQLText	想要绑定的语句的 SQL 文本。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

CancelBind()

short CancelBind(BSTR *CollectionName*, BSTR *PackageName*)

描述
此函数取消刚刚初始化的一个绑定操作。有关这个已命名包的所有信息都被舍弃。

参数

名称	描述
CollectionName	想要绑定的包的集合标识。
PackageName	想要绑定的包的名称。

返回值
如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

ChangePassword()

short ChangePassword(BSTR *NewPassword*)

描述
此函数更改先前在 InitializeServer() 调用时指定的用户标识的口令。

注: 并非所有类型的数据库服务器都支持更改口令的。如果 InitializeServer() 调用时指定的服务器不支持更改口令，则返回出错信息，并且口令不改变。

参数

名称	描述
NewPassword	新口令。

返回值
如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

ClearList()

short ClearList(short *Type*)

描述
此函数重新初始化由 *Type* 参数指定的内部列表。

参数

名称	描述
Type	RSL_SERVER 或 RSL_QUERY 值。

返回值

如果成功则为零，否则为 RS_ERROR_OUTOFRANGE。

相关主题

Open()

Close()

short Close(long *QueryID*)

描述

此函数用于关闭查询并使 *QueryID* 无效。如果查询有打开的游标，则关闭该游标，并将数据库释放给其它用户使用。此函数不终止与数据库服务器的连接。由于连接保持打开，因此不执行回滚或提交。

注: 此函数的名称与 Microsoft Access 2.0 的关键字 Close 冲突。如果在使用 MS Access 2.0，则应将函数名放在方括号 [ ] 内。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

Execute()
Open()

Commit()

short Close(long *QueryID*)

描述

此函数提交您在当前工作单元中所作的任何更改、结束当前工作单元、关闭打开的所有游标、并使所有查询标识无效。

注: 此函数的名称与 Microsoft Access 2.0 的关键字 Commit 冲突。如果在使用 MS Access 2.0，则应将函数名放在方括号 [ ] 内。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

Rollback()

CompleteQuery()

short CompleteQuery(long *QueryID*)

描述

此函数读取结果集中的所有行，并将它们保存到 QMF for Windows 内部。如果查询有打开的游标，则关闭该游标，并将数据库释放给其它用户使用。可使用 FetchNextRow() 或 FetchNextRows() 检索这些行。完成此查询后，可调用 Close() 函数。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

CopyToClipboard()

short CopyToClipboard(long *QueryID*, long *FirstRow*, long *FirstCol*, long *LastRow*, long *LastCol*, BOOL *IncludeColHeadings*, [VARIANT *DateTimeFomat*])

描述

此函数将指定范围内的行和列复制到剪贴板上。如果还没有检索想要复制到剪贴板的所有行中的行数据，则可在调用这个函数之前调用 CompleteQuery()。如果您试图复制不是从数据库中检索到的行，则此函数将返回出错信息。

参数

名称	描述

QueryID	查询的标识，由 InitializeQuery() 返回。
FirstRow	想要复制的第一行。
FirstCol	想要复制的第一列。
LastRow	想要复制的最后一行；如果已经包含了所有行，则为 -1。
LastCol	复制中想要包括的最后一列；如果包含了所有列，则为 -1。
IncludeColHeadings	这个参数值非零时第一行中包含列标题，为零则表示不包含列标题。
DateTimeFormat	用于日期和时间值的格式，可任选。有效值是 0（ISO 格式）、1（USA 格式）、2（EUR 格式）、3（JIS 格式）或 4（Windows 控制面板格式）。缺省值是 4。

注: 结果集中第一行的值为 0，最后一行的值比总行数小一。结果集中第一列的值为 0，最后一列的值比总列数小一。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。如果结果集为空或在数据库中没有检索到任何行，将返回非零值，除非 *FirstRow*=0 并且 *LastRow*=1。如果发生这种情况，将返回 0，并将空字符串复制到剪贴板中。

DeleteQMFObject()

short DeleteQMFObject(BSTR *OwnerAndName*)

描述

此函数删除 QMF 对象(查询、表单、过程或表)。

参数

名称	描述
OwnerAndName	包含欲删除对象的拥有者和名称的字符串，所有者和名称之间用句点隔开。例如， `John.Query2`

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

EndBind()

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

描述

此函数完成静态 SQL 包的绑定过程。对这个函数的调用将使 QMF for Windows 把当前包的完整信息发送到数据库中以待处理。

参数

名称	描述
CollectionName	先前调用 StartBind() 时所用的集合名。
PackageName	先前调用 StartBind() 时所用的包名。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

Execute()

short Execute(long *QueryID*)

描述

此函数执行使用 SQL 动词（不是 SELECT）的 SQL 语句。当此语句不返回任何结果时使用 Execute()。对于返回结果的语句，使用 ExecuteEx()。对于使用 SELECT 动词的语句，使用 Open() 代替 Execute() 或 ExecuteEx()。要确定查询使用的动词，请调用 GetQueryVerb()。

注: 此函数的名称与 Microsoft Access 2.0 的关键字 Execute 是冲突的。如果在使用 MS Access 2.0，则应将函数名放在方括号 [ ] 内。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

Execute()
Open()

ExecuteEx()

short ExecuteEx(long *QueryID*,VARIANT* *Result*)

描述

此函数执行使用 SQL 动词（不是 SELECT）的 SQL 语句。当语句返回结果（例如，带一条 SELECT INTO 语句）时使用 ExecuteEx()。对于不返回任何结果的语句，使用 Execute()。对于使用 SELECT 动词的语句，使用 Open() 代替 Execute() 或 ExecuteEx()。要确定查询使用的动词，请调用 GetQueryVerb()。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Result	指向存放结果的 VARIANT 的指针。结果是一数组（变量类型为 VT_ARRAY | VT_BSTR），包含了结果中每个列的值。 每个值都是以它的的本机数据类型或最接近的变体数据类型来指定的。所支持的返回类型有：字符串（变体类型 VT_BSTR）、浮点（变体类型 VT_R4）、双精度浮点型（变体类型 VT_R8）、短整型（变体类型 VT_I2）、长整型（变体类型 VT_I4）和二进制（变体类型 VT_UI1 | VT_ARRAY）。 调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

ExecuteStored Procedure()

short ExecuteStoredProcedure(long *QueryID*,[VARIANT *vaCommitOK*], [VARIANT *vaMaxResultSets*], [VARIANT *vaColumnNames*], [VARIANT *vaColumnLabels*], [VARIANT *vaColumnComments*])

描述

此函数执行 SQL 语句，它使用 SQL 动词 CALL 在数据库服务器运行存储的过程。当此存储过程不返回任何结果（而不是结果集，或者除结果集以外）时使用 ExecuteStoredProcedure()。对于不返回结果的存储过程，使用 ExecuteStoredProcedureEx()。

为使用 ExecuteStoredProcedure() 执行操作而初始化存储过程，请先调用 InitializeQuery() 指定使用 CALL 语句的 SQL 语句。在 CALL 语句中，存储过程名称必须指定为字面量。CALL 语句中指定的任何参数（常量或其它）均被忽略。取而代之的是使用 AddHostVariable() 指定输入和输出变量。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。该查询的 SQL 文本应当指定调用语句。
vaCommitOK	可选的布尔值，指定存储过程可提交工作单元，或指定此操作应被禁止。缺省值是“真”。
vaMaxResultSets	可选的数值，它指定该存储过程应当允许返回的结果集的最大数目。如果不想要该存储过程返回任何结果集，或者数据库服务器不支持从 DRDA 上的存储过程返回结果，则指定零。
vaColumnNames	可选的布尔值，指定数据库是否应当在每个返回的结果集中返回列的名称。
vaColumnLabels	可选的布尔值，指定数据库是否应当在每个返回的结果集中返回列的标号。
vaColumnComments	可选的布尔值，指定数据库是否应当在每个返回的结果集中返回列的注释。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

ExecuteStored ProcedureEx()

short ExecuteStoredProcedureEx(long *QueryID*,VARIANT* *Result*,[VARIANT *vaCommitOK*], [VARIANT *vaMaxResultSets*], [VARIANT *vaColumnNames*], [VARIANT *vaColumnLabels*], [VARIANT *vaColumnComments*])

描述

此函数执行 SQL 语句，它使用 SQL 动词 CALL 在数据库服务器运行一个存储的过程。当此存储过程返回结果（而不是结果集，或者除结果集以外）时使用 ExecuteStoredProcedureEx()。对于不返回结果的存储过程，使用 ExecuteStoredProcedureEx()。

为使用 ExecuteStoredProcedure() 执行操作而初始化存储过程，请先调用 InitializeQuery() 指定一个使用 CALL 语句的 SQL 语句。在 CALL 语句中，存

储过程名称必须指定为字面量。CALL 语句中指定的任何参数（常量或其它）均被忽略。取而代之的是使用 AddHostVariable() 指定输入和输出变量。

如果存储过程返回结果集，则调用 GetStoredProcedureResultSets() 在结果集中检索查询标识。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。该查询的 SQL 文本应当指定调用语句。
Result	指向存放结果的 VARIANT 的指针。结果是一数组（变量类型为 VT_ARRAY | VT_BSTR），包含了结果中每个列的值。 每个值都是以它的的本机数据类型或最接近的变体数据类型来指定的。所支持的返回类型有: 字符串（变体类型 VT_BSTR）、浮点（变体类型 VT_R4）、双精度浮点型（变体类型 VT_R8）、短整型（变体类型 VT_I2）、长整型（变体类型 VT_I4）和二进制（变体类型 VT_UI1 | VT_ARRAY）。 调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。
vaCommitOK	可选的布尔值，指定存储过程可提交工作单元，或指定此操作应被禁止。缺省值是 “真”。
vaMaxResultSets	可选的数值，它指定该存储过程应当允许返回的结果集的最大数目。如果不想要该存储过程返回任何结果集，或者数据库服务器不支持从 DRDA 上的存储过程返回结果，则指定零。
vaColumnNames	可选的布尔值，指定数据库是否应当在每个返回的结果集中返回列的名称。
vaColumnLabels	可选的布尔值，指定数据库是否应当在每个返回的结果集中返回列的标号。
vaColumnComments	可选的布尔值，指定数据库是否应当在每个返回的结果集中返回列的注释。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

Export()

short Export(long *QueryID*, long *FirstRow*, long *FirstCol*, long *LastRow*, long *LastCol*, short *Format*, short *StringDelimiter*, short *ColumnDelimiter*, BOOL *IncludeColHeadings*, BSTR *FileName*, [VARIANT *DateTimeFomat*])

描述

此函数将指定范围内的行和列复制到剪贴板上。如果还没有检索想要复制到剪贴板的所有行中的行数据，则可在调用这个函数之前调用 CompleteQuery()。如果您试图复制不是从数据库中检索到的行，则此函数将返回出错信息。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
FirstRow	想要导出的第一列。
FirstCol	想要导出的第一列。
LastRow	想要复制的最后一行；如果已经包含了所有行，则为 -1。
LastCol	复制中想要包括的最后一列；如果包含了所有列，则为 -1。
IncludeColHeadings	这个参数值非零时第一行中包含列标题，为零则表示不包含列标题。
Filename	包含要写入导出内容的文件名的字符串。
DateTimeFormat	用于日期和时间值的格式，可任选。有效值是 0（ISO 格式）、1（USA 格式）、2（EUR 格式）、3（JIS 格式）或 4（Windows 控制面板格式）。缺省值是 4。

注: 结果集中第一行的值为 0，最后一行的值比总行数小一。结果集中第一列的值为 0，最后一列的值比总列数小一。

名称	描述
Format	指定输出格式。

值	含义
0 (RSEF_TEXT)	以纯文本格式写入输出文件。
1 (RSEF_HTML)	以 HTML 格式写入输出文件，并且数据组织到一份 HTML 表中。
2 (RSEF_CSV)	以 CSV（以逗号分隔值）格式写入输出文件。
3 (RSEF_PCIXF)	以 PC/IXF 格式写入输出文件。
4 (RSEF_S370IXF)	以 System/370 IXF 格式写入输出文件。

名称	描述

字符串定界符	指定字符串定界符。如果 *Format* 指定为 RSEF_HTML，此参数将被忽略。

值	含义
0 (RSSD_NONE)	不使用字符串定界符。
1 (RSSD_SINGLEQUOTE)	字符串以单引号 (') 来定界。
2 (RSSD_DOUBLEQUOTE)	字符串以双引号 (″) 来定界。

名称	描述
列定界符	指定列定界符。如果 *Format* 指定为 RSEF_HTML，此参数将被忽略。

值	含义
0 (RSCD_SPACE)	列以空格( )定界。
1 (RSCD_TAB)	列以制表符 (\t) 来定界。
2 (RSCD_COMMA)	列以逗号(,)定界。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。如果结果集为空或在数据库中没有检索到任何行，将返回非零值，除非 *FirstRow*=0 并且 *LastRow*=1。如果发生这种情况，函数将返回 0，并写一个空文件。

相关主题

CopyToClipboard()

ExportForm()

short ExportForm(BSTR *OwnerAndName*, BSTR *FileName*)

描述

此函数将指定 QMF 表单导出到指定文件中。

参数

名称	描述
OwnerAndName	包含欲导出表单的拥有者和名称的字符串，所有者和名称之间用句点隔开。例如， `John.Query2`
FileName	包含要写入所导出表单的文件名的字符串。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

PrintReport()

ExportReport()

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

描述

此函数为指定查询创建报表并将它写入文件。可在 QMF 表单中指定该报表的格式和布局。输出文件是个 ASCII 文本文件，它的每一行用一对回车符和换行字符隔开，页之间用换页字符隔开。最好以等距字体查看输出文件。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Source	想要使用的表单的名称(文件名或 Owner.name)。
OutputFileName	想要写入报表的文件的名称。
PageLength	报表每一页的行数。*PageLength* 为 -1 表示继续输出（不分页，除非报表比 *PageWidth* 宽）。
IncludeDateTime	非零表示日期和时间包括在每一页的底部。为零则表示不包括日期和时间。
IncludePageNumbers	非零时表示每页底部显示页号。为零则表示不显示页号。
Format	任选的，指定导出报表的格式。如果是零，则是明文，指定应当精确地输出表单所产生的(文本或 HTML，取决于表单的类型)。如果非零，则格式是 HTML，指定输出应当是 HTML。对于非 HTML 的表单，通过在输出的开头和结尾添加 HTML 标志，输出被转换为 HTML。缺省值是零。
DateTimeFormat	用于日期和时间值的格式，可任选。有效值是 0（ISO 格式）、1（USA 格式）、2（EUR 格式）、3（JIS 格式）或 4（Windows 控制面板格式）。缺省值是 4。
Format	输出文件的格式。

UseFormPageSetup	任选的，如果非零，则*PageLength*、*PageWidth*、*IncludeDateTime* 和 *IncludePageNumbers* 参数应当被忽略，并且它们的值改为从按指定格式保存的值中获得。缺省值是零。

值	含义
0 (RSF_DEFAULT)	使用缺省表单。*FormName* 应当是个空串。
1 (RSF_DATABASE)	使用来自数据库的表单。在 *FormName* 参数中指定表单拥有者和名称 (Owner.Name)。要使用位于其它数据库服务器上的表单，请先使用 ExportForm() 将表单导出为文件，然后指定 RSF_FILE 的 *SourceType*。
2 (RSF_FILE)	使用包含在文件中的表单。指定 *FormName* 参数中的文件名。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

ExportForm()

FastSaveData()

short FastSaveData(long *QueryID*, BOOL *Replace*, BSTR *Tablename*, BSTR *TableSpaceName*, [VARIANT *Comment*])

描述

此函数为指定查询创建报表并将它写入文件。可在 QMF 表单中指定该报表的格式和布局。输出文件是个 ASCII 文本文件，它的每一行用一对回车符和换行字符隔开，页之间用换页字符隔开。最好以等距字体查看输出文件。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Replace	如果想要用指定数据来代替表中的原有数据，则将这个参数取为非零值。如果希望将指定数据附加到表中现有数据的后面，则将这个参数值取为零。
TableName	要用来存放数据的表的名称。如果此表不存在，QMF for Windows 将创建它。

TableSpaceName	表位于或创建于的表空间名称。如果 *TableSpaceName* 为 NULL 或空字符串， QMF for Windows 将使用缺省表空间。如果已经将 QMF for Windows 配置为使用缺省表空间，则此参数将被忽略。请参阅 GetResourceLimit() 描述中的 RSR_SDDIFFERENTTS。
Comment	任选的字符串，它为保存数据的表指定注释。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

FetchNextRow()

short FetchNextRow(long *QueryID,* VARIANT* *Row*)

描述

此函数从数据库中读取数据的下一行。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Row	指向存放结果的 VARIANT 的指针。结果是一数组（变量类型为 VT_ARRAY | VT_BSTR），包含了结果中每个列的值。可调用 GetColumnCount() 来确定数组中值的数量。 每个值都是以它的的本机数据类型或最接近的变体数据类型来指定的。所支持的返回类型有: 字符串（变体类型 VT_BSTR）、浮点（变体类型 VT_R4）、双精度浮点型（变体类型 VT_R8）、短整型（变体类型 VT_I2）、长整型（变体类型 VT_I4）和二进制（变体类型 VT_UI1 | VT_ARRAY）。 当到达结果集的末尾（没有其它行可读取）或结果集为空时，则结果为空（变体类型 VT_EMPTY），而不是数组。 调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。

注: 由于 Microsoft Excel 7.0 和 Microsoft Access 7.0（可能还有在应用程序中使用 Visual Basic 的其它 32 位 Microsoft 产品）中的程序错误，从 QMF for Windows 接收到的 Variant 变量中的字符串数据可能没有从 Unicode（OLE 所采用的）转换成 ANSI（VBA 所采用的）。发出这种情况时，只有字符串的第一个字符被显示出来。为解决这个问题，可在调用使用该变量的 QMF for Windows 函数前将变量设置为空字符串。

返回值

如果成功则为零，否则为非零值。到达结果集的末尾时，返回值为 -1。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

FetchNextRows()

FetchNextRowEx()

short FetchNextRowEx(long *QueryID*)

描述

此函数从数据库中读取数据的下一行。可在不支持 VARIANT 数组的环境中（如 Microsoft Access 2.0）使用此函数。将此函数与 GetColumnValue() 一起使用，检索当前行中每一列的数据。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

返回值

如果成功则为零，否则为非零值。到达结果集的末尾时，返回值为 -1。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

FetchNextRowsEx()

FetchNextRows()

short FetchNextRows(long *QueryID*, VARIANT* *Rows*, long* *NumRows*)

描述

此函数从数据库中读取数据的下一 *NumRows*。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

Row	指向存放结果的 VARIANT 的指针。结果是二维数组（变量类型为 VT_ARRAY | VT_VARIANT），包含了每行中每个列的一个值。可调用 GetColumnCount() 来确定数组中的列数。尽管结果集中未读取的行数小于 *NumRows*，数组的维数还是为 [*NumRows*][*ColumnCount*]（在此情况下，数组将包含额外的、未使用的条目）。 每个值都是以它的的本机数据类型或最接近的变体数据类型来指定的。所支持的返回类型有：字符串（变体类型 VT_BSTR）、浮点（变体类型 VT_R4）、双精度浮点型（变体类型 VT_R8）、短整型（变体类型 VT_I2）、长整型（变体类型 VT_I4）和二进制（变体类型 VT_UI1 | VT_ARRAY）。 当到达结果集的末尾（没有其它行可读取）或结果集为空时，则结果为空（变体类型 VT_EMPTY），而不是数组。 调用这个函数之前，必须正确地初始化 VARIANT。Visual Basic 能够自动地完成这个操作。Visual C++ 程序员则必须调用 VariantInit()。
NumRows	指向包含欲读取行数的长整型数的指针。如果结果集中未读取的行数小于 *NumRows*，则 *NumRows* 将被重新设置为结果中所包含的实际行数。

注： 由于 Microsoft Excel 7.0 和 Microsoft Access 7.0（可能还有在应用程序中使用 Visual Basic 的其它 32 位 Microsoft 产品）中的程序错误，从 QMF for Windows 接收到的 Variant 变量中的字符串数据可能没有从 Unicode（OLE 所采用的）转换成 ANSI（VBA 所采用的）。发出这种情况时，只有字符串的第一个字符被显示出来。为解决这个问题，可在调用使用该变量的 QMF for Windows 函数前将变量设置为一个空字符串。

返回值

如果成功则为零，否则为非零值。到达结果集的末尾时，返回值为 -1。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

FetchNextRow()

FetchNextRowsEx()

short FetchNextRowsEx(long *QueryID*, long* *NumRows*)

描述

此函数从数据库中读取数据的下一 *NumRows*。可在不支持 VARIANT 数组的环境中（如 Microsoft Access 2.0）使用此函数。将此函数与 GetColumnValueEx() 一起使用以检索给定行中每一列的数据。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
NumRows	指向包含欲读取行数的长整型数的指针。如果结果集中未读取的行数小于 *NumRows*，则 *NumRows* 将被重新设置为结果中所包含的实际行数。

返回值

如果成功则为零，否则为非零值。到达结果集的末尾时，返回值为 -1。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

FetchNextRowEx()

FlushQMFCache()

void FlushQMFCache()

描述

此函数告诉 QMF for Windows 刷新它的高速缓冲存储器中的 QMF 信息，舍弃它的内容。下一次 QMF for Windows 需要 QMF 信息时，将从数据库中获取。通常，QMF for Windows 高速存取从数据库中得到的 QMF 信息，以减少数据库通信量并提高系统性能。调用 GetQMFObjectInfo()、GetQMFQueryText() 或 GetQMFObjectList() 之前先调用此函数，以确保返回的信息是最新的。

返回值

无。

GetColumnCount()

long GetColumnCount(long *QueryID*)

描述

此函数返回结果集中的列数。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

返回值

如果成功，则为每一行中的列数。如果不成功，则为 0 或 -1。如果返回值是 0 或 -1，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

GetColumnDataValue()

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

描述

此函数返回 *Index* 中为当前数据行指定的列的数据值。调用此函数后，可以询问返回值的 *Value* 属性。将 FetchNextRowEx() 与此函数一起使用以访问在单独数据行中的数据。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Index	想要检索的数据行的索引，它以零为基础。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

GetColumnHeader()

BSTR GetColumnHeader(long *QueryID*, long *Index*, short* *Result*)

描述

此函数返回与索引 *Index* 相关的列标题（列名）。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Index	想要检索的数据行的索引，以零开始。

结果	如果成功则为零，否则为非零值。如果结果不是零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取额外的出错信息。

注: 列标题对静态 SQL 语句无效。对于从 InitializeStaticQuery() 返回的查询标识，GetColumnHeader 返回格式为 “Coln” 的字符串，其中 “n” 表示列号。

返回值

所返回的字符串代表了 *Index* 参数中指定的列名。

GetColumnHeaderEx()

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

描述

此函数返回与索引 *Index* 相关的列标题（列名）。调用此函数后，可以询问返回值的 *Value* 属性。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Index	想要检索的数据行的索引，它以零为基础。

注: 列标题对静态 SQL 语句无效。对于从 InitializeStaticQuery() 返回的查询标识，GetColumnHeader 返回格式为 “Coln” 的字符串，其中 “n” 表示列号。

返回值

如果成功则为零，否则为非零值。如果返回值为零，则查询代表列名的字符串的 *Value* 属性。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

GetColumnHeadings()

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

描述

此函数返回列标题（也称作列名）。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

Headings	指向存放结果的 VARIANT 的指针。结果是一字符串数组（变量类型为 VT_ARRAY | VT_BSTR），包含了每个列标题的字符串。 调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。

注: 由于 Microsoft Excel 7.0 和 Microsoft Access 7.0（可能还有在应用程序中使用 Visual Basic 的其它 32 位 Microsoft 产品）中的程序错误，从 QMF for Windows 接收到的 Variant 变量中的字符串数据可能没有从 Unicode（OLE 所采用的）转换成 ANSI（VBA 所采用的）。发出这种情况时，只有字符串的第一个字符被显示出来。为解决此问题，可在调用使用该变量的 QMF for Windows 函数前将变量设置为空字符串。

注: 列标题对静态 SQL 语句无效。对于从 InitializeStaticQuery() 返回的查询标识，GetColumnHeadings 返回字符串 “Col1”、“Col2”等。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

GetColumnValue()

short GetColumnValue(long *QueryID*, long *Index*, VARIANT* *Value*)

描述

此函数返回*Index* 中为当前数据行指定的列的数据值。将 FetchNextRowEx() 与此函数一起使用以访问在单独数据行中的数据。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Index	想要检索的数据行的索引，它以零为基础。
Value	指向要用来存放结果的 VARIANT 的指针。结果是基于变体类型的数据值。 调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

GetColumnValueEx()

short GetColumnValueEx(long *QueryID*, long *RowIndex*, long *ColIndex*, VARIANT* *Value*)

描述

此函数返回 *RowIndex* 指定的数据行 *ColIndex* 指定的列的数据值。可以和 FetchNextRowsEx() 一起使用这个函数来访问单个数据行中的数据。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
RowIndex	想要检索的行的索引，它以零为基础。
ColIndex	想要搜索的列的索引，它以零为基础。
Value	指向要用来存放结果的 VARIANT 的指针。可查询结果变体来找出数据类型，以便作进一步处理。 调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

GetDefaultServerName()

BSTR GetDefaultServerName()

描述

此函数返回包含缺省服务器名称的字符串。

返回值

指定缺省服务器名称的字符串。

GetGlobalVariable()

BSTR GetGlobalVariable(BSTR *Name*)

描述
此功能检索指定全局变量的值。

参数

名称	描述
Name	包含了想要设置的变量名的字符串。

返回值
字符串，包含全局变量的值；如果该变量没有值和发生错误，则为 NULL。

GetHostVariableNames()

short GetHostVariableNames(long *QueryID*, VARIANT* *Names*)

描述
此函数返回在指定查询中引用的所有主变量名的数组。此查询必须是引用主变量（或由 QMF 查询存储，或由AddHostVariable() 创建）的静态查询。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Names	指向要用来存放结果数组的 VARIANT 的指针。

返回值
如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 获取更多出错信息。

GetHostVariableTypes()

short GetHostVariableTypes(long *QueryID*, VARIANT* *Types*)

描述
此函数返回在指定查询中引用的所有主变量的数据类型的数组。此查询必须是引用主变量（或由 QMF 查询存储，或由 AddHostVarialble() 创建）的静态查询。请参阅 AddHostVariable()，获取可返回的数据类型的列表。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Types	指向要用来存放结果数组的 VARIANT 的指针。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 获取更多出错信息。

GetLastErrorString()

BSTR GetLastErrorString()

描述

此函数返回包含了最近发生的出错信息的字符串。如果您在某个函数成功地执行后（没有出错）调用此函数，则此函数将返回前一函数调用中发生的最后一个错误。为避免混淆，通常总是在函数调用返回错误后立即调用此函数。

返回值

包含出错信息的字符串。如果自创建 QMF API 对象以来没有发生过错误，则返回 NULL。

相关主题

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

GetLastErrorType()

short GetLastErrorType()

描述

此函数返回最近发生的错误的类型。如果您在某个函数成功地执行后（没有出错）调用此函数，则此函数将返回前一函数调用中发生的最后一个错误。为避免混淆，通常总是在函数调用返回错误后立即调用此函数。

返回值

返回值表明错误的类型:

值	含义
0 (RS_ERROR_NONE)	自 QMF for Windows API 对象创建以来没有出现过错误。
1 (RS_ERROR_SQL)	发生 SQL 错误。如果错误是在调用将 *QueryID* 用作参数的函数时发生的，那么就调用 *Close()* 来关闭该查询。不执行回滚。您可以继续使用 QMF for Windows API 对象，但可能还会遇到其它错误。

2 (RS_ERROR_USER_CANCEL)	用户取消了该操作，通常是通过在忙窗口单击“取消”来取消的。这将导致 QMF for Windows 执行隐式的回滚操作（使所有显式查询标识无效），并毁坏与数据库的连接。如果想要继续，则必须调用 InitializeServer() 或 ReinitializeServer()。
3 (RS_ERROR_FATAL_GOV)	发生了致命的管理错误。一种可能性是 QMF for Windows API 超时，因为已经超出所允许的最大空闲时间。这将导致 QMF for Windows 执行隐式的回滚操作（使所有显式查询标识无效），并毁坏与数据库的连接。如果想要继续，则必须调用 InitializeServer() 或 ReinitializeServer()。
4 (RS_ERROR_NONFATAL_GOV)	发生了不致命的管理错误。可能是超出了允许读取的最大行数，或者 SQL 动词是不允许的。如果错误是在调用以 *QueryID* 作为参数的函数时发生的，那么就调用 Close() 来关闭该查询。这样将不重新执行该函数，与数据库之间的连接也不受影响，因此您可以继续使用 QMF for Windows API 对象。
5 (RS_ERROR_OTHER)	发生了一般错误。不执行回滚。您可以继续使用 QMF for Windows API 对象，但可能还会遇到其它错误。

相关主题

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

GetLastSQLCode()

long GetLastSQLCode()

描述

此函数返回最近发生的错误的 SQL 代码。如果您在某个函数成功地执行后（没有出错）调用此函数，则此函数将返回前一函数调用中发生的最后一个错误。为避免混淆，通常总是在函数调用返回错误后立即调用这个函数。

返回值

最近发生的错误的 SQL 代码。如果自创建 QMF for Windows API 对象以来没有出现过错误，或者最近发生的错误不是 SQL 错误，则返回零。

相关主题

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

GetLastSQLError()

VARIANT GetLastSQLError()

描述

此函数返回最近发生的错误的详细 SQL 出错信息。如果您在某个函数成功地执行后（没有出错）调用此函数，则此函数将返回前一函数调用中发生的最后一个错误。为避免混淆，通常总是在函数调用返回错误后立即调用这个函数。

返回值

包含了出错信息的数组（变体类型 VT_ARRAY | VT_VARIANT）。如果自创建 QMF for Windows API 对象以来没有出现过错误，或者最近发生的错误不是 SQL 错误，则返回空（变体类型 VT_EMPTY）。数组有以下格式:

元素	类型	目录
0	长整型 (VT_I4)	Code
1	字符串 (VT_BSTR)	State
2	字符串 (VT_BSTR)	ErrProc
3	字符串 (VT_BSTR)	RDBName
4	长整型 (VT_I4)	ErrD1
5	长整型 (VT_I4)	ErrD2
6	长整型 (VT_I4)	ErrD3
7	长整型 (VT_I4)	ErrD4
8	长整型 (VT_I4)	ErrD5
9	长整型 (VT_I4)	ErrD6
10	字符串 (VT_BSTR)	Warn0
11	字符串 (VT_BSTR)	Warn1
12	字符串 (VT_BSTR)	Warn2
13	字符串 (VT_BSTR)	Warn3
14	字符串 (VT_BSTR)	Warn4
15	字符串 (VT_BSTR)	Warn5
16	字符串 (VT_BSTR)	Warn6

17	字符串 (VT_BSTR)	Warn7
18	字符串 (VT_BSTR)	Warn8
19	字符串 (VT_BSTR)	Warn9
20	字符串 (VT_BSTR)	WarnA
21	字符串 (VT_BSTR)	MessageTokens

相关主题

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

GetLastSQLState()

BSTR GetLastSQLState()

描述

此函数返回最近发生的错误的 SQL 状态。如果您在某个函数成功地执行后（没有出错）调用此函数，则此函数将返回前一函数调用中发生的最后一个错误。为避免混淆，通常总是在函数调用返回错误后立即调用这个函数。

返回值

包含最近所发生错误的 SQL 代码的字符串。如果自创建 QMF for Windows API 对象以来没有出现过错误，或者最近发生的错误不是 SQL 错误，则返回 NULL。

相关主题

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

GetOption()

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

描述

获取 QMF for Windows 中的指定选项值。

参数

名称	描述
Option	指定要检索的选项。

值	含义
0 (RSO_SERVER_DEFINITION_FILE)	服务器定义文件名。
1 (RSO_CPIC_DLL)	CPI-C 提供者 DLL 文件名。
2 (RSO_CPIC_TIMEOUT_WARNING)	CPI-C 警告超时（以秒为单位）。QMF for Windows API 不使用此限制。
3 (RSO_CPIC_TIMEOUT_CANCEL)	CPI-C 取消超时（以秒为单位）。
4 (RSO_TCP_TIMEOUT_WARNING)	TCP 警告超时（以秒为单位）。QMF for Windows API 不使用此限制。
5 (RSO_TCP_TIMEOUT_CANCEL)	TCP 警告超时（以秒为单位）。
6 (RSO_DISPLAY_NULLS_STRING)	用来显示空值的字符串。
7 (RSO_ENTER_NULLS_STRING)	用来输入空值的字符串。
8 (RSO_ENTER_DEFAULTS_STRING)	用来输入缺省值的字符串。
9 (RSO_TRACE_FILE_1)	跟踪文件 1 名称。
10 (RSO_TRACE_FILE_2)	跟踪文件 2 名称。
11 (RSO_TCP_TRACE_LEVEL)	TCP 跟踪级。
12 (RSO_CPIC_TRACE_LEVEL)	CPI-C 跟踪级。
13 (RSO_DDM_TRACE_LEVEL)	DDM 跟踪级。

Value	指向存放结果的 VARIANT 的指针。结果是一数组（变量类型为 VT_ARRAY | VT_BSTR），包含了结果中每个列的值。可调用 GetColumnCount() 来确定数组中值的数量。调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。

注: 由于 Microsoft Excel 7.0 和 Microsoft Access 7.0（可能还有在应用程序中使用 Visual Basic 的其它 32 位 Microsoft 产品）中的程序错误，从 QMF for Windows 接收到的 Variant 变量中的字符串数据可能没有从 Unicode（OLE 所采用的）转换成 ANSI（VBA 所采用的）。发出这种情况时，只有字符串的第一个字符被显示出来。为解决这个问题，可在调用使用该变量的 QMF for Windows 函数前将变量设置为空字符串。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

相关主题

SetOption()

GetOptionEx()

short GetOptionEx(short *Option*)

描述

获取 QMF for Windows 中的指定选项值。当选项值返回时，必须查询 *Option* 属性来获取该数据。

参数

名称	描述
Option	这些选项值与 GetOption() 调用的选项值相同。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

相关主题

GetOption()
SetOption()

GetProcText()

BSTR GetProcText(long *ProcID*)

描述

此函数返回的文本，在变量替换后，为指定的过程而被执行。调用此函数之前，您应当用 SetProcVariable() 设置在该过程中使用的所有变量的值。

参数

名称	描述
ProcID	过程的标识，由 InitializeProc() 返回。

返回值

如果成功，则是包含了所返回过程文本的字符串。如果不成功，则返回 NULL。如果返回值为 NULL，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

GetProcVariables()

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

描述

获取 QMF for Windows 中的指定选项值。

参数

名称	描述
ProcID	过程的标识，由 InitializeProc() 返回。
Value	指向存放结果的 VARIANT 的指针。结果是一字符串数组（变体类型 VT_ARRAY | VT_BSTR），每个字符串都包含了一个变量的名称。如果过程中没有变量，则结果为空（变体类型 VT_EMPTY）。调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。

注: 由于 Microsoft Excel 7.0 和 Microsoft Access 7.0（可能还有在应用程序中使用 Visual Basic 的其它 32 位 Microsoft 产品）中的程序错误，从 QMF for Windows 接收到的 Variant 变量中的字符串数据可能没有从 Unicode（OLE 所采用的）转换成 ANSI（VBA 所采用的）。发出这种情况时，只有字符串的第一个字符被显示出来。为解决这个问题，可在调用使用该变量的 QMF for Windows 函数前将变量设置为空字符串。

返回值

如果成功则为零，否则为非零值。如果在过程中没有变量，则返回值为 RS_NO_ERROR_NO_DATA (-1)。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

GetQMFObjectInfo()

short GetQMFObjectInfo(BSTR *OwnerAndName*, short *Type*, short *Time*, VARIANT* *Value*)

描述

此函数返回有关 QMF 对象（表单或查询）的信息。返回的信息由 *Type* 和 *Time* 参数指定。

参数

名称	描述
OwnerAndName	字符串，此字符串中包含了欲检索其信息的对象的拥有者和名称，所有者和名称之间用句点隔开 。例如， `John.Query2`

Value	指向存放结果的 VARIANT 的指针。对于 RSI_TIMEUSED、RSI_TIMESRUN、RSI_TIMESCANCELLED 和 RSI_LEVEL，结果是短整型（变体类型 VT_I2）。对于 RSI_RESTRICTED，此结果是布尔型（变体类型 VT_BOOL）。对于其它，结果则是字符串（变体类型 VT_BSTR）。调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。

注: 由于 Microsoft Excel 7.0 和 Microsoft Access 7.0（可能还有在应用程序中使用 Visual Basic 的其它 32 位 Microsoft 产品）中的程序错误，从 QMF for Windows 接收到的 Variant 变量中的字符串数据可能没有从 Unicode（OLE 所采用的）转换成 ANSI（VBA 所采用的）。发出这种情况时，只有字符串的第一个字符被显示出来。为解决这个问题，可在调用使用该变量的 QMF for Windows 函数前将变量设置为空字符串。

Type	指定要获取信息的类型。

值	含义
0 (RSI_COMMENT)	注释
1 (RSI_LEVEL)	级别
2 (RSI_TYPE)	类型
3 (RSI_SUBTYPE)	子类型
4 (RSI_RESTRICTED)	限制
5 (RSI_MODEL)	型号。
6 (RSI_TIMESUSED)	使用次数。
7 (RSI_TIMESRUN)	运行次数。
8 (RSI_TIMESCANCELLED)	取消的次数。
9 (RSI_DATE)	首次使用、最后使用、最后修改的日期。
10 (RSI_TIME)	首次使用、最后使用、最后修改的时间。
11 (RSI_USERID)	首次使用、最后使用、最后修改的用户标识。
12 (RSI_SQLID)	首次使用、最后使用、最后修改的 SQL 标识。
13 (RSI_ENVIRONMENT)	首次使用、最后使用、最后修改的环境。
14 (RSI_MODE)	首次使用、最后使用、最后修改的方式。
15 (RSI_COMMAND)	首次使用、最后使用、最后修改的命令。

Time	指定首次使用、最后使用或最后修改的时间。

值	含义
0 (RST_FIRSTUSED)	首次使用。

1 (RST_LASTUSED)	最后使用。
2 (RST_LASTMODIFIED)	最后修改。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

GetQMFObjectInfoEx()

short GetQMFObjectInfoEx(BSTR *OwnerAndName*, short *Type*, short *Time*)

描述

此函数返回有关 QMF 对象的信息。返回的信息由 *Type* 和 *Time* 参数指定。调用此函数后，可以询问返回值的 *QMFObjectInfo* 属性。

参数

名称	描述
OwnerAndName	字符串，此字符串中包含了欲搜索其信息的对象的拥有者和名称，所有者和名称之间用句点隔开 。例如, `John.Query2`
Type	指定要获取信息的类型。

值	含义
0 (RSI_COMMENT)	注释
1 (RSI_LEVEL)	级别
2 (RSI_TYPE)	类型
3 (RSI_SUBTYPE)	子类型
4 (RSI_RESTRICTED)	限制
5 (RSI_MODEL)	型号。
6 (RSI_TIMESUSED)	使用次数。
7 (RSI_TIMESRUN)	运行次数。
8 (RSI_TIMESCANCELLED)	取消的次数。
9 (RSI_DATE)	首次使用、最后使用、最后修改的日期。
10 (RSI_TIME)	首次使用、最后使用、最后修改的时间。
11 (RSI_USERID)	首次使用、最后使用、最后修改的用户标识。
12 (RSI_SQLID)	首次使用、最后使用、最后修改的 SQL 标识。
13 (RSI_ENVIRONMENT)	首次使用、最后使用、最后修改的环境。

14 (RSI_MODE)	首次使用、最后使用、最后修改的方式。
15 (RSI_COMMAND)	首次使用、最后使用、最后修改的命令。

Time	指定首次使用、最后使用或最后修改的时间。

值	含义
0 (RST_FIRSTUSED)	首次使用。
1 (RST_LASTUSED)	最后使用。
2 (RST_LASTMODIFIED)	最后修改。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

GetQMFObjectList()

short GetQMFObjectList(BSTR *Owner*, BSTR *Name*, short *Type*, VARIANT* *List*)

描述

此函数返回与 *Owner* 和 *Name* 参数所指定模式匹配的所有 QMF 对象的名称的数组。

参数

名称	描述
Owner	包含您希望放入返回列表中的对象拥有者的字符串。
Name	包含您希望放入返回列表中的对象名称的字符串。
List	指向存放结果的 VARIANT 的指针。结果是一字符串数组（变体类型 VT_ARRAY | VT_BSTR），每个字符串格式为 Owner.Name。如果找不到与 QMF for Windows 查询匹配的结果，结果将为空（变体类型 VT_EMPTY）。调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。

注: 由于 Microsoft Excel 7.0 和 Microsoft Access 7.0（可能还有在应用程序中使用 Visual Basic 的其它 32 位 Microsoft 产品）中的程序错误，从 QMF for Windows 接收到的 Variant 变量中的字符串数据可能没有从 Unicode（OLE 所采用的）转换成 ANSI（VBA 所采用的）。发出这种情况时，只有字符串的第一个字符被显示出来。为解决这个问题，可在调用使用该变量的 QMF for Windows 函数前将变量设置为空字符串。

Type	指定想要包含到列表中的 QMF 对象的类型。这些值可以加到一起，以指定多个对象类型。

值	含义
2048 (RSQ_MASK_QUERIES)	在列表中包含 QMF 查询。
1024 (RSQ_MASK_FORMS)	在列表中包含 QMF 表单。
512 (RSQ_MASK_PROCS)	在列表中包含 QMF 过程。
256 (RSQ_MASK_TABLES)	在列表中包含表。

返回值

如果成功则为零，否则为非零值。如果没有找到匹配的 QMF 对象，则返回值为零。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

GetQMFObjectListEx()

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

描述

此函数返回与 *Owner* 和 *Owner* 参数所指定模式匹配的 QMF 对象的名称，这两个参数由 *Index* 参数引用。调用此函数后，可以询问返回值的 *Value* 属性。

参数

名称	描述
Owner	包含您希望放入返回列表中的对象拥有者的字符串。
Name	包含您希望放入返回列表中的对象名称的字符串。
Index	与模式匹配的 QMF 对象的列表的索引。

Type	指定想要包含到列表中的 QMF 对象的类型。这些值可以加到一起，以指定多个对象类型。

值	含义
2048 (RSQ_MASK_QUERIES)	在列表中包含 QMF 查询。
1024 (RSQ_MASK_FORMS)	在列表中包含 QMF 表单。
512 (RSQ_MASK_PROCS)	在列表中包含 QMF 过程。
256 (RSQ_MASK_TABLES)	在列表中包含表。

返回值

如果成功则为零，否则为非零值。如果没有找到匹配的 QMF 对象，则返回值为 RS_ERROR_OUTOFRANGE。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

GetQMFProcText()

BSTR GetQMFProcText(BSTR *OwnerAndName*)

描述

此函数返回的文本，在变量替换后，为指定的过程而被执行。调用此函数之前，您应当用 SetProcVariable() 设置在该过程中使用的所有变量的值。

参数

名称	描述
OwnerAndName	包含欲删除对象的拥有者和名称的字符串，所有者和名称之间用句点隔开。例如， `John.Proc2`

返回值

包含所检索过程的文本的字符串；如果没有检索到该过程，则返回 NULL。如果返回值为 NULL，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError 或 GetLastSQLState() 获取更多出错信息。

GetQMFQueryText()

BSTR GetQMFQueryText(BSTR *OwnerAndName*)

描述

此函数检索存储在指定查询中的 SQL 文本。

参数

名称	描述
OwnerAndName	包含欲删除对象的拥有者和名称的字符串，所有者和名称之间用句点隔开。例如， `John.Query2`

返回值

包含所检索查询的文本的字符串；如果没有检索到该查询，则返回 NULL。如果返回值为 NULL，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError 或 GetLastSQLState() 获取更多出错信息。

GetQueryText()

BSTR GetQueryText(long *QueryID*)

描述

变量替换之后，此函数返回 SQL 文本，此文本是指定的查询执行的结果。调用此函数之前，应当用 SetVariable() 设置在该查询中使用的所有变量的值。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

注: 此查询文本对静态 SQL 语句无效。对于 InitializeStaticQuery() 返回的查询标识，GetQueryText() 将返回空的字符串。

返回值

如果成功，则是包含了所返回 SQL 文本的字符串。如果不成功，则返回 NULL。如果返回值为 NULL，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

GetQueryVerb()

BSTR GetQueryVerb(long *QueryID*)

描述

此函数返回包含查询中所用 SQL 动词的字符串。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

注: 查询动词对静态 SQL 语句无效。对于 InitializeStaticQuery() 返回的查询标识，GetQueryVerb() 将返回空字符串。

返回值

如果成功，将返回包含 SQL 动词的字符串。如果不成功，则返回 NULL。如果返回值为 NULL，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

GetResourceLimit()

short GetResourceLimit(short *Resource*, long* *Value*)

描述

此函数获取所请求的资源限制。调用这个函数之前必须先调用 InitializeServer()，因为资源限制是在每个服务器的基础上进行的。

参数

名称	描述
Resource	资源值包括:

值	含义
0 (RSR_IDLE_CONNECTION_TIMEOUT)	空闲连接超时（以秒为单位）。
1 (RSR_IDLE_QUERY_TIMEOUT_CANCEL)	空闲查询超时（以秒为单位）。
2 (RSR_IDLE_QUERY_TIMEOUT_WARNING)	空闲查询超时（以秒为单位）。这是警告限制；它对 QMF for Windows API 不起作用。
3 (RSR_SERVER_RESPONSE_TIMEOUT_CANCEL)	服务器超时（以秒为单位）。
4 (RSR_SERVER_RESPONSE_TIMEOUT_WARNING)	服务器超时（以秒为单位）。这是警告限制；它对 QMF for Windows API 不起作用。
5 (RSR_MAX_ROWS_TO_FETCH_CANCEL)	可读取的最大行数。
6 (RSR_MAX_ROWS_TO_FETCH_WARNING)	可读取的最大行数。这是警告限制；它对 QMF for Windows API 不起作用。
7 (RSR_MAX_BYTES_TO_FETCH_CANCEL)	可读取的最大字节数。
8 (RSR_MAX_BYTES_TO_FETCH_WARNING)	可读取的最大字节数。这是警告限制；它对 QMF for Windows API 不起作用。

9 (RSR_MAX_CONNECTIONS)	所允许的连接到数据库服务器的最大数目。
10 (RSR_ALLOW_SERVER_ACCESS_UI)	允许来自 QMF for Windows 接口的对数据库服务器的访问吗？
11 (RSR_ALLOW_SERVER_ACCESS_API)	允许从 QMF for Windows API 访问数据库服务器吗?
12 (RSR_FETCH_ALL_ROWS)	自动读取所有行吗?
13 (RSR_CONFIRM_UPDATES)	确认数据库服务器更新吗? 此选项对 QMF for Windows API 没有任何作用; QMF for Windows API 并不对数据库的更新进行确认。
14 (RSR_SUMMARY_TRACKING)	启用汇总对象跟踪吗?
15 (RSR_DETAILED_TRACKING)	启用详细对象跟踪吗?
16 (RSR_SQL_TRACKING)	启用 SQL 文本跟踪吗?
17 (RSR_ADHOC_TRACKING)	启用特设对象跟踪吗?
18 (RSR_ALLOW_ACQUIRE)	允许 SQL 动词 ACQUIRE 吗?
19 (RSR_ALLOW_ALTER)	允许 SQL 动词 ALTER 吗?
20 (RSR_ALLOW_COMMENT)	允许 SQL 动词 COMMENT 吗?
21 (RSR_ALLOW_CREATE)	允许 SQL 动词 CREATE 吗?
22 (RSR_ALLOW_DELETE)	允许 SQL 动词 DELETE 吗?
23 (RSR_ALLOW_DROP)	允许 SQL 动词 DROP 吗?
24 (RSR_ALLOW_EXPLAIN)	允许 SQL 动词 EXPLAIN 吗?
25 (RSR_ALLOW_GRANT)	允许 SQL 动词 GRANT 吗?
26 (RSR_ALLOW_INSERT)	允许 SQL 动词 INSERT 吗?
27 (RSR_ALLOW_LABEL)	允许 SQL 动词 LABEL 吗?
28 (RSR_ALLOW_LOCK)	允许 SQL 动词 LOCK 吗?
29 (RSR_ALLOW_REVOKE)	允许 SQL 动词 REVOKE 吗?
30 (RSR_ALLOW_SELECT)	允许 SQL 动词 SELECT 吗?
31 (RSR_ALLOW_SET)	允许 SQL 动词 SET 吗?

32 (RSR_ALLOW_SIGNAL)	允许 SQL 动词 SIGNAL 吗?
33 (RSR_ALLOW_UPDATE)	允许 SQL 动词 UPDATE 吗?
34 (RSR_ALLOW_CALL)	允许 SQL 动词 CALL 吗?
35 (RSR_ALLOW_SAVE_DATA)	允许“保存数据”命令吗？
36 (RSR_SAVE_DATA_TABLE_SPACE_NAME)	绑定包的缺省集合名称？
37 (RSR_SAVE_DATA_TABLE_SPACE_NAME_OVERRIDE)	可由用户覆盖“保存数据”命令的缺省表空间名称吗?
38 (RSR_ALLOW_BIND_PACKAGE)	允许绑定包吗?
39 (RSR_DEF_COLLECTION)	绑定包的缺省集合名。
40 (RSR_DEF_COLLECTION_OVERRIDE)	可由用户覆盖绑定包的缺省集合名吗?
41 (RSR_DEF_ISOLATION_LEVEL)	绑定包的缺省隔离级别。
42 (RSR_DEF_ISOLATION_LEVEL_OVERRIDE)	可由用户覆盖绑定包的缺省隔离级别吗？
43 (RSR_ALLOW_TABLE_EDIT)	允许使用表编辑器吗?
44 (RSR_ALLOW_EXPORT)	允许导出数据吗？
45 (RSR_ALLOW_SAVED_QUERIES_ONLY)	允许用户只运行保存的查询吗?
46 (RSR_ALLOW_DROP_PACKAGE)	允许卸下包吗?
47 (RSR_QUERY_ISOLATION_LEVEL)	运行查询时使用的隔离级别。
48 (RSR_ACCOUNT_STRING)	在连接数据库服务器时传送帐户信息的字符串。
49 (RSR_ACCOUNT_OVERRIDE)	可由用户覆盖在连接数据库服务器时传送的帐户信息的字符串吗?

Value	指向保存结果的长整型整数的指针。结果是所请求的资源限制的值。对于布尔型值，结果非零表示真，为零表示假。对于 RSR_SAVE_DATA_TABLE_SPACE_NAME、RSR_DEF_COLLECTION 和 RSR_ACCOUNT_STRING，返回 -1 ，可对返回的字符串值询问 *ResourceLimit* 属性。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

GetResourceLimitEx()

short GetResourceLimitEx(short *Resource*)

描述

此函数获取所请求的资源限制。调用这个函数之前必须先调用 InitializeServer()，因为资源限制是在每个服务器的基础上进行的。调用这个函数后，可查询 *ResourceLimit* 属性来获取结果。

参数

名称	描述
Resource	Resource 的值与 GetResourceLimit() 所调用的相同。

注: 查询动词对静态 SQL 语句无效。对于 InitializeStaticQuery() 返回的查询标识，GetQueryVerb() 返回空字符串。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

GetRowCount()

long GetRowCount(long *QueryID*)

描述

此函数返回当前在 QMF for Windows 内部缓冲区中的行数。此值可能大于 FetchNextRow() 或 FetchNextRows() 检索到的行数，因为 QMF for Windows 缓存了从数据库接收的数据。

此函数返回已经从数据库检索到的行数。如果想要检索结果集中的总行数，可:

- 调用 CompleteQuery()，并用 FetchNextRow() 或 FetchNextRows() 来读取所有行。
- 调用 Open() 时指定 FetchAllRows = TRUE。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

返回值

如果成功则为行数（如果一行都没有检索到，则为 0），不成功则为 -1。如果为 1，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

GetServerList()

short GetServerList(VARIANT* *List*)

描述

此函数返回包含 QMF for Windows 服务器定义文件 (SDF) 中定义的数据库服务器名称的数组。如果希望用 QMF for Windows API 访问数据库服务器，则必须在 SDF 文件中定义此数据库服务器。

参数

名称	描述
列表	指向存放结果的 VARIANT 的指针。结果是一字符串数组（变体类型 VT_ARRAY | VT_BSTR），每个字符串都包含了数据库服务器的名称。如果没有定义任何数据库服务器，则结果为空（变体类型 VT_EMPTY）。调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。

注: 由于 Microsoft Excel 7.0 和 Microsoft Access 7.0（可能还有在应用程序中使用 Visual Basic 的其它 32 位 Microsoft 产品）中的程序错误，从 QMF for Windows 接收到的 Variant 变量中的字符串数据可能没有从 Unicode（OLE 所采用的）转换成 ANSI（VBA 所采用的）。发出这种情况时，只有字符串的第一个字符被显示出来。为解决这个问题，可在调用使用该变量的 QMF for Windows 函数前将变量设置为空字符串。

返回值

如果成功则为零，否则为非零值。如果没有定义任何数据库服务器，则返回值为零。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

GetServerListEx()

short GetServerListEx(short *Index*)

描述

此函数检索被 *Index* 参数引用的服务器名称。调用此函数后，可以询问返回值的 *Value* 属性。

参数

名称	描述
Index	服务器列表的索引。

返回值

如果成功则为零，当索引大于可用服务器个数时 RS_OUTOFRANGE 为真，不成功则为非零。如果没有定义任何数据库服务器，则返回值为 RS_OUTOFRANGE。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

GetStoredProcedureResultSets()

short GetStoredProcedureResultSets(long *QueryID*, VARIANT* *ResultSets*)

描述

此函数检索由用原来的 *QueryID* 执行的存储过程返回的结果集的查询标识。返回的每个查询标识可与 FetchNextRow() 或 FetchNextRows() 一起用于检索结果集中的行，当到达每个结果集结尾时与 Close() 一起使用。

参数

名称	描述
QueryID	原始查询的标识，由 InitializeQuery() 返回。
ResultSets	指向一个 VARIANT，结果集的查询标识就存储在其中。结果是一个长整数数组（变体类型 VT_ARRAY | VT_I4），每个整数都是相应结果集的查询标识。如果存储过程没有返回任何结果集，则结果为空（变体类型 VT_EMPTY）。调用这个函数之前，必须正确地初始化 VARIANT。Visual Basic 能够自动地完成这个操作。Visual C++ 程序员则必须调用 VariantInit()。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

GetVariables()

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

描述

此函数返回查询 SQL 文本中变量名称的数组。使用 Open() 或 Execute() 运行查询之前，您必须通过调用 SetVariable() 对这些变量赋值。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

Variables	指向存放结果的 VARIANT 的指针。结果是一字符串数组（变体类型 VT_ARRAY | VT_BSTR），每个字符串都包含了变量的名称。如果 SQL 语句中没有变量，则结果为空（变体类型 VT_EMPTY）。调用此函数之前，必须正确地初始化 VARIANT。Visual Basic 可自动完成此操作。Visual C++ 程序员则必须调用 VariantInit()。

注: 由于 Microsoft Excel 7.0 和 Microsoft Access 7.0（可能还有在应用程序中使用 Visual Basic 的其它 32 位 Microsoft 产品）中的程序错误，从 QMF for Windows 接收到的 Variant 变量中的字符串数据可能没有从 Unicode（OLE 所采用的）转换成 ANSI（VBA 所采用的）。发出这种情况时，只有字符串的第一个字符被显示出来。为解决这个问题，可在调用使用该变量的 QMF for Windows 函数前将变量设置为空字符串。

返回值

如果成功则为零，否则为非零值。如果 SQL 语句中没有变量，则返回值为 RS_ERROR_NO_DATA(-1)。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多信息。

GetVariablesEx()

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

描述

此函数返回在查询的 SQL 文本中被 *Index* 参数引用的变量名。调用此函数后，可以询问返回值的 *Value* 属性。在使用 Open() 或 Execute() 运行查询之前，必须先通过调用 SetVariable() 为此变量（和其它所有 SQL 文本中的变量）赋值。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Variables	变量的内部列表的索引。可查询 *Value* 属性来获取与所传递的索引相对应的字符串。如果 SQL 语句中没有变量，此函数将返回 RS_ERROR_NO_DATA。

返回值

如果成功则为零，否则为非零值。如果 SQL 语句中没有变量，则返回值为 RS_ERROR_NO_DATA(-1)。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多信息。

InitializeProc()

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

描述

此函数设置要在过程中使用的文本。可将文本作为参数传递到这个函数、从文本文件中读取它、或者从现有过程获取它。

参数

名称	描述
SourceType	指定过程文本的来源。

值	含义
0 (RSS_STRING)	在*Source* 参数中包含的文本。
2 (RSS_FILE)	此文本包含在由 *Source* 参数指定名称的那个正文文件中。
3 (RSS_QMFPROC)	此文本包含在由 *Source* 参数指定拥有者和名称的那个过程中。

Source	字符串，包含文本、过程的拥有者和名称 (Owner.Name)、或者包含过程文本的文件的名称。

返回值

如果成功，则为过程的标识 (ProcID)。如果不成功，则为 -1。在所有要求 *ProcID* 参数的接口调用中都必须使用这个值。

InitializeQuery()

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

描述

此函数设置您想要在查询中使用的文本。可将 SQL 文本作为参数传递到这个函数、从文本文件中读取它、或者从现有查询获取它。完成查询时调用 close()。

参数

名称	描述
SourceType	指定 SQL 语句文本的源。

值	含义
0 (RSS_STRING)	SQL 文本包含在 *Source* 参数中。
1 (RSS_QMFQUERY)	此 SQL 文本包含在由 *Source* 参数指定拥有者和名称的那个查询中。

2 (RSS_FILE)	此 SQL 文本包含在由 *Source* 参数指定名称的那个正文文件中。

返回值

如果成功，则为查询的标识。如果不成功，则为 -1。您必须在所有需要 *Query* 参数的接口调用中使用此值。

InitializeServer()

short InitializeServer(BSTR *ServerName*, BSTR *UserID*, BSTR *Password*, BOOL *ForceDialog*, [VARIANT *Account*], [VARIANT *SuppressDialog*])

描述

此函数初始化与数据库服务器的连接。调用 QMF for Windows API 中的任何其它函数之前必须先调用这个函数。可多次调用这个函数。但是，如果调用此函数但没有调用 Commit() 或 Rollback() 来结束，则将导致隐式回滚。

参数

名称	描述
ServerName	包含您想要使用的数据库服务器名称的字符串。此名称必须与 QMF for Windows 的服务器定义文件中定义的某个名称相匹配。可调用 GetServerList() 检索有效服务器列表。
UserID	包含您希望使用的用户标识的字符串。如果 UserID 为 NULL 或空字符串，则 QMF for Windows 将使用最近一次查询中的 UserID（如果可得到）。否则，QMF for Windows 将显示“用户信息”对话框以获取用户标识和口令。
Password	包含指定用户标识的口令的字符串。如果 Password 为 NULL 或空字符串，QMF for Windows 将使用所保存的口令（如果可得到，并且需要 Windows for Workgroups）。如果没有可用的口令，QMF for Windows 将显示“用户信息”对话框以获取口令。
ForceDialog	非零表示 QMF for Windows 显示“用户信息”对话框，而不管是否指定了 UserID 和 Password。这就给了用户一个机会在使用这些信息前更改它们。零则表示 QMF for Windows 应当只在需要的时侯显示“用户信息”对话框。
Account	任选的字符串，它指定连接时传送给服务器的帐户管理信息。服务器可能在作业帐户管理系统中使用这些信息。
SuppressDialog	非零表示即使用户标识和口令没有指定，QMF for Windows 也不显示“用户信息”对话框。当在没有用户响应“用户信息”对话框的环境中执行操作时，此选项将是非常有用的，例如，在万维网服务器上。

返回值
如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题
SetParent()

InitializeStaticQuery()

long InitializeStaticQuery(BSTR *CollectionName*, BSTR *PackageName*, BSTR *ConsistencyToken*, short *SectionNumber*)

描述
此函数指定要作为静态查询运行的包的节。

参数

名称	描述
CollectionName	先前绑定集合的名称。
PackageName	先前绑定包的名称。
ConsistencyToken	前面命名的集合和包所用的令牌。
SectionNumber	想要允许的集合和包内语句的段号。

返回值
如果成功，则为查询的标识。如果不成功，则为 -1。在所有要求 *QueryID* 参数的接口调用中都必须使用这个值。

IsStatic()

BOOL IsStatic(long *QueryID*)

描述
此函数确定指定的查询标识是否引用静态查询或动态查询。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 或 InitializeStaticQuery() 返回。

返回值
如果成功，则返回非零，*QueryID* 引用静态查询，否则为零。

Open()

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

描述

使用此函数，对使用 SELECT 动词的查询，通过在数据库中打开它的游标来运行它。使用 FetchNextRow() 或 FetchNextRows() 检索查询的数据，然后在完成时调用 Close()。如果 QMF for Windows 配置为自动读取所有行（请参阅 GetResourceLimit() 描述中的 RSR_FETCHALLROWS），或者 FetchAllRows 参数不为零，则 QMF for Windows 在从这个调用返回前将结果集中的所有行读取到它的内部缓冲区。

注: 此函数的名称与 Microsoft Access 2.0 中的关键字 Open 冲突。如果在使用 MS Access 2.0，则应将函数名放在方括号 [ ] 内。

注: 仅在包含 SQL 动词 SELECT 的语句中使用这个函数。对于包含其它动词的语句，例如 SET，则调用 Execute() 取代它。要确定查询使用的动词，请调用 GetQueryVerb()。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
RowLimit	表示从数据库检索的最大行数的数值。若为零，则表示除 QMF for Windows 管理员程序中的限制外，没有其它行数限制。
FetchAllRows	布尔值，指出结果集中的所有行是否被自动读取到 QMF for Windows 内部缓冲区中。如果非零，则自动读取所有行，关闭游标并释放数据库让其它用户使用。这与调用 CompleteQuery() 是相同的。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

Prepare()

short Prepare(long *QueryID*)

描述

此函数准备由 *QueryID* 指定的查询。此语句由数据库服务器检查，看对象是否存在以及所需的授权等。如果查询是 SELECT 语句，则此语句返回的有关列的信息将在 Prepare() 完成之后得到。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

Execute()
Open()

PrintReport()

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

描述

PrintReport() 与 ExportReport() 函数同义。

ReinitializeServer()

short ReinitializeServer()

描述

此函数重新初始化与数据库服务器之间的连接。通常，当其它某个 QMF for Windows API 函数返回错误时，只需要调用这个函数就可以了。调用这个函数将会引起隐式回滚，关闭所有打开的游标并使所有显式的查询标识无效。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

Rollback()

short Rollback()

描述

此函数取消当前工作单元中所作的任何更改、结束当前工作单元、关闭打开的所有游标、并使所有查询标识无效。

注: 此函数的名称与 Microsoft Access 2.0 中的关键字 Rollback 冲突。如果在使用 MS Access 2.0，则应将函数名放在方括号 [ ] 内。

注: 回滚只影响调用 Open() 或 Execute() 所运行的 SQL 更改。回滚不会影响其它 QMF for Windows API 函数（例如 FastSaveData()、SaveData() 或 DeleteQMFObject()）所作的更改。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

Commit()

RunProc()

short RunProc(long *ProcID*)

描述

此函数运行指定的过程。此过程一直运行到完成或出错。您不能通过此程序设计接口访问过程的任何结果（例如，来自一个运行的查询的数据）。然而，在运行之后，任何该过程导出的文件和保存的数据都是可用的。

参数

名称	描述
ProcID	过程的标识，由 InitializeProc() 返回。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

SaveData()

short SaveData(long *QueryID*, long *FirstRow*, long *FirstCol*, long *LastRow*, long *LastCol*, BOOL *Replace*, BSTR *TableName*, BSTR *TableSpaceName*, BSTR *ServerName*, BSTR *UserID*, BSTR *Password*, BOOL *ForceDialog*, [VARIANT *Account*], [VARIANT *Comment*], [VARIANT *CommitScope*])

描述

此函数将指定范围的行和列保存到指定表空间的指定表中。如果还未完成对想要保存到表中的所有行的行数据的检索，则调用此函数前必须先调用

CompleteQuery()。如果要保存的行不是从数据库中检索到的，则保存操作将失败。如果该表已经存在，则新数据的列数和列类型必须与原有表中的完全一样。

此函数不同于其它 API 函数，它是在单独的工作单元中运行的，它的结果被自动提交。调用 Commit() 或 Rollback() 不会影响使用此函数所做的更改。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
FirstRow	想要保存的第一行。结果集中第一行的值为零。
FirstCol	想要保存的第一列。结果集中的第一列的值为零。
LastRow	想要保存的最后一行，如果包含了所有行，则为 -1。结果集中最后一行的值是一个小于总行数的数值。
LastCol	想要保存的最后一列；如果包含了所有列，则为 -1。结果集中最后一列的值是一个小于总列数的数值。
Replace	此参数值非零则表示指定数据将替换表中的所有现有数据。如果为零则表示指定数据将附加到表中现有数据的后面。
TableName	数据所存储的表的名称。如果此表不存在，将创建它。
TableSpaceName	表所位于或创建于的表空间的名称。如果 *TableSpaceName* 为 NULL 或空字符串，则使用缺省表空间。如果已经将 QMF for Windows 配置为总是使用缺省表空间（请参阅 GetResourceLimit() 的描述中的 RSR_SDDIFFERENTTS），则此参数将被忽略。
ServerName	存放表的数据库服务器的名称。如果 *ServerName* 为 NULL 或空字符串，则将使用在 InitializeServer() 的调用中指定的服务器，并且 *UserID*、*Password*、 *ForceDialog* 和 *Account* 将被忽略。
UserID	如果在 *ServerName* 中指定了其它服务器，则 *UserID* 将是用于该服务器的用户标识。如果没有指定用户标识，QMF for Windows 将使用指定的最后一个用户标识（如果可得到），或者在得不到的情况下显示对话框。如果 *ServerName* 为 NULL 或空字符串，此参数将被忽略。
Password	如果在 *ServerName* 中指定了其它服务器，则 *Password* 是用于该服务器的口令。如果不指定口令，QMF for Windows 将使用为此服务器指定的最后一个口令（如果可得到），或者在得不到此口令的情况下显示对话框。如果 *ServerName* 为 NULL 或空串，此参数将被忽略。

ForceDialog	如果您在 *ServerName* 中指定了不同的服务器，非零强制 QMF for Windows 显示对话框提示输入注册信息，不管已经指定了或有其它可用的用户标识和口令。零表示 QMF for Windows 仅在必要的时候显示此对话框。如果 *ServerName* 为 NULL 或一个空串，此参数将被忽略。
Account	如果您在 *ServerName* 中指定不同的服务器，则是任选的字符串，它指定连接时传送给服务器的帐户管理信息。服务器可能在作业帐户管理系统中使用这些信息。如果 *ServerName* 为 NULL 或空串，此参数将被忽略。
Comment	任选的字符串，它为保存数据的表指定注释。
CommitScope	任选的，在每次提交工作单元之前，插入多少行到表中。指定零，表示在提交前所有行都应当插入。指定 10（举例说），表示每插入十行后就该提交一次。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。如果结果集为空，或者从数据库中没有检索到行，则返回一个非零值，除非 FirstRow = 0 并且 LastRow = -1。如果是这样的情况，系统将返回零，并且创建空表。

SaveQMFProc()

short SaveQMFProc(BSTR *OwnerAndName*, BSTR *Text*, BSTR *Comment*, BOOL *Replace*, BOOL *Share*)

描述

此函数在数据库服务器保存过程。

参数

名称	描述
OwnerAndName	包含要保存过程的拥有者和名字的字符串，所有者和名字之间用句点隔开。例如， `John.Proc2`
Text	包含想要保存到过程中的文本的字符串。
Comment	包含要与该过程一起保存的任何注释的字符串。如果没有注释，则将这个参数作为空串或 NULL 进行传递。
Replace	非零则表示用同一名称代替现有过程。为零则表示当同一名称的过程已经存在时异常中止。
Share	非零则与其他用户共享该过程。为零则不与其他用户共享该过程。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

SaveQMFQuery()

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

描述

此函数在数据库服务器保存查询。

参数

名称	描述
OwnerAndName	包含要保存查询的拥有者和名字的字符串，所有者和名字之间用句点隔开。例如， `John.Query2`
Text	包含想要保存到查询中的文本的字符串。
Comment	包含要与该查询一起保存的任何注释的字符串。如果没有注释，则将这个参数作为空串或 NULL 进行传递。
Replace	非零则表示用同一名称代替现有查询。为零则表示当同一名称的查询已经存在时异常中止。
Share	非零则与其他用户共享该查询。为零则不与其他用户共享该查询。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

SetBindOption()

short SetBindOption(BSTR *CollectionName*, BSTR *PackageName*, short *Option*, short *Value*)

描述

此函数在调用 EndBind() 之前先设置集合和包的选项。

参数

名称	描述

CollectionName	想要设置其选项的包的集合标识。
PackageName	想要设置其选项的包的名称。
Option	下面列出的某个选项。
Value	非零则表示用同一名称代替现有查询。为零则表示当同一名称的查询已经存在时异常中止。
Share	指定选项的某个值，这些值也都在下面列出。

各选项的含义和值如下:

Option	含义	描述
DDM_PKGRPLOPT(0x211C)	指示是否要替换同名集合标识和名称的现有包的标志。	DDM_PKGRPLALW (0x241F) 是 DDM_PKGRPLNA (0x2420) 否
DDM_STTDECDEL(0x2121)	包的 SQL 语句中用于小数点的分界符。	DDM_DECDELPRD (0x243C) 句点 DDM_DECDELCMA (0x243D) 逗号
DDM_STTSTRDEL(0x2120)	包的 SQL 语句中用于字符串值的分界符。	DDM_STRDELAP (0x2426) 单引号 DDM_STRDELDQ (0x2427) 双引号
DDM_PKGISOLVL(0x2124)	包的隔离级别。	DDM_ISOLVLALL (0x2443) 全部 DDM_ISOLVLCHG (0x2441) 更改 DDM_ISOLVLCS (0x2442) 光标稳定性 DDM_ISOLVLNC (0x2445) 没有提交 DDM_ISOLVLRR (0x2444) 可重复读取
DDM_PKGATHOPT(0x211E)	指示是否保持包的原有授权的标志。	DDM_PKGATHKP (0x2425) 保持 DDM_PKGATHRVK (0x2424) 撤消
DDM_QRYBLKCTL(0x2132)	读取包中查询的数据行时所采用的方法。	DDM_FIXROWPRC (0x2418) 每次一行 DDM_LMTBLKPRC (0x2417) 每次一块

DDM_RDBRLSOPT(0x2129)	何时释放运行包时获取的数据库资源。	DDM_RDBRLSCMM (0x2438) 提交 DDM_RDBRLSCNV (0x2439) Conversation Deallocation
DDM_STTDATFMT(0x2122)	所检索日期值的格式。	DDM_ISODATFMT (0x2429) ISO DDM_USADATFMT (0x242A) 美国 DDM_EURDATFMT (0x242B) 欧洲 DDM_JISDATFMT (0x242C) 日本工业标准
DDM_STTTIMFMT(0x2123)	所检索时间值的格式。	DDM_ISOTIMFMT (0x242E) ISO DDM_USATIMFMT (0x242F) 美国 DDM_EURTIMFMT (0x2430) 欧洲 DDM_JISTIMFMT (0x2431) 日本工业标准

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多信息。

SetBindOwner()

short SetBindOwner(BSTR *CollectionName*, BSTR *PackageName*, BSTR *OwnerID*)

描述

此函数可让您为正在绑定的包指定不同于您用户标识的拥有者。当您的用户标识不具备绑定包所要求的授权，而指定的拥有者却具有此授权时，这个函数可能很有用处。

参数

名称	描述
CollectionName	想要指定其拥有者的包的集合标识。
PackageName	想要指定其拥有者的包的名称。
Comment	包含要与该查询一起保存的任何注释的字符串。如果没有注释，则将这个参数作为空串或 NULL 进行传递。
OwnerID	正在绑定的包的期望拥有者标识。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

SetBusyWindowButton()

void SetBusyWindowButton(BSTR *Text*)

描述

此函数指定在忙窗口中“取消”按钮上显示的文本。

参数

名称	描述
Text	指定忙窗口的“取消”按钮上所显示文本的字符串。缺省值为“取消”。如果您指定了一个空串，该按钮将被隐藏。无论指定什么样的文本，此按钮总是执行取消或关闭窗口的操作。

返回值

无。

相关主题

SetBusyWindowMessage()
SetBusyWindowMode()
SetBusyWindowTitle()
ShowBusyWindow()

SetBusyWindowMessage()

void SetBusyWindowMessage(BSTR *Message*)

描述

此函数指定在忙窗口的消息区域显示的文本。

参数

名称	描述
Message	指定忙窗口的“取消”消息区域中所显示文本的字符串。

返回值

无。

相关主题

SetBusyWindowButton()
SetBusyWindowMode()
SetBusyWindowTitle()
ShowBusyWindow()

SetBusyWindowMode()

void SetBusyWindowMode(short *Mode*)

描述

此函数确定 QMF for Windows 是否显示忙窗口：忙窗口很有用处，它可以向用户提供反馈信息，并允许用户取消暂挂的数据库操作。您所作的更改将在 QMF for Windows 下一次执行引起忙窗口显示或隐藏的操作时生效。

参数

名称	描述
Mode	指定 QMF for Windows 何时显示忙窗口：

值	含义
0 (RSM_NEVER)	该窗口不显示。这是缺省值。
1 (RSM_WHENBUSY)	该窗口在 QMF for Windows 与数据库进行通信时出现。QMF for Windows 在适当的时侯自动地显示这个窗口。
2 (RSM_CLIENTCONTROLLED)	该窗口在调用 ShowBusyWindow(TRUE) 和 ShowBusyWindow(FALSE) 之后出现。由客户来决定该窗口何时出现。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多信息。

相关主题

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

SetBusyWindowTitle()

void SetBusyWindowTitle(BSTR *Title*)

描述
此函数指定在忙窗口的标题栏上显示的文本。

参数

名称	描述
Title	指定忙窗口的标题栏上所显示文本的字符串。

返回值
无。

相关主题
SetBusyWindowButton()
SetBusyWindowMode()
SetBusyWindowMessage()
ShowBusyWindow()

SetGlobalVariable()

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

描述
此函数为指定全局变量赋值。此值在查询、表单和过程中都是可用的。

参数

名称	描述
Name	包含了想要设置的变量名的字符串。
Value	包含了想要赋值给指定变量的值的字符串。

返回值
如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多信息。

SetHostVariable()

short SetHostVariable(long *QueryID*, VARIANT *Index*, VARIANT *Value*)

描述
此函数为被该查询引用的特定主变量赋值。此查询必须是引用主变量（或由 QMF 查询存储，或由AddHostVariable() 创建）的静态查询。*Index* 可以指定为主变量的数字索引或名称。

参数

名称	描述
QueryID	查询的标识，由 InitializeStaticQuery() 返回。
Index	或者是指定该查询中主变量索引的数值（变体类型 VT_I2），或者指定主变量名称的一个字符串（变体类型 VT_BSTR）。
Value	主变量的值。要指定空值，就要将它的类型设置为 VT_EMPTY。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

SetOption()

short SetOption(short *Mode*, VARIANT *Value*)

描述

此函数设置 QMF for Windows 中的指定选项值。对于某些选项，所做的更改直到 QMF for Windows 重新启动时才会生效。正常情况下，您不会重新启动 QMF for Windows，除非您已毁坏了 QMF for Windows API 对象的所有实例。

参数

名称	描述
Option	指定要设置的选项:

值	含义
0 (RSO_SERVER_DEFINITION_FILE)	服务器定义文件名。
1 (RSO_CPIC_DLL)	CPI-C 提供者 DLL 文件名。
2 (RSO_CPIC_TIMEOUT_WARNING)	CPI-C 警告超时（以秒为单位）。此限制不用于 QMF for Windows API。
3 (RSO_CPIC_TIMEOUT_CANCEL)	CPI-C 取消超时（以秒为单位）。
4 (RSO_TCP_TIMEOUT_WARNING)	TCP 警告超时（以秒为单位）。此限制不用于 QMF for Windows API。
5 (RSO_TCP_TIMEOUT_CANCEL)	TCP 警告超时（以秒为单位）。
6 (RSO_DISPLAY_NULLS_STRING)	用于显示空值的字符串。
7 (RSO_ENTER_NULLS_STRING)	用于输入空值的字符串。
8 (RSO_ENTER_DEFAULTS_STRING)	用于输入缺省值的字符串。
9 (RSO_TRACE_FILE_1)	跟踪文件 1 名称。
10 (RSO_TRACE_FILE_2)	跟踪文件 2 名称。

11 (RSO_TCP_TRACE_LEVEL)	TCP 跟踪级。
12 (RSO_CPIC_TRACE_LEVEL)	CPI-C 跟踪级。
13 (RSO_DDM_TRACE_LEVEL)	DDM 跟踪级。

名称	描述
Value	用于设置选项的值。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

相关主题

GetOption()

SetParent()

short SetParent(long *ParentWnd*)

描述

此函数设置对话框的父窗口。通常，当 QMF for Windows 显示对话框时（在忙窗口中或“用户信息”对话框中），它是居中的，并且以 QMF for Windows 主窗口为模式。此函数允许您将 QMF for Windows 强制放在对话框的中央，并以客户应用程序窗口为模式显示。

参数

名称	描述
ParentWnd	新父窗口的 HWND。指定 NULL，以将 QMF for Windows 主窗口作为父窗口。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

相关主题

ShowBusyWindow()

SetProcVariable()

short SetProcVariable(long *ProcID*, BSTR *Name*, BSTR *Value*)

描述

此函数为指定变量赋值。这个值在运行过程语句前替换掉变量。如果您的过程中有一个或多个变量，则在调用 RunProc() 之前必须调用此函数以设置这些变量的值。

参数

名称	描述
ProcID	过程的标识，由 InitializeProc() 返回。
Name	字符串，包含了想要设置的变量的名称。
Value	字符串，包含想要赋给指定变量的值。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

SetVariable()

short SetVariable(long *QueryID*, BSTR *Name*, BSTR *Value*)

描述

此函数为指定变量赋值。这个值在运行 SQL 语句前替换掉变量。如果您的 SQL 语句中有一个或多个变量，则必须先调用此函数设置变量值，然后调用 Open() 或 Execute()。

参数

名称	描述
QueryID	查询的标识，由 InitializeQuery() 返回。
Name	字符串，包含了想要设置的变量的名称。
Value	字符串，包含想要赋给指定变量的值。

返回值

如果成功则为零，否则为非零值。如果返回值不是零，则可调用 GetLastErrorString() 或 GetLastErrorType() 获取更多出错信息。

ShowBusyWindow()

void ShowBusyWindow(BOOL *Show*)

描述

此函数让 QMF for Windows 显示或隐藏忙窗口。忙窗口很有用处，它可以向用户提供反馈信息，并允许用户取消一个暂挂的数据库操作。此函数仅在以

RSM_CLIENTCONTROLLED 方式调用 SetBusyWindowMode() 时生效。如果是通过调用 SetParent() 设置父窗口，则忙窗口将是指定窗口的模式。

参数

名称	描述
Show	非零则表示显示忙窗口；为零则表示隐藏忙窗口。如果非零，则忙窗口出现，直至您调用 ShowBusyWindow() 将 *Show* 设置为零。

返回值

无。

StartBind()

short StartBind(BSTR *CollectionName*, BSTR *PackageName*, BSTR *ConsistencyToken*)

描述

此函数开始数据库中包的绑定过程。

参数

名称	描述
CollectionName	包的期望集合标识。
PackageName	包的期望名称。
ConsistencyToken	一个 16 个字符长度的字符串，它包含一个 8 位令牌的十六进制表示，用于确保数据库中所绑定包和使用该包的应用程序之间的一致性。当一个段在包中执行时，必须提供与此相同的值。

返回值

如果成功则为零，否则为非零值。如果返回值不为零，则可调用 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 获取更多出错信息。

相关主题

EndBind()
CancelBind()

附录. 注意事项

本信息适用于在美国和中国提供的产品和服务。 IBM 可能不在其它国家提供本文档中讨论的产品、服务或功能组件。有关您所在地区当前产品和服务的情况，可咨询您的本地 IBM 代表。任何对 IBM 产品、程序或服务的引用并不说明或暗示只能使用 IBM 的产品、程序或服务。任何不侵犯 IBM 知识产权、具有同等功能的产品、程序或服务均可用来代替 IBM 产品、程序或服务。当然，对任何其它非 IBM 产品、程序或服务的评价和验证，均由用户自行负责。

本文档描述的议题可能涉及 IBM 的某些专利或正在申请中的专利的应用。这份文档并未给予您运用这些专利的许可。您可以用书面形式将特许查询寄往:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10594-1785
U.S.A.

有关双字节 (DBCS) 信息的特许查询，请与您所在国家的 IBM 知识产权部门联系，或以书面形式将查询寄往:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

下面这一段落不适用于英国或任何其它这些条款与本地法律不一致的国家: 国际商业机器公司 (IBM) 『按现样』出版本书，不做任何明确或暗示的担保，包括（但不局限于）非侵权性、可销售性或适合于特殊目的的暗示担保。有些地区不允许在某些事务中放弃明确或暗示的担保，因此本条款可能不适合您。

本信息中可能包含不准确内容或者印刷错误。我们定期更改这些信息；这些更改将并入本出版物的新版本中。IBM 任何时侯可能对本出版物中描述的产品和程序作改进和更改，不另行通知。

在本出版物中，仅为方便而引用了一些非 IBM Web 站点，但并没有保证这些 Web 站点的服务器的存在。这些 Web 站点上的资料不属本 IBM 产品的资料，对这些站点的使用由您自己负责。

如果希望获取有关以下事项的信息: (i) 在各自创建的程序和其它程序（包括本程序）之间进行信息交换, (ii) 交换后信息的相互使用, 则有关本程序的许可证信息, 请与以下地址联系:

IBM Corporation
J74/G4
555 Bailey Avenue
P.O. Box 49023
San Jose, CA 95161-9023
U.S.A.

根据包括在一些情况下的项目和条件, 只要支付一些费用, 就可以使用这些信息。

本信息中描述的特许程序及其全部可用的特许资料, 由 IBM 按照 IBM 客户协定、IBM 国际程序特许协定或我们之间的任何等效协定提供。

这里包含的任何性能数据都在受控环境中确定。因此, 在其它操作环境中所获结果可能有明显变化。有些测量可能是在开发级的系统上进行的, 因此无法保证在一般可用系统上进行相同的测量。此外, 有些测量通过归纳估算。实际结果可能有变化。本文档的用户应当验证其特定环境的适用数据。

有关非 IBM 产品的信息, 可从其供应商、发行宣告或其它有效的出版源获得。IBM 不对这些产品作测试, 也无法确认其性能、兼容性或其它有关非 IBM 产品声明的准确度。非 IBM 产品功能上的问题应向那些产品的供应商提出。

涉及 IBM 未来方向或意向的所有语句可随时更改或抽去, 它们只表示目标。

给出的所有 IBM 价格是 IBM 的建议零售价, 是当前价格, 随时可更改。推销价可能有变化。

本信息仅为了计划目的。在描述的产品成为可用之前, 此处的信息随时可更改。

本信息包含日常商务操作所使用的数据和报表的例子。为了尽可能完整地说明它们, 这些例子中包含了个人、公司、商标和产品的名称。所有这些名称都是虚构的, 若与实际商业企业中用到的名称相同, 则纯属巧合。

版权特许:

本信息包含源语言的示例应用程序, 以便说明各种不同的操作平台上的程序设计技术。您可以为了开发、使用、销售或分发符合编写这些示例程序的操作平台的

应用程序编程接口的应用程序的目的，以任何形式复制、修改和分发这些示例程序，不必向 IBM 作支付。这些例子未在所有条件下经过测试。因此，IBM 无法保证或暗示其可靠性、可用性或功能。

如果您正在查看本信息的软拷贝，则可能不会有照片或彩色说明。

商标

以下术语是 IBM 公司在美国或其他国家的注册商标:

ACF/VTAM
Advanced Peer-to-Peer Networking
AIX
AIX/6000
AS/400
CICS
CICS/ESA
CICS/MVS
CICS/VSE
COBOL/370
DATABASE 2
DataJoinerDB2
DB2 Universal Database
Distributed Relational Database Architecture
DRDA
DXTGDDM
IBM
IBMLink
IMS
Language Environment
MVS/ESA
MVS/XA
OfficeVision/VM
OS/2
OS/390
PL/I
PROFS
QMF
RACF
S/390
SQL/DS
Virtual Machine/Enterprise Systems Architecture
Visual Basic
VM/XA
VM/ESA
VSE/ESAVTAM

Java 或所有基于 Java 的商标和徽标以及 Solaris 是 Sun Microsystems, Inc. 在美国和其它国家的商标。

Lotus 和 1-2-3 是 Lotus Development Corporation 在美国和其它国家的商标。

Microsoft、Windows 和 Windows NT 是 Microsoft Corporation 的商标或注册商标。

其他公司、产品和服务名称（可能以双星号 ** 标出）可能是其他公司的注册商标或服务标志。

索引

[Z]

A

B

C

D

E

F

G

H

I

O

P

R

S

读者意见表

Query Management Facility
QMF for Windows 入门

SB84-0253-00

姓名

地址

单位及部门

电话号码

读者意见表

SB84-0253-00

请沿此线
撕下或折起

折起并封口　　**请勿使用钉书机**　　折起并封口

在此
贴上
邮票

IBM CORPORATION
Department BWE/H3
P.O. Box 49023
San Jose, CA 95161-9023
U.S.A.

折起并封口　　**请勿使用钉书机**　　折起并封口

SB84-0253-00

请沿此线
撕下或折起

文件号码:
程序编号: 5675-DB2
5697–F42
5697-G24
5697-G22
5648–D35
5697-G23

Printed in China

SB84-0253-00

Spine information:

QMF

QMF for Windows 入门

版本 7