



**GUPTA® SQLBase® NLS Migration to SQLBase  
International**

By Ana Paula Bonani  
Technical Support Engineer

June, 2006

**GUPTA™**

<b>Abstract</b> .....	<b>3</b>
<b>Introduction</b> .....	<b>3</b>
<b>What is SQLBase NLS?</b> .....	<b>4</b>
<b>How Unicode features replace NLS?</b> .....	<b>4</b>
<b>COLLATION keyword</b> .....	<b>6</b>
<b>CHARACTERSET keyword</b> .....	<b>6</b>
<b>How to migrate data from NLS to SQLBase International</b> .....	<b>6</b>
<b>Recommendations</b> .....	<b>7</b>

## **Abstract**

This paper will describe National Language Support (NLS) features of SQLBase versions prior to SQLBase International and explain how to migrate the database to conform to the new provided Unicode features.

## **Introduction**

SQLBase versions prior to SQLBase International have always supported National Language Support (NLS) so customers could define a language other than English as their database standard language.

SQLBase International introduces Unicode support plus other capabilities which handle the features previously provided by NLS and support an unlimited number of languages instead of a single language per database server. For that reason NLS is no longer available.

Therefore customers using NLS databases will need to migrate them when moving to SQLBase International.

**NLS defines the use of double-byte character settings among other features**

## What is SQLBase NLS?

English is the standard language supported by SQLBase. NLS was introduced to provide a way for customers around the world to create databases built upon other languages as well.

In order to achieve this purpose, databases should use country-specific settings defined in a file called `country.sql`. This file contained information necessary to customize SQLBase for non-English languages. Other settings in files like `sql.ini`, `error` and `message.sql` would also compound the steps required to enable National Language Support from SQLBase.

Some NLS features include:

1. Specify double-byte characters for countries using double-byte character settings
2. Specify additional alphabetic and numeric characters allowed in column, table and index names
3. Specify alternative ways for converting a string to lowercase
4. Specify alternative ways for sorting characters or strings of characters

However, SQLBase's NLS has a limitation: mixing different country databases on the same server is not supported. In other words, all databases on a server must support the same language.

But in today's world of internationalized applications, this is not acceptable. Databases are required to support several languages and this is where Unicode comes into picture.

## How Unicode features replace NLS?

SQLBase International introduces Unicode support. This means that it now not only supports the features previously provided by NLS in an easier and more flexible way but it also introduces new related features.

For instance, here's a comparison between the NLS features presented in the above section and the new ones from SQLBase International:

1. [Specify double-byte characters for countries using double-byte character settings](#)  
✓ With SQLBase International it's now possible to store and retrieve Unicode data from the database automatically. This means that it now recognizes characters outside of the ASCII set with no extra database configuration efforts.
2. [Specify additional alphabetic and numeric characters allowed in column, table and index names](#)  
✓ SQLBase International automatically allows that characters outside the ASCII set are used both for data and object names, for example table names, columns, indexes, etc.

**The need for NLS features has been eliminated with the introduction of Unicode support**

3. Specify alternative ways for converting a string to lowercase

√ With SQLBase International this is not necessary anymore as it provides built-in support for upper and lower casing of all Unicode characters as well as characters in the ASCII set.

4. Specify alternative ways for sorting characters or strings of characters

√ SQLBase International introduces the concept of Collations. Collations are sets of rules applied to define how characters compare to each other. With NLS, SQLBase only supported a binary collation for which the numeric value of a certain character was compared to the numeric value of another character. This limitation is now eliminated as with SQLBase International customers can choose among several other collations as well.

Also, with NLS it was possible to set up a specific country.sql file containing a "translate" section which defined the order to use for sorting. The disadvantage of that process was that it forced all databases on the server to use the same sorting order. With SQLBase International customers have total flexibility when it comes to sorting as they can now define specific collations even at column level.

Please refer to the "COLLATION Keyword" section below for more information on SQLBase International collations.

Besides, SQLBase International has also introduced new data types like NCHAR and NVARCHAR in order to store national characters. With that, the need for Multiple-Byte Character Sets has been eliminated. Those new data types can also be used in indexes, stored procedures, stored commands and triggers.

All of the above features are part of SQLBase International and available for your use after installing the product. This means that all the steps previously required to setup and configure a NLS database are now eliminated and software applications accessing SQLBase International can now be internationalized without any configuration efforts.

## **COLLATION keyword**

Collations can be assigned at server and client side and are used whenever SQL queries involving string comparison are executed. If no collation is assigned, the BINARY collation is used.

To assign a collation to be used at server side, use the keyword COLLATION in the server portion of the sql.ini file. To assign a client side collation, use the COLLATION keyword in the client portion of the sql.ini.

Client applications can change collations during run-time by calling the SQLSET SQLBase API function with the SQLPCCOL parameter.

## **CHARACTERSET keyword**

When added to the server portion of the sql.ini file the new CHARACTERSET keyword serves as the default character set used by database operations like the @CHAR function. This is also true when the client's character set cannot be determined, for instance in the case of an older SQLBase client connecting to SQLBase International.

When added to the client portion of the sql.ini it defines the character set used by client applications.

If SQLBase can't find this keyword in the server or client portion of the sql.ini it automatically selects the character set in use by the operating system. This is actually the recommended way for customers using SQLBase International both on the server and the client. Customers accessing SQLBase International from older clients please see "Recommendations" section below.

Customers familiar with NLS databases might find it useful to compare the server side CHARACTERSET keyword to the old COUNTRY setting, as they're similar in use.

## **How to migrate data from NLS to SQLBase International**

Here are the steps to migrate NLS databases to a SQLBase International server:

1. UNLOAD your NLS database;
2. If your NLS database had binary data stored in character columns, edit the UNLOAD file to change those columns datatypes to one of the new binary datatypes introduced in SQLBase International;
3. On the SQLBase International server, add the keyword CHARACTERSET to the sql.ini. Assign a value equivalent to the COUNTRY setting specified in your NLS database server;
4. If you're using a SQLBase API prior to 10.0.0 on the client machines, make sure that their character set matches the server character set;
5. LOAD the data into SQLBase International.

**Kommentar [cm1]:** This section will need to be modified once the migration tool becomes available.

Alternative migration process using Collate when creating the SQLBase 10 database. This process gives you more flexibility and control over the default collation used in a SQLBase 10 database.

1. UNLOAD your NLS database;
2. If your NLS database had binary data stored in character columns, edit the UNLOAD file to change those columns datatypes to one of the new binary datatypes introduced in SQLBase International (Varchar -> binary, long Varchar -> long binary);
3. If you are using pre SQLBase 10 clients, add the keyword CHARACTERSET to the server sql.ini. Assign a value equivalent to the COUNTRY setting specified in your NLS database server;
4. If you're using a SQLBase API prior to 10.0.0 on the client machines, make sure that their character set matches the server character set;
5. Create a new SQLBase 10 database in SQLTalk using the collate keyword: create database newdb collate german\_cs\_as; Where german\_cs\_as is a sample for a German collation. See the SQLBase 10 documentation for a list of all available collations.
6. LOAD the data into the newly created SQLBase International database.

### Recommendations

Customers who want to connect to a SQLBase International database server using a SQLBase client API prior to version 10.0.0 are recommended to add the CHARACTERSET keyword to the server portion of the sql.ini configuration file.

For clients older than 10.0.0, SQLBase International server expects the message traffic to be in the server character set. If the server character set is not specified in the sql.ini, the character set of the operating system is used. So if data sent by clients do not match the character set in use by the server, the results are unpredictable.

Either setting the CHARACTERSET keyword appropriately or upgrading client applications to SQLBase API 10.0.0 will solve the problem.

Please note that connecting to SQLBase International from old NLS clients is not supported.