

The Embarcadero logo, featuring a red circle with a white lowercase 'e' inside, followed by the word 'embarcadero' in white lowercase letters with a registered trademark symbol.

embarcadero[®]



**C++ Builder 10.1 Berlin
FireDAC**

数据库开发手册

New! 支持 No-SQL 数据库 - MongoDB
更新驱动程序

序

FireDAC 应该算是 C++Builder 从 Borland 时代开始的第 3 代数据存取技术和框架了, C++Builder 历经了 BDE/IDAPI, dbExpress 一直到现在的 FireDAC, 这也代表了 C++Builder 从桌面开发, C/S, Web, Multi-Tier 转变到现在着重跨平台和移动开发的需求演进。

FireDAC 是第一个完全由 C++Builder 程序语言撰写的数据库存取框架, 以前的 BDE/IDAPI 和 dbExpress 是混合了 Object Pascal 和 C 语言撰写的数据库存取框架, 因此随着 C++Builder 程序语言支持多平台, FireDAC 也可以轻易的在多个平台中执行。但除了程序语言的原因之外, 到底为什么要使用 FireDAC 来取代 BDE/IDAPI 和 dbExpress 呢?

最主要的原因就是这 BDE/IDAPI, dbExpress 和 FireDAC 设计的目标和架构, BDE/IDAPI 在近 20 年前设计的目标是让 C++Builder 在桌面和稍后出现的 C/S 架构中使用, 而 dbExpress 设计的目标则是让 C++Builder 除了能够在原本的桌面和 C/S 架构中使用之外, 也能够 Web 和 Multi-Tier 架构中使用。但随着移动和穿戴式设备的出现和开发的需求, C++Builder 也需要一个能够适用在所有平台的数据存取技术和框架, 而 FireDAC 正好能够满足这个目标和需求。

FireDAC 的功能其实非常类似 BDE/IDAPI 和 dbExpress 的结合体, 它在使用上非常接近 BDE/IDAPI, 但又具备 dbExpress 的联机存取和脱机数据处理的能力, 再加上 FireDAC 不需要部署额外 DLL 档案而能够直接连结客户端程序代码的特性以及精简型数据集的功能, 让 FireDAC 也非常适合使用在移动和穿戴式设备的应用。因为如果您需要使用 C++Builder 开发任何需要处理数据的应用程序, 那么您绝对应该认真考虑使用 FireDAC。

本书的目的是希望让读者能够快速学习和使用 FireDAC 来开发 C++Builder 的数据库应用程序, 希望在您阅读完本书的内容之后就具备了足够的知识和技术善用 FireDAC 开发出跨平台的数据库应用程序。

目录

第 1 章 开始学习使用 FireDAC 开发数据库应用程序吧	8
1-1 使用 FireDAC 链接数据库	9
1-1-1 链接数据库的方式	19
使用组态档	19
1-1-2 直接使用程序代码	24
1-2 处理数据	26
1-2-1 主从关连资料	26
1-2-1-1 使用客户端范围机制	27
1-2-1-2 使用伺服器端动态查询机制	29
1-3 开发移动数据库 App	30
1-3-1 开发和部署 iOS/Android 手机 App	31
1-3-2 直接在 iOS/Android 手机中建立数据库	39
1-4 结论	45
第 2 章 处理数据	46
2-1 使用 Array DML 处理大量数据	46
2-2 搜寻数据	50
2-2-1 Locate 和 LocateEx	51
Locate 单字段搜寻	54
Locate 多字段搜寻	56
使用 LocateEx 搜寻数据	58
2-2-2 Lookup 和 LookupEx	62
单字段搜寻	63

多字段搜寻	64
使用 LookupEx	65
2-2-3 在客户端动态排序	68
2-2-4 使用过滤器	72
使用过滤器的场合	75
2-2-5 使用 SetRange	76
2-2-6 使用 FireDAC 在手机中搜寻数据	76
2-3 快储机制	77
2-3-1 使用 FireDAC 快储功能	83
SavePoint	92
RevertRecord 方法	95
CommitUpdates 方法	96
UndoLastChange 方法	97
2-3-2 处理 FireDAC 快储更新错误	99
2-3-3 处理 FireDAC 快储执行效率	104
2-4 监督数据处理	104
2-5 在移动平台使用快储功能	108
2-6 结论	109
第 3 章 使用内存数据组件	110
3-1 使用 TFDMemTable	110
3-1-1 使用 TFDMemTable 组件提供快速查询	111
3-1-2 使用 TFDMemTable 处理 SOAP/REST 取得的数据	117
3-1-3 使用 TFDMemTable 处理数据	123

3-2 结论	127
第 4 章 FireDAC 进阶功能	128
4-1 存取 MetaData	128
4-1-1 使用 TFDConnection 组件存取 MetaData.....	128
4-1-2 使用 TFDMetaInfoQuery 组件存取 MetaData	129
4-2 宏功能(Marco).....	133
4-3 Update SQL 处理客制化数据	141
4-3-1 使用 TUpdateSQL 组件产生 DML	142
4-3-2 使用 TUpdateSQL 组件客制化数据更新	145
4-3-3 使用 OnUpdateRecord 事件客制化数据更新	149
4-3-4 使用 TUpdateSQL 组件处理复杂数据更新	151
4-4 异步处理数据.....	158
4-5 结论	165
第 5 章 FireDAC 更多的功能	166
5-1 批处理	166
5-2 控制数据的显示和更新.....	169
5-3 数据转换	180
5-3-1 数据换文字格式.....	181
5-3-2 在不同数据源中转换数据	185
5-4 处理自动增加值字段(Auto-Increment Field).....	188
5-5 使用计算字段.....	195
5-6 结论	201
第 6 章 MongoDB 数据库开发.....	202

6-1 MongoDB 的基本介绍	202
6-2 下载和安装 MongoDB.....	204
6-3 FireDAC 对 MongoDB 的支援.....	207
6-3-1 使用 C++Builder 类别处理 MongoDB	208
存取 TMongoConnection 和 TMongoEnv 物件.....	209
6-3-2 使用 FireDAC 组件处理 MongoDB.....	215
6-3-3 使用 TMongoQuery 搜寻数据	219
第 7 章 开发第 1 个实时数据系结应用程序	225
7-1 开发第一个 FireMonkey 数据库应用程序	227
7-1-1 浅尝系结表达式	242
7-2 使用 TBindSourceDBX 组件.....	247
7-3 使用 TPrototypeBindSource 组件	250
7-4 结论	259
第 8 章 更多的实时数据系结技术.....	260
8-1 使用实时数据系结技术的 Lookup 功能.....	260
8-2 什么是实时数据系结	267
简单的系结表达式(Simple Expressions)	269
拖管系结表达式(Managed Bindings)	269
未拖管系结表达式(Unmanaged Bindings)	269
8-3 进阶 Lookup 功能.....	270
8-4 结论	277
第 9 章 实时数据系结框架	278
9-1 建立实时数据系结概念.....	278

9-1-1 使用 TBindExpression 组件	290
9-1-2 未拖管系结表达式	295
9-1-3 拖管系结表达式	301
9-2 数据型态转换函式	303
9-3 实时数据系结相关的类别	306
9-3-1 使用 TBindExprItems 类别	308
步骤 1-系结 TListBox 和 TEdit 组件	309
步骤 2-系结 TListBox 和 TTrackBar 组件	311
步骤 3-系结 TTrackBar 和 TEdit 组件	315
步骤 4-系结 TEdit 和 TTrackBar 组件	315
9-4 TBindingsList 提供的可呼叫方法	320
9-5 系结编辑器，观察元和系结范例组件	323
9-6 使用实时数据系结设定	325
9-7 TBindScope 组件	329
9-8 结论	333