

SQLBase™

New Features

Product Version 12



The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Last updated: July 2015

Legal Notice

Copyright © 2014-2015 Gupta Technologies, Inc. All rights reserved.

Gupta, Gupta Technologies, the Gupta logo, Gupta Powered, the Gupta Powered logo, ACCELL, Centura, Centura Ranger, the Centura logo, Centura Web Developer, Component Development Kit, Connectivity Administrator, DataServer, DBIntegrator, Development Kit, eWave, Fast Facts, NXJ, Object Nationalizer, Quest, Quest/Web, QuickObjects, RDM, Report Builder, RPT Report Writer, RPT/Web, SQL/API, SQLBase, SQLBase Exchange, SQLBase Resource Manager, SQLConsole, SQLGateway, SQLHost, SQLNetwork, SQLRouter, SQLTalk, Team Developer, Team Object Manager, TD Mobile, Velocis, VISION, Web Developer and WebNow! are trademarks of Gupta Technologies and may be registered in the United States of America and/or other countries.

SQLWindows is a registered trademark, and TeamWindows, ReportWindows and EditWindows are trademarks exclusively used and licensed by Gupta Technologies.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Gupta Technologies Corporation and its licensors, if any.

THE DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. GUPTA TECHNOLOGIES, INC. SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

This document may describe features and/or functionality not present in your software or your service agreement. Contact your account representative to learn more about what is available with this Gupta Technologies® product.

Gupta Technologies, Inc.
1420 Rocky Ridge Drive, Suite 380
Roseville, CA 95661

www.guptatechnologies.com

Table of Contents

| | |
|---|----------|
| Chapter 1 – What’s New in SQLBase 12.0 | 6 |
| Multithreaded Server | 6 |
| Embedded Deployment Pack Usability Refinement..... | 6 |
| Signed Executables..... | 6 |
| Removal of Support for Partitioned Databases | 6 |
| Ready For Windows 10 | 7 |
| Documentation | 7 |
| Chapter 2 – New Features in SQLBase 11.7 | 8 |
| SQLBase 64-bit for Windows..... | 8 |
| New SQLBase Client Drivers | 8 |
| No Configuration File Required (sql.ini) | 8 |
| Support for IPv6 (Internet Protocol Version 6) | 8 |
| Network/Windows Authentication/External Authentication..... | 9 |
| Requirements | 9 |
| Creating an Externally Authenticated User..... | 9 |
| Connecting Using External Authentication..... | 10 |
| New Client SQL.INI Parameters | 10 |
| TRANSPORTS..... | 10 |
| New Server SQL.INI Parameters..... | 11 |
| NAMED PIPES..... | 11 |
| SQLBase 11.7 Debug Tracing | 11 |
| New Fully Managed .NET Data Provider | 12 |
| New Security Features Including SSL..... | 12 |
| Improved Database Encryption | 13 |
| Improved Communication Encryption (SSL) | 13 |
| New Functions and Command Line Features..... | 13 |
| @REVERSE | 14 |
| @SCAN | 14 |
| sqlget SQLPCUN..... | 14 |
| show auto_increment <tablename> | 14 |
| DBNTRSV Service Registration | 14 |
| SQLConsole Download..... | 15 |
| Database Migration..... | 15 |
| New to SP1: Visual Studio Integration – DDEX Provider..... | 16 |

| | |
|--|-----------|
| Chapter 3 – New Features in SQLBase 11.6..... | 17 |
| 64-bit Client Support..... | 17 |
| Column Widening/Lengthening..... | 17 |
| Improved Crash Reporting..... | 17 |
| Isolation Levels..... | 18 |
| Roles..... | 18 |
| Chapter 4 – New Features in SQLBase 11.5 SP1 | 19 |
| Alter Table – LONG VARBINARY/LONG VARCHAR | 19 |
| Chapter 5 – New Features in SQLBase 11.5..... | 20 |
| Collation..... | 20 |
| Command Center | 20 |
| External Cache Manager | 20 |
| LONG VARBINARY | 20 |
| Longer Password Support..... | 21 |
| Migration Wizard..... | 21 |
| National Character (Unicode) and Multi-Language Support..... | 21 |
| Server Console | 21 |
| Chapter 6 – New Features in SQLBase 11.0..... | 23 |
| Database Version Utility..... | 23 |
| New Locking Statements | 23 |
| New Auto-Increment Option..... | 23 |
| New Multi-Versioning concurrency control..... | 23 |
| SQLBase Installer upgrades automatically..... | 23 |
| Chapter 7 – New Features in SQLBase 9.0 | 24 |
| Linux Compatibility..... | 24 |
| Database Migration..... | 24 |
| Console Mode | 24 |
| Chapter 8 – New Features in SQLBase 8.5 | 25 |
| Multiple SQLBase Installations | 25 |
| SQLBase Configuration File (SQL.INI)..... | 25 |
| SQLBase Installer..... | 25 |
| Windows Registry | 25 |
| COM+ Support and SQLBase Resource Manager | 26 |
| SQLBase API..... | 26 |
| Connectivity Administrator | 26 |
| Database Drivers and Providers..... | 30 |

| | |
|--|----|
| SQLTalk | 30 |
| SQLBase Management Console | 30 |
| SQL99 ANSI Join Syntax Support..... | 30 |
| New keywords..... | 30 |
| SQL99 ANSI Join Support is Optional..... | 31 |
| Keyword Check Tool..... | 31 |
| API Enhancements | 31 |
| Multiple Outer Joins..... | 31 |
| Database Engine Performance | 31 |
| SQL Language Enhancements..... | 32 |
| JDBC Driver Enhancements | 32 |
| New API Methods | 32 |
| OLE DB Data Provider Changes..... | 33 |
| .NET Data Provider Changes | 34 |
| ODBC Driver Changes..... | 34 |

Chapter 1 – What's New in SQLBase 12.0

Multithreaded Server

The SQLBase 12 database server has been re-engineered to allow the server to process queries on separate operating system threads. This design allows SQLBase to take full advantage of the CPU and increase its throughput for the clients. The reliance on the OS threads also permits the underlying operating system to schedule and prioritize the threads.

When the multithreaded operation is combined with read committed isolation level, the reduced level of locking allows many of the threads to process queries with very little lock overhead.

SQLBase 12 uses the same database file format and configuration file format as SQLBase 11.7. This convenience allows the user to compare the advantages of SQLBase 12 over SQLBase 11.7.

SQLBase 12 does not support Windows XP. If you have to run some servers on XP, knowing that SQLBase 11.7 can run with those database files on XP gives you the ability to choose when you migrate machines forward.

Embedded Deployment Pack Usability Refinement

The Embedded Deployment Pack (EDP) was created for customers who embed SQLBase into their product and distribute it. The EDP Install Maker is a new tool with a new GUI interface that helps you gather the needed files for a custom configuration and assists you in building a Windows install image. The EDP Install Maker allows you to click through and build a simple install image, and enables you to step through the process and provide customizations at each step.

Signed Executables

All installed executables from Gupta are signed with a code signing certificate. This allows the user to verify that they have not been tampered with since they were built. The signature can be checked by viewing the properties of the file from the file explorer.

Removal of Support for Partitioned Databases

Partitioned databases are no longer supported by SQLBase. Partitioning databases was a mechanism for spreading a database file across multiple disks. We recommend using SSD drives or RAID arrays for replacing the use of partitioned databases.

Ready For Windows 10

SQLBase 12 is ready for Windows 10 usage and deployment.

SQLBase 12 for Linux supports RedHat Linux 6 and 7.

Documentation

For the most current version of this document and all the documents available for SQLBase, go to:

http://support.guptatechnologies.com/supportwiki/index.php/SQLBase_12_Documentation

Chapter 2 – New Features in SQLBase 11.7

SQLBase 64-bit for Windows

SQLBase 11.7 is available as a true 64-bit Windows application.

- SQLBase 64-bit Server
Use the 64-bit installer to install SQLBase 64-bit for Windows. Database files are compatible between SQLBase 64-bit Servers and SQLBase 32-bit Servers. When you first open a 32-bit file on a 64-bit server, it will automatically be converted to 64 bits, and vice versa.
- SQLBase 64-bit Client Drivers
All client drivers of SQLBase 11.7 are available as 32-bit and as 64-bit drivers.

New SQLBase Client Drivers

The new SQLBase 11.7 client drivers offer a host of new easy configuration features. These include connections with or without an sql.ini file, Windows authentication and IPv6 support. In addition, all of the client drivers now feature a single-file format for ease of deployment.

No Configuration File Required (sql.ini)

The new SQLBase 11.7 client drivers can connect to databases without a client configuration file (sql.ini) being present. The connect function just requires a connection string providing the connection parameters. If no values are provided, the new drivers will try to guess the right settings and connect.

The new drivers do provide backward compatibility, that is, they can use existing sql.ini configurations.

Support for IPv6 (Internet Protocol Version 6)

SQLBase can now use IPv6 to connect to databases. You can configure SQLBase to listen for IPv6 connections on the same port as IPv4 or on different ports.

The following new configuration options can be added to the “dbntsrv” section of the ini file.

- IPV4LISTENADDRESS
The tcp listen address for ipv4 can be different than for ipv6. If this option is specified, it becomes the listen address for ipv4 for the server.
IPV4LISTENADDRESS=<ipaddress>[:<port>] - Listen on one specific ipv4 network address
IPV4LISTENADDRESS=*[:<port>] - Listen on all ipv4 network addresses
- IPV6LISTENADDRESS

This is the same as for IPV4LISTENADDRESS, except that the network address gets interpreted as ipv6 only network addresses.

For complete details on how to configure SQLBase to use IPV6, see the documentation for IPV4LISTENADDRESS and IPV6LISTENADDRESS in Chapter 4 of the *SQLBase Database Administrator's Guide*. See also the Transports sections in that guide (and later in this document).

Network/Windows Authentication/External Authentication

External authentication allows users that are already authenticated on their Microsoft Windows systems to log in to a SQLBase database without having to specify a separate username and password. Operating system user IDs are mapped to SQLBase user IDs, and known users can be automatically logged in to the SQLBase database.

Requirements

- External authentication is only available when both the client and server are running Microsoft Windows.
- The user must have accounts on both the server and client machines. The account can be an Active Directory account. If the account is on a domain, both the server and client machines must be on the same domain.
- If non-Active Directory accounts are used, the passwords for the user on both the server and client machines must be the same.

Creating an Externally Authenticated User

Before a user can use external authentication, a mapping must be made between the Microsoft Windows user ID and the SQLBase user ID. This is done using the **GRANT CONNECT** statement with the **IDENTIFIED BY EXTERNAL** clause:

```
GRANT CONNECT TO sqlbase_user_name IDENTIFIED BY EXTERNAL 'windows_user_id'
```

The *windows_user_id* is of the form *domain\user*, where *domain* is either an Active Directory domain name or a host name and *user* is the user name within that domain. The *domain* portion is optional. If the *domain* portion is not specified, a user with name *user* on any domain or host may log in to the system, as long as Windows can correctly authenticate the user; for example:

```
GRANT CONNECT TO bill IDENTIFIED BY EXTERNAL 'EXCORP\bsmith';  
GRANT CONNECT TO alice IDENTIFIED BY EXTERNAL 'ajones';
```

Mapping from Windows users to SQLBase users is one to one. Only one SQLBase user may be mapped to a specific Windows user and only one Windows user may be mapped to a SQLBase user. The same user names in different domains are considered different users, as is the user name without a domain or host name. Thus the following is allowed:

```
GRANT CONNECT TO joe1 IDENTIFIED BY EXTERNAL 'EXCORP\jbarnes';  
GRANT CONNECT TO joe2 IDENTIFIED BY EXTERNAL 'MYMACH\jbarnes';  
GRANT CONNECT TO joe3 IDENTIFIED BY EXTERNAL 'jbarnes';
```

When the server attempts external authentication, specific system user names (including the domain or host name) override general user names (not including domain or host names). Given the example users above, user "EXCORP\jbarnes" will connect as database user "joe1", while user "ACCOUNTING\jbarnes" will connect as database user "joe3".

Connecting Using External Authentication

To connect to the database using external authentication in the SQLTalk connect dialog, check the "Use External Authentication" checkbox. If using the CONNECT statement, specify the user name as "*". Either of these methods triggers the Microsoft Windows authentication mechanisms. This provides the server with the user's Windows domain (or host name) and username. The SQLBase server will then use this information to determine what SQLBase user is mapped to the Windows user and automatically log the user into the database as that SQLBase user.

The command line syntax in SQLTalk is "connect <database> */*;" as the following example shows:

```
"connect island */*;"
```

For API calls such as sqlcnc, the connect string syntax is "<database>*/" as the following example shows:

```
SQLTRCD rcd;
SQLTCUR cursor;
char * connect_string = "ISLAND/*/*";
rcd = sqlcnc(&cursor, (SQLTDAP) connect_string, strlen(connect_string));

// . . . Check error or use cursor
```

New Client SQL.INI Parameters

TRANSPORTS

The TRANSPORTS setting replaces the [Win32client.dll] section of the sql.ini file. The reason for the change is that .DLLs are no longer used to implement communication interfaces. Support for tcp and named pipes is built in to the server. Support for anonymous pipes is also built in, but it should only be used when needed to communicate with older versions of SQLBase. The named pipe interface is intended to replace the anonymous pipe interface.

If a TRANSPORTS entry is included, it takes precedence over the old style entries. If TRANSPORTS is not included, the [Win32client.dll] section is interpreted as in older versions. If neither is included, TRANSPORTS is implied to be:

```
[Win32client]
TRANSPORTS=npipe,tcpv4,tcpv6,apipe
```

To remain compatible with older sql.ini files, if the [Win32client.dll] section is present and no TRANSPORTS entry is present, the [Win32client.dll] will be used as it was before. If there is a TRANSPORTS section, it will be given precedence. The valid TRANSPORTS entries are:

- NPIPE - Named pipes
- TCPV4 (synonyms are TCP and WS32) - Tcp ipv4
- TCPV6 - Tcp ipv6
- APIPE - Anonymous pipes (only for compatibility with older versions of SQLBase); superseded by NPIPE.

New Server SQL.INI Parameters

The server side configuration settings that are affected in SQLBase 11.7 are stored under the [dbntsrv] section on Windows or [dblxsrv] section under Linux.

NAMED PIPES

SQLBase 11.7 supports named pipes (a socket connection) as a message transport. In prior versions, a mechanism using anonymous pipes was supported. The named pipe interface can be used to replace the anonymous pipe interface. Named pipes are required for using Windows authentication. Windows authentication is not supported under anonymous pipes.

```
[dbntsrv]
NAMEDPIPES=[ 0 | 1 | 2 | OFF | ON | LOCALONLY ]
```

0, OFF - Named pipe support is disabled

1, ON - Named pipe support is enabled for local and remote connections.

2, LOCALONLY - Named pipe support is enabled for local connections only.

When using named pipes over a remote connection, the client section needs to specify a serverpath, just as it does for a pipe.

```
[win32client.npipe]
serverpath=server1,dbmachine
```

You can also specify database names, as for TCP; for example:

```
[win32client.npipe]
serverpath=server1,dbmachine/ISLAND,DEMO
serverpath=server1,dbmachine/*
```

Furthermore, named pipe connections require that Windows recognize the client connection, according to its rules. If you cannot connect, check to make sure you can browse to the machine using Windows Explorer. Once you can do that, you will probably be able to also connect to SQLBase via a named pipe connection.

SQLBase 11.7 Debug Tracing

New debug tracing functionality has been added to SQLBase.

Tracing is configured using the **SQLBASE_TRACE** environment variable. The environment variable is of the form

```
SQLBASE_TRACE=option=value;facility.option=value;facility.option=value;...
```

The following is an example:

```
SQLBASE_TRACE=connection.level=verbose;connection.file=C:\logs\connection.log;
```

In this example, this environment variable will log verbose tracing information about connectivity to the log file C:\logs\connection.log.

Detailed information about the debug tracing feature, including the various options that can be set, is included in Chapter 7 (DBA Operations) of the *SQLBase Database Administrator's Guide*, "SQLBase Debug Tracing" section.

New Fully Managed .NET Data Provider

As of SQLBase 11.7, the new all-C# .NET data provider supports a number of new connectivity features such as connecting to a database without a configuration file (sql.ini), as well as support for many new 11.7-only features. This driver is also referred to as “the fully managed driver” because it does not rely on the C API as does the earlier .NET Data Provider.

In addition to being fully written in managed code, the .NET Data Provider offers the convenience of being able to ship as a single file, making deployment much easier.

To keep both drivers available and ease the transition to the new driver, both drivers are now available. Code compiled against the old driver can be used without modification in most cases.

It is recommended that you consider moving to the new provider, because it is the focus of new development. The provider based on the C API is considered deprecated. Moving to the new driver is relatively simple and involves making the changes shown in the following table:

| Environment / Code | New Managed Driver | Earlier Driver |
|--|---|---|
| All Visual Studio Projects | Replace Gupta.SQLBase.Data.dll with Gupta.SQLBase.Data.dll in your project references (for example, by right-clicking on your project in the Solution Explorer, then “Add Reference”). | Continue to use Gupta.SQLBase.Data.dll, adding the new file to your project to have Visual Studio update the reference. |
| Connection Strings using “Data Source” | In the new driver, “Data Source” is in line with other .NET drivers to mean the server or hostname; for example, “localhost”. To specify your database, use “Initial Catalog” (or “database” or “db”, which are synonyms). “Data Source=localhost;Initial Catalog=island” | No code changes needed. |
| VB.NET Projects | In your project source, replace: Imports Gupta.SQLBase.Data with: Imports Gupta.SQLBase.Data | No code change needed. |
| C# Projects | In your project source, replace: using Gupta.SQLBase.Data; with: using Gupta.SQLBase.Data; | No code change needed. |

For more information about the new .NET driver, including complete documentation of all connection string options, see chapter 7 in *SQLBase Connecting*.

New Security Features Including SSL

With SQLBase 11.7, all SQLBase products include the full security and encryption technology. Security and encryption is no longer limited to the SQLBase Treasury version.

Improved Database Encryption

SQLBase 11.7 introduces very strong 256-bit AES encryption for all database files. This updates the previous 128-bit Triple DES encryption. The new encryption level doubles the encryption key length and uses the latest industry-standard AES encryption technology to keep your data as safe as possible.

- New AES database encryption support

When specifying the database security level with ALTER DBSECURITY, there are two new options added to the previous NONE, LOW, MEDIUM, and HIGH encryption levels. AES128 will encrypt the database using 128-bit Advanced Encryption Standard (AES) and AES256 will encrypt the database using 256-bit AES. Both AES levels will use cipher chaining within each database block. Database backups of databases encrypted with the new encryption levels will be encrypted with the same encryption level as the database.

- Better log file encryption

In 11.6 and lower releases of SQLBase, log files for encrypted database were always encrypted using MEDIUM encryption. In 11.7 the same encryption level as the database will be used. The only exception is for databases encrypted with LOW encryption - in this case the log file will be encrypted using MEDIUM encryption.

- Longer encryption key support

The keys specified for database or backup encryption in the ALTER DBSECURITY and ALTER EXPORTKEY statements can now be longer than 36 characters, if specified as a single-quoted string. Single-quoted strings can contain spaces and punctuation. Although extended or Unicode characters can be used this is not recommended.

Improved Communication Encryption (SSL)

Client-server communication can be encrypted using OpenSSL including AES 256-bit encryption technology. The server and client will negotiate the highest encryption available, up to AES256.

SSL support is enabled in SQLBase by specifying the SSL parameter in the configuration file. Several parameters are needed by the SSL option to fully configure it. SSL communication is supported over both tcp ipv4 and ipv6.

```
SSL=<ssl parameter>;<ssl parameter> ...
```

- SSLONLY=[0|1]
- KEYFILE=<key filename>
- CERTIFICATEFILE=<certificate filename>
- KEYPASSWORD=<certificate key password>

The filenames must be included in quotes if they contain spaces.

New Functions and Command Line Features

SQLBase 11.7 features several new functions you can use in your queries along with new SQLTalk convenience functions and new command line options to make using SQLBase as a Windows service easier than ever.

@REVERSE

The new @REVERSE function reverses the order of a string argument; for example, the following query:

```
select @REVERSE(first_name) from employee where (first_name = 'Pat');
gives this output:
@REVERSE(FIRST_NAME)
=====
taP
```

@SCAN

The @SCAN function has been improved to work with long nvarchar as well as long varchar.

sqlget SQLPCUN

The new sqlget option returns the name of the current user. For example, given a typical connection to ISLAND, running “sqlget SQLPCUN;” in SQLTalk returns “SYSADM”.

show auto_increment <tablename>

The new “show auto_increment” command takes a table name as the sole argument and displays the SEED, STEP, and NEXT values for the auto_increment field, where:

SEED = starting auto_increment value for the table

STEP = amount to increment each time

NEXT = the value of the auto_increment field as of the next insert

For example, given the statements:

```
create table t1(
  c1 integer not null auto_increment(5,2),
  c2 char(20) not null);
insert into t1(c2) values('Hello');
show auto_increment t1;
```

The show auto_increment command displays:

```
***** SEED:5  STEP:2  NEXT:7 *****
```

DBNTRSRV Service Registration

- /register and /unregister

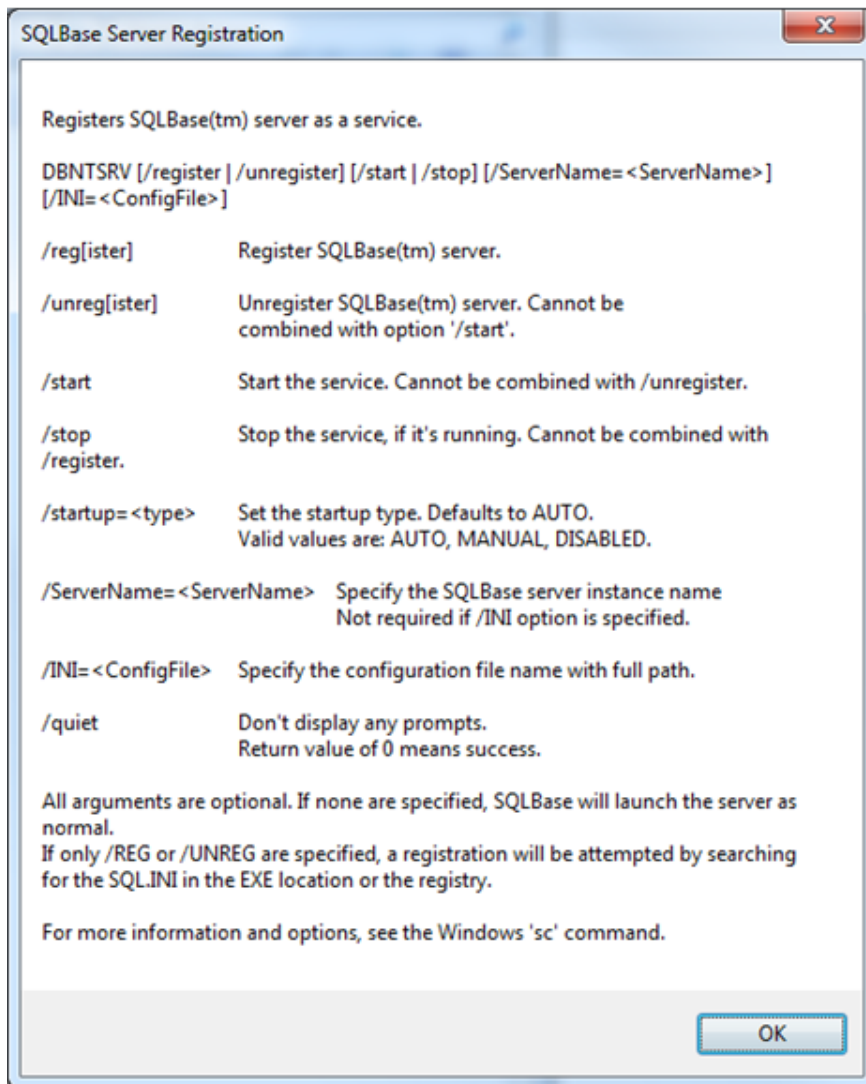
The DBNTRSRV.EXE server executable now features command line switches for registering and unregistering the server as a Windows service.

- /start and /stop

The SQLBase 11.7 service can be started and stopped by using “DBNTRSRV.EXE /start” and “dbntrsv.exe /stop”, respectively.

- Use the /? Switch for the new command line help screen

To make the command line easier to use, running “dbntrsv /?” displays the command line help screen, as the following shows.



SQLConsole Download

SQLBase Console is no longer installed by the main SQLBase product installer but is instead available for download from the support download page:

<http://support.guptatechnologies.com/supportwiki/index.php/Downloads>

Database Migration

Database files are automatically converted from the SQLBase 11.6 format to the SQLBase 11.7 format. Conversion from 32-bit 11.6 to 64-bit 11.7 also occurs automatically.

If database files are from a version of SQLBase 11.5 or earlier, the database must be unloaded and reloaded into 11.7. This conversion can be simplified by using the migration wizard utility included in this release. See the notes about the migration wizard under [Column Widening/Lengthening](#).

New to SP1: Visual Studio Integration – DDEX Provider

Visual Studio's Data Designer Extensibility (DDEX) specification allows database-provider vendors to integrate ADO.NET compliant .NET drivers in Visual Studio. This lets you take advantage of the Server Explorer, table browsing, query building, and other database features in Visual Studio.

Chapter 3 – New Features in SQLBase 11.6

64-bit Client Support

SQLBase 11.6 contains support for 64-bit drivers and clients, including SQLTalk, ODBC, OLEDB, JDBC, and .NET connections. 32-bit support will continue as well.

Column Widening/Lengthening

Changes have been made to SQLBase programming interface functions and data structures in order to support wider/lengthened columns and more columns in a row.

For information about programming interface (API) changes, see “Extension Functions” in the *SQLBase API* document.

For information about data structure changes, see “Column Widening/Lengthening” in the *SQLBase Advanced Topics Guide*.

Due to these changes, a migration wizard has been provided for migrating databases from SQLBase 7.6, 8.5, 9.0, 11.0, and 11.5, including all service packs. The wizard unloads the entire database and then re-loads it into an 11.6 version. Thus, it is very important to **back up your database** before migrating in case you run into any errors.

The migration wizard is easy to use, and it can be launched from the Start menu (Start/All Programs/Gupta/SQLBase 11.6/SQLBase 11.6 Database Migration Wizard).

Note: The migration wizard will not work for a Treasury Edition database. If you need to migrate a Treasury Edition database, or want to manually unload and load your database, see UNLOAD and LOAD for information on these commands. Or you can use the following format:

SQLTalk 11.5:

```
CONNECT island;  
UNLOAD DATABASE 'c:\island.unl'; DISCONNECT ALL;
```

SQLTalk 11.6 (the database can be renamed if desired):

```
SET SERVER server1/password; CREATE DATABASE island; CONNECT island;  
LOAD SQL 'c:\island.unl'; UPDATE STATISTICS ON DATABASE; COMMIT;  
DISCONNECT ALL;
```

Improved Crash Reporting

The new crash reporter provides a stack trace and other information to assist in diagnosing and fixing a problem that results in a crash. This will be especially helpful should you report a SQLBase crash that Gupta Support cannot reproduce.

Isolation Levels

Isolation level behavior can now be overridden so that a given driver-specified isolation level will correspond with an allowed SQLBase isolation level. Any values set can be used during the lifetime of the connection to set transaction isolation levels as well.

Additionally, CS and RC isolation levels are now supported for the .NET, OLEDB, ODBC, and JDBC drivers.

See the *SQLBase Connecting* documentation for more information:

- ODBC - How do I set isolation levels?
- OLEDB - How do I set isolation levels?
- .NET - How do I set isolation levels?
- JDBC - How do I set isolation levels?

Roles

Support for "roles" has been added. This feature allows you to easily grant privileges to a group of users rather than using numerous GRANT/REVOKE statements. Corresponding SQLTalk commands have been added for implementation of this feature.

See "Roles" in the SQLBasebase Administrator's Guide.

Chapter 4 – New Features in SQLBase 11.5

SP1

Alter Table – LONG VARBINARY/LONG VARCHAR

ALTER TABLE now allows you to change the column datatype LONG VARBINARY to LONG VARCHAR and vice-versa.

Chapter 5 – New Features in SQLBase 11.5

Collation

Collation is used in any SQL query that involves comparison of strings. In earlier versions of SQLBase, the system had only BINARY collation available. SQLBase 11.5 provides many additional collations for sorting Unicode data by language. These collations can be set for a server (and all of the server's databases and connections to those databases), a specific database (and all connections to that database), and a specific connection to a database.

See “Collations” in the *SQLBase Language Reference*.

Command Center

SQLBase Command Center is a tool for managing and maintaining SQL databases. SQLBase Command Center consists of the following two tools:

- SQLBase Command Center (GUI-based application that simplifies database management)
- Service for MS-Windows™

Automated and time-specific background tasks for performing scheduled tasks, such as database index updating.

See the *SQLBase Command Center User Guide*.

External Cache Manager

The new external cache manager takes advantage of the processing capabilities of multi-core machines. It includes both a main database process and a cache manager daemon process (cmdmn), which work together to maximize concurrency when caching pages.

By default, SQLBase uses the existing internal cache manager on single-core machines and the new external cache manager on multi-core machines. This default setting can be changed in SQL.ini.

See “External Cache Manager” in the *SQLBase Database Administrator's Guide*.

Note: The external cache manager is not supported in the Treasury edition.

LONG VARBINARY

The new LONG VARBINARY data type stores character data. When migrating, existing columns using LONG VARCHAR become LONG VARBINARY. These columns behave exactly the same way as they did in previous releases.

See “LONG VARCHAR (or LONG or CLOB)” in the *SQLBase Language Reference*.

Longer Password Support

The new maximum password length is 32 characters. The same password restrictions apply as in SQLBase 10. Maximum username length has not changed.

Migration Wizard

The new migration wizard enables you to migrate a SQLBase 10.0 database to SQLBase 11.5. The wizard (sb10mgwz.exe) is located in the Gupta\SQLBase115\migration directory. The installation adds a shortcut menu item.

National Character (Unicode) and Multi-Language Support

SQLBase supports Unicode data. Previous versions of SQLBase limited a database server to a single language, but in SQLBase 11.5 the number of languages supported is theoretically unlimited. The ICU library from IBM Corporation provides some of the support for this feature.

There are many new datatypes. The datatypes beginning with the letter “N” are referred to as “national character types”. NCHAR and NVARCHAR are intended to store national characters (Unicode). They are compatible with most of the functions and operations that apply to CHAR and VARCHAR datatypes (which usually store “ANSI” or 8-bit characters) and can be used in indexes, stored procedures, stored commands, and triggers. There are also functions to convert data from CHAR/ VARCHAR to NCHAR/NVARCHAR and vice versa. SQLBase also supports CLOB (synonym for LONG VARCHAR) and BLOB (synonym for LONG VARBINARY).

See “Character data types” in the *SQLBase Language Reference*.

Server Console

Version 11.5 introduces the concept of a Server Console that runs independent of the server itself. Thus, the Server Console can be opened and closed without actually affecting the server.

If the database server was not started as a Windows service, the initial launch of "SQLBase Database Engine" will start the server *and* open the console. Closing this console will shut down the server, as in previous versions.

Unlike previous versions, multiple instances of "SQLBase Database Engine" can be launched (as Server Consoles), although each window will perform the same functions. If you launch multiple instances, be careful to note which window was launched first, because closing that window will shut down the server. The other windows will only act as Server Consoles.

If the database server was started as a Windows service, the initial launch of "SQLBase Database Engine" will **not** affect the server. Instead, it will open a Server Console, which can be used just like the Database Engine

and then safely closed. Closing the console will not shut down the server, and it can be relaunched and closed as needed.

Version 11.5 also enables text highlighting in the console, as well as copy and paste (into other applications) using CTRL+C and CTRL+V.

Note: The default console port value is 2696. To change this value add the entry "consoleport" under [dbntrsv] and set its port value accordingly.

Chapter 6 – New Features in SQLBase 11.0

Database Version Utility

The dbversion utility displays the version number of the database files. If it does not recognize the file or version number it will report that it cannot determine the version.

New Locking Statements

SQLBase now supports locking tables. You can use the new LOCK TABLE statement to improve performance, or you can set the autolocktable option to rely on SQLBase to automatically lock and release the table at the right time to minimize deadlocks and maximize performance.

See the description of the new LOCK TABLE statement in the *Database Administrator's Guide*, chapter 4 (Databases) in the section "LOCK table statement." Also see chapter 3 (Configuration: SQL.ini and Registry) for the "autolocktable" sql.ini option described under the "Keyword" section.

New Auto-Increment Option

SQLBase now has a column attribute for columns to automatically assign an increasing numeric value for specific columns. You can use this functionality to improve your application's performance when unique incrementing keys are needed in an application.

See the description of the new auto-increment option in the "Databases" chapter of the *Database Administrator's Guide* under the "Auto-Increment Option" section.

New Multi-Versioning concurrency control

This is a protocol for concurrency control that allows old values of data to be kept when the item is updated. See the "SQLBase Multi-Versioning" section of chapter 19 in *the SQLBase Advanced Topics Guide*.

SQLBase Installer upgrades automatically

With SQLBase version 11.0 you can now upgrade the following versions of SQLBase: 7.5.1, 7.6, 8.0, 8.5 and 9.0.1.

Chapter 7 – New Features in SQLBase 9.0

Linux Compatibility

SQLBase 9.0 now runs on several versions of Linux operating systems from SUSE and Red Hat. See the release notes for the exact versions supported.

Database Migration

Databases from Windows versions of SQLBase can be copied as files and restored directly to a SQLBase installation on Linux. These databases can be used on Linux without any further migration or conversion. Note the following prerequisites:

- The database must have been in a consistent state before being copied (no open log files.)
 - The database must be from version 8.0 or later.
 - The database file must be placed in a directory of the same name. In other words, database file OPERATIONS.DBS must reside in directory OPERATIONS, beneath the main database directory.
 - File sql.ini must be modified to contain entries for the new database name, so that it is accessible to clients. Note that Linux is case-sensitive and "sql.ini" must be in lowercase. You are still free to use the feature introduced in version 8.5 that allows you to specify your own name and location for sql.ini. See the next chapter for details.
-

Console Mode

The status display of the server is in console mode, as is the SQLTalk utility when it is run from the Linux machine. However, you can still run GUI-based tools on Windows client machines that connect to the database server on Linux.

Chapter 8 – New Features in SQLBase 8.5

Multiple SQLBase Installations

SQLBase 8.5 now supports more than one installation of SQLBase on a computer. Multiple instances of the SQLBase database engine, different versions of SQLBase, and multiple client configurations can run simultaneously without interfering with each other.

You can run one or more SQLBase 8.5 database engines concurrently with one earlier version of SQLBase, but you cannot run multiple earlier versions simultaneously.

SQLBase Configuration File (SQL.INI)

To support multiple installations, the configuration file (always named SQL.INI in versions prior to 8.5) now has a flexible name and path specification. You can use whatever name you like in place of SQL.INI. The name SQL.INI continues to be used throughout SQLBase documentation, but your actual file name might be different.

Multiple installations are optional. During installation, if you choose not to take the extra steps to accommodate multiple installations, the configuration file will continue to be named SQL.INI, and the search algorithm used to find the file will be identical to that used in SQLBase versions 7.x, 8.0, and 8.1.

The multiple installation feature requires several new keywords in several sections of SQL.INI. These are ordinarily created in response to your answers during the install process. For more detailed information about the keywords, see chapter 3 of the Database Administrator's Guide. The new keywords are AUTOSTARTSERVERPATH, CLIENTRUNTIMEDIR. Keyword SERVERPATH, previously only in the WS32 protocol section, also now appears in sections SPX32, WSSPX, MPIPE, and APIPE.

SQLBase Installer

By default, the installer still uses the SQL.INI behavior of earlier versions - a file named SQL.INI in the same directory as the server executables. After installation, you can use the Connectivity Administrator utility to specify a different file name and different directory in place of the default SQL.INI. More details and examples are in the following Connectivity Administrator section.

Windows Registry

Previous versions of SQLBase wrote information to the HKEY_LOCAL_MACHINE portion of the Windows registry. Version 8.5 writes instead to HKEY_CURRENT_USER. This means that the user doing the installation of SQLBase does not need Administrator privileges. However, if SQLBase is to be registered as a Windows service, the user does need appropriate privileges for creating a new service.

Use of registry information has been greatly reduced. The primary purpose of the entries in HKEY_CURRENT_USER is to allow utilities such as Connectivity Administrator and SQLBase Management

Console to locate and enumerate server installations. The database server itself does not use registry information at all.

COM+ Support and SQLBase Resource Manager

Some features in SQLBase use COM+ to surface their functionality. For example, SQLBase Management Console (SMC) displays statistics about SQLBase Resource Manager (SQLBrm) transactions, and obtains those statistics from SQLBrm by calling methods from the SQLBrm COM+ interfaces. SQLBrm itself allows your client applications to enroll the SQLBase engine as part of a distributed COM+ transaction.

COM+ servers must be registered with Windows before they can be used. Each component name must be unique. If you install multiple SQLBase 8.5 servers, each using the same component names, only one of those servers can have its COM+ components registered with Windows at a given time. For this reason, only one of the installations will be able to use SQLBrm at a time.

SQLBase API

The name and path of the server configuration file (SQL.INI in previous versions) are now available by using parameter SQLPSINI in a call to the `sqlget` method. Similarly, parameter SQLPCINI will return the name and path of the client configuration file currently in use.

There is a new API method called `sqliniEx`. This new method is implemented in SQLWNTM.DLL.

```
SQLTRCD sqliniEx (SQLTDAP pINI, SQLTDAL nLength) ;
```

This new method has the functionality of method `sqlini` in earlier versions of SQLBase. It also accepts two new parameters. The first is a string argument that provides the configuration file name and location for the client. The second is a 2-byte integer that specifies the number of characters passed in the first argument. If this is 0, the first parameter is presumed to be null-terminated.

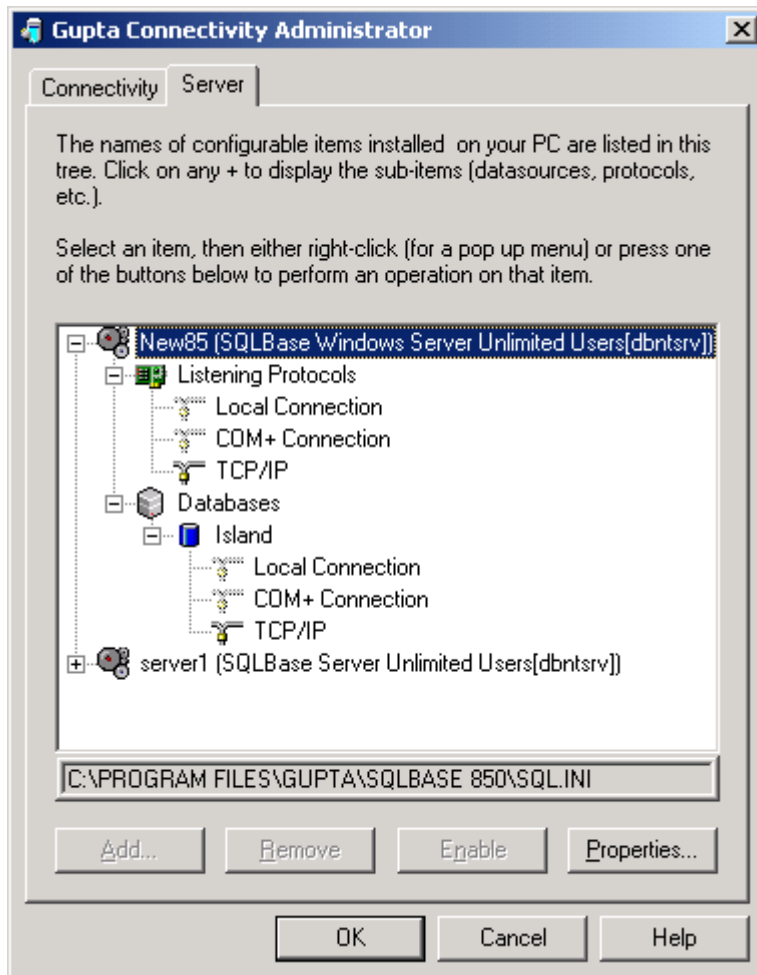
Note: This method is not absolutely required. If it is omitted, the SQL/API will look for the SQL.INI file using the search algorithms of earlier versions of SQLBase. However, to take advantage of version 8.5's ability to use a flexible name and path for the configuration file, this method must be called before any other methods in the SQL/API are called.

If the caller provides just a file name without any location information, the API expects that file to be available in the current directory of the application. If the file is not available at that location, the call to `sqliniEx` will fail.

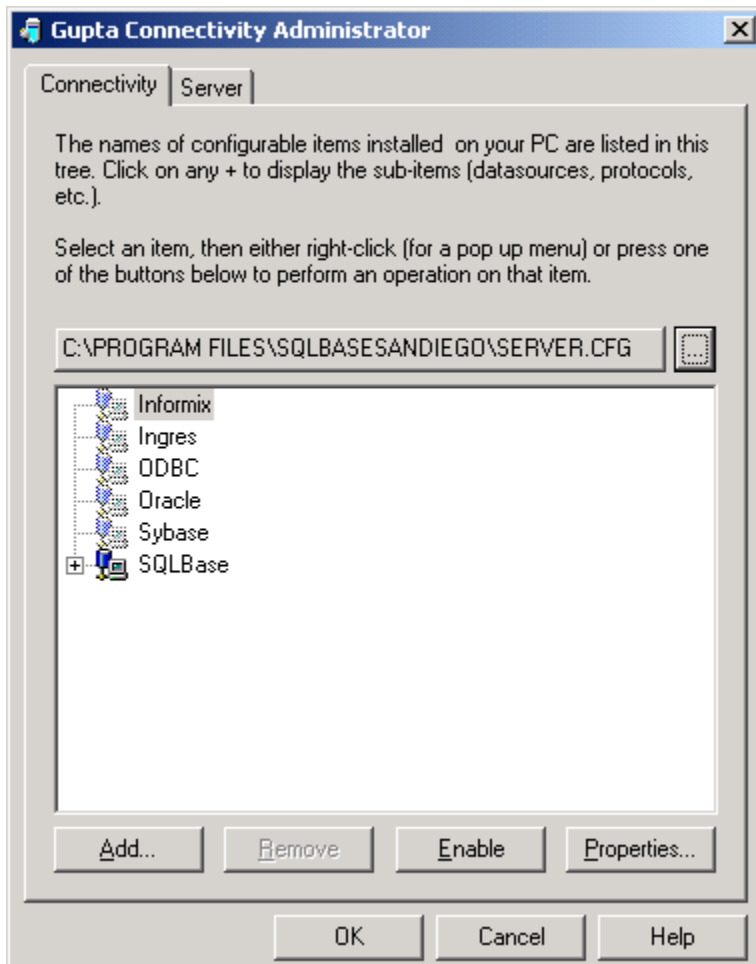
Programmers can call this method multiple times in a program. The requirement is that before a second or subsequent call to `sqliniEx`, method `sqldon` should have been called. If `sqldon` was not called prior, the call to `sqliniEx` will return an error.

Connectivity Administrator

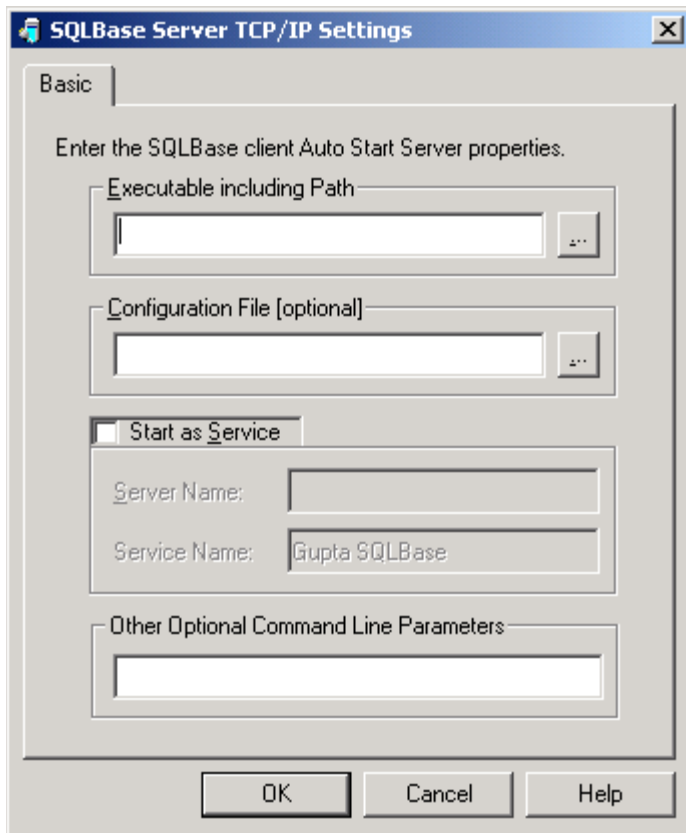
Connectivity Administrator now displays all installed servers, regardless of version.



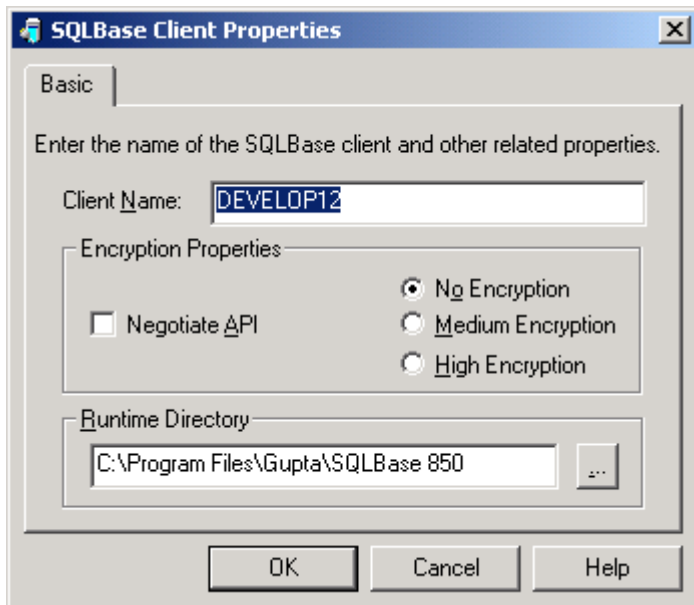
On the Client tab, a new pushbutton enables you to specify the name and path of the configuration file to be edited in your current session, should you want to change it from the default value chosen when Connectivity Administrator starts. In the following example, the path has been left unchanged, but the file name is now SERVER.CFG.



Connectivity Administrator now enables you to manage multiple servers on the client side for the local, SPX, and TCP/IP protocols. New right-click menu options on the Connectivity tab allow you to add server names to a specific protocol and to invoke the Properties dialog for a listening protocol. The 8.5 enhancement keyword `AUTOSTARTSERVERPATH` is accessible through this dialog, as shown in following example.



The client properties dialog now enables you to specify the runtime directory (new keyword CLIENTRUNTIMEDIR) where SQLBase executables can be found.



Database Drivers and Providers

Even though a user can have multiple SQL/API clients installed in a machine, only the most recently installed ODBC, OLE DB & .NET drivers and providers will be known to the system. All these drivers are registered with Windows upon installation, and each installation will overwrite previous information.

The OLE DB data provider now supports multiple installs with a new property that supports specifying a configuration file before connecting to the data source. To make use of this feature, new property TDataSource.DBPROP_INIT_PROVIDERSTRING must be given a valid string value. The format of this string is the following:

```
INI=<Configuration Filename with Full Path>;ims=<Value>;oms=<Value>
```

ims (input message buffer) and oms (output message buffer) must have values between 1 and 32000.

You can also specify the string in the UDL/Data Link property dialog box. The configuration file name with the above format could be entered as a value for "Extended Properties" in the "All" page.

SQLTalk

A new command-line argument for SQLTalk supports specifying a configuration file before starting a session. This choice is available in a dialog box that is accessed from menu items Options, Session Settings. You can also supply the file name as a command-line argument; for example:

```
SQLTalk.exe "c:\program files\km.sql" "ini=c:\program files\sql.ini"
```

The example shows a SQL script as the first argument and the configuration file as the second argument. Both are optional.

SQLBase Management Console

SMC has changed to reflect the existence of multiple installations. All the operations available in earlier versions of SMC (starting and stopping servers, etc..) are available with any of the installed servers. In addition, SMC allows you to search for SQLBase executables and register the executables found as Windows services. You can specify the name and location of the configuration file as part of the registration process. You can unregister existing server installations. You can also rename servers using SMC in version 8.5. The online help in SMC has full details on these new features.

SQL99 ANSI Join Syntax Support

Version 8.5 offers extensive support for the SQL99 ANSI join syntax.

New keywords

Supported elements that are new to this version include: Keywords NATURAL, JOIN, ON, and USING Keywords INNER and OUTER and Keywords LEFT and RIGHT.

SQL99 ANSI clauses CROSS JOIN and FULL OUTER JOIN are not supported in SQLBase 8.5.

SQL99 ANSI Join Support is Optional

The new keywords could potentially conflict with the names of tables, views, columns, or other database objects in your existing databases. For this reason, SQL99 ANSI join compatibility is optional—not automatic—for a specific SQLBase server.

You can select it at installation, select/deselect it later through Connectivity Administrator, or control it using the new SQL.INI keyword "AnsiJoinSyntax".

Keyword Check Tool

If you are unsure whether the new keywords are already being used for other purposes in your existing databases, you can use the new Keyword Check tool to analyze those databases and detect potential keyword conflicts. You will find Keyword Check in the Gupta, SQLBase program group in the Start menu.

API Enhancements

The SQLBase API has been enhanced so that client applications can query whether the server to which they are connected is currently supporting SQL99 ANSI join compatibility, and then use this information to construct their queries. Call function sqlget with parameter SQLPAJS to retrieve 0 (unsupported) or 1 (supported).

Multiple Outer Joins

If you elect to use ANSI join syntax, you may also use multiple outer joins in queries. This feature is not available when using SQLBase native syntax.

Database Engine Performance

SQLBase 8.5 is much faster at sorting than previous versions. You will notice improved query performance, particularly for queries using GROUP BY or other features involving sorting.

The maximum size of the database cache has been increased from 32,767 pages (a page is one kilobyte) to 1,000,000 pages. There is a minimum size requirement of 15 pages. If your system hardware has sufficient memory to support the increased cache size, you will see improved performance. To set the cache size, use Connectivity Administrator to edit the cache keyword setting in SQL.INI.

Other areas of improvement include lock management and query optimization.

SQL Language Enhancements

The new @COALESCE function returns the first non-null value found in a list of two or more values. The values in the list can be mixed data types. The return value's data type matches the data type of the first value in the list. If every value in the list is null, the function returns NULL.

For example, assume that the column AUTHOR_NAME does not contain a value of 'Chopin' and the @DECODE expression does not contain a default. In such a case, the following expression returns the string 'ABC' (the second value in the list) because the first value evaluates to NULL:

```
@COALESCE(@DECODE(AUTHOR_NAME, 'Chopin', '007'), 'ABC', 'more')
```

JDBC Driver Enhancements

The JDBC driver is built with, and supports, JKD version 1.3.1.

Before the JDBC sample programs can be run, an additional SQL script must be run against the ISLAND sample database. For more information, see README.HTML in the samples/jdbc folder beneath your database server install folder.

JDBC Driver setTransactionIsolation() functional behavior has been changed to map isolation as follows:

| | |
|------------------------------|---------------|
| TRANSACTION_NONE | "RL" |
| TRANSACTION_REPEATABLE_READ | "RR" |
| TRANSACTION_READ_COMMITTED | "CS" |
| TRANSACTION_SERIALIZABLE | Not Supported |
| TRANSACTION_READ_UNCOMMITTED | Not Supported |

The JDBC Driver connect URL has been extended to specify the SQLBase IMS (InputMessageSize) and OMS (OutputMessageSize) values. These are optional. The new URL format looks like the following:

```
jdbc:sqlbase://localhost[:2157]/island;ims=dddd;oms=dddd
```

- ims and oms are case-insensitive keywords
- ims and oms are optional parameters and can be any order and positioned after the database name, separated by semicolons.

The value of dddd should be greater than zero and less than or equal to 32000. If the value specified does not fall in this range, default values IMS=2000 and OMS=1000 are set.

New API Methods

The SQLBase JDBC driver now supports a larger number of JDBC 1.2 API functions. Here is a summary of the methods that are newly supported by SQLBase 8.5.

Class SqlbaseStatement

(inherits from Statement)

- int getResultSetConcurrency

- int getResultSetType()
- Connection getConnection()

Class `SqlBaseConnection`

(inherits from `Connection`)

Statement `createStatement(int resultSetType, int resultSetConcurrency)`

- `PreparedStatement` `prepareStatement(String sql, int resultSetType, int resultSetConcurrency)`
- `CallableStatement` `prepareCall(String sql, int resultSetType, int resultSetConcurrency)`

Class `SqlBaseResultSet`

(inherits from `ResultSet`)

- int getType()
- int getConcurrency()
- Statement getStatement()

Class `SqlBaseDatabaseMetaData`

(inherits from `DatabaseMetaData`)

- boolean supportsResultSetType(int type)
- boolean supportsResultSetConcurrency(int type, int concurrency)
- boolean ownUpdatesAreVisible(int type)
- boolean ownDeletesAreVisible(int type)
- boolean ownInsertsAreVisible(int type)
- boolean othersUpdatesAreVisible(int type)
- boolean othersDeletesAreVisible(int type)
- boolean othersInsertsAreVisible(int type)
- boolean updatesAreDetected(int type)
- boolean deletesAreDetected(int type)
- boolean insertsAreDetected(int type)
- boolean supportsBatchUpdates()
- Connection getConnection()

OLE DB Data Provider Changes

The OLE DB Data Provider datatypes have changed. The provider data type "long" has been renamed "long varchar" and the type has been changed from `DBTYPE_BYTES` to `DBTYPE_STR`. The datatype for "number" has been changed from `DBTYPE_NUMERIC` to `DBTYPE_R8`, to be consistent with the `SQLBASE` ODBC driver. `FLOAT` type has been removed from the providers list. The default value of property `DBPROP_UPDATABILITY` has been set to 0. Client applications should set the value to one or more of `DBPROPVAL_UP_CHANGE |`

DBPROPVAL_UP_DELETE | DBPROPVAL_UP_INSERT. Otherwise the methods return DB_E_NOTSUPPORTED depending on the value set.

OLEDB Provider now supports ODBC escape sequences in the SQL statement.

A new argument in the DBPROP_INIT_PROVIDERSTRING property allows you to specify a particular configuration file as part of the connection. The argument should take the form "ini=filename". The filename can include a path specification.

.NET Data Provider Changes

.NET Data Provider now supports ODBC escape sequences in the SQL statement. Enhancements to the .NET Data Provider now support scrollable result sets. To make use of this feature, new property `SQLBaseCommand.ResultSetMode` must be set to `TRUE`. The `SQLBaseDataReader` object now implements interface `IEnumerable`, with a single method, `GetEnumerator`, which returns an `IEnumerator` object. That object has methods `MoveNext` and `Reset`, and `get/set` property `RowPos` (current row of result set), which allow you to move to a specific row in the result set.

A new ini argument in the connect string allows you to specify a particular configuration file as part of the connection. The following is a Visual Basic example:

```
SQLBaseConnection conn = new SQLBaseConnection(  
"data source=island; uid=sysadm; pwd=sysadm; ini=c:\\sqlbase\\sql.ini");
```

ODBC Driver Changes

ODBC Driver property `DBPROP_SQLSUPPORT` value returned has been changed. For versions prior to 8.5 it always returns 0. For version 8.5, if the server is running with `ANSIJOINSYNTAX` value set to 1, the value returned will be `DBPROPVAL_SQL_ANSI92_ENTRY`; otherwise it will be 0.

ODBC Driver property `SQL_OJ_CAPABILITIES` value returned has been changed. If the server is running with `ANSIJOINSYNTAX` value set to 1, then value will have the additional bit masks `SQL_OJ_NESTED` | `SQL_OJ_FULL`.

The ODBC Driver "Gupta SQLBase Data Source Properties" dialog box has been extended to allow you to optionally enter the configuration filename.