

Wonderware[®] FactorySuite[™] SQL Access Manager

用户指南

修订版

2001 年 6 月

Wonderware Corporation

保留所有权利。未经 Wonderware 公司的事先书面许可，不得以电子、机械、影印、录制或其它的任何方式复制、传输本说明文件的任何部分或将其储存在检索系统中。使用此处包含的信息不应承担版权或专利责任。虽然在本说明文件编制过程中采取了一切预防措施，错误或疏漏在所难免，出版商和作者不对此承担任何责任，亦不对因使用此处包含的信息而引起的任何损害负责。

本说明文件的信息如有变更，恕不另行通知，亦不代表 Wonderware 公司一方的承诺。本说明文件所述的软件在遵守许可证或许可协议的前提下提供。本软件的使用或复制须遵从这些协议规定的条款。

© 2001 Wonderware 公司。版权所有。

100 Technology Drive
Irvine, CA 92618
U.S.A.
(949) 727-3200
<http://www.wonderware.com>

商标

本书所有作为商标或服务标志的专门名词均采用大写字母印刷。Wonderware 公司并不保证本书信息的准确性。使用本书中的任何术语不应视为影响任何商标或服务标志的有效性。

Wonderware、FactorySuite 和 InTouch 是 Wonderware 公司的注册商标。

WindowMaker、WindowViewer、SQL Access Manager、Recipe Manager、SPCPro、DBDump、DBLoad、HDMerge、HistData、Wonderware Logger、Alarm Logger、InControl、InBatch、IndustrialSQL、FactoryOffice、FactoryFocus、License Viewer、SuiteLink、SuiteVoyager 和 NetDDE 是 Wonderware 公司的商标。

目录

第 1 章 - SQL Access Manager	1-1
简介	1-2
关于本手册.....	1-3
技术支持	1-4
您的 FactorySuite 许可证	1-5
ODBC 兼容性	1-6
 第 2 章 - 配置和连接数据库	2-1
使用 Oracle 6.....	2-2
配置客户端	2-2
使用 Oracle 7.2.....	2-4
配置客户端	2-4
在 Oracle 日期字段中记录日期和时间	2-7
使用 Sybase 或 Microsoft SQL Server	2-8
配置客户端	2-8
支持的数据类型	2-9
使用 dBASE.....	2-10
使用 Microsoft Access	2-12
使用 Paradox	2-13
支持的数据库的数据类型值	2-14
 第 3 章 - 配置 SQL Access Manager	3-1
SQL Access Manager 概述	3-2
配置绑定列表	3-3
使用特殊分隔符	3-7
配置表格模板	3-8
SQL.DEF 文件	3-11
 第 4 章 - 使用 SQL 函数	4-1
SQL 函数	4-2
SQL 参数	4-5
在 InTouch QuickScript 中使用 SQL 函数	4-8
指定复合查询	4-8
动态建立查询	4-9
从文件中读取 SQL 语句	4-9
修改扩展 SQL 语句	4-10
执行扩展 SQL 语句	4-11
支持存储过程	4-13
 第 5 章 - 错误排解	5-1
函数错误排解	5-2
结果码错误消息	5-2
特定数据库错误消息	5-4

附录 A - SQL Access 与 ODBC 的保留关键字A-1

索引..... I-1

第 1 章

SQL Access Manager

Wonderware® FactorySuite® SQL Access Manager 允许您访问、修改、建立和删除数据库表格。数据库在一个共享公共属性或字段的表格中存储信息。

“结构化查询语言”（SQL）即是用于访问该信息的语言。

目录

- 简介
- 关于本手册
- 技术支持
- 您的 FactorySuite 许可证
- ODBC 兼容性

简介

InTouch SQL Access Manager 附加程序的设计，可以将批配方等数据从 SQL 数据库轻易传送到 InTouch 应用程序。它也可以简化运行时数据、报警状态或历史数据从 InTouch 到 SQL 数据库的传送。例如，当一个机械循环完成后，公司可能需要存储几套数据，每套数据需要一个不同的应用程序。SQL 数据库提供了在一个或多个第三方应用程序之间轻易传送信息的能力。SQL Access Manager 允许在任何 InTouch 应用程序中访问和显示这些数据。

InTouch SQL Access 产品由 SQL Access Manager 程序和 SQL 函数组成。SQL Access Manager 程序用于创建数据库列并将其与标记名字典中的标记名相关联。将数据库列与 InTouch 数据库标记名相关联的过程称为“绑定”。将 InTouch 数据库标记名与数据库列绑定允许 SQL Access Manager 直接控制数据库中的数据。SQL Access Manager 将数据库字段名及其关联保存在一个名为 "SQL.DEF." 的逗号分隔变量 (.CSV) 格式的文件中（此文件位于 InTouch 应用程序目录中，可使用 SQL Access Manager 或记事本之类的文本编辑器查看或修改）。SQL Access Manager 还可创建表模板来定义数据库的结构和格式。

有关绑定列表和表格模板的详细信息，请参阅第 3 章“配置 SQL Access Manager”。

SQL 函数可用于任何 InTouch 动作脚本。这些函数可以根据运算符输入、标记名值改变或特定条件集的出现自动执行。例如，如果报警条件存在时，应用程序将执行 `SQLInsert()` 或 `SQLUpdate()` 命令，以保存所有可应用的数据点和报警状态。SQL 函数可用于建立新表格、在表格插入新记录、编辑现有表格记录、清除表格、删除表格、选择记录和滚动浏览记录等。

InTouch SQL Access 使用 Intersolv/Q+E Software 的数据库程序库 QELIB（它支持许多不同的数据库系统），并且包含与 ODBC 兼容的数据库驱动程序连接所需的所有代码。这些驱动程序不包括在 InTouch SQL Access 软件产品中，而必须从 Intersolv/Q+E 和 Microsoft 公司这样的制造商那里单独购买。ODBC 驱动程序要求与数据库实际“连接”。例如，如果要在 Microsoft® Access 数据库中应用 SQL 函数，您必须单独购买、安装和配置 Microsoft® ODBC Desktop Driver Set。本章的下一节将介绍有关驱动程序配置的一般示例。

注意：本用户指南中未论及的数据库系统不受支持。

关于本手册

本手册由逻辑上连贯的多个章节组成，就使用 SQL Access Manager 的各方面问题分别进行介绍。本书用“程序化”格式写成，通过连续的步骤阐述如何执行大多数的功能和任务。

如果您阅读的是本手册的联机版，则可以在看到绿色文本时，单击该文本来跳到相关的章节。如果您要在跳转后回到原来的章节，可以使用所提供的“后退”选项。

☞ 这些“提示”可以告诉您如何用更简便快捷的方法来完成某项功能或任务。

《*InTouch 用户指南*》将帮助您熟悉 WindowMaker 开发环境及其工具，请参阅第 1 章“WindowMaker 程序元素”。要了解如何使用窗口、图形对象、向导和 ActiveX 控件等元素，请阅读第 2 章“使用 WindowMaker”。

有关 InTouch 运行时环境 (WindowViewer) 的详细信息，请参阅您的联机《*InTouch 运行时用户指南*》。

此外，《*InTouch 参考指南*》为您提供了有关 InTouch 脚本语言、系统标记名以及标记名点域的深入参考资料。

有关附加程序 SPC Pro 的详细信息，请参阅您的《*SPC Pro 用户指南*》。

有关附加程序“配方管理器”的详细信息，请参阅您的《*配方管理器用户指南*》。

《*FactorySuite 系统管理员指南*》也为您提供了有关 FactorySuite 中的其它组件程序、系统要求、网络连接注意事项、产品集成和技术支持等方面的完整信息。。

FactorySuite 软件包还提供了所有 FactorySuite 组件的联机版手册。

注意：要查看或打印联机手册，您必须安装 Adobe Acrobat Reader（4.0 或以上版本）。

假设

本手册假设您：

- 已经熟悉 Windows 2000 和（或）Windows NT 操作系统工作环境。
- 懂得如何使用鼠标、Windows 菜单、选择选项和访问联机帮助。
- 有使用编程语言和宏语言的经验。最好理解一些编程概念，如变量、语句、函数和方法等。

技术支持

Wonderware 技术支持提供许多支持选项，来回答有关 Wonderware 产品及其实施的问题。

在联系技术支持前，请参考您的《*SQL Access Manager 用户指南*》中的相关章节，寻找使用系统时所遇问题的可能答案。如果您觉得有必要求助于技术支持，请提供以下信息：

1. 您的软件序列号。
2. 您所运行的 InTouch 版本。
3. 您所使用的操作系统类型和版本。例如 Microsoft Windows NT 4.0 workstation。
4. 描述所遇系统错误消息的准确用词。
5. 来自 Wonderware Logger、Microsoft Diagnostic utility (MSD) 或任何其它诊断应用程序的任何相关输出列表。
6. 您所尝试的解决方法的细节和结果。
7. 有关如何重现问题的细节。
8. 如果所遇问题是老问题，请提供指定的 Wonderware 技术支持案例号。

有关技术支持的详细信息，请参阅您的联机《*FactorySuite 系统管理员指南*》。

您的 FactorySuite 许可证

要查看您的 FactorySuite 系统许可证信息，可以从 WindowMaker 帮助菜单的“关于”对话框中启动许可证查看程序进行查看。

➤ **要打开许可证实用程序：**

1. 在 WindowMaker 的“帮助”菜单上，选择“关于”命令。
2. 单击“**View License**”（查看许可证）。“**License Utility - LicView**”（许可证实用程序 - LicView）对话框出现。

有关查看许可证实用程序的详细信息，请参阅您的《*FactorySuite 系统管理员指南*》。

ODBC 兼容性

SQL Access Manager 是一个 ODBC 兼容应用程序，它可与任何数据库系统通信，只要该数据库安装有 ODBC 驱动程序。在使用 ODBC 驱动程序之前，它必须通过 Microsoft ODBC 管理器程序进行配置，以设定 ODBC 兼容应用程序与数据库间的链接。

➤ 要配置 ODBC 驱动程序：

1. 运行 Microsoft ODBC 管理器程序。
2. 选定驱动程序或数据源，然后单击“添加新名称”、“设置缺省”或“配置”，“ODBC 驱动程序设置”对话框出现。

选项	描述
数据源名	用于确定数据源的用户自定义名称。
描述	该数据源的用户自定义说明。
数据库目录	确定包含数据库文件的目录。如果未指定目录，系统将使用当前工作目录。
	☞ 输入配置所选驱动程序所需的任何其它信息。

3. 单击“确定”。

☞ 驱动程序会将每个字段的值写入 ODBC.INI 文件。这些值是连接到数据源的缺省值。缺省值可以通过修改数据源字段进行更改。对于任何“ODBC 驱动程序设置”对话框不支持的属性，可在 ODBC.INI 文件的相应数据源部分中手工插入。

连接要求

在 InTouch SQL Access Manager 4.x 版本中的 **SQLCoonect()** 函数所使用的连接字符串是“DRV=database_d11”约定。

因为 5.6（或以后）版本是 ODBC 兼容应用程序，所以“DRV=database_d11”中的 database_d11 字段必须与使用 ODBC 管理器程序配置的数据源名完全相同（如上所述）。您也可以使用“DSN”来代替“DRV”。

第 2 章

配置和连接数据库

本章中包括的数据库经过测试并当前可被支持。每个数据库的要求都有所不同。本章介绍每个数据库的不同部分，向您说明如何配置特定的数据库来与 SQL Access Manager 进行通讯。

目录

- 使用 Oracle 6
- 使用 Oracle 7.2
- 使用 Sybase 或 Microsoft SQL Server
- 使用 dBASE
- 使用 Microsoft Access
- 使用 Paradox
- 支持的数据库的数据类型值

使用 Oracle 6

➤ 要与 Oracle 6 进行通讯：

1. 配置您的 Windows 数据库客户端。
2. 启动 SQL*Net TSR 和 NETINIT.EXE 程序。
3. 通过执行 InTouch 动作脚本中的 **SQLConnect()** 函数，连接到 Oracle®。

有关 **SQLConnect()** 的详细信息，参见第 4 章“使用 SQL 函数”。

配置客户端

通过在本地计算机安装 Oracle SQL*Net 产品可以访问 Oracle 数据库。Oracle SQL*Net 产品包括 SQL*Net TSR 和 NETINIT.EXE 程序。为了建立与 Oracle 服务器的通讯，必须运行 SQL*Net TSR 和 NETINIT.EXE。

启动 SQL*Net TSR 和 NETINIT.EXE

在运行 Windows 之前，必须从 DOS 加载与所使用的网络兼容的 SQL*Net TSR。为了建立与 Oracle 服务器的通讯，必须启动 NETINIT.EXE 程序。要在 Windows 启动时自动执行 NETINIT.EXE，将其图标放入 Windows 启动程序组中。

SQLConnect() 格式

SQLConnect() 函数用于连接到 Oracle 数据库。**SQLConnect()** 函数所使用的连接字符串格式如下：

```
SQLConnect (ConnectionId, "<attribute>=<value>;  
            <attribute>=<value>;...");
```

下面介绍 Oracle 所使用的属性。这些属性按下列顺序指定：

属性	值
DSN	在 Microsoft ODBC 管理器中配置的数据源名称。
DRV	为确保与 SQL Access Manager 的兼容，如果链接字符串中没有数据源名称 (DSN)，则将使用此值。QELIB 会将其改为数据源名称。
UID	用户名。
PWD	口令。
SRVR	SQL*Net 连接字符串指定所访问的服务器计算机和数据库。

实例：

```
SQLConnect (ConnectionId, "DSN=Oracle_Data;UID=SCOTT;  
PWD=TIGER;SRVR=B:MKTG_SRV");
```

此处：**B**:指定 NetBIOS 通信层的前缀（此字符串中的 **SRVR=B:MKTG_SRV** 部分），或

T:指定 TCP/IP 的前缀，或

P:指定所命名管道的前缀

MKTG_SRV 是指定给 Oracle 数据库服务器的名称。

注意：SRVR 属性值较难指定。所需信息根据 SQL*Net 驱动程序的不同而变化。这些信息可以在 SQL*Net 文档中（或可能位于 CONFIG.ORA 文件的 "remote=" 行上）找到。

支持的数据类型

SQL Access Manager 支持 Oracle 6 数据库的两种数据类型。其中字符数据类型包含可变长度的字符串。InTouch 消息标记名要求使用字符数据类型。如果未指定长度，则缺省值为一 (1) 个字符。Oracle 支持最长达 255 个字符的字符字段。但 InTouch 消息标记名则限制为 131 字符。如果消息型变量的字符长度超过数据库字段中指定的长度，该字符串在插入数据库时会被截短。

数字数据类型表示 InTouch 整型和实型标记名。如果未指定长度，数值将用一个精确到 38 位小数的浮点数值来表示。如果指定了长度，格式为“宽度.小数”。其中宽度值决定列的最长位数；小数值则指定小数点右边的位数。此数据类型不要求指定字段长度。

使用 Oracle 7.2

➤ 要与 Oracle 7.2 进行通讯:

1. 在 InTouch 节点上安装 Oracle 标准客户端。
2. 运行 SQL_Net 简易配置，找到 SQL 连接字符串的数据库别名。
3. 建立数据源名称。
4. 通过执行 InTouch 动作脚本中的 **SQLConnect()** 函数，连接到 Oracle。

有关使用 **SQLConnect()** 的详细信息，请参阅第 4 章“使用 SQL 函数”。

配置客户端

➤ 要运行 Oracle 7 安装程序:

1. 在 Oracle 7 Workgroup 服务器屏幕上，选择“**标准客户**”作为您要执行的安装类型，然后单击“**确定**”。
2. 选择“**应用程序用户安装**”作为您要执行的客户端安装类型，然后单击“**确定**”。
3. 从“**数据库连接安装**”屏幕，输入安装 Oracle7 Workgroup 服务器的主机名。例如：WWServer。
4. 单击“**确定**”。

配置 SQL_Net

1. 在 Windows 任务栏上，单击“**开始**”。指向“**程序**”，然后指向“**Oracle**”，单击“**SQL_Net 简易配置**”。
2. 缺省情况下，服务器别名以 **wgs_ServerName_orcl** 开头。但是，您可以更改此名称。
 - ☞ 数据库别名将用于 InTouch 的 **SQLConnect()** 函数中。
3. 要修改数据库别名，选择服务器名，然后单击“**确定**”。
4. 单击“**改变数据库别名选择网络协议**”。“已命名管道服务器”是 Oracle 服务器的计算机名。SQL_Net 现在即已完成配置。
 - ☞ 确保记下数据库别名，以备日后使用。

配置数据源名称

➤ 要在客户端节点上安装 ODBC 驱动程序：

1. 在 Windows 任务栏上，单击“开始”。指向“设置”，然后单击“控制面板”。Windows 控制面板打开。
2. 双击“ODBC”。“ODBC 数据源管理器”对话框出现。



3. 单击“系统 DSN”标签。
4. 单击“添加”，“创建新数据源”对话框出现。
5. 选择 Oracle7 ODBC 驱动程序，然后单击“完成”。“ODBC Oracle 驱动程序设置”对话框出现。
6. 在“数据源名称”框中，输入 Oracle 服务器名。
7. 单击“高级”。在“ODBC Oracle 高级驱动程序设置”对话框上使用缺省设置。单击“关闭”。“ODBC 数据源管理器”对话框出现。
8. 单击“确定”。

SQLConnect() 格式

SQLConnect() 函数用于连接到 Oracle 数据库。**SQLConnect()** 函数所使用的连接字符串格式如下：

```
SQLConnect (ConnectionId, "<attribute>=<value>;  
            <attribute>=<value>;...");
```

下面介绍 Oracle 所使用的属性。这些属性按下列顺序指定：

属性	值
DSN	在 Microsoft ODBC 管理器中配置的数据源名称。
UID	用户名。
PWD	口令。
SRVR	SQL_NET 数据库别名。

实例：

```
SQLConnect (ConnectionId, "DSN=Oracle;UID=SCOTT;  
                        PWD=TIGER;SRVR=wsg_wwServer_orcl");
```

注意：如果数据库别名已更改其缺省格式 wsg_ServerName_orcl，请使用由 SQL_NET 简易配置创建的新名称 SRVR=WWServer。

在 Oracle 日期字段中记录日期和时间

要在 Oracle 日期字段中记录日期和时间，您必须使用分隔函数来配置绑定列表。

- 要在 Oracle 日期字段中记录日期和时间：
1. 在应用程序浏览器的 **SQL Access Manager** 下，双击“绑定列表”。“绑定列表配置”对话框出现。



2. 在“标记名.域名”框中，输入您要使用的标记名。
3. 在“列名”框中，输入 `DATE_TIME delim()` 函数。

注意：在“列名”字段中，必须用 `delim()` 函数输一个空格。例如，`DATE_TIME`。DATE 和 TIME 之间必须输入空格。

4. 在 InTouch 应用程序中建立一个 QuickScript，以准备输入当前日期和时间的数据。例如：

```
DATE_TIME_TAG = "TO_DATE(' " + $DateString + " " +  
StringMid($TimeString,1,8) + "','mm/dd/yy hh24:mi:ss')";
```

☞ `Date_Time_Tag` 在运行时显示如下：

```
TO_DATE('08/22/97 23:32:18' , 'mm/dd/yy hh24:mi:ss')
```

使用 Sybase 或 Microsoft SQL Server

➤ 与 Sybase 或 Microsoft SQL Server 进行通讯

1. 配置 Windows 数据库客户端。
2. 通过执行 InTouch QuickScript 中的 **SQLConnect()** 函数，连接到 Sybase® 或 Microsoft® SQL Server。

有关使用 **SQLConnect()** 的详细信息，请参阅第 4 章“使用 SQL 函数”。

配置客户端

➤ 连接到数据库

要连接到数据库，必须在 WINDOWS/SYSTEM 目录中安装下列客户端 DLL：

- DBNMP3.DLL
- W3DBLIB.DLL

注意：如果应用程序使用 Winsock (TCP/IP) 与数据库连接，则不需要 DBNMP3.DLL。

SQLConnect() 格式

SQLConnect() 函数用于连接到 Sybase 或 Microsoft SQL Server。执行此函数可以登录到数据库服务器并打开一个允许执行其它 SQL 函数的连接。

SQLConnect() 函数所使用的连接字符串格式如下：

```
SQLConnect (ConnectionId, "<attribute>=<value>;  
<attribute>=<value>;...");
```

以下介绍 Sybase 或 Microsoft SQL Server 所使用的属性。这些属性按下列顺序指定：

属性	值
DSN	在 Microsoft ODBC 管理器中配置的数据源名称。
DRV	为确保与 InTouch SQL Access 的兼容，如果链接字符串中没有数据源名称 (DSN)，则将使用此值。QELIB 会将其改为数据源名称。
UID	登录 ID，区分大小。
PWD	口令，区分大小写。
SRVR	包含要访问的数据库表格的服务器计算机名称。
DB	要访问的数据库名称。

实例：

```
SQLConnect (ConnectionId, "DSN=SQL_Data;UID=OPERATOR;PWD=XYZZ");
```

支持的数据类型

SQL Access Manager 支持 Sybase 和 Microsoft SQL Server 数据库的三种数据类型。其中字符数据类型包含可变长度的字符串。InTouch 消息标记名要求使用字符数据类型。必须指定字段长度。Sybase 和 Microsoft SQL Server 数据库支持最长达 255 个字符的字符字段。但 InTouch 消息标记名则限制为 131 字符。如果消息型变量的字符长度超过数据库字段中指定的长度，该字符串在插入数据库时会被截短。

整型数据类型表示 InTouch 整型标记名。如果未指定字段长度，其长度将设为数据库的缺省值。如果指定了长度被指定，则其格式为“宽度”。“宽度”决定列的最大位数。

浮点数据类型表示 InTouch 实型标记名。字段长度设置由数据库设定。此数据类型不要求指定字段长度。

注意：在使用 Sybase 数据库服务器时，数据类型（字符、整型、浮点）区分大小写，并且必须是小写。

Sybase 或 Microsoft SQL Server 对每个 ConnectionID 只支持一个有效语句。当执行 **SQLSelect()** 或 **SQLNext()** 时，您可以浏览，但是不能插入、删除或更新数据。要插入、删除或更新数据，请对现 **SQLInsert()**、**SQLUpdate()** 和 **SQLDelete()** 不同的 ConnectionId 执行 **SQLSelect()**。列名和表格名也是区分大小写的。表格名必须与表格所有者完全相称（例如，dbo.MyTable）。

使用 dBASE

要与 dBASE[®] 通讯，您必须通过执行 InTouch QuickScript 中的 **SQLConnect()** 函数来连接到 dBASE。

SQLConnect() 格式

SQLConnect() 函数用于连接到 dBASE 数据库。执行此函数可以登录到数据库服务器并打开一个允许执行其它 SQL 函数的连接。**SQLConnect()** 使用的连接字符串格式如下：

```
SQLConnect(ConnectionId,"<attribute>=<value>;  
<attribute>=<value>;...");
```

下面介绍 dBASE 所使用的属性。这些属性按指定顺序列出如下：

属性	值
DSN	在 Microsoft ODBC 管理器中配置的数据源名称。
DRV	为确保与 InTouch SQL Access 的兼容，如果链接字符串中没有数据源名称 (DSN)，则将使用此值。QELIB 会将其改为数据源名称。
CS	字符集通知驱动程序是以 IBM PC 字符集还是以 ANSI 字符集来存储数据。缺省为 IBMPC。
DB	指定 dBASE 文件的存储目录。如果未指定，则将使用当前工作目录。
FOC	在缓冲器中可打开的最大未使用文件数目。缺省值为 0。dBASE 驱动程序支持文件打开缓存以提高系统性能。当 FOC 大于 0 时，最近使用的文件都将保持打开状态。如果您尝试用另一应用程序独占打开这些文件，则会显示一则错误消息，表明文件当前正在使用。
LCK	确定数据库文件的记录锁定级别。有效值是 FILE、RECORD（缺省）或 NONE（例如，LCK=FILE）。
CSZ	驱动程序用于缓存数据库记录的 64K 区块的数目。区块数目越多，性能越好。缺省为 4 个区块。允许设置的最大区块数目由系统可用内存决定。 当反向浏览时，在重新执行 SQLSelect() 语句之前，将不能看见其他用户所做的更新。
USF	决定驱动程序何时更新 DOS 目录项。如果此选项设为 1，则驱动程序会在每次 COMMIT 时更新目录项。这就降低了系统性能。缺省值为 0，这意味着驱动程序在关闭文件时更新目录项。这种情况下，在文件关闭以前发生的机器“崩溃”会导致新插入的记录丢失。
MS	此选项提供了与以前的 Q+E 软件产品的反向兼容性。指定 MS=0，以让驱动程序了解以前的 Q+E 软件驱动程序中 SQL 术语。缺省为 1。

属性	值
LCOMP	决定是使用 dBASE-兼容锁定还是使用 Q+E-兼容锁定。指定 LCOMP=DBASE 以让驱动程序使用 dBASE-兼容锁定；指定 LCOMP=Q+E 以让驱动程序使用 Q+E-兼容锁定。缺省为 dBASE。 如果该锁定与以前版本的 Q+E 兼容，则可使用 Q+E-兼容锁定。
COMP	提供与以前 Q+E 软件产品的反向兼容性。如需反向兼容，使用 COMP=DBASE；如需可移植性，使用 COMP=ANSI。缺省为 ANSI。

实例：

```
SQLConnect(ConnectionId,"DSN=DBASE_FILES;  
CS=ANSI;DB=C:\ODBC\EMP;  
LCK=NONE");
```

支持的数据类型

SQL Access Manager 支持 dBASE 数据库的三种数据类型。其中字符数据类型包含可变长度的字符串。InTouch 消息标记名要求使用字符数据类型。必须指定长度。dBASE 数据库支持最长达 254 个字符的字符字段。但 InTouch 消息标记名则限制为 131 字符。如果消息型变量的字符长度超过数据库字段中指定的长度，该字符串在插入数据库时会被截短。

数字和浮点数据类型表示 InTouch 整型或实型标记名。必须指定长度。格式为“宽度.小数”。其中宽度值决定列的最长位数；格式为“宽度.小数”。其中宽度值决定列的最长位数；

使用 Microsoft Access

要与 Microsoft Access 通讯，您必须通过执行 InTouch QuickScript 中的 **SQLConnect()** 函数来与之连接。

SQLConnect() 格式

SQLConnect() 函数用于连接到 Microsoft Access 数据库。执行此函数可以登录到数据库服务器并打开一个允许执行其它 SQL 函数的连接。**SQLConnect()** 使用的连接字符串格式如下：

```
SQLConnect(ConnectionId, "<attribute>=<value>;  
<attribute>=<value>;...");
```

以下介绍 Microsoft Access 所使用的属性。这些属性按指定顺序列出如下：

属性	值
DSN	在 Microsoft ODBC 管理器中配置的数据源名称。
DRV	为确保与 InTouch SQL Access 的兼容，如果链接字符串中没有数据源名称 (DSN)，则将使用此值。QELIB 会将其改为数据源名称。

实例：

```
SQLConnect(ConnectionId, "DSN=MSACC");
```

支持的数据类型

SQL Access Manager 支持 Microsoft Access 数据库的五种数据类型。有效数据类型取决于当前使用的 ODBC 驱动程序版本。文本数据类型包含固定长度字符串并支持 InTouch 消息型标记名。必须指定长度。Microsoft Access 数据库支持最长 255 个字符的文本字段。InTouch 消息型标记名则限制在 131 个字符内。如果消息型变量的字符长度超过数据库字段中指定的长度，该字符串在插入数据库时会被截短。Microsoft Access ODBC 驱动程序支持每个列名最长 17 个字符。当使用 **SQLSetStatement(Select Col1, Col2, ...)** 时，最多能支持 40 列。

对于使用 Microsoft Access 第 1.1 版（包括 Excel 5.0、Microsoft Word 6.0 等）的“SIMBA.DLL”驱动程序的程序，有效数字类型是短或长整型标记名及单精度浮点数或双精度浮点实型标记名；

对于使用 Microsoft Access 第 2.0 版的“ODBCJT16.DLL”驱动程序的程序，有效数字类型是短或长整型标记名，以及带单或双修正符数字的实型标记名。长度根据数据类型自动计算，因此不得在表模板定义中指定。

使用 Paradox

要与 Paradox[®] 通讯，您必须通过执行 InTouch QuickScript 中的 **SQLConnect()** 函数与之连接。

SQLConnect() 格式

SQLConnect() 函数用于连接到 Paradox 数据库。执行此函数可以登录到数据库服务器并打开一个允许执行其它 SQL 函数的连接。**SQLConnect()** 使用的连接字符串格式如下：

SQLConnect(ConnectionId,"<attribute>=<value>;<attribute>=<value>;...");

以下介绍 Paradox 所使用的属性。这些属性按指定顺序列出如下：

属性	值
DSN	在 Microsoft ODBC 管理器中配置的数据源名称。
或	
DRV	为确保与 SQL Access Manager 的兼容，如果链接字符串中没有数据源名称 (DSN)，则将使用此值。QELIB 会将其改为数据源名称。

实例：

```
SQLConnect (ConnectionId, "DSN=PARADOX_FILE") ;
```

支持的数据类型

InTouch SQL 支持 Paradox 数据库的三种数据类型。字母数字数据类型包含固定长度的字符串，并且支持 InTouch 消息型标记名。必须指定字段长度。Paradox 数据库支持最长达 255 个字符的字母数字字段。但 InTouch 消息标记名则限制为 131 字符。如果消息型变量的字符长度超过数据库字段中指定的长度，该字符串在插入数据库时会被截短。

有效的数字类型是整型、实型或离散型标记名的数字，短整型（值范围在 -32,767 和 32,767 之间）只可用于整型和离散型标记名。不得指定字段长度。

支持的数据库的数据类型值

Oracle

数据类型	长度	缺省	范围	标记类型
字符	255 个字符	1 个字符		消息型
数字	38 位	38 位		整型

Sybase 或 Microsoft SQL Server

数据类型	长度	缺省	范围	标记类型
字符	255 个字符			消息型
整数			-2,147,483,647 to 2,147,483,647	整型
浮点	15 位		1.7E ⁻²⁹⁴ 到 1.7E ⁺³⁰⁸	实型

dBASE

数据类型	长度	缺省	范围	标记类型
字符	254 个字符			消息型
数字或 浮点	19 位			整型或实型

Microsoft Access 1.1 (Simba.DLL)

数据类型	长度	缺省	范围	标记类型
文本	254 个字符			消息型
整数				整型

Microsoft Access 2.0 (ODBCJT16.DLL), 7.0 (ODBCJT32.DLL)

数据类型	长度	缺省	范围	标记类型
文本	254 个字符			消息型
数字				整型
数字				实型

Paradox

数据类型	长度	缺省	范围	标记类型
字母数字	255 个字符			消息整型、实型或离散型
短			-32,767 到 32,767	整型或离散型

第 3 章

配置 SQL Access Manager

SQL Access Manager 实用程序可以建立绑定列表和表格模板。绑定列表将数据库列与 InTouch 标记名数据字典中的标记名相关联。表格模板则定义数据库中新表格的结构和格式。

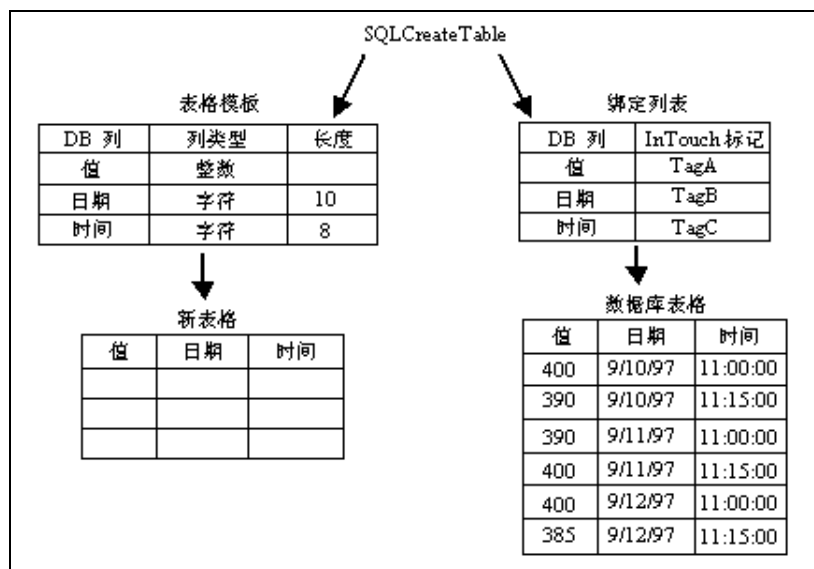
目录

- SQL Access Manager 概述
- 使用特殊分隔符
- 配置表格模板
- SQL.DEF 文件

SQL Access Manager 概述

当 InTouch 应用程序执行 **SQLCreateTable()** 命令时，表格模板参数会定义新数据库文件中的结构。

当 **SQLInsert()**、**SQLSelect()** 或 **SQLUpdate()** 执行时，绑定列表参数将决定所使用的 InTouch 标记名及所关联的数据库列。

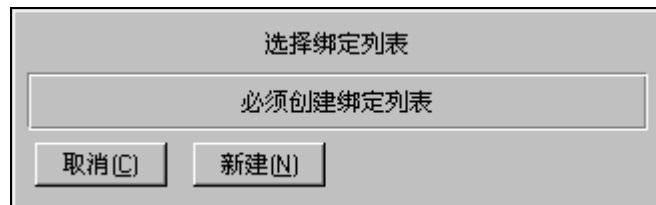


配置绑定列表

绑定列表将数据库列与 InTouch 标记名数据字典中的标记名相关联。

➤ 要建立新的绑定列表：

1. 在“特别”菜单，指向“SQL Access Manager”，然后单击“绑定列表”，或在“应用程序浏览器”的“SQL Access Manager”下，双击“绑定列表”。



2. 单击“新建”。
3. “绑定列表配置”对话框出现。



- ☞ 右击任意文本输入框，将出现一个菜单，显示您可应用到选定文本的命令。

4. 在“绑定列表名称”框中，输入绑定列表名称。

☞ 绑定列表名称最长可达 32 个字符。新绑定列表会将数据库列与 InTouch 标记名相链接。例如，如果生成了某个雇员人数统计列表，您可以在其中输入与雇员信息有关的绑定列表名称。

注意：SQLInsert()、SQLSelect() 和 SQLUpdate() 函数使用 Bind List 参数。在使用 SQLExecute() 函数时，标记名次序很重要，因为 SQLExecute() 函数将按所列顺序写入标记名。

5. 在“标记名.域名”框中，输入 InTouch 标记名点域名称。

☞ 标记名字典将标记名点域与数据库中的列名相关联。如果此标记名当前尚未在标记名字典中定义，双击该标记名，打开“标记名字典”对话框以现在进行定义。

6. 单击“标记名”以选择现有的标记名。标记浏览器出现。

☞ 标记浏览器会为当前选定的标记源显示标记名。要选择标记名，双击或选择它，然后单击“确定”。要选择标记名点域，单击“点域”箭头，选择您要使用的点域，然后单击“确定”。

注意：如果 I/O 型标记名未在应用程序中使用，但已在 SQLAccess 绑定列表中指定，则它会在 WindowViewer 启动时立即激活（提供给 I/O 服务器）。无需连接到数据库就可查看此行为。

有关标记浏览器的详细信息，请参阅您的联机《InTouch 用户指南》。

7. 单击“域名”以在标记名后添加 .field。“选择域名”对话框出现。

8. 单击您要使用的点域。对话框将关闭，.field 会自动地添加到 Tagname.FieldName 字段中的标记名后面。

有关标记名点域的详细信息，请参阅《InTouch 用户指南》中的第 4 章。

9. 在“列名”框中，输入列名称。

☞ 列名最长可为 30 个字符。列名与数据库列名直接关联，如果列名包含空格，请在绑定表及脚本中使用时将列名用方括号括起。例如：

```
WHERE EXPR= "[Pipe Flow} = " text (tagname,"#");
```

也可以使用特殊分隔符将列名与数据库关联。

有关特殊分隔符的详细信息，请参阅“使用特殊分隔符”。

10. 单击“向上移动”把列表中的所选标记名上移一级。
 11. 单击“向下移动”把列表中的所选标记名下移一级。
 12. 单击“添加项”以在绑定列表中添加新的标记名、点域和列名。
 13. 单击“删除项”以从绑定列表中删除选定的标记名、点域和列名。
 14. 单击“修改项”以修改该绑定表中选定的标记名、点域或列名。
 15. 单击“确定”以保存新的绑定列表配置并关闭对话框。
- ☞ 您可以单击“保存”来保存您的设置而不关闭对话框。

➤ 要修改绑定列表：

1. 在“特别”菜单，指向“SQL Access Manager”，然后单击“绑定列表”，或在“应用程序浏览器”的“SQL Access Manager”下，双击“绑定列表”。
2. “选择绑定列表”对话框出现。



3. 选择您要更改的绑定列表名称，然后单击“修改”。“绑定列表配置”对话框出现。
4. 修改所需项目。
5. 单击“确定”以保存更改并关闭对话框。

有关配置绑定列表的详细信息，请参阅“建立新的绑定列表”。

➤ 要删除绑定列表：

1. 在“特别”菜单，指向“SQL Access Manager”，然后单击“绑定列表”，或在“应用程序浏览器”的“SQL Access Manager”下，双击“绑定列表”。
2. “选择绑定列表”对话框出现。



3. 选择您要删除的绑定列表名。
4. 单击“删除”，屏幕出现一个要求您确认删除的对话框。单击“是”以删除选定的绑定列表，或单击“否”以取消删除。“绑定列表配置”对话框出现。
5. 单击“确定”关闭对话框。

使用特殊分隔符

SQLInsert() 和 **SQLUpdate()** 函数使用以单引号括起消息型字符串的缺省格式。某些 SQL 数据库需要收到被另一类型分隔符括起的消息字符串。例如，Oracle 需要收到被中括号括起的日期字符串。当出现这种情况时，必须使用 **Delim()** 函数，方法如下：

在“绑定列表配置”对话框的“列名”字段中，在列名后输入关键字“**delim**”（不分大小写）。关键字“**delim**”必须后跟：

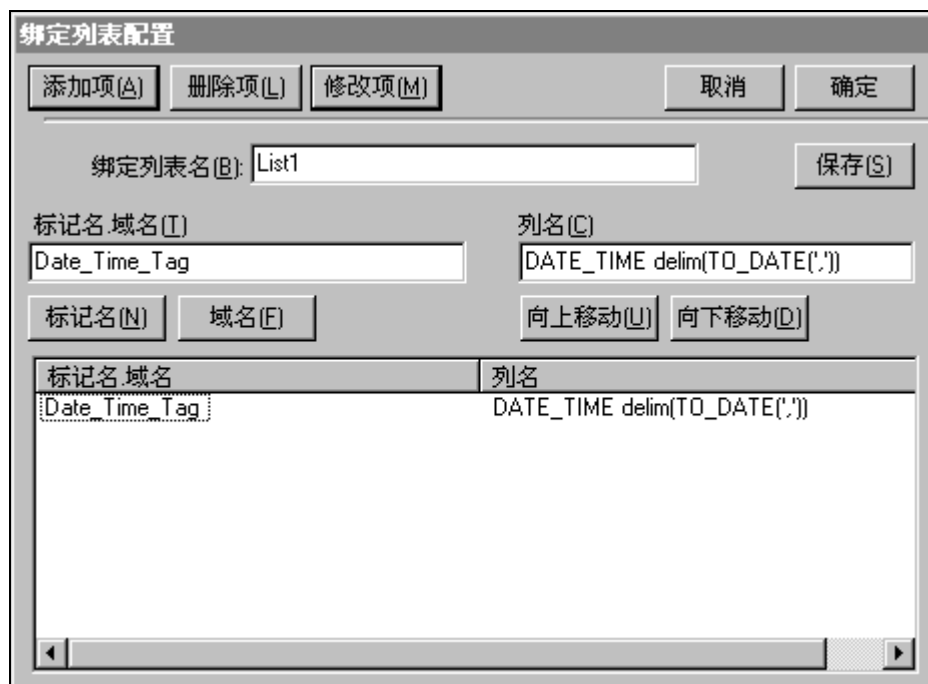
- 左括号
- 左分隔符
- 逗号
- 右分隔符
- 右括号

实例： **datestring delim (‘,‘)**

若要左右使用同一分隔符，只需在括号内指定没有逗号的分隔符。

实例： **datestring delim (‘,‘)**

下面的例子使用不同的左、右分隔符。注意此处的 **date delim (‘,’)** 在“列名”字段中输入：



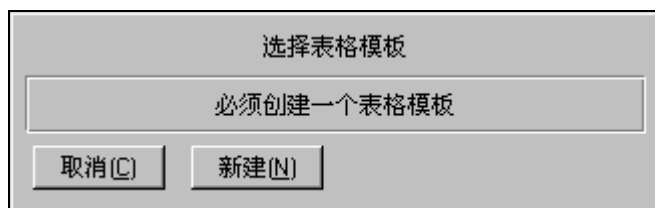
有关在 Oracle 日期字段中登录日期和时间的详细信息，请参阅第 2 章“使用 Oracle”。

配置表格模板

此命令可以建立一个用于定义数据库中的新表格结构和格式的表格模板。

➤ 要建立新的表格模板：

1. 在“特别”菜单上，指向“SQL Access Manager”，然后单击“表格模板”，或者在“应用程序浏览器”的“SQL Access Manager”下，双击“表格模板”。



2. 单击“新建”。
3. “表格模板配置”对话框出现。

列名	列类型	长度	允许输入空值
EmployeeID	Decimal	7.2	空值 NULL

- 右击任意文本输入框，将出现一个菜单，显示您可应用到选定文本的命令。

4. 在“表格模板名”对话框中，输入表格模板名称。

注意：表格模板名最长可为 32 个字符。如果您要创建一个唯一或非唯一的索引，表格模板名不得超过 **24 个字符**。表格模板名由 **SQLCreateTable()** 函数用于确定数据库的结构。

5. 在“列名”框中，输入表格模板的列名。列名最长可为 30 个字符。
6. 在“列类型”框中，输入列的数据类型。数据类型选择根据所使用的数据库而变化。

有关特定数据库的数据类型的详细信息，请参阅第 2 章“支持的数据库的数据类型值”。

7. 如下选择“索引类型”：

- | | |
|-----|----------------|
| 唯一 | 该列内的每个值必须是唯一的。 |
| 非唯一 | 该列内的每个值不必是唯一的。 |
| 无 | 没有索引。 |

TIP 当您执行 **SQLCreateTable()** 时，系统会自动创建一个索引文件。

8. 选择“允许输入空值”，以允许在此列中输入空数据。

注意：InTouch 不支持空数据。当插入数据时，如果某个标记名没有输入值，其值将如下指定：

- | | |
|-----|----------|
| 离散型 | 0 |
| 整型 | 0 |
| 消息型 | 没有字符的字符串 |

在选择数据时，空值将按照上述数据类型转换。

9. 单击“添加项”以在表格模板中添加新的列名、列类型、长度和索引类型。
10. 单击“删除项”以从表格模板中删除选定的列名、列类型、长度和索引类型。
11. 单击“修改项”以修改表格模板中的选定列名、列类型、长度和索引类型。
12. 单击“确定”以保存新表格模板配置并关闭对话框。
- ☞ 您可以单击“保存”来保存您的设置而不关闭对话框。

➤ 要修改表格模板：

1. 在“特别”菜单上，指向“SQL Access Manager”，然后单击“表格模板”，或者在“应用程序浏览器”的“SQL Access Manager”下，双击“表格模板”。
2. “选择表格模板”对话框出现。



3. 选择您要修改的表格模板名，然后单击“修改”。“表格模板配置”对话框出现。

修改所需项目。

单击“确定”以保存更改并关闭对话框。

有关配置表格模板的详细信息，请参阅“建立新表格模板”。

➤ 要删除表格模板：

1. 在“特别”菜单上，指向“SQL Access Manager”，然后单击“表格模板”，或者在“应用程序浏览器”的“SQL Access Manager”下，双击“表格模板”。
2. “选择表格模板”对话框出现。



3. 选择您要删除的表格模板名。
4. 单击“删除”，屏幕出现一个要求您确认删除的对话框。单击“是”以删除选定的绑定列表，或单击“否”以取消删除。“表格模板配置”对话框再次出现。
5. 单击“确定”关闭对话框。

SQL.DEF 文件

SQL Access Manager 会将绑定列表和表格模板的配置信息保存到一个名为“SQL.DEF”的文件中。此文件采用逗号分离变量 (CSV) 格式。您可以使用 SQL Access Manager 或任何文本编辑器（如记事本）来查看或修改 SQL.DEF 文件。数据按如下格式出现在文件中：

:BindListName,*BindListName*

Tagname1.FieldName,ColumnName1

Tagname2.FieldName,ColumnName2

Tagname3.FieldName,ColumnName3

:TableTemplateName,*TableTemplateName*

ColumnName1,ColumnType,[ColumnLength],Null,Index

ColumnName2,ColumnType,[ColumnLength],Null,Index

ColumnName3,ColumnType,[ColumnLength],Null,Index

第 4 章

使用 SQL 函数

InTouch 使用 SQL 函数与数据库中的信息通讯。这些函数是标准 InTouch QuickScript 函数的扩展并可用于任何脚本中。它们允许您选择、修改、插入或删除所选表格中的记录。

目录

- SQL 函数
- SQL 参数
- 在 InTouch QuickScript 中使用 SQL 函数

SQL 函数

本节列出所有 SQL 函数。记住 SQL 动作是同步的。在 SQL 活动完成之前，控制权不会返回到 InTouch（InTouch 趋势、轮询等将暂停）。

所有 SQL 函数（**SQLNumRows()** 除外）均会返回 *ResultCode*。如果 *ResultCode* 非零，则表示函数失败，您应采取其它措施。*ResultCode* 可由 **SQLErrorMsg()** 函数使用。

SQL 函数的通用格式如下：

SQLFunction(Parameter1, Parameter2, ...)

有关每个 SQL 函数及其使用实例的详细信息，请参阅《*InTouch 参考指南*》。

函数	描述
SQLAppendStatement	使用 <i>SQLStatement</i> 的内容来追加 SQL 语句。
SQLClearParam	清除指定参数的值。
SQLClearStatement	释放与 <i>SQLHandle</i> 指定的语句相关联的资源。
SQLClearTable	删除数据库表格中的所有记录，但保留表格。
SQLCommit	SQLCommit() 命令定义了一组操作命令的结束。
SQLConnect	将 InTouch 连接到 <i>ConnectionString</i> 中指定的数据库。
SQLCreateTable	利用指定的表格模板中的参数在数据库中建立表格。
SQLDelete	删除一条或多条记录。
SQLDisconnect	断开用户与数据库的连接。
SQLDropTable	删除表格。
SQLEnd	在 SQLSelect() 函数后使用，以释放存储结果表格所用的资源。
SQLErrorMsg	检索与特定 <i>ResultCode</i> 关联的文本错误消息。 <i>ErrorMsg</i> 是与 <i>ResultCode</i> 关联的 InTouch 内存消息型标记名（最长 131 个字符）。
SQLExecute	执行 SQL 语句。如果语句是 Select 语句， <i>BindList</i> 参数将指定绑定表的名称，以将数据库中的列与标记名关联。如果 <i>BindList</i> 为空，则没有形成标记名关系。
SQLFirst	选定由最后一个 SQLSelect() 函数创建的结果表格中的首条记录。
SQLGetRecord	从当前选择缓冲区中检索由 <i>RecordNumber</i> 指定的记录。

函数	描述
SQLInsert	根据提供的 <i>BindList</i> 中的标记名值，将一条新记录插入参考表中。 <i>BindList</i> 参数定义使用哪些 InTouch 标记名及与之关联的数据库列。
SQLInsertEnd	释放语句。
SQLInsertExecute	执行已预备的语句。
SQLInsertPrepare	创建并准备一个要执行的插入语句。
SQLLast	选择由最近一个 SQLSelect() 函数创建的结果表格中的最后一条记录。
SQLLoadStatement	读取 <i>FileName</i> 中包含的语句。在这一点止，此语句类似于由 SQLSetStatement() 创建的语句，并且能通过 SQLAppendStatement() 来追加记录，或由 SQLExecute() 执行。每个文件只允许包含一条语句。但是，如果尚未调用 SQLPrepareStatement() 或 SQLExecute() ，就可以用 SQLAppendStatement() 在语句中追加内容。
SQLManageDSN	运行 Microsoft ODBC 管理器配置程序。可用于添加、删除和修改所有的数据源名称。
SQLNext	选择由最后的 SQLSelect() 函数创建的结果表的下一条记录。
SQLNumRows	指示有多少行满足最后的 SQLSelect() 函数中指定的条件。
SQLPrepareStatement	SQLPrepareStatement() 预备由函数 SQLSetParam() 使用的现有 SQL 语句。可通过使用 SQLSetStatement() 或 SQLLoadStatement() 来建立语句。
SQLPrev	选择由最后的 SQLSelect() 函数创建的结果表的上一条记录。
SQLRollback	SQLRollback() 命令会反转或“回滚”最近提交的事务集。事务集是由 SQLTransact() 命令和 SQLCommit() 命令，或 SQLRollback() 命令之间发出的一组命令。事务集的处理方式和单个事务的一样。在发出 SQLTransact() 命令后，所有后续操作必须等待 SQLCommit() 命令发出后，才能提交给数据库。
SQLSelect	通知数据库从表格中检索信息。当执行 SQLSelect() 函数时，会在内存中创建一个临时结果表，其中含有可用 SQLFirst() 、 SQLLast() 、 SQLNext() 和 SQLPrev() 函数浏览的记录。
SQLSetParamChar	将指定参数的值设置为指定的字符串。 SQLSetParamChar() 可在执行前多次调用，从而使参数值设为全部发送值的串接。

函数	描述
SQLSetParamDate	将指定日期参数的值设为指定的字符串。
SQLSetParamDateTime	将指定日期-时间参数的值设为指定的字符串。
SQLSetParamDecimal	将指定的小数参数值设为指定的字符串。
SQLSetParamFloat	将指定的参数值设为指定的 <i>ParameterValue</i> 。
SQLSetParamInt	将指定的参数值设为指定的 <i>ParameterValue</i> 。
SQLSetParamLong	将指定的参数值设为指定的 <i>ParameterValue</i> 。
SQLSetParamNull	将指定的参数值设为 NULL。
SQLSetParamTime	将指定的时间参数值设为指定的字符串。
SQLSetStatement	在已建立的连接 <i>ConnectionID</i> 上，根据 <i>SQLStatement</i> 的内容创建一个 SQL 语句缓冲区。每个 <i>ConnectionID</i> 可以有一个 SQL 语句缓冲区。
SQLTransact	SQLTransact() 命令定义一组操作命令的开始。在 SQLTransact() 命令和 SQLCommit() 命令之间执行的一组命令称为事务集。事务集的处理方式和单个事务的一样。在发出 SQLTransact() 命令后，所有后续操作必须等待 SQLCommit() 命令发出后才能提交给数据库。
SQLUpdate	修改记录以用当前标记名值来更新记录。
SQLUpdateCurrent	取出当前所选记录并用任何新的 InTouch 值更新它。

SQL 参数

下面介绍每个 SQL 函数所需的参数。当脚本中输入的参数用引号括起时，例如 “Parameter1”，则程序将使用整个字符串。如果没有用引号括起，那么 Parameter1 将假定为标记名，系统就会从 Intouch 标记名字典访问标记名 Parameter1 的值。

实例：

"c:\main\file" vs. Location

此处：location 是一个 InTouch 消息型标记名。
"c:\main\file" is a literal string

大多数 SQL 函数所使用的参数为下列参数中的一个或多个：

参数	描述
<i>BindList</i>	对应 SQL.DEF 文件中的某个绑定列表名。
<i>ConnectionID</i>	由用户创建的内存整型标记名，用于存储由 SQLConnect 函数赋给每个数据库的编号 (ID)。
<i>ConnectionString</i>	SQLConnect() 用于识别数据库及任何附加登录信息的字符串。
<i>ErrorMsg</i>	包含错误消息的文本描述的消息变量。 有关错误消息描述的详细信息，请参阅第 5 章“错误排解”。
<i>FileName</i>	包含消息的文件名。
<i>MaxLen</i>	与此参数关联的最大列尺寸。此设置决定参数是可变字符类型还是长可变字符类型。如果 <i>MaxLen</i> 小于或等于数据库所允许的最大字符串，那么此参数为可变字符类型。否则为长可变字符类型。

参数	描述
<i>OrderByExpression</i>	定义排序的列和方向。只能使用列名进行排序。表达式格式必须为： ColumnName [ASC DESC] 按列名（如 manager），以升序对所选表格排序： "manager ASC" 按多列排序，表达式格式为： ColumnName [ASC DESC], ColumnName [ASC DESC] 某一列名（如 temperature）按升序，另一列名（如 time）按降序，对所选表格排序： "temperature ASC, time DESC"
<i>ParameterNumber</i>	语句中的实际参数数目。
<i>ParameterType</i>	指定参数的数据类型。有效值是：
类型	描述
字符	可填空格的固定长度字符串
可变字符	可变长度字符串
小数	BCD 数字
整型	4 字节有符号整数
小整数	2 字节有符号整数
浮点	4 字节浮点
双精度浮点	8 字节浮点
日期时间	8 字节日期时间值
日期	4 字节日期时间值
时间	4 字节日期时间值
无类型	无数据类型

参数	描述
<i>ParameterValue</i>	实际设置值。
<i>Precision</i>	小数值精度、字符最大大小，或日期时间值的字节长度。
<i>RecordNumber</i>	要检索的实际记录数值。
<i>ResultCode</i>	大多数 SQL 函数返回的整型变量。如果函数成功， <i>ResultCode</i> 返回为零 (0)，如果失败则返回一个负整数。 更关详细信息，请参阅第 5 章“错误排解”。
<i>Scale</i>	小数值比例。仅当适用参数设为空时，才需要使用此值。
<i>SQLHandle</i>	当使用高级功能语句时，SQL 返回供内部使用的 SQLHandle。
<i>SQLStatement</i>	实际语句，例如： ResultCode=SQLSetStatement(ConnectionID,"Select LotNo, LotName from LotInfo");
<i>TableName</i>	您要访问的数据库表格名称。
<i>TemplateName</i>	您要使用的模板定义名称。
<i>WhereExpression</i>	为表格的任意一行定义真假条件。此命令只从表格中提取条件为真的行。表达式必须为下列格式： ColumnName comparison_operator expression 注意：如果列为字符数据类型，则表达式必须用单引号括起。 下面的例子将选择其 name 列包含值 EmployeeID 的所有行： name='EmployeeID' 下面的例子将选择包含从 100 到 199 的 partno 的所有行： partno>=100 and partno<200 下面的例子将选择其 temperature 列包含大于 350 的值的行： temperature>350

在 InTouch QuickScript 中使用 SQL 函数

您可以在 InTouch QuickScript 中自动插入 SQL 函数，方法是单击 QuickScript 编辑器对话框中的“附加”按钮，SQL 函数会自动插入到脚本中的当前光标位置。

有关 InTouch QuickScript 的详细信息，请参阅《InTouch 用户指南》中的第 6 章“在 InTouch 中创建 QuickScript”。

指定复合查询

SQL Access Manager 允许您指定复合查询和您自己设计的 SQL 语句。这些查询可动态建立，也可包含在一个外部文件中。此外，这些查询可以包含“传递”到查询所需的参数，随后执行这些查询并可能返回结果集。SQL Access Manager 允许您执行数据库可处理的任何 SQL 语句，并检索该查询的结果。并且还可伴随执行存储过程（存储过程并不完全受支持）。

有关存储过程的详细信息，请参阅“支持存储过程”。

对于复合查询及超过 131 字符的字符串表达式，必须使用 SQLSetStatement 函数。当字符串表达式超过 131 个字符时，使用 SQLAppend。例如：

```
SQLSetStatement( Connect_Id, "Select Speed, Ser_No from  
tablename where Ser_No =' " + Serial_input + "'");
```

在下面的例子中，*SQLHandle* 设为零，这样在执行语句前就不必调用 **SQLPepare**(Connect_Id,SQLhandle)。由于 *SQLhandle* 不是由 **SQLPepare**() 建立的，要正确结束此选择，使用 **SQLEnd**() 函数，而不要使用 **SQLClearStatement**() 函数。

```
SQLExecute (Connect_Id,0);  
  
SQLSetStatement( Connect_Id, "Select Speed, Ser_No from  
tablename where Ser_No =' " + Serial_input + "'");  
  
SQLPrepareStatement (Connect_Id,SqlHandle);
```

在下面的例子中，*SQLHandle* 由 **SQLPrepareStatement**() 创建并用于 **SQLExceute**()函数中。要结束此选择语句，请使用 **SQLClearStatement**() 来释放资源和 *SqlHandle*。

```
SQLExecute (Connect_Id,SqlHandle);
```

动态建立查询

要动态地建立查询，需要用到两个附加函数：**SQLSetStatement()** 和 **SQLAppendStatement()**。**SQLSetStatement()** 启动一条新的 SQL 语句。这可以是任何有效的 SQL 语句，包括存储过程名。由于 InTouch 只支持 131 个字符的字符串，因而可以使用 **SQLAppendStatement()** 在语句中串联附加字符串。

注意：粗体文本指 SQL 查询语言命令。

实例

```
ResultCode = SQLSetStatement ( ConnectionID, "Select LotNo,
LotName, LotDescription, LotQuantity from LotInfo,
ProductionInfo" );

ResultCode = SQLAppendStatement (ConnectionID, " where
LotInfo.LotNo = ProductionInfo.LotNo" );

ResultCode = SQLAppendStatement (ConnectionID, " order by
LotNo, LotName, LotQuantity" );
```

现在可执行该语句。

注意：许多数据库列名和表名区分大小写。为让以上脚本正常工作，输入的列名和数据库名必须与数据库表格中的完全相同。

从文件中读取 SQL 语句

您可以在其它软件包，如 Microsoft Access 和其它第三方数据库工具中建立查询，然后用 InTouch 的 SQL Access 来执行查询。由于某些软件包会生成 SQL 语句，因此您可以使用 **SQLLoadStatement()** 轻易地调用该 SQL 语句并将其存储到文件中。

实例

```
ResultCode = SQLLoadStatement ( ConnectionID,
"c:\myappdir\lotquery.sql" );
```

现在可执行该语句。

修改扩展 SQL 语句

为提供完整的 SQL 功能，SQL Access Manager 允许您指定包含 InTouch 标记名值的 where 从句。为了能够在运行时指定 SQL 参数，系统提供了下列函数：

- SQLPrepareStatement()
- SQLSetParamType()
- SQLClearStatement()
- SQLClearParam()

要在 SQL 语句上执行参数替换，在需要以后指定参数的 SQL 语句位置放一个“?”号。在“预备”语句，并在语句中“设置”参数后，就可以执行该语句了。

SQLPrepareStatement() 预备语句的执行。它并不执行语句，而只是激活语句以便您设置参数值。**SQLSetParamType()** 是允许您设置 SQL 语句的参数值的函数集。

实例

```
ResultCode = SQLSetStatement ( ConnectionID, "Select LotNo,
LotName, LotDescription, LotQuantity from LotInfo,
ProductionInfo" );

ResultCode = SQLAppendStatement (ConnectionID, " where
LotInfo.LotNo = ?");

ResultCode = SQLAppendStatement (ConnectionID, " order by
LotNo, LotName, LotQuantity" );

ResultCode = SQLPrepareStatement (ConnectionID, SQLHandle );
{return the statement handle into tag 'SQLHandle'}

ResultCode = SQLSetParamInt ( SQLHandle, 1, tagLotNumber );
{put the value of tagLotNumber into param}
```

由于该语句只有一个参数，因而可以现在执行。

一旦执行语句并且您已完成预备语句，即可调用 **SQLClearStatement()** 来释放与该语句关联的资源。

注意：调用 **SQLEnd()** 可释放“未命名的”SQL 语句（那些由现有 SQL Access 函数生成的语句），以及由 **SQLSetStatement()** 和 **SQLLoadStatement()** 创建但未预备的参数。

执行扩展 SQL 语句

在动态建立语句或从文件中读取语句，并且有选择地预备和修改语句后，您就可以开始执行该语句了。SQL Access API 使用 **SQLExecute()** 函数来执行。**SQLExecute()** 会执行当前活动状态下的语句（即由 **SQLSetStatement()** 或 **SQLLoadStatement()** 生成的语句），或执行以前预备的且由语句句柄参数指定的语句。

实例 1:

```
ResultCode = SQLLoadStatement ( ConnectionID,
"c:\myappdir\lotquery.sql" );

ResultCode = SQLExecute (ConnectionID, "BindList", 0);
{将 Select 的结果置入 BindList 中的指定标记名，预备语句句柄为零}

ResultCode = SQLNext ( ConnectionID );
{取得 Select 结果}
```

实例 2:

```
ResultCode = SQLSetStatement ( ConnectionID, "Select LotNo,
LotName, LotDescription, LotQuantity from LotInfo,
ProductionInfo" );

ResultCode = SQLAppendStatement (ConnectionID, " where
LotInfo.LotNo = ?");
{问号表示会返回值}

ResultCode = SQLAppendStatement (ConnectionID, " order by
LotNo,NotName,LotQuantity" );

ResultCode = SQLPrepareStatement (ConnectionID, SQLHandle );
{返回语句句柄到标记 'SQLHandle'}

ResultCode = SQLSetParamInt ( SQLHandle, 1, tagLotNumber );
{将 tagLotNumber 值置入 param}

ResultCode = SQLExecute (ConnectionID, "BindList",
SQLHandle); {将 Select 的结果置入 BindList 中指定的标记名，预备语句句柄位于 SQLHandle 中}

ResultCode = SQLNext ( ConnectionID );
{取得 Select 结果}
```

实例 3:

SQLSetStatement – 此语句必须用于复合查询及超过 131 个字符的字符串表达式。当字符串表达式超过 131 个字符时，请使用 SQLAppend。

```
SQLSetStatement( ConnectionID, "Select  Speed, Ser_No from  
tablename where Ser_No =' " + Serial_input + "'");
```

```
SQLExecute(ConnectionID, "BindList", 0);
```

在上面的例子中，SQLHandle 设为零，这样在执行语句前就不必调用 SQLPepare(Connection_Id, SQLhandle)。由于 SQLhandle 不是由 SQLPepare 建立的，要正确结束此选择，使用 SQLEnd 函数，而不要使用 SQLClearStatement() 函数。

```
SQLSetStatement( Connection_Id, "Select  Speed, Ser_No from  
tablename where Ser_No =' " + Serial_input + "'");
```

```
SQLPrepareStatement(Connection_Id, SQLHandle);
```

```
SQLExecute(Connection_Id, SqlHandle);
```

在上面的例子中，SQLhandle 由 SQLPrepareStatement 创建并用在 SQLExceute 函数中。要结束此选择语句，请使用 SQLClearStatment 来释放资源和 SqlHandle。

支持存储过程

SQLExecute() 函数支持某些存储过程的执行。如，假设您在数据库服务器上创建了一个名为“LotInfoProc”的存储过程，它包含以下语句：“Select LotNo,LotName from LotInfo”。编写下面的 InTouch QuickScript 来执行程序并得到结果：

使用 Microsoft SQL Server

```
ResultCode = SQLSetStatement ( ConnectionID,
"LotInfoProc" );

ResultCode = SQLExecute(ConnectionID, "BindList", 0);

ResultCode = SQLNext ( ConnectionID );

{取得 Select 结果}
```

使用 Oracle 或 Microsoft Access

```
ResultCode = SQLSetStatement ( ConnectionID, "{CALL
LotInfoProc}" );

ResultCode = SQLExecute(ConnectionID, "BindList", 0);

ResultCode = SQLNext ( ConnectionID );

{取得 Select 结果}
```

第 5 章

错误排解

本章介绍如何使用由 SQL 函数返回的结果代码来排除 SQL 应用程序错误。第一节介绍 **SQLErrorMsg()** 函数，并列出 SQL 结果代码表及其相应的错误消息。第二节介绍包含特定数据库错误消息的消息表。

目录

- 函数错误排解
- 特定数据库错误消息

函数错误排解

所有 SQL 函数均会返回一个可用于除错的 *ResultCode*。**SQLException()** 函数会返回与 *ResultCode* 关联的错误消息。

实例：

```
ErrorMsg=SQLException(ResultCode);
```

此处：**ErrorMsg** 是内存消息型标记。
ResultCode 是从先前 SQL 函数获得的整型值。

结果码错误消息

对于此处没有记录的结果码，请参考您的特定数据库手册并确保检查 Wonderware Logger，以获得任何附加信息。

SQLException() 函数将设置 InTouch 消息标记 *ErrorMsg* 的值。下面列出可能出现的一些 SQL 结果码及其相应的错误消息和描述：

结果码	错误消息	描述
0	没有发生错误。	命令成功执行
-5	无更多的行可取用	已到达表格中的最后一条记录
-1001	内存不够	没有足够的内存来执行此函数
-1002	无效连接	传递给函数的 ConnectionId 无效
-1003	未找到绑定表	指定的绑定表名称不存在
-1004	找不到模板	指定的表格模板不存在。
-1005	内部错误	出现内部出错，请致电话“技术支持”。

结果码	错误消息	描述
-1006	字符串为空	警告 - 从数据库读取的字符串是空的。
-1007	字符串被截短	警告 - 从数据库中读取的字符串超过 131 个字符，所选字符串将被截短。
-1008	无 Where 子句	Delete 上没有 Where 子句。
-1009	连接失败	欲知关于连接失败的更多详细描述，参见 Wonderware 日志。
-1010	连接字符串的“DB=部分”上指定的数据库不存在	指定的数据库不存在。
-1011	没有选择行	未先执行 SQLSelect() 命令前，而试图调用 SQLNumRows() 、 SQLFirst() 、 SQLNext() 或 SQLPrev() 。
-4149	所给连接、语句或查询句柄无效	列类型可能定义错误。假如，如果表格模板定义为字符列类型，而不是 dBASE 文件字符，则会返回错误。

特定数据库错误消息

Oracle

错误消息	解决方法
ORA-03112 – 主机字符串语法错误	如果您没有运行 NETINIT.EXE ，请将它置入 Windows Startup 组中。如果您正在运行 NETINIT.EXE ，并希望建立一个以上的连接或会话，则需要分配内存。为此，将 CONFIG.ORA 文件中的 WIN_REMOTE_SESSIONS 参数值设置为所需的连接数。下面的例子分配 4 个连接的内存： WIN_REMOTE_SESSIONS=4
ORA-3121 没有连接接口驱动程序	在进入 Windows 并使用 InTouch SQL Access <u>之前</u> ，为网络系统适当启动的 SQL*NET TSR。
ORA-6435 - NetBIOS:不能添加本地名称到名称表格	必须运行网络软件（Novell、LAN Manager 等）
ORA-09301 -本地核心只在标准模式中被支持	在连接字符串中指定服务器名 ("SRVR=")
ORA-06430 -不能连接	确认连接字符串中的属性准确无误，并且次序正确。

Sybase 或 Microsoft SQL Server

错误消息	解决方法
您不能一次使用多条语句	在执行 SQLSelect() 后您正试图执行一个 SQL 命令。执行 SQLEnd() 以从 SQLSelect() 释放系统资源，或在第二条语句使用不同的 ConnectionId 。
处理命令的内存不够	尝试重新启动客户端工作站。
无效对象名称，表名称	表名称不存在于您当前使用的数据库中。尝试 DB=database name。

dBASE

错误消息	解决方法
文件或 DLL 没有发现	对于 Windows，QEDBF.DLL 必须位于您的当前目录下或 DOS 路径中的 \windows\system 目录下。
无效连接	确定您的路径中具有适当的 DLL。若要 DBF 支持，您将需要 QLDBF.DLL。
连接或所给语句句柄无效	有关详细消息，请查阅 Wonderware Logger，从中可能发现 SQL 语句中的更多语法错误。

附录 A

SQL Access 与 ODBC 的保留关键字

本附录列出了保留用于 SQL Access 绑定列表、表格模板及开放式数据库连接 (ODBC) 接口的关键字。

如果某个保留关键字被用作绑定列表或表格模板中的列名，则 WWLogger 会产生一条错误信息。产生的错误类型取决于所使用的 ODBC 驱动程序和关键字的位置。例如，最常见的错误之一是使用 DATE 和 TIME 作为绑定列表和表格模板中的列名。为避免出现此类错误，请使用稍微不同的名称，如 “aDATE” 和 “aTIME”。

这些保留的关键字定义了 InTouch SQL Access 所使用的结构化查询语言 (SQL)。关键字也可以由所使用的特定 ODBC 驱动程序识别。SQL Access 将包含一个或多个保留关键字的 SQL 命令传送给 ODBC.DLL 文件。如果 SQL 命令不能被正确解译，SQL Access 会在 Wonderware Logger 中产生一条错误信息。

保留关键字按字母顺序列出如下：

ABSOLUTE	CHECK	ENTF
ADA	CLOSE COALESCE	DESC
ADD	COBOL	DESCRIBE
ALL	COLLATE	DESCRIPTOR
ALLOCATE	COLLATION	DIAGNOSTICS
ALTER	COLUMN	DICTIONARY
AND	COMMIT	DISCONNECT
ANY	CONNECT	DISPLACEMENT
ARE	CONNECTION	DISTINCT
AS	CONSTRAINT	DOMAIN
ASC	CONSTRAINTS	DOUBLE
ASSERTION	CONTINUE	DROP
AT	CONVERT	ELSE
AUTHORIZATION	CORRESPONDING	ENDEESC
AVG	COUNT	EXCEPT
BEGIN	CREATE	EXCEPTION
BETWEEN	CURRENT	EXEC
BIT	CURRENT_DATE	EXECUTE
BIT_LENGTH	CURRENT_TIME	EXISTS
BY	CURRENT_TIMESTAMP	EXTERNAL
CASCADE	CURSOR	EXTRACT
CASCADED	DATE	FALSE
CASE	DAY	FETCH
CAST	DEALLOCATE	FIRST
CATALOG	DEC	FLOAT
CHAR	DECIMAL	FOR FOREIGN
CHAR_LENGTH	DECLARE	FORTRAN
CHARACTER	DEFERRABLE	FOUND
CHARACTER_LENGTH	DEFERRED	FROM FULL

GET	NONE	SQLWARNING
GLOBAL	NOT	SUBSTRING
GO	NULL	SUM
GOTO	NULLIF	SYSTEM
GRANT	NUMERIC	TABLE
GROUP	OCTET_LENGTH	TEMPORARY
HAVING	OF	THEN
HOURL	OFF	TIME
IDENTITY	ON	TIMESTAMP
IGNORE	ONLY	TIMEZONE_HOUR
IMMEDIATE	OPEN	TIMEZONE_MINU
IN	OPTION	TO
INCLUDE	OR	TRANSACTION
INDEX	ORDER	TRANSLATE
INDICATOR	OUTER	TRANSLATION
INITIALLY	OUTPUT	TRUE
INNER	OVERLAPS	UNION
INPUT	PARTIAL	UNIQUE
INSENSITIVE	PASCAL	UNKNOWN
EINFÜGEN	PLI	UPDATE
INTEGER	POSITION	UPPER
INTERSECT	PRECISION	USAGE
INTERVALL	PREPARE	OPRN
INTO	PRESERVE	USING
IS	PRIMARY	WERT
ISOLATION	PRIOR	VALUES
JOIN	PRIVILEGES	VARCHAR
KEY	PROCEDURE	VARING
LANGUAGE	PUBLIC	VIEW
LAST	RESTRICT	WHEN
NACH-LINKS	REVOKE	WHENEVER
LEVEL	NACH RECHTS	WHERE
LIKE	ROLLBACK	WITH
LOCAL	ROWS	WORK
LOWER	SCHEMA	YEAR
MATCH	SCROLL	
MAX	SECOND	
MIN	SECTION	
MINUTE	SELECT	
MODULE	SEQUENCE	
MONTH	SET	
MUMPS	SIZE	
NAMES	SMALLINT	
NATIONAL	SOME	
NCHAR	SQL	
NEXT	SQLCA	
	SQLCODE	
	SQLERROR	
	SQLSTATE	

索引

B

BindListName, 3-11,4-5

C

ConnectionID, 4-5

ConnectionString, 4-5

CSV, 1-2,3-11

D

dBASE

支持的数据类型, 2-11,2-14

连接要求, 2-10

DBNMP3.DLL, 2-8

E

ErrorMsg, 4-5

F

FileName, 4-5

M

MaxLen, 4-5

Microsoft Access

支持的数据类型, 2-14,2-15

支持的数据类型, 2-12

连接要求, 2-12

Microsoft SQL Server

支持的数据类型, 2-9

连接要求, 2-8

配置客户端, 2-8

N

NETINIT.EXE 程序, 2-2

O

ODBC Compliant, 1-6

ODBC 管理员程序, 1-6

ODBC.INI, 1-6

Oracle

支持的数据类型, 2-14

Oracle 6, 2-2

支持的数据类型, 2-3

启动 SQL*Net TSR 和 NETINIT.EXE, 2-2

连接要求, 2-2

配置 Windows 数据库客户端, 2-2

Oracle 6 支持的数据类型, 2-3

Oracle 7, 2-4

配置数据源名称, 2-5

OrderByExpression, 4-6

P

Paradox

支持的数据类型, 2-13,2-15

连接要求, 2-13

Parameter

BindListName, 4-5

ParameterNumber, 4-6

ParameterType, 4-6

ParameterValue, 4-7

Precision, 4-7

Q

QuickScripts, 4-8

R

RecordNumber, 4-7

ResultCode, 4-7,5-2

S

Scale, 4-7

SQL Access Manager 概述, 3-2

SQL Access Manager 简介, 1-2

SQL 函数格式, 4-2

SQL 函数疑难排除, 5-2

SQL 参数, 4-5

SQL.DEF, 1-2,3-11

SQLAppendStatement, 4-2

SQLClearParam, 4-2

SQLClearStatement, 4-2

SQLClearTable, 4-2

SQLCommit, 4-2

SQLConnect, 2-2,2-8,4-2

SQLCreateTable, 4-2

SQLDelete, 4-2

SQLDisconnect, 4-2

SQLDropTable, 4-2

SQLEnd, 4-2

SQLErrorMsg, 4-2,5-2

SQLExecute, 4-2

SQLFirst, 4-2

SQLGetRecord, 4-2

SQLHandle, 4-7

SQLInsert, 3-7,4-2

SQLInsertEnd, 4-3

SQLInsertExecute, 4-3

SQLInsertPrepare, 4-3

SQLLast, 4-3

SQLLoadStatement, 4-3

SQLManageDSN, 4-3

SQLNext, 4-3

SQLNumRows, 4-3
SQLPrepareStatement, 4-3
SQLPrev, 4-3
SQLRollback, 4-3
SQLSelect, 4-3
SQLSetParamChar, 4-3
SQLSetParamDate, 4-3
SQLSetParamDateTime, 4-3
SQLSetParamDecimal, 4-4
SQLSetParamFloat, 4-4
SQLSetParamInt, 4-4
SQLSetParamLong, 4-4
SQLSetParamNull, 4-4
SQLSetParamTime, 4-4
SQLSetStatement, 4-4
SQLStatement, 4-7
SQLTransact, 4-4
SQLUpdate, 3-7,4-4
SQLUpdateCurrent, 4-4
Sybase
 支持的数据类型, 2-9,2-14
 连接要求, 2-8
 配置客户端, 2-8
Sybase 或 Microsoft SQL Server
 支持的数据类型, 2-14

T

Table Template Name, 3-11
TableName, 4-7
Tagname.FieldName, 3-4
TemplateName, 4-7

W

WhereExpression, 4-7
Wonderware 技术支持, 1-4

三划

与 Oracle 6 通讯, 2-2

四划

从文件中读取 SQL 语句, 4-9
分隔函数, 3-7
分隔符, 3-7
支持存储过程, 4-13
支持的数据库
 dBASE, 2-10
 Microsoft Access, 2-12
 Microsoft SQL Server, 2-8
 Oracle 6, 2-2
 Paradox, 2-13
 Sybase, 2-8
支持的数据类型, 2-14

六划

关于本手册, 1-3
列名, 3-5
动态建立查询, 4-9
在 InTouch 中使用 SQL 函数, 4-8
在 Oracle 日期字段中记录日期和时间, 2-7
执行扩展 SQL 语句, 4-11

七划

删除表格模板, 3-10
删除绑定列表, 3-6

八划

使用 Microsoft Access, 2-12
使用 Oracle 7.2, 2-4
使用 Paradox, 2-13
使用 SQL 函数, 4-1
使用dBASE, 2-10
使用Sybase或Microsoft SQL Server, 2-8
使用特殊分隔符, 3-7
函数

 SQLAppendStatement, 4-2
 SQLClearParam, 4-2
 SQLClearStatement, 4-2
 SQLClearTable, 4-2
 SQLCommit, 4-2
 SQLConnect, 2-2,2-8,4-2
 SQLCreateTable, 4-2
 SQLDelete, 4-2
 SQLDisconnect, 4-2
 SQLDropTable, 4-2
 SQLEnd, 4-2
 SQLErrorMsg, 4-2
 SQLExecute, 4-2
 SQLFirst, 4-2
 SQLGetRecord, 4-2
 SQLInsert, 4-2
 SQLInsertEnd, 4-3
 SQLInsertExecute, 4-3
 SQLInsertPrepare, 4-3
 SQLLast, 4-3
 SQLLoadStatement, 4-3
 SQLManageDSN, 4-3
 SQLNext, 4-3
 SQLNumRows, 4-3
 SQLPrepareStatement, 4-3
 SQLPrev, 4-3
 SQLRollback, 4-3
 SQLSelect, 4-3
 SQLSetParamChar, 4-3
 SQLSetParamDate, 4-3
 SQLSetParamDateTime, 4-3
 SQLSetParamDecimal, 4-4
 SQLSetParamFloat, 4-4

- SQLSetParamInt, 4-4
- SQLSetParamLong, 4-4
- SQLSetParamNull, 4-4
- SQLSetParamTime, 4-4
- SQLSetStatement, 4-4
- SQLTransact, 4-4
- SQLUpdate, 4-4
- SQLUpdateCurrent, 4-4

参数, 4-5

- ConnectionID, 4-5
- ConnectString, 4-5
- ErrorMsg, 4-5
- FileName, 4-5
- MaxLen, 4-5
- OrderByExpression, 4-6
- ParameterNumber, 4-6
- ParameterType, 4-6
- ParameterValue, 4-7
- Precision, 4-7
- RecordNumber, 4-7
- ResultCode, 4-7
- Scale, 4-7
- SQLHandle, 4-7
- SQLStatement, 4-7
- TableName, 4-7
- TemplateName, 4-7
- WhereExpression, 4-7

命令

- 表格模板, 3-9
- 表格模板, 3-8
 - 删除, 3-10
 - 命令, 3-9
 - 修改, 3-10
 - 新建, 3-8
- 表格模板名, 3-9

九划

- 保留关键字, A-1
- 修改扩展 SQL 语句, 4-10
- 修改表格模板, 3-10
- 修改绑定列表, 3-6
- 指定复合查询, 4-8
- 查询
 - Complex, 4-9
 - 动态建立, 4-9
- 标记浏览器, 3-4
- 绑定列表, 3-3
 - 删除, 3-6
 - 修改, 3-6
 - 标记浏览器, 3-4
 - 新建, 3-3
- 结构化查询语言, 1-1
- 结果码错误消息, 5-2

十划

- 特定数据库错误消息
 - dBASE, 5-5
 - Oracle, 5-4
 - Sybase 或 Microsoft SQL Server, 5-5
- 配置 SQL Access Manager, 3-1
- 配置和连接数据库, 2-1
- 配置表格模板, 3-8
- 配置绑定列表, 3-3

十一划

- 您的 FactorySuite 许可证, 1-5

十二划

- 联机手册, 1-3

十三划

- 数据库, 1-1
- 错误排解, 5-1

