

# 序

FireDAC 应该算是 Delphi 从 Borland 时代开始的第 3 代数据存取技术和框架了, Delphi 历经了 BDE/IDAPI, dbExpress 一直到现在的 FireDAC, 这也代表了 Delphi 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 Delphi 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 Delphi 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 Delphi 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 Delphi 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, Delphi 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 Delphi 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 Delphi 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

1-1 使用 FireDAC 链接数据库 .....	9
1-1-1 链接数据库的方式 .....	18
使用组态档 .....	19
1-1-2 直接使用程序代码 .....	23
1-2 处理数据 .....	25
1-2-1 主从关连资料 .....	25
1-2-1-1 使用客户端范围机制 .....	26
1-2-1-2 使用伺服器端动态查询机制 .....	28
1-3 开发移动数据库 App .....	29
1-3-1 开发和部署 iOS/Android 手机 App .....	30
1-3-2 直接在 iOS/Android 手机中建立数据库 .....	35
1-4 结论 .....	39
2-1 使用 Array DML 处理大量数据 .....	40
2-2 搜寻数据 .....	44
2-2-1 Locate 和 LocateEx .....	46
Locate 单字段搜寻 .....	48
Locate 多字段搜寻 .....	49
使用 LocateEx 搜寻数据 .....	51
2-2-2 Lookup 和 LookupEx .....	54
单字段搜寻 .....	55
多字段搜寻 .....	56
使用 LookupEx .....	57

2-2-3 在客户端动态排序 .....	59
2-2-4 使用过滤器 .....	64
使用过滤器的场合 .....	66
2-2-5 使用 SetRange .....	67
2-2-6 使用 FireDAC 在手机中搜寻数据 .....	68
2-3 快储机制 .....	69
2-3-1 使用 FireDAC 快储功能 .....	74
SavePoint .....	82
RevertRecord 方法 .....	85
CommitUpdates 方法 .....	86
UndoLastChange 方法 .....	87
2-3-2 处理 FireDAC 快储更新错误 .....	89
2-3-3 处理 FireDAC 快储执行效率 .....	93
2-4 监督数据处理 .....	94
2-6 结论 .....	98
3-1 使用 TFDMemTable .....	100
3-1-1 使用 TFDMemTable 组件提供快速查询 .....	101
3-1-2 使用 TFDMemTable 处理 SOAP/REST 取得的数据 .....	107
3-1-3 使用 TFDMemTable 处理数据 .....	112
3-2 结论 .....	116
4-1 存取 MetaData .....	118
4-1-1 使用 TFDCONNECTION 组件存取 MetaData .....	118
4-1-2 使用 TFDMetaInfoQuery 组件存取 MetaData .....	119

4-3 Update SQL 处理客制化数据 .....	131
4-3-1 使用 TUpdateSQL 组件产生 DML .....	131
4-3-2 使用 TUpdateSQL 组件客制化数据更新 .....	135
4-3-3 使用 OnUpdateRecord 事件客制化数据更新 .....	138
4-3-4 使用 TUpdateSQL 组件处理复杂数据更新 .....	140
4-4 异步处理数据 .....	147
4-5 结论 .....	153
5-1 批处理 .....	154
5-2 控制数据的显示和更新 .....	157
5-3 数据转换 .....	168
5-3-1 数据换文字格式 .....	169
5-3-2 在不同数据源中转换数据 .....	173
5-4 处理自动增加值字段(Auto-Increment Field) .....	176
5-5 使用计算字段 .....	183
5-6 结论 .....	189
6-1 开发第一个 FireMonkey 数据库应用程序 .....	193
6-1-1 浅尝系结表达式 .....	208
6-2 使用 TBindSourceDBX 组件 .....	213
6-3 使用 TPrototypeBindSource 组件 .....	216
6-4 结论 .....	225
7-1 使用实时数据系结技术的 Lookup 功能 .....	226
7-2 什么是实时数据系结 .....	233
简单的系结表达式(Simple Expressions) .....	235

拖管系结表达式(Managed Bindings) .....	235
未拖管系结表达式(Unmanaged Bindings) .....	235
7-3 进阶 Lookup 功能.....	236
7-4 结论 .....	242
8-1 建立实时数据系结概念.....	243
8-1-1 使用 TBindExpression 组件.....	253
8-1-2 未拖管系结表达式 .....	258
8-1-3 拖管系结表达式.....	263
8-2 数据型态转换函式 .....	264
8-3 实时数据系结相关的类别.....	268
8-3-1 使用 TBindExprItems 类别 .....	269
步骤 1-系结 TListBox 和 TEdit 组件.....	271
步骤 2-系结 TListBox 和 TTrackBar 组件.....	272
步骤 3-系结 TTrackBar 和 TEdit 组件 .....	276
步骤 4-系结 TEdit 和 TTrackBar 组件 .....	277
8-4 TBindingsList 提供的可呼叫方法 .....	281
8-5 系结编辑器, 观察元和系结范例组件.....	284
8-6 使用实时数据系结设定.....	286
8-7 结论 .....	290
9-1 TPrototypeBindSource 组件和维型开发.....	291
9-2 系结引擎 Adapter 类别.....	298
9-2-1 TObjectBindSourceAdapter 类别.....	300
9-2-2 TListBindSourceAdapter 类别 .....	302

9-3 执行范围组件- TBindScope.....307

9-4 结论 ..... 316

\* →