



Product Documentation

---

# ToGo Edition Quick Start

InterBase XE3 Update 4

March, 2014

---

© 2014 Embarcadero Technologies, Inc. Embarcadero, the Embarcadero Technologies logos, and all other Embarcadero Technologies product or service names are trademarks or registered trademarks of Embarcadero Technologies, Inc. All other trademarks are property of their respective owners.

Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, CodeGear™ RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at [www.embarcadero.com](http://www.embarcadero.com).

April 2, 2014

- Introduction.....1
- Comparing IBLite, ToGo, and InterBase Server Edition .....1
- Using InterBase ToGo and IBLite ..... 1
- System Requirements.....1
- ToGo Edition Users.....2
- ToGo Edition Platforms .....3
- Windows Platforms.....3
  - ToGo Requirements and Constraints ..... 4
  - Developing with the ToGo Edition for Windows..... 4
  - Deploying with the ToGo Edition for Windows..... 5
  - File List for Windows 32-bit and 64-bit ToGo Editions:..... 5
- Mac OS X Platform.....7
  - Developing with the ToGo Edition for Mac..... 7
  - Converting Mac Server Applications to ToGo..... 8
  - File List for Mac OS X 32-bit ToGo Edition ..... 8
- iOS Application.....10
  - Requirements and Constraints ..... 10
  - Licensing ToGo and IBLite for iOS ..... 11
  - InterBase Runtime File Locations for Apple iOS ..... 11
  - Developing with ToGo Edition for iOS..... 11
  - Deploying InterBase with your applications on iOS ..... 11
  - File List for iOS ToGo Edition ..... 12
- Android Application.....12
  - Requirements and Constraints ..... 13
  - Character Set Supported by InterBase ToGo on Android ..... 13
  - Licensing ToGo and IBLite for Android ..... 14
  - InterBase Runtime File Locations for Android..... 14
  - Developing with ToGo Edition for Android..... 15
  - Deploying InterBase with your Applications on Android ..... 15
  - File List for Android ToGo Edition ..... 15
  - Development Dependencies..... 16
  - InterBase Required Permissions ..... 17
- Using the ToGo Edition .....18
  - Licensing ToGo..... 18
  - InterBase ToGo Test Deployment Licenses..... 19
  - How would I use an InterBase ToGo Test Deployment License?..... 19
  - How do I get InterBase ToGo Test Deployment Licenses? ..... 20

>

InterBase ToGo and IBLite Default Folders .....	21
InterBase ToGo Temp Directory .....	21
Developing Applications for InterBase ToGo .....	21
Using IBConsole with InterBase ToGo .....	22

# INTERBASE TOGO AND IBLITE QUICK START

The InterBase ToGo has two different platforms: ToGo and IBLite. Developers targeting database application deployment to an iOS can deploy either with a (for cost) "ToGo" license, or (no cost) "Lite" license. ToGo and IBLite have the same development model; both are embedded applications using the ToGo library.

## INTRODUCTION

The InterBase ToGo database engine is embedded in applications by directly using the InterBase database engine library. It does not have to be installed on any server or end-user workstation and it does not require that InterBase be installed on a machine before being used like InterBase Server and Desktop.

Applications built for iOS and Android can embed InterBase ToGo with IBLite licensing. IBLite is a "free" deployment . This application has limited capabilities when compared to the InterBase ToGo Edition on the same platform. However, for a fee, you can upgrade IBLite to the InterBase ToGo edition that has additional features.

**NOTE:** IBLite is available for the iOS and Android platforms.

The InterBase ToGo Test Deployment is available for Windows, MacOSX, iOS, and Android.

This Quick Start provides information necessary to run both InterBase ToGo and IBLite. For information on using InterBase ToGo for iOS and Android click the links below:

- [iOS Application](#)
- [Android Application](#)

## COMPARING IBLITE, TOGO, AND INTERBASE SERVER EDITION

The following table lists the features of each application. This table lists the features of the Server Edition as well.

Attribute Description	IBLite Edition	InterBase ToGo Edition	InterBase Server Edition
Licensing of CPU Cores	1	4	8
Licensed simultaneous users	1	2	Unlimited
Number of connections per user	1	4	4
Remote client access	Yes	Yes	Yes
Pre-compiler languages	Yes	Yes	Yes

<b>Attribute Description</b>	<b>IBLite Edition</b>	<b>InterBase ToGo Edition</b>	<b>InterBase Server Edition</b>
Metadata Updates (ddl operations)	Yes	Yes	Yes
External Tables Access	Yes	Yes	Yes
Database Access	Yes	Yes	Yes
Performance Monitoring	No	No	Yes
TCP Listening Server	No	No	Yes
Strong (AES) Database and Column Encryption	No encryption of any kind (weak or strong) allowed	Yes	Yes
OTW/SSL Support (Strong Network (SSL) Encryption)	No	Yes	Yes
Able to Import Add-on Licenses	No	No	Yes
Database File Size Limitation	100MB	No limit	No limit
Simultaneous Transaction per Connection	1	No limits	No limits
Services API Enabled	No	Yes	Yes

# USING INTERBASE TOGO AND IBLITE

The InterBase ToGo database engine can be embedded in applications by directly using the InterBase database engine library. It does not have to be installed on any server or end-user workstation and it does not require that InterBase be installed on a machine before use like InterBase Server and Desktop.

ToGo has no separate database server process running, and the database engine executes in the application process space. The embedded InterBase engine runs faster with these types of database applications because network connections are not used and data is not passed between the application and a separate database server. This gives application developers more choice and flexibility in their database design and deployment. Some applications for the embedded service include small devices and public kiosks, and the lowest end version of Value Added Resellers (VAR) applications which are built using InterBase.

For information on becoming a Value Added Retailer see <http://www.embarcadero.com/products/interbase/faq#oem>.

Only one application is allowed to connect to the database; the application exclusively locks the database file. However this application is allowed to make multiple connections to the database, so a multi-threaded application can make a single connection or multiple connections and access the database simultaneously. This also means that no other external client can access the database when it is being used (connected to) by the application using the embedded database engine.

## SYSTEM REQUIREMENTS

The InterBase ToGo edition is available on the following Windows platforms and can access any InterBase database created by InterBase desktop or server editions, versions 2007 and later.

### **System Requirements for Windows**

- Microsoft Windows Vista
- Microsoft Windows 8
- Microsoft Windows 7 (32-bit and 64-bit)
- Microsoft Windows XP (SP2)
- Microsoft Windows Server 2003, 2008
- Microsoft Windows Server 2008 R2 (64-bit)
- Microsoft Windows Server 2012
- Microsoft Windows 2000 (SP4)

### System Requirements for Mac

- Apple MAC OS X

### System Requirements for iOS

- iPhone 4 and 5 or newer running iOS 5.1 or newer
- iPod 4 or 5 or newer running iOS 5.1 or newer
- iPad 2 and 3 or newer running iOS 5.1 or newer
- Mountain Lion, Lion and their latest supported XCode versions. XCode 4.3 for iOS 5.1, 4.5 for iOS 6, 4.6 for 6.1
- Devices running earlier versions of the OS are not supported

### System Requirements for Android

- Android 2.3.3 and above; specifically Android OS versions Gingerbread, Ice Cream Sandwich, and Jelly Bean. Android devices running earlier versions of the OS are not supported.

## TOGO EDITION USERS

VARs and OEMs can install and embed the InterBase database service within their applications. They can now include InterBase within their applications by linking in with the new ToGo edition library. Also, a VAR can access an InterBase database from the CD or DVD without the need to install anything on the machine. This means InterBase does not access anything from the Windows Registry. There is no need for registry changes because all needed settings are provided by options in the InterBase configuration file.

The InterBase log files, lock files, and event files now go to a [specified temporary directory](#) on the machine as opposed to the install directory, as the install directory can now be on a read-only media. The ToGo edition can be used in conjunction with Embedded User Authentication (EUA), to control user access to a database.

Application Developers can link their applications with an included library containing the InterBase database service so nothing is needed other than this library to access InterBase database files. Once the library is included there is no need to start the InterBase database service, or to stop the service once the application exits, as the database service is in process.

The database file is compatible across all the editions of InterBase, so a VAR can upgrade a particular customer installation to the Desktop or Server edition with minimal changes to his application.

For more information on developing applications see: [Developing Applications for InterBase ToGo](#).



## TOGO EDITION PLATFORMS

The ToGo editions can be used on four different platforms

- [Windows Platforms](#)
- [Mac OS X Platform](#)
- [iOS Application](#)
- [Android Application](#)

## WINDOWS PLATFORMS

The winxx\_min.zip file (where xx = 32 or 64) is the minimum file set required to deploy InterBase ToGo. This file set is primarily intended for VARs (Value Added Retailers) deployment.

The winxx\_full.zip file (where xx = 32 or 64) includes the full file set for InterBase ToGo, and can be used for development with the ToGo edition.

- 1 You can unzip these two files into a directory such as c:\Interbase\ToGo\_Min and c:\InterBase\ToGo\_full so you can copy them later.
- 2 Depending on which type you want to deploy (minimum or full file set), copy the required InterBase config and interbase.msg file in an interbase sub-directory under your application directory.

The location of the InterBase configuration file and other files such as interbase.msg and license sub-directory are limited to the "interbase" sub-directory relative to the application. So if the application's running directory is <app\_dir> then the configuration file location is always restricted to "<app\_dir>\interbase". The INTERBASE environment variable is ignored.

**NOTE:** It is recommended that you use EUA which means that you will not need the admin.ib. If you do not want to use EUA, copy your admin.ib to the interbase sub-directory.

The following topics are covered for using ToGo on the Windows Platform:

- [ToGo Requirements and Constraints](#)
- [Developing with the ToGo Edition for Windows](#)
- [Deploying with the ToGo Edition for Windows](#)
- [File List for Windows 32-bit and 64-bit ToGo Editions:](#)

## TOGO REQUIREMENTS AND CONSTRAINTS

- Listed below is other functionality not available on the ToGo edition:
  - performance monitoring
  - database point in time and disaster recovery
  - database linger
  - shadowing
  - on-line dumps.
- If a function is not present in the Desktop edition, it is also not present in the ToGo Edition.
- No Installer, IBManager, or IBConsole are provided for this edition of InterBase. Installer and IBManager are not needed for ToGo.
- No JDBC access is available for the ToGo edition.
- If you need access to the above features, you can upgrade to Desktop or Server Edition(s). This is on the same lines as the restrictions we have placed with performance monitoring, where only the server edition can access this feature.

## DEVELOPING WITH THE TOGO EDITION FOR WINDOWS

- 1 In order to develop your application to work with the ToGo edition, replace the "gds32.lib" on your link path with the following files:
  - For Embarcadero-based linkers the file is located in the  
<app\_dir>\interbase\sdk\lib\ibtogo.lib
  - For Microsoft-based linkers the file is located in the  
<app\_dir>\interbase\sdk\lib\_ms\ibtogo\_ms.lib
- 2 If you want to use the ToGo edition and ibtogo.dll with older applications or applications linked with gds32.lib, you need to copy the ibtogo.dll to gds32.dll and make sure this is the first in your application's path. ibtogo.lib and gds32.lib both surface and support the same API so they can be moved from one to another.

**NOTE:** If you are developing with the ToGo edition you will need the complete ToGo Edition set (The winxx\_full.zip file includes the full file set for InterBase ToGo).

## DEPLOYING WITH THE TOGO EDITION FOR WINDOWS

- 1 When deploying with the ToGo edition, make sure the `ibtogo.dll` is available to your application when it is launched. The easiest way to ensure this is to include it in the same directory as your application.

The DLL is a file specially created to contain the function code and the definitions; a DLL is used by linking in a `.lib` file. An application linked using the DLL needs the DLL to be present at run time on the deployment machine, and be in the search `PATH` of the application.

- 2 Also create a InterBase sub-directory under the directory where "`ibtogo.dll`" is located (if you followed the recommendation above this should be your application directory) and copy the required InterBase config and `interbase.msg` file. InterBase licenses need to be made available in the InterBase/license folder. It is recommended that you use EUA which means that you will not need the `admin.ib`. If you do not want to use EUA copy your `admin.ib` to the interbase sub-directory.
- 3 You might need the internationalization, UDF, and other InterBase-provided libraries if you use them in your application.
- 4 For ToGo deployment you just need the runtime file set defined in the following table; you do not need to deploy with the full file set.

### FILE LIST FOR WINDOWS 32-BIT AND 64-BIT TOGO EDITIONS:

File Name	32-Bit	64-Bit	Min. (runtime) kit	Full Kit	Notes
<code>ibtogo.dll</code>	Yes		Yes	Yes	Can be located either in the <run directory> or the application directory or anywhere in one of the directories included in the <code>PATH</code>
<code>ibtogo64.dll</code>		Yes	Yes	Yes	Can be located either in the <run directory> or the application directory or anywhere in one of the directories included in the <code>PATH</code>
<code>gbak.exe</code>	Yes	Yes		Yes	
<code>gfix.exe</code>	Yes	Yes		Yes	
<code>gpre.exe</code>	Yes	Yes		Yes	
<code>gsec.exe</code>	Yes	Yes		Yes	
<code>gstat.exe</code>	Yes	Yes		Yes	
<code>isql.exe</code>	Yes	Yes		Yes	
<code>InterBase\interbase.msg</code>	Yes	Yes		Yes	
<code>InterBase\ibconfig</code>	Yes	Yes		Yes	

File Name	32-Bit	64-Bit	Min. (runtime) kit	Full Kit	Notes
InterBase\admin.ib	Yes	Yes	Yes	Yes	Not needed if you are using embedded user authentication, which is recommended.
InterBase\License.txt	Yes	Yes	Yes	Yes	
InterBase\oss_license_notice.txt	Yes	Yes	Yes	Yes	
InterBase\intl\gdsintl.dll	Yes	Yes	Yes	Yes	
ib_util.dll	Yes			Yes	
ibxml.dll	Yes			Yes	
ib_util64.dll		Yes		Yes	
ib_xml64.dll		Yes		Yes	
InterBase\bin\LicenseManager.exe	Yes	Yes		Yes	
InterBase\bin\LicenseReg.exe	Yes	Yes		Yes	
InterBase\bin\sanctuarylib.dll	Yes	Yes		Yes	
InterBase\license\ibxe3togo.slip	Yes	Yes	Yes	Yes	InterBase license file
InterBase\SDK\lib_ms\ibtogo_ms.lib	Yes			Yes	
InterBase\SDK\lib_ms\ib_util_ms.lib	Yes			Yes	
InterBase\SDK\lib_ms\ibxml_ms.lib	Yes			Yes	
InterBase\SDK\lib_ms\ibtogo64_ms.lib		Yes		Yes	
InterBase\SDK\lib_ms\ib_util64_ms.lib		Yes		Yes	
InterBase\SDK\lib_ms\ibxml64_ms.lib		Yes		Yes	
InterBase\SDK\lib\ibtogo.lib	Yes			Yes	
InterBase\SDK\lib\ib_util.lib	Yes			Yes	
InterBase\SDK\lib\ibxml.lib	Yes			Yes	
InterBase\SDK\include\ibase.h	Yes	Yes		Yes	
InterBase\SDK\include\iberror.h	Yes	Yes		Yes	
InterBase\SDK\include\ibxml.h	Yes	Yes		Yes	
InterBase\SDK\include\ibxml_proto.h	Yes	Yes		Yes	
InterBase\UDF\ib_udf.dll	Yes	Yes		Yes	
InterBase\examples\database\employee.gbk	Yes	Yes		Yes	
InterBase\examples\database\employee.gdb	Yes	Yes		Yes	
InterBase\examples\database\intlemp.gbk	Yes	Yes		Yes	
InterBase\examples\database\intlemp.gdb	Yes	Yes		Yes	

File Name	32-Bit	64-Bit	Min. (runtime) kit	Full Kit	Notes
InterBase\Doc\APIGuide.pdf	Yes	Yes		Yes	
InterBase\Doc\DataDef.pdf	Yes	Yes		Yes	
InterBase\Doc\DevGuide.pdf	Yes	Yes		Yes	
InterBase\Doc\EmbedSQL.pdf	Yes	Yes		Yes	
InterBase\Doc\LangRef.pdf	Yes	Yes		Yes	
InterBase\Doc\OpGuide.pdf	Yes	Yes		Yes	
InterBase\Doc\ReleaseNotes.pdf	Yes	Yes		Yes	

## MAC OS X PLATFORM

The following topics are covered for using ToGo on the Mac Platform:

- [Developing with the ToGo Edition for Mac](#)
- [Converting Mac Server Applications to ToGo](#)
- [File List for Mac OS X 32-bit ToGo Edition](#)

## DEVELOPING WITH THE TOGO EDITION FOR MAC

- 1 Open the Application bundle (show Package Contents) where the Application is located.
- 2 Navigate to Contents/MacOS and extract the InterBase ToGo zip file contents.
  - The osx32\_min.zip file is the minimum file set required to deploy InterBase ToGo.
  - The osx32\_full.zip file includes the full file set for InterBase ToGo, and can be used for development with the ToGo edition.
- 3 The MacOS folder looks like the following with <interbase> as the location of the extracted files:

```

yourapplication
<interbase>
libcgunwind.1.0.dylib
libib_util.dylib
libibxml.dylib
    
```

## CONVERTING MAC SERVER APPLICATIONS TO TOGO

When deploying with the ToGo edition, you just need to make sure that libibtogo.dylib is installed and the environment variable DYLD\_LIBRARY\_PATH has been updated to include the location of the libibtogo.dylib

You also need to create an **interbase** sub-directory under your application directory and copy the required "ibconfig" and interbase.msg file. The license sub-directory must also be located here.

The location of the InterBase configuration file and other files such as interbase.msg are limited to the "interbase" sub-directory relative to the application. The interbase.log file also defaults to the interbase sub-directory on Mac OS X. You can control the location of the interbase.log file by setting the following environment variable, in order of precedence.

```
# INTERBASE_TMP environment variable
```

```
# TMP environment variable
```

It is recommended you use EUA which means that you do not need the admin.ib. If you do not want to use EUA, copy your admin.ib to the interbase sub-directory.

You might need the i18n (gdsintl), UDF (ib\_udf) and other InterBase provided libraries if you use them in your application. Please update your configuration file, ibconfig, to include the location of the UDF directory if it is not the default interbase/UDF.

For ToGo deployment you just need the runtime fileset mentioned below; you can choose to deploy the full set should you need to use the InterBase command-line tools and documentation.

## FILE LIST FOR MAC OS X 32-BIT TOGO EDITION

File Name	Min. (runtime) kit	Full Kit	Notes
libibtogo.dylib	Yes	Yes	Can be located either in the <run directory> or the application directory. The location must be included in DYLD_LIBRARY_PATH environment variable.
gbak		Yes	
gfix		Yes	
gpre		Yes	
gsec		Yes	
gstat		Yes	
isql		Yes	
interbase\interbase.msg	Yes	Yes	
interbase\ibconfig		Yes	

File Name	Min. (runtime) kit	Full Kit	Notes
interbase\admin.ib	Yes	Yes	Not needed if you are using Embedded User Authentication (EUA), which is recommended
interbase\license.txt	Yes	Yes	
interbase\oss_license_notice.txt	Yes	Yes	
interbase\readme.html		Yes	
interbase\registration_FAQ.html		Yes	
interbase\IBsetup.html		Yes	
interbase\intl\gdsintl	Yes	Yes	
libib_util.dylib		Yes	
libibxml.dylib		Yes	
interbase\bin\LicenseManagerLauncher		Yes	
interbase\bin\lmlauncher.jar		Yes	
interbase\bin\cglm.ini		Yes	
interbase\license\ibxe3togo.slip	Yes	Yes	InterBase license file
interbase\include\ibase.h		Yes	
interbase\include\iberror.h		Yes	
interbase\include\ibxml.h		Yes	
interbase\include\ibxml_proto.h		Yes	
interbase\UDF\ib_udf		Yes	
interbase\examples\employee.gbk		Yes	
interbase\examples\employee.gdb		Yes	
interbase\examples\intlemp.gbk		Yes	
interbase\examples\intlemp.gdb		Yes	
interbase\Doc\APIGuide.pdf		Yes	
interbase\Doc\DataDef.pdf		Yes	
interbase\Doc\DevGuide.pdf		Yes	
interbase\Doc\EmbedSQL.pdf		Yes	
interbase\Doc\LangRef.pdf		Yes	
interbase\Doc\OpGuide.pdf		Yes	
interbase\Doc\ReleaseNotes.pdf		Yes	

## IOS APPLICATION

Both ToGo and IBLite are available for the iOS platform via RAD Studio XE4/XE5. The databases deployed on the mobile device can either have centralized user authentication via admin.ib, or use EUA to control user access within the database.

VARs and OEMs can now use InterBase on mobile devices by linking their applications with InterBase ToGo.

The following topics are covered for using ToGo on the iOS Platform:

- [Requirements and Constraints](#)
- [Licensing ToGo and IBLite for iOS](#)
- [InterBase Runtime File Locations for Apple iOS](#)
- [Developing with ToGo Edition for iOS](#)
- [Deploying InterBase with your applications on iOS](#)
- [File List for iOS ToGo Edition](#)

## REQUIREMENTS AND CONSTRAINTS

InterBase ToGo and IBLite have the following requirements and constraints:

- The iOS devices must use iOS 5.1, 6.0 or above. Devices running earlier versions of the OS are not supported.
- The file size for IBLite is limited to 100MB.
- IBLite will not allow any database and column encryption (weak or strong).
- InterBase UDF and Filter functionality are not supported due to limitations imposed by Apple for explicit loading of dynamic libraries by applications.
  - Where ToGo or IBLite is used as a client to connect to a remote InterBase server, you can use UDF and Filter libraries on the remote InterBase server.
- InterBase applications on iOS can work in a pure "Client" mode connecting to remote InterBase servers on the network. For this to work, InterBase server versions need to be 2009 (9.x) or higher.
- InterBase applications on iOS can only connect with embedded database files with ODS version  $\geq 13$ . If the database file is copied from a non-iOS InterBase platform, this implies that only databases created (or restored) with InterBase 2009 versions or greater are supported.
- ibconfig has the following limitations:
  - The parameter DATABASE\_CACHE\_PAGES defaults to 75 pages.
  - The parameter SORTMEM\_BUFFER\_SIZE defaults to 128KB.



## LICENSING TOGO AND IBLITE FOR IOS

RAD Studio XE4/XE5 users only get InterBase ToGo Test Deployment and IBLite licenses with their product. Both InterBase ToGo Test Deployment and IBLite can be used for development and testing.

When you are ready to deploy your applications to customers/production/appstore, you can either use the free IBLite license you have, or, contact [Embarcadero](#) for an InterBase ToGo license.

## INTERBASE RUNTIME FILE LOCATIONS FOR APPLE IOS

The InterBase related files are separated into the following categories:

- **Temporary files** are created under the <Application Home>/tmp directory.
- **Transient files** such as isc\_lock, isc\_init and temporary sort files are created here.
- **interbase.log** (informational and error messages from InterBase) file is delivered under <Application Home>/Documents/interbase directory.
- **Read/Write files** and files used by InterBase for setup are located in the <Application\_Home>/Documents/interbase directory.

The following files should be delivered in this directory: ibconfig, interbase.msg, license/ folder and related license files, and, admin.ib (if used).

For OTW/SSL connections, the recommended folder for providing supporting files is <Application\_Home>/Documents/interbase. In your OTW connection, be sure to provide relative argument values. For example:

```
10.20.10.20?ssl?serverPublicFile=Documents/interbase/
ibserverCAfile.pem??:<database_file_path>
```

## DEVELOPING WITH TOGO EDITION FOR IOS

The RAD Studio IDE from Embarcadero now supports developing applications for multi-device/ OS deployments including Windows, MacOSX, iOS, and Android.

## DEPLOYING INTERBASE WITH YOUR APPLICATIONS ON IOS

The databases deployed on the mobile device can either have centralized user authentication via admin.ib, or, use EUA to control user access within the database. EUA use could limit the need to deploy admin.ib on the target mobile device for database access.

## FILE LIST FOR IOS ToGO EDITION

File Name	Min. (runtime) kit	Full Kit	Notes
libibtogo.a		Yes	This is the InterBase ToGo static library which should be used to build Emulator and Device apps.
libibtogo.dylib		Yes	This is a mixed mode binary and should be used to build iPhoneSimulator apps only. Apple does not allow apps on the device to perform dynamic loading.
interbase/interbase.msg		Yes	Must be deployed to the app.
interbase/ibconfig		Yes	Optional: May be deployed with the app.
interbase/admin.ib		Yes	Optional; may be deployed with the app. This is not available if your deployed database is using EUA. You must deploy admin.ib with your application if you plan to use non-EUA databases or create a new database.
interbase/license.txt		Yes	Must be deployed on the app.
interbase/oss_license_notice.txt		Yes	Must be deployed on the app.
interbase/readme.html			
<i>interbase/registration_FAQ.html</i>			
<i>interbase/IBsetup.html</i>			
<i>interbase\license\libxe3togo.slip</i>		Yes	This is just a placeholder; this will eventually be replaced with a proper license by the developer/customer
InterBase\examples\employee.gbk			
InterBase\examples\employee.gdb			
InterBase\examples\intlemp.gbk			
InterBase\examples\intlemp.gdb			

## ANDROID APPLICATION

Both ToGo and IBLite are available for the Android platform via RAD Studio XE4/XE5. VARs and OEMs can use InterBase on mobile devices by linking their applications with InterBase ToGo.

The databases deployed on the mobile device can either have centralized user authentication through the admin.ib, or use EUA to control user access within the database. Note that EUA use may limit the need to deploy admin.ib on the target mobile device for database access.

The following topics are covered for using ToGo on the Android Platform:

- [Requirements and Constraints](#)

- [Character Set Supported by InterBase ToGo on Android](#)
- [Licensing ToGo and IBLite for Android](#)
- [InterBase Runtime File Locations for Android](#)
- [Developing with ToGo Edition for Android](#)
- [Deploying InterBase with your Applications on Android](#)
- [File List for Android ToGo Edition](#)
- [Development Dependencies](#)
- [InterBase Required Permissions](#)

## REQUIREMENTS AND CONSTRAINTS

- InterBase targets Android 2.3.3 and above; more specifically Android OS versions Gingerbread, Ice Cream Sandwich, and Jelly Bean. Android devices running earlier versions of the OS will not be supported.
- InterBase on Android will NOT support InterBase UDF and Filters to be in line with the feature set on all supported Mobile platforms (iOS and Android).
- InterBase applications on Android can work in a pure "Client" mode connecting to remote InterBase servers on the network. For this, InterBase server versions need to be 2009 (9.x) or higher.
- InterBase applications on Android can only connect with embedded database files with ODS version  $\geq 13$ . If the database file is copied from a non-Android InterBase platform, this implies that only databases created (or restored) with InterBase 2009 versions or greater are supported.
- ibconfig limitations: Android devices are equipped with limited RAM as compared to other Desktop/Server machines, and thus drive some changes to keep InterBase footprint smaller. This is similar to our support on Apple iOS.
  - parameter DATABASE\_CACHE\_PAGES defaults to 75 pages
  - parameter SORTMEM\_BUFFER\_SIZE defaults to 128KB

## CHARACTER SET SUPPORTED BY INTERBASE TOGO ON ANDROID

Character Set	Char. set ID	Max. Char. Size	Min. Char Size
NONE	0	1 byte	1 byte
ASCII	2	1 byte	1 byte

Character Set	Char. set ID	Max. Char. Size	Min. Char Size
OCTETS	1	1 byte	1 byte
UNICODE_FSS	3	3 bytes	1 byte
UTF_8/UTF8	59	4 bytes	1 byte
UNICODE_BE/UCS2BE	8	2 bytes	2 bytes
UNICODE_LE/US2LE	64	2 bytes	2 bytes

\*No COLLATION support for iOS and Android.

\*\*UTF8 is only supported client attachment's character set (LC\_CTYPE) for iOS and Android.

\*\*\*May need "Unsupported database" or kind like exception when detect unsupported character set.

## LICENSING TOGO AND IBLITE FOR ANDROID

RAD Studio XE4/XE5 users only get InterBase ToGo Test Deployment and IBLite licenses with their product. Both InterBase ToGo Test Deployment and IBLite can be used for development and testing.

When you are ready to deploy your applications to customers/production/appstore, you can either use the free IBLite license you have, or, contact [Embarcadero](#) for an InterBase ToGo license.

## INTERBASE RUNTIME FILE LOCATIONS FOR ANDROID

The InterBase related files are separated into two categories: Temporary files created by InterBase at run time and Read/Write files and files used by InterBase for setup.

- Temporary files created by InterBase at run time
  - These files are created under the *external storage* location mentioned above, /data/data/<package\_name>/files/interbase directory. interbase.log and other transient files such as isc\_lock, isc\_init and temporary sort files are also created here.
- Read/Write files and files used by InterBase for setup:
  - These files are created under the *external storage* location mentioned above, /data/data/<package\_name>/files/interbase directory. The following files should be delivered to this directory: ibconfig, interbase.msg, license/ folder and related license files, and, admin.ib (if used).
  - For OTW/SSL connections, our recommended folder for providing supporting files is /data/data/<package\_name>/files/interbase. In your OTW connection, please be sure to provide relative argument values. For example:

```
10.20.10.20?ssl?
```

```
serverPublicFile=/data/data/<package_name>/files/interbase/  
ibserverCAfile.pem??:<database_file_path>
```

## DEVELOPING WITH TOGO EDITION FOR ANDROID

The RAD Studio IDE from Embarcadero now supports developing applications for multi-device/ OS deployments including Windows, MacOSX, iOS, and Android.

## DEPLOYING INTERBASE WITH YOUR APPLICATIONS ON ANDROID

The databases deployed on the mobile device can either have centralized user authentication via admin.ib, or, use EUA to control user access within the database. EUA use could limit the need to deploy admin.ib on the target mobile device for database access.

## FILE LIST FOR ANDROID TOGO EDITION

File Name	Min. (runtime) kit	Full Kit	Notes
libibtogo.a		Yes	This is the InterBase ToGo static library which should be used to build Emulator and Device apps.
interbase/interbase.msg		Yes	Must be deployed with the app.
interbase/ibconfig		Yes	Optional: May be deployed with the app.
interbase/admin.ib		Yes	Optional; may be deployed with the app. This is not available if your deployed database is using EUA. You must deploy admin.ib with your application if you plan to use non-EUA databases or create a new database.
interbase/license.txt		Yes	Must be deployed on the app.
interbase/oss_license_notice.txt		Yes	Must be deployed on the app.
interbase/readme.html			
<i>interbase/registration_FAQ.html</i>			
<i>interbase/IBsetup.html</i>			
<i>interbase\license\ibxe3togo.slip</i>		Yes	This is a placeholder; this will eventually be replaced with a proper license by the developer/customer
InterBase\examples\employee.gbk			
InterBase\examples\employee.gdb			
InterBase\examples\intlemp.gbk			
InterBase\examples\intlemp.gdb			

## DEVELOPMENT DEPENDENCIES

When a native application is build for Android, InterBase expects the following requirements to be set before the application is deployed to the Android Emulator/Device. InterBase works with **“android-9” platform targeting ARMv7** devices.

Requirement	RAD/Delphi applications with FireDAC/dbExpress/IBX	RAD/Delphi applications with custom database connectivity
<p><b>INTERBASE environment variable in application process</b></p> <p>InterBase ToGo on Android needs to know the application folder where it can write its temporary files and look for license and other supporting database files. For this, the application needs to copy the InterBase assets to a writable location, and, set INTERBASE env. variable to that location. Typically, this is the "external" storage area for Android NDK apps.</p> <p>For e.g. &lt;package_name&gt; might be "com.example.native_activity"</p> <p>For e.g. /data/data/&lt;package_name&gt;/files</p> <p>Set INTERBASE env. variable to "/data/data/&lt;package_name&gt;/files/interbase"</p>	<p>This is automatically set by these database components. No need for the application to do anything specific.</p>	<p>Application needs to set this in their initialization code before calling any InterBase api.</p> <p>You may want to add code similar to the following in your application:</p> <pre data-bbox="995 808 1383 1354"> {\$IFDEF ANDROID}  procedure SetInterBaseVariable; var     M: TMarshaller; begin     setenv('INTERBASE', M.AsUtf8(GetFilesDir + PathDelim + 'interbase').ToPointer, 1); end;  {\$ENDIF} </pre>

Requirement	RAD/Delphi applications with FireDAC/dbExpress/IBX	RAD/Delphi applications with custom database connectivity
<p><b>AndroidManifest permissions</b> InterBase requires certain permissions set in the AndroidManifest file.</p> <p><b>INTERNET:</b> Allows applications to access network sockets. If you are building Client/Server apps accessing remote InterBase databases, this is a must have. This is also required by InterBase "call home" silent license registration.</p> <p><b>READ/WRITE EXTERNAL STORAGE:</b> This is the external storage location where InterBase files (including your database files) are delivered for applications to perform read/write operations.</p>	<p>See <a href="http://docwiki.embarcadero.com/RADStudio/XE5/en/Uses_Permissions">http://docwiki.embarcadero.com/RADStudio/XE5/en/Uses_Permissions</a> for further details.</p> <p><b>Permissions:</b></p> <ul style="list-style-type: none"> <li>• Internet</li> <li>• Read external storage</li> <li>• Write external storage</li> </ul>	<p>See <a href="http://docwiki.embarcadero.com/RADStudio/XE5/en/Uses_Permissions">http://docwiki.embarcadero.com/RADStudio/XE5/en/Uses_Permissions</a> for further details.</p> <p><b>Permissions:</b></p> <ul style="list-style-type: none"> <li>• Internet</li> <li>• Read external storage</li> <li>• Write external storage</li> </ul>
<p><b>Linking dependency on external NDK libraries</b></p> <p>In order to support C++ exception handling used by InterBase, the developer needs to link the libstlport_static.a with their application.</p> <p>libstlport_static.a is available at &lt;NDK&gt;/sources/cxx-stl/stlport/libs/armeabi-v7a/</p>	<p>This is automatically set by these database components. No need for the application to do anything specific.</p>	<p>Application needs to request the linker to include this library.</p>

## INTERBASE REQUIRED PERMISSIONS

Android applications require certain permissions to work. The following InterBase permissions must be set in the Project Options dialog:

- **INTERNET:** Allows applications to access network sockets. If you are building Client/Server apps accessing remote InterBase databases, this permission is a must-have. This permission is also required by InterBase "call home" silent license registration.
- **READ/WRITE EXTERNAL STORAGE:** Allows access to the external storage location where InterBase files, including your database files, are delivered. This permission is required for applications to perform read/write operations.

## USING THE TOGO EDITION

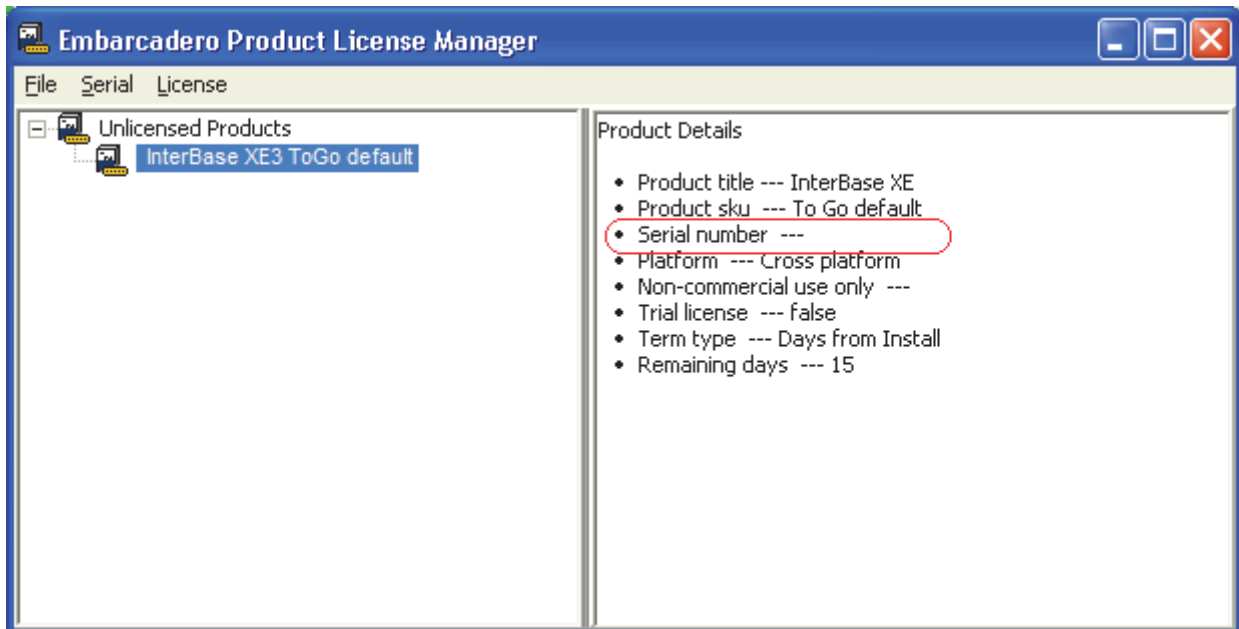
This section covers the following topics:

- [Licensing ToGo](#)
- [InterBase ToGo Test Deployment Licenses](#)
- [InterBase ToGo Temp Directory](#)
- [Developing Applications for InterBase ToGo](#)
- [Using IBConsole with InterBase ToGo](#)

## LICENSING TOGO

If you are an InterBase VAR and have received a superkey slip license file for distribution, please deliver the license file into the <interbase>/license folder. The following steps are only required if you want to register an individual S/N.

- 1 To license InterBase ToGo, execute the License Manager tool (licensemanager.exe on Windows, LicenseManagerLauncher on Mac OS X) from the bin folder. This opens the Embarcadero Product License Manager dialog.
- 2 When you downloaded either a trial version or purchase a licensed version, you will receive an email with a serial number. You can then copy this serial number into License Manager dialog.



- 3 All a VAR must do is copy their .slip file into the license folder and they have all the files they need.



## INTERBASE TOGO TEST DEPLOYMENT LICENSES

The test deployment licenses enable InterBase ToGo on your test deployment machines, not just your development machine. These licenses are only for testing and not actual deployment or production. You must purchase deployment licenses to deploy InterBase ToGo or use it in production.

## HOW WOULD I USE AN INTERBASE TOGO TEST DEPLOYMENT LICENSE?

Assume that you use Windows 32 bit for your RAD development. When you install RAD Studio XE3/XE4/XE5 (other than the starter edition), you use the included InterBase ToGo license on this development machine. As you progress in your application development and are ready to test your application on Windows 64-bit, Mac OSX, and other Windows 32-bit machines, you can use the InterBase ToGo Test Deployment licenses to enable InterBase on these machines and test your InterBase application deployments.

## HOW DO I GET INTERBASE TOGO TEST DEPLOYMENT LICENSES?

If you purchased an eligible RAD Studio XE3/XE4/XE5 product, you should have the InterBase ToGo Test Deployment serial number included with your RAD Studio serial numbers.

If you received your RAD Studio serial numbers via email, you will find the InterBase ToGo Test Deployment serial number in the same email.

You must install RAD Studio XE3/XE4/XE before completing the following instructions:

- 1 Go to <https://reg.embarcadero.com/srs6/activation.do> and enter your InterBase ToGo Test Deployment Serial Number.

If you are a RAD Studio, Delphi, or C++Builder XE3/XE4/XE5 user, enter the Registration Code: 737479.

All other users must enter the Registration Code they received from Embarcadero.

- 2 Once you have confirmed that the serial number and registration code match select the option to download the activation file. This generates an activation file which can be downloaded or emailed to be included with your distribution.

**RAD Studio XE5** users should copy the downloaded file to the %PUBLIC%\InterBase\redist\InterBaseXE3 folder. Then follow the steps in the Deployment Wizard. RAD Studio XE5 no longer creates an IBREDISTDIR system environment variable.

**NOTE:** %PUBLIC% is not a known path on XP, but RAD Studio XE5 does not support XP. However, %PUBLIC% is a known path on all operating systems supported by RAD Studio XE5. RAD Studio creates \InterBase\redist\InterBaseXE3 below %PUBLIC%, so the rest of the path will be known after the RAD Studio install is complete.

**RAD Studio XE4** users should copy the downloaded file to the %IBREDISTDIR% folder. Then follow the steps in the Deployment Wizard.

For **RAD Studio XE3** users, copy the activation file to the platform specific sub-folder of %IBREDISTDIR% folder.

- To do this open a cmd.exe window and "cd" to the directory you downloaded the activation file to. Then execute the following commands:

```
copy reg*.txt
%IBREDISTDIR%\osx32_togo\interbase\license\reg_ibtogo.txt
```

```
copy reg*.txt
%IBREDISTDIR%\win32_togo\InterBase\license\reg_ibtogo.txt
```

```
copy reg*.txt
%IBREDISTDIR%\win64_togo\InterBase\license\reg_ibtogo.txt
```

You might have to replace the reg\*.txt with your specific filename in case the copy does not work.

3 Save the activation file to a location of your choice. Depending on your platform choose one of the following files:

- For an IBToGo license choose reg\_ibtogo.txt
- For an IBLite license choose reg\_iblite.txt
- Otherwise, the activation file format is: regXXXX\_XXXXXXXX.txt.

Please make a note of the activation file name.

Now you are ready to use the Deployment project wizard in RAD Studio XE3/XE4/XE5, with InterBase ToGo.

## INTERBASE TOGO AND IBLITE DEFAULT FOLDERS

The location of the InterBase configuration file and other files such as interbase.msg are limited to the "interbase" sub-directory relative to the application. The interbase.log file also defaults to the interbase sub-directory on Mac OS X. You can control the location of the interbase.log file by setting the following environment variable, in order of precedence.

- 1 INTERBASE\_TMP environment variable
- 2 TMP environment variable.

## INTERBASE TOGO TEMP DIRECTORY

In order to allow read-only media to access and create temporary files, InterBase relies on a configuration parameter called "INTERBASE\_TMP". The "INTERBASE\_TMP" controls location of write files such as "interbase.log". If the "INTERBASE\_TMP" configuration parameter is not set, then the following order is used to check for the existence of environment variables and use the first one found:

- The path specified by the TMP environment variable.
- The path specified by the TEMP environment variable.
- The path specified by the USERPROFILE environment variable.
- The Windows directory.

For more information on specifying a Temp Directory see: *Configuring Sort Files* in the **Operations Guide**.

## DEVELOPING APPLICATIONS FOR INTERBASE TOGO

The simplest way to develop an application for InterBase ToGo is to do development using InterBase Developer edition. The InterBase Developer Edition is included with Embarcadero IDE's and can be downloaded for free from Embarcadero's web site.

When you are ready to use your application with InterBase ToGo, all you need to do is copy your executable to the same directory as `ibtogo.dll`. If you use this strategy for development you do not need to worry about paths. Also, if you use this strategy you will not need to worry whether any other installations of InterBase are on the same computer.

InterBase ToGo gets loaded using the Windows standard search order. This is important information if you elect to not place your executable in the same directory as `ibtogo.dll`. If you are unfamiliar with the Windows standard search order, it is strongly recommended that you read about it [here](#).

**NOTE:** The interface to InterBase ToGo is via `ibtogo.dll`. This is different than InterBase Server/Desktop editions which uses `gds32.dll`. However, `ibtogo.dll` and `gds32.dll` have the same interface. This means applications written for InterBase Desktop edition and 98% of all applications written for InterBase Server edition work unmodified by simply renaming `ibtogo.dll` to `gds32.dll`.

## USING IBCONSOLE WITH INTERBASE TOGO

While IBConsole is not included with InterBase ToGo, it works fine with InterBase ToGo if you require a graphical user interface tool to manage your database. IBConsole may be obtained from the InterBase Developer Edition. Depending on the version of IBConsole you are using, you may need to rename/copy `ibtogo.dll` to `gds32.dll` as noted above. Keep in mind that InterBase ToGo does not allow multiple applications to use the same database at the same time, so you cannot use IBConsole with your database while your application is using it at the same time.