



Lesson 3: Accessing Local Storage and Databases

David Intersimone "David I"
Vice President of Developer Relations and Chief Evangelist
davidi@embarcadero.com

Mobile App Development

- Lesson 1 – Hello World! My First Multi-Device App
- Lesson 2 – Turning up the Style and Data!
- **Lesson 3 – Accessing Local Storage and Databases**
- Lesson 4 – Building Multi-tier, Multi-device Apps
- Lesson 5 – Connecting Mobile and Desktop using Tethering
- Lesson 6 – Accessing REST and BaaS Cloud Services

Replay links and lesson slides will appear on my blog
<http://blogs.embarcadero.com/davidi/>

Lesson 3 Agenda

- Mobile application local file system access
- INI files
- FireDAC – SQL databases and in-memory datasets
- InterBase®
- Samples and Snippets
- Continue development of the mobile business app
- Review, Homework and Next Time
- Q&A

Mobile application local file system access

- System.IOUtils, System::Ioutils
- Contains a large number of directory, file and path methods
 - GetHomePath
 - GetDocumentsPath
 - GetDirectories, GetFiles
 - Exists, Delete, Copy
- Use device APIs and 3rd party components to get to other non-app file areas
 - Music – look at Mobile Samples | Media | Music Player

```
{$IF DEFINED(IOS) or DEFINED(ANDROID)}
FireTaskList.Params.values['Database'] := TPath.GetDocumentsPath + PathDelim + 'TASKS.GDB';
{$ENDIF}
#if defined(TARGET_OS_IPHONE) || defined(__ANDROID__)
FireTaskList->Params->values["Database"] = System::Ioutils::TPath::GetDocumentsPath() + PathDelim + "TASKS.GDB";
#endif
```

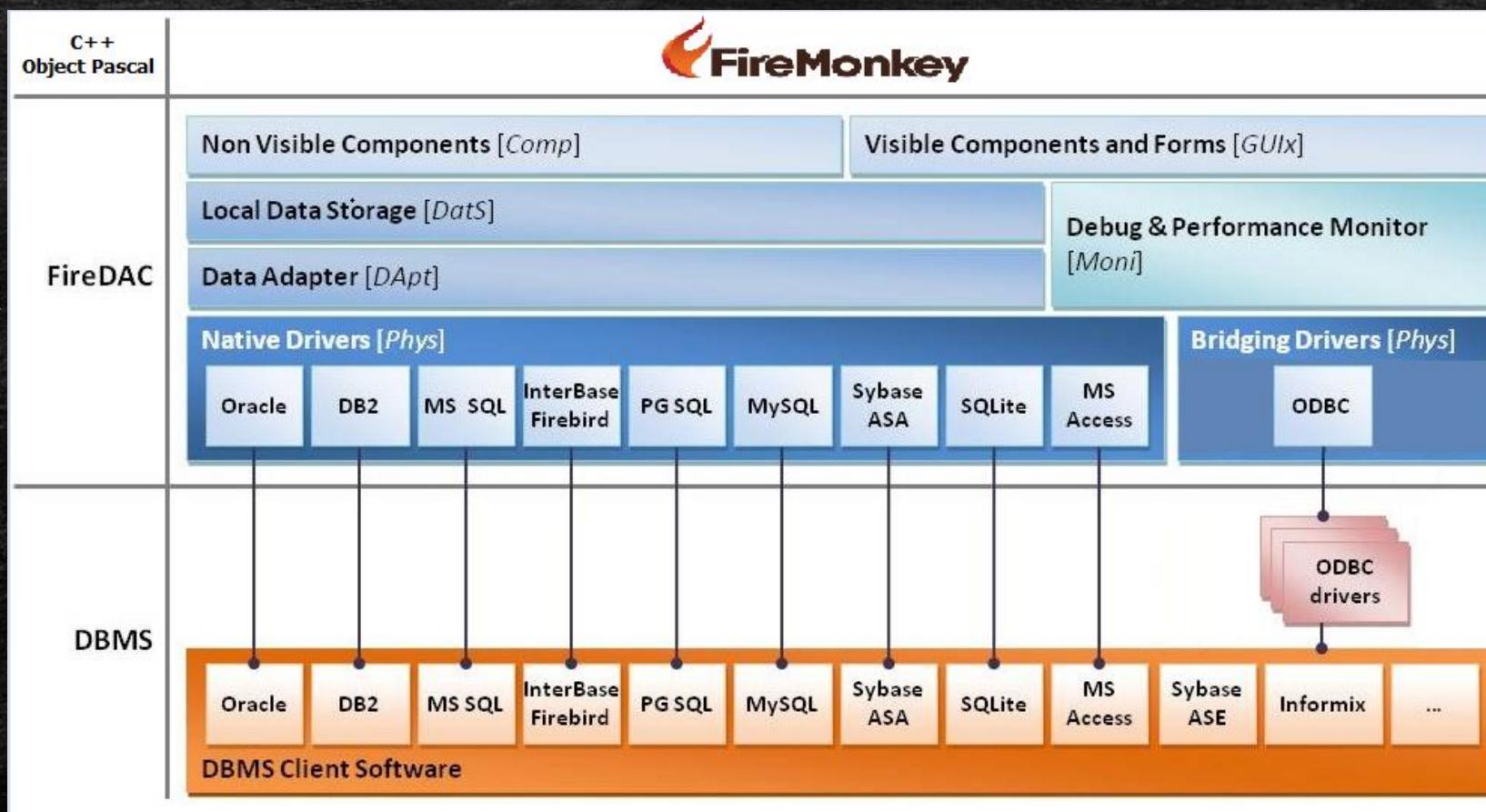
INI files

- Used to store and retrieve application-specific information and settings
- Stores information in logical groupings, called sections: [Example Section Name]
- Within each section, actual data values are stored in named keys. Keys take the form: <keyname> = <value>
- The FileName is passed to the TIniFile constructor and identifies the INI file that the object accesses
- Methods are provided for working with INI files
 - ReadString / WriteString
 - ReadInteger / WriteInteger, ReadFloat / WriteFloat
 - ReadBool / WriteBool, ReadBinaryStream / WriteBinaryStream
 - ReadDate / WriteDate, ReadTime / WriteTime
 - ReadSection, ReadSections, SectionExists, EraseSection, DeleteKey, UpdateFile, ValueExists

```
[MASTSQL]
Database=C:\Users\Public\Documents\Embarcadero\Studio\14.0\Samples\Data\MASTSQL.GDB
User_Name=SYSDBA
Password=masterkey
DriverID=IB
```

FireDAC – Multi-Device Data Access Library

- Direct, high speed access to mobile, desktop and enterprise databases

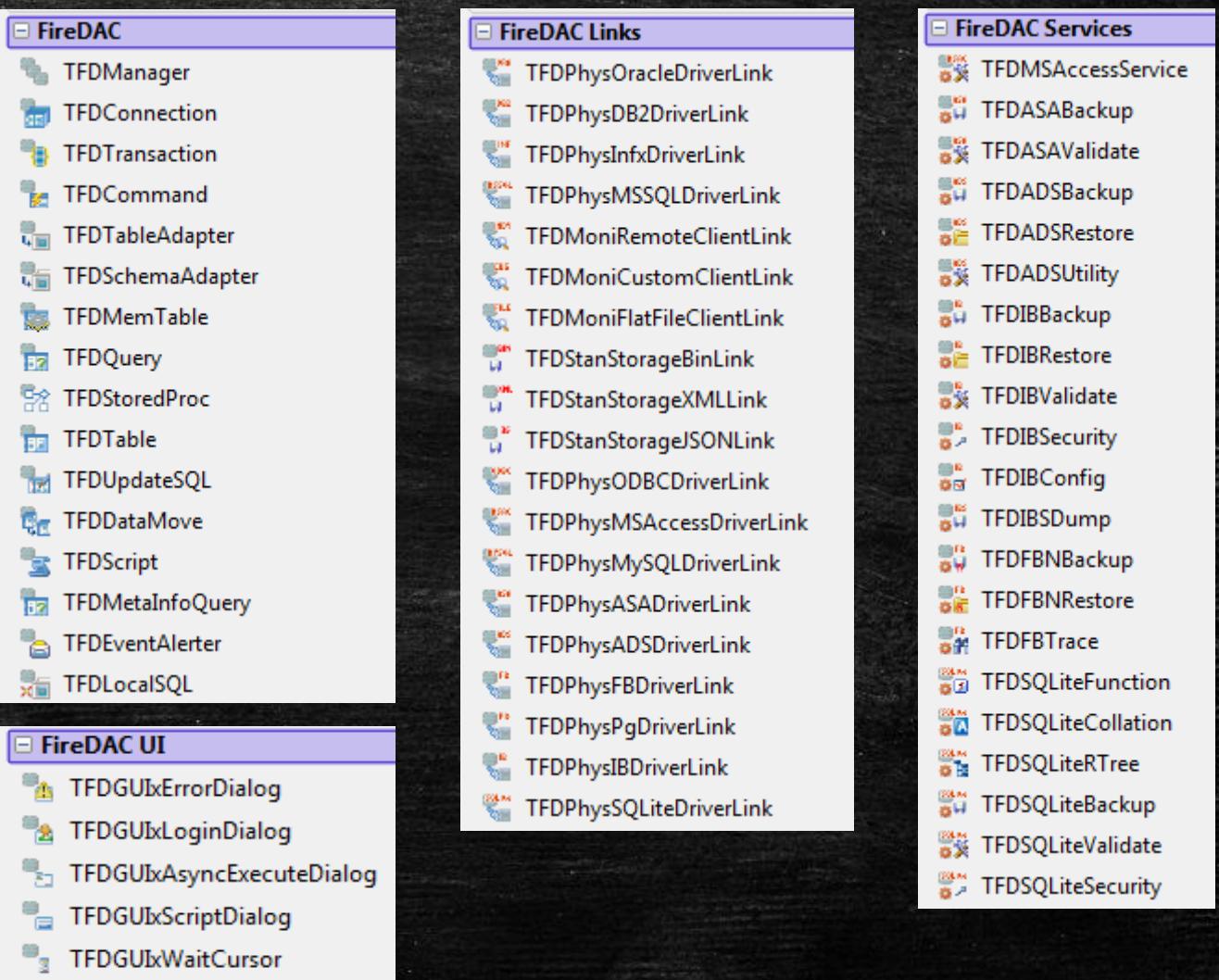


FireDAC – SQL Databases

- Mobile
 - InterBase XE3 – IBLite, IBToGo
 - SQLite
- Desktop and Enterprise
 - InterBase
 - SQLite
 - MySQL
 - SQL Server
 - Oracle
 - PostgreSQL
 - DB2
 - SQL Anywhere, Advantage DB
 - Firebird
 - Access
 - Informix
 - and more (via ODBC)

FireDAC Components

- Core
- UI
- Driver& Monitor Links
- Services



TFDConnection

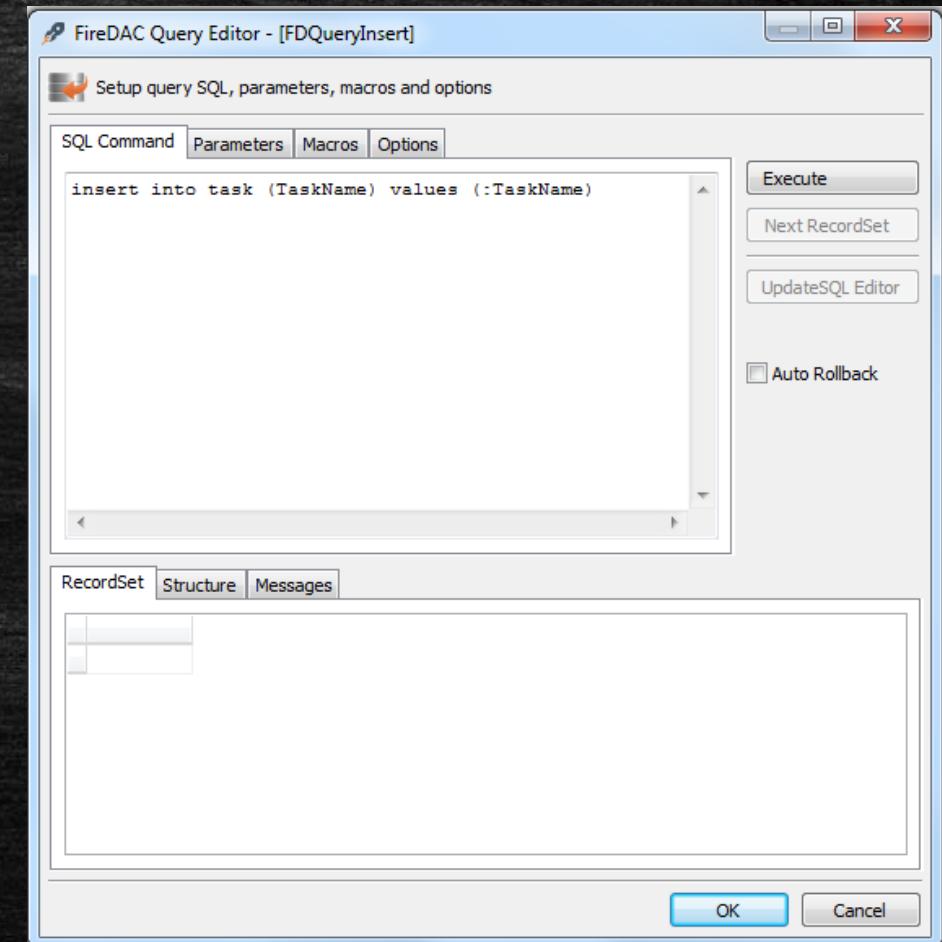
- FireDAC supports 3 connection definition kinds:
 - Persistent - Has a unique name, is managed by the FDManager, and is stored in a connection definition file.
 - Private - Has a unique name, is managed by the FDManager, but is NOT stored in a connection definition file.
 - Temporary - Has no name, is not stored in a connection definition file, and is not managed by the FDManager.
- Persistent connection definitions are stored in the connection definition file.
 - C:\Users\Public\Documents\Embarcadero\Studio\FireDAC\FDConnectionDefs.ini

TFDTable

- The TFDTable has two main operation modes:
 - Live Data Window mode. Allows bidirectional navigation through large data volumes with a minimal memory usage.
 - Standard mode. This mode is similar to TFDQuery. TFDTable generates a single SELECT command and uses the result set to walk through table records.
- Live Data Window Mode
 - Queries and keeps in memory only $2 * \text{FetchOptions.RowsetSize}$ of records.
 - When the application navigates through the table data, FireDAC automatically scrolls or positions the LDW to the required position.
 - Minimizes memory usage and allows you to work with large data volumes, similarly to an unidirectional dataset.
 - Enables bidirectional navigation, in contrast to an unidirectional dataset.
 - Gives always fresh data, reducing the need to refresh the dataset.
 - Does not give a delay to fetch all result set data, required to perform sorting, record location, jumping to last record, etc.

TFDQuery

- SQL query statement component
 - Double-click TFDQuery to invoke the FireDAC Query Editor
 - Use the Object Inspector to set the Connection, SQL and Active properties
 - Write code to set the properties and call the ExecSQL() method
- You can use the Params property for parameterized queries



TFDMemTable

- Use **TFDMemTable** to manage data in the client memory and optionally exchange the data with a **DBMS**
- You can create and populate a TFDMemTable.
- You can load and save TFDMemTable metadata and data to local storage (XML, binary)
- You can copy FireDAC datasets to TFDMemTable

```
with FDMemTable1.FieldDefs do begin
  with AddFieldDef do begin
    Name := 'f1';
    DataType := ftInteger;
  end;
  with AddFieldDef do begin
    Name := 'f2';
    DataType := ftString;
    Size := 50;
  end;
end;
```

```
with FDMemTable1 do begin
  Open;
  Append;
  Fields[0].AsInteger := ...;
  Fields[1].AsString := ...;
  Post;
end;
```

```
FDQuery1.Open;
FDQuery1.FetchAll;
FDMemTable1.Data := FDQuery1.Data;
FDMemTable1.First;

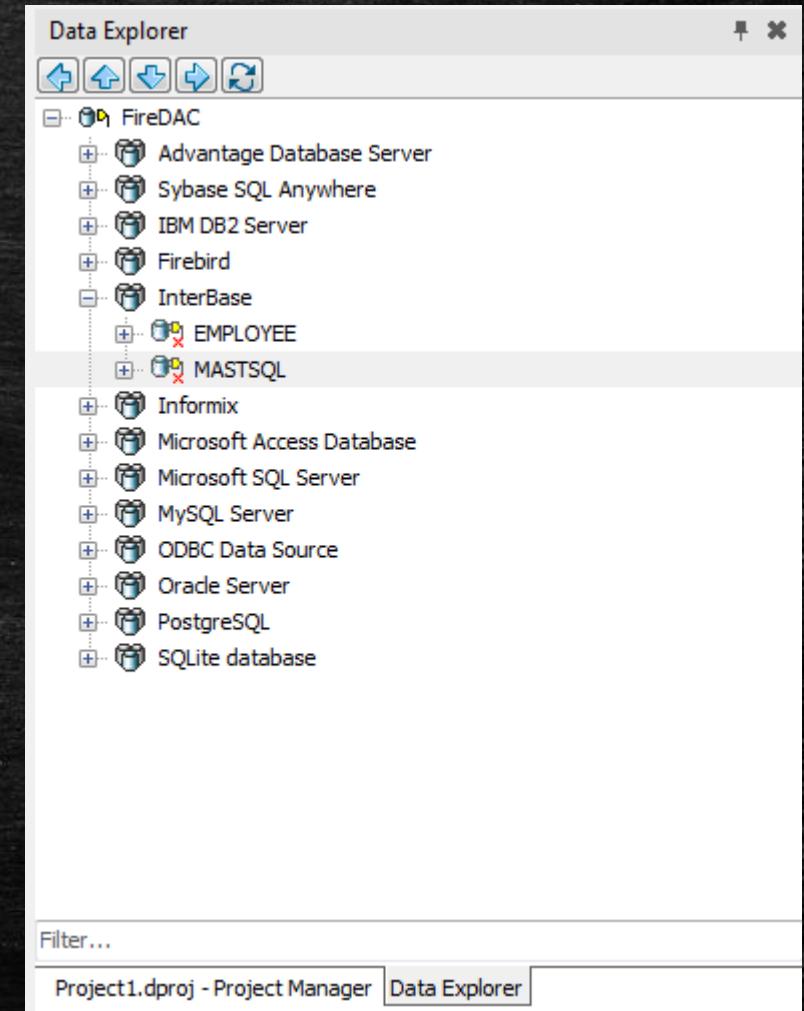
while not FDMemTable1.Eof do begin
  FDMemTable1.Edit;
  .....
  FDMemTable1.Post;
  FDMemTable1.Next;
end;
```

FireDAC Master-Detail Relationships

- The master-detail relationship allows you to automatically filter a detail dataset based on a current master dataset record. For example, the master dataset has "Order" records, and the detail dataset has "Order Line" records. So the detail dataset shows only lines for the current order.
- Special setup is not required for a master dataset.
- FireDAC offers two base methods to set up a detail dataset in a master-detail relationship:
 - Parameter-based. The master dataset field values are assigned to the detail TFDQuery or TFDStoredProc parameters, and then the detail dataset query is reexecuted.
 - Range-based. The master dataset field values are used to apply a range to the detail dataset. The detail dataset might be any FireDAC dataset with current active index.
- These methods can be combined.
- FireDAC offers two cached updates modes for the datasets in a master-detail relationship:
 - Decentralized Cached Updates mode: Each dataset tracks the changes independently of others.
 - Centralized Cached Updates mode: A few datasets in a master-detail relationship share a single change log. The master dataset can propagate changes to the detail datasets in cascading fashion, including auto-incrementing field values.

Data Explorer

- Add new connections; modify, delete, or rename your connections.
- Browse database server-specific schema objects, including tables, fields, stored procedure definitions, stored functions, triggers, and indexes.
- Drag and Drop schema objects to the Design Window to create FireDAC components with their connection properties



Local Databases

| SQLite | IBLite | InterBase ToGo |
|---------------------|-----------------------|-----------------------|
| Free | Free | Commercial |
| Feature light | Feature light | Fully featured |
| No security | No security | Secure Encryption |
| Simple Data Storage | Full SQL-92 RDBMS | Full SQL-92 RDBMS |
| Single read/write | Fast multi read/write | Fast multi read/write |





InterBase® - Suited to Mobile, Desktop, Server

- Overview
 - Highly scalable, powerful, embeddable database
 - Small memory and on-disk footprint
 - Simple installation / deployment
 - Near zero administration
 - Encryption built into all paid versions*
 - Low TCO
- Notable Features
 - Full-featured - SQL92 compliant, SQL Domains, ODBC
 - ACID compliant / Multi-version architecture
 - Automatic crash recovery
 - SMP - Multiple core support
 - Self-tuning engine
 - Fully featured in all editions
 - *Some features limited in run-time royalty free IBLite*
 - Database monitoring and statistics built in
-  Security
 - Embedded user & role based authentication
 - Full database, table or column encryption
 - DES (weak)
 - AES (strong 256bit) encryption
 - Decryption by user / user role
 - Allows developers to focus on creating fast, resilient applications.
- Disaster Recovery
 - Journaling
 - Data dumps
 - Partial Backups
- Also....
 - Window Direct I/O support & Amazon EC2 certified

InterBase - Editions

| INSTALLED | TRIAL | DEVELOPMENT |
|--|---|--|
| | | |
| <ol style="list-style-type: none">Server Edition<ul style="list-style-type: none">- 1 to unlimited connectionsDesktop Edition<ul style="list-style-type: none">- Same as server, but local connections only. | <ol style="list-style-type: none">Free 90 day Server trial<ul style="list-style-type: none">- Full featured server edition- 20 users / 80 connectionsToGo Edition<ul style="list-style-type: none">- Embedded Version | <ol style="list-style-type: none">Developer Edition<ul style="list-style-type: none">- 20 users / 80 connections- Weak encryption only- Time-limited connectionSDK Pack<ul style="list-style-type: none">- Full featured licenses- Time-limited (Similar to trial) |
| <ol style="list-style-type: none">ToGo Edition<ul style="list-style-type: none">- Embedded versionIBLite Edition<ul style="list-style-type: none">- Feature limited version of ToGo<ul style="list-style-type: none">▪ 100mb on disk▪ No encryption▪ Single thread connection |   | |

*ToGo and IBLite do not offer UDFs on iOS / Android due to platform requirements excluding external libraries in application bundles for security reasons.

Project Deployment

The screenshot displays two windows of the Deployment FireBLite application. Both windows have a header bar with tabs: 'Welcome Page', 'uMain', and 'Deployment FireBLite'. The top window is titled 'Debug configuration - Android platform' and lists files for deployment to the Android platform. The bottom window is titled 'Debug configuration - iOS Device platform' and lists files for deployment to the iOS Device platform. In both windows, several files are selected for deployment, indicated by checked checkboxes in the first column. Blue arrows point from the right side of the image towards the 'Remote Path' column in both tables, highlighting the deployment paths for the selected files.

Debug configuration - Android platform

| Local Path | Local Name | Type | Platforms | Remote Path |
|--|---------------------------|------------------|-------------------|-------------------------------------|
| <input checked="" type="checkbox"/> Android\Debug\ | TASKS.GDB | File | [Android,iOSD...] | \assets\internal |
| <input checked="" type="checkbox"/> Android\Base\ | AndroidManifest.xml | ProjectAndroi... | [Android] | \ |
| <input checked="" type="checkbox"/> \${IBRESTDIR}\andr... | AndroidManifest.xml | ProjectAndroi... | [Android] | \ |
| <input checked="" type="checkbox"/> \${IBRESTDIR}\andr... | oss_license_notice.txt | File | [Android] | \assets\internal\interbase\ |
| <input checked="" type="checkbox"/> \${IBRESTDIR}\andr... | reg_iblite.txt | File | [Android] | \assets\internal\interbase\license\ |
| <input checked="" type="checkbox"/> Android\Debug\ | libFireBLite.so | ProjectOutput | [Android] | library\lib\armeabi-v7a\ |
| <input checked="" type="checkbox"/> \${IBRESTDIR}\andr... | license.txt | File | [Android] | \assets\internal\interbase\ |
| <input checked="" type="checkbox"/> \${BDS}\bin\Artwork\... | FM_LauncherIcon_36x36... | Android_Laun... | [Android] | res\drawable-ldpi\ |
| <input checked="" type="checkbox"/> c:\program files (x86... | libnative-activity.so | AndroidLibnat... | [Android] | library\lib\armeabi\ |
| <input checked="" type="checkbox"/> ..\..\..\..\..\InterBas... | reg4059_1405107045000.... | File | [Android,iOSD...] | \assets\internal\interbase\license\ |

Debug configuration - iOS Device platform

| Local Path | Local Name | Type | Platforms | Remote Path |
|--|---------------------------|-----------------|--------------------|--------------------------------------|
| <input checked="" type="checkbox"/> \${BDS}\bin\Artwork\... | FM_LaunchImagePortra... | iPad_Launch7... | [iOSDevice] | \ |
| <input checked="" type="checkbox"/> \${BDS}\bin\Artwork\... | FM_SettingIcon_29x29.p... | iPad_Setting29 | [iOSDevice] | \ |
| <input checked="" type="checkbox"/> \${BDS}\bin\Artwork\... | FM_LaunchImageLands... | iPad_Launch1... | [iOSDevice] | \ |
| <input checked="" type="checkbox"/> \${BDS}\bin\Artwork\... | FM_ApplicationIcon_11... | iPhone_AppIc... | [iOSDevice] | \ |
| <input checked="" type="checkbox"/> \${BDS}\bin\Artwork\... | FM_ApplicationIcon_60x... | iPhone_AppIc... | [iOSDevice] | \ |
| <input checked="" type="checkbox"/> \${BDS}\bin\Artwork\... | FM_LaunchImage_320x4... | iPhone_Launc... | [iOSDevice] | \ |
| <input checked="" type="checkbox"/> ..\..\..\..\..\InterBas... | reg4059_1405107045000.... | File | [Android,iOSD...] | StartUp\Documents\interbase\license\ |
| <input checked="" type="checkbox"/> \${BDS}\bin\Artwork\... | FM_LaunchImagePortra... | iPad_Launch1... | [iOSDevice] | \ |
| <input checked="" type="checkbox"/> | TASKS.GDB | File | [iOSDevice,iOS...] | StartUp\Documents |
| <input checked="" type="checkbox"/> \${IBRESTDIR}\ios... | ibconfig | File | [iOSDevice] | StartUp\Documents\interbase\ |

FireDAC Samples

- Folders
 - C:\Users\Public\Documents\Embarcadero\Studio\14.0\Samples\CPP\Mobile Snippets
 - C:\Users\Public\Documents\Embarcadero\Studio\14.0\Samples\Object Pascal\Mobile Snippets
- FireDACSQLite
- FireDAC_IBLite
 - Note: As part of your purchase, you will receive an email that includes InterBase XE3 ToGo Test Deployment license and IBLite XE3 Deployment for Mobile Platforms license
 - “This Embarcadero product also entitles you to a free IBLite for iOS database deployment license. To activate your deployment license, go to <https://reg.embarcadero.com/srs6/activation.do> and enter the IBLite for iOS serial number above and Registration Code: xxx.”

Next Steps for our Business Mobile App

- Connect to MASTSQL database using FireDAC
- Customer table
 - Master Query for Customer List
 - Detail Query for Orders by Customer – connect to ListView
 - Customer List binding to ListView
 - Customer selection brings up TabItem with Customer Info (ListBox) and Orders ListView
 - Order selection brings up TabItem with a ListBox
- Employee table
- Parts table

Lesson 3 Review

- You can access the local storage for your mobile app
- Use INI files for application settings persistence
- FireDAC for local SQL database access – SQLite and IBLite/IBToGo
- Continued work on the business mobile app

Resources

- IOUtils - Docwiki
 - http://docwiki.appmethod.com/appmethod/1.14/topics/en/StandardRTLPathFunctions_across_the_Supported_Target_Platforms
 - <http://docwiki.appmethod.com/appmethod/1.14/topics/en/DiskAndDirectorySupportRoutines>
 - <http://docwiki.appmethod.com/appmethod/1.14/topics/en/PathManipulationRoutines>
- INI Files – Docwiki
 - http://docwiki.appmethod.com/appmethod/1.14/topics/en/UsingTIniFile_and_TMemIniFile
- FireDAC - DocWiki
 - <http://docwiki.appmethod.com/appmethod/1.14/topics/en/FireDAC>
 - [http://docwiki.appmethod.com/appmethod/1.14/topics/en/Overview_\(FireDAC\)](http://docwiki.appmethod.com/appmethod/1.14/topics/en/Overview_(FireDAC))
 - [http://docwiki.appmethod.com/appmethod/1.14/topics/en/GettingStarted_\(FireDAC\)](http://docwiki.appmethod.com/appmethod/1.14/topics/en/GettingStarted_(FireDAC))
 - [http://docwiki.appmethod.com/appmethod/1.14/topics/en/Components_\(FireDAC\)](http://docwiki.appmethod.com/appmethod/1.14/topics/en/Components_(FireDAC))
 - [http://docwiki.appmethod.com/appmethod/1.14/topics/en/Master-Detail_Relationship_\(FireDAC\)](http://docwiki.appmethod.com/appmethod/1.14/topics/en/Master-Detail_Relationship_(FireDAC))
 - http://docwiki.appmethod.com/appmethod/1.14/topics/en/Data_Explorer
 - [http://docwiki.appmethod.com/appmethod/1.14/topics/en/Mobile_Tutorial:_Using_FireDAC_and_SQLite_\(iOS_and_Android\)](http://docwiki.appmethod.com/appmethod/1.14/topics/en/Mobile_Tutorial:_Using_FireDAC_and_SQLite_(iOS_and_Android))
- Blogs
 - <http://blogs.embarcadero.com/>
 - Jim McKeeth - <http://delphi.org/>
 - Sarina Dupont - <http://blogs.embarcadero.com/sarinadupont/>

Note: <http://docwiki.appmethod.com/appmethod/1.14/topics/en/...> = <http://docwiki.embarcadero.com/RADStudio/XE6/en/...>

Homework & Next Time

- Take a look at more mobile samples and snippets for FireDAC
- Explore the Docwiki articles and tutorials listed on the Resources page
- Continue work on the business mobile app
- Lesson 4 – Building Multi-tier, Multi-device Apps
 - Multi-Tier architectures
 - DataSnap REST Application server
 - Stand-alone FireMonkey application
 - Stand-alone console application
 - ISAPI dynamic library
 - Apache dynamic link mobile
 - Mobile and Desktop DataSnap client applications

Q&A

Thank You 😊

davidi@embarcadero.com