



Administration: Globalization

SAP Sybase IQ 16.0 SP01

DOCUMENT ID: DC01772-01-1601-01

LAST REVISED: May 2013

Copyright © 2013 by SAP AG or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices.

Contents

About International Language Data	1
International Languages and Character Sets	1
What is ICU, and when is it needed?	2
Character sets	3
Code Pages in Windows	4
How the Collation Sequence Sorts Characters	7
Collations	8
SAP Sybase IQ Collation Algorithm (SACA)	8
Unicode Collation Algorithm (UCA)	9
Collations in an SAP Sybase IQ database	11
Alternate collations	12
Turkish character sets and collations	15
Locales	18
Locale language	18
Locale character set	20
Understanding Character Set Translation	20
Character Translation for Database Messages	20
Connection Strings and Character Sets	21
Avoiding Character Set Translation	22
ANSI and OEM Code Pages	23
ANSI ISO_1 Collation	23
ANSI 1252LATIN1 Collation	23
ANSI ISO1LATIN1 Collation	23
ANSI ISO9LATIN1 Collation	24
Displaying Collations	25
Multibyte Collations	27
Japanese Language Support	27
Thai Language Support	27
Locale information	29
Selecting a Collation to Support a Specific Locale	31

Choosing Character Sets and Collation Sequences to Maximize Performance	33
Setting a Locale	35
List of Language Label Values	35
Setting the Locale for an INSERT...LOCATION Statement	36
Disabling Character Set Translation on a Database Server	39
Creating a Database with the Default Collation	41
Changing the database collation	43
Index	45

About International Language Data

When you create a database, you specify a collating sequence, or *collation*, to be used by the database. A collation is a combination of a *character set* and a *sort order* for characters in the database.

International Languages and Character Sets

Configure your SAP® Sybase® IQ installation to handle international language issues.

The database collation sorts and compares all character data types in the database, including object names, such as table and column names.

SAP Sybase IQ takes advantage of the space efficiency and speed of the SAP® Sybase SQL Anywhere Collation Algorithm.

- The database option `Sort_Collation` allows implicit use of the **SORTKEY** function on **ORDER BY** expressions. When the value of this option is set to a valid collation name or collation ID, any string expression in the **ORDER BY** clause is automatically treated as if the **SORTKEY** function has been invoked.
- The **SORTKEY** function uses the International Components for Unicode (ICU) library, instead of the Sybase Unicode Infrastructure Library (Unilib®). Sort-key values created using a version of SAP Sybase IQ earlier than 15 do not contain the same values created using later versions. Regenerate any sort key values in your database that were generated using a version of SAP Sybase IQ earlier than 15.
- The **CREATE DATABASE** parameter **COLLATION** supports specification of a collation for a database.
The collation of the database must match the collation used by the operating system. The default character set for **CREATE DATABASE** is `ISO_BINENG`.
- The **CP874toUTF8** utility converts data in the CP874 character set into UTF8 collation, supported by SAP Sybase IQ for the Thai language. The **CP874toUTF8** utility calls the ICU library to perform data conversion. You can also use this utility to load data in the CP874 character set without converting the data to UTF8.

SAP Sybase IQ no longer supports custom collations. If you rebuild a database with a custom collation, rebuild in a single step to preserve the custom collation. If you unload the database and then load the schema and data into a database you create, you must use one of the supplied collations.

For more information about changes to database collations, and a list of collations deprecated in SAP Sybase IQ 15, see *New Features in SAP Sybase IQ 15.0*.

Use the **iqunload** utility to migrate to the current version of SAP Sybase IQ from an existing 12.7 database that was created with a deprecated collation.

What is ICU, and when is it needed?

ICU, or International Components for Unicode, is an open source library developed and maintained by IBM. ICU facilitates software internationalization by providing Unicode support. SAP Sybase IQ implements certain character set conversions and collation operations using ICU.

When is ICU needed on the database server? (all platforms except Windows Mobile)

Ideally, ICU should always be available for use by the database server. The following table specifies when and why ICU is needed:

ICU is needed when...	Notes
Unicode Collation Algorithm (UCA) is used as the collation for the NCHAR or CHAR character set.	UCA requires ICU.
The database character set is not UTF-8, but is a multi-byte character set.	For password conversion from the database character set to UTF-8 (database passwords are stored in UTF-8, internally).
The client and database character sets are different, and when either of them is multi-byte (including UTF-8). This includes Unicode ODBC, OLE DB, ADO.NET, and SAP Sybase IQ JDBC applications, regardless of the database character set where at least one of these clients does not have ICU.	Proper conversion to and from a multi-byte character set requires ICU.
The database character set is not UTF-8 and conversion between CHAR and NCHAR values is required.	The database server requires ICU to convert UTF-8 to another character set.
An embedded SQL client uses an NCHAR character set other than UTF-8.	The database server requires ICU to convert UTF-8 to another character set. The default embedded SQL client NCHAR character set is the same as the initial client CHAR character set. This can be changed using the <code>db_change_nchar_charset</code> function.
The CSCONVERT or SORTKEY functions are used. The CSCONVERT function is called to convert between character sets that conform to the requirements of the third point above.	Character set conversion to and from a multi-byte character set requires ICU. Sortkey generation for many sortkey labels requires UCA, which, in turn, requires ICU.

When can I get correct character set conversion on the database server without ICU?

You can get correct character set conversion without ICU when both the database character set and client character set are single-byte and `sqlany.cvf` is available (all platforms), or if the operating system supports the conversion (Windows only). This is because single-byte to single-byte conversions can be processed without ICU if the `sqlany.cvf` file is available, or the host operating system has the appropriate converters installed.

When is ICU needed on the client? (all platforms)

For Unicode client applications, you are likely to get better combined client and database server performance when all clients have ICU installed, regardless of the database character set. This is because some of the required conversion activity may be offloaded from the database server to the client, and because fewer conversions are required.

Also, if you are using ODBC on Windows platforms, you must have ICU installed on the client, even for ANSI applications. This is because the driver manager converts ANSI ODBC calls to Unicode ODBC calls.

Character sets

Each piece of software works with a **character set**. A character set is a set of symbols, including letters, digits, spaces, and other symbols. An example of a character set is ISO-8859-1, also known as Latin1.

To properly represent these characters internally, each piece of software employs an **encoding**, also known as **character encoding**. An encoding is a method by which each character is mapped onto one or more bytes of information, and is presented as a hexadecimal number. An example of an encoding is UTF-8.

Sometimes the terms character set and encoding are used interchangeably, since the two aspects are so closely related.

A **code page** is one form of encoding. A code page is a mapping of characters to numeric representations, typically an integer between 0 and 255. An example of a code page is Windows code page 1252.

In this documentation, the terms encoding, character encoding, character set encoding, and code page are synonymous.

Database servers, which sort characters (for example, listing names alphabetically), use a **collation**. A collation is a combination of a character encoding (a map between characters and their representation) and a **sort order** for the characters. There may be more than one sort order for each character set; for example, a case sensitive order and a case insensitive order, or two languages may sort the same characters in a different order.

About International Language Data

Characters are printed or displayed on a screen using a **font**, which is a mapping between characters in the character set and their appearance. Fonts are handled by the operating system.

Operating systems also use a **keyboard mapping** to map keys or key combinations on the keyboard to characters in the character set.

Code Pages in Windows

Many languages have few enough characters to be represented in a single-byte character set. In such a character set, each character is represented by a single *byte*: a two-digit hexadecimal number.

A maximum of 256 characters can be represented in a single byte. No single-byte character set can hold all of the characters used internationally, including accented characters. This problem is addressed through the use of *code pages*, each of which describes a set of characters appropriate for one or more national languages. For example, code page 869 contains the Greek character set, and code page 850 contains an international character set suitable for representing many characters in a variety of languages.

ANSI and OEM code pages in Windows

For Windows users, there are two code pages in use. Applications using the Windows graphical user interface use the Windows code page. Windows code pages are compatible with ISO character sets, and also with ANSI character sets. They are often referred to as **ANSI code pages**.

Character-mode applications (those using a command prompt window) in Windows use code pages that were used in DOS. These are called **OEM code pages** (Original Equipment Manufacturer) for historical reasons.

SAP Sybase IQ supports collations based on both OEM and ANSI code pages. The OEM collations are provided for compatibility, but they should not be used for new databases.

Recommended character sets and collations

While SAP Sybase IQ recognizes the names of hundreds of character sets, code pages, encodings, and collations, this topic lists those that are recommended for use with Windows and Unix platforms, depending on the language in use.

Use the `iqinit -le` option to obtain a list of all the available character set labels for an SAP Sybase IQ database. Use the `iqinit -l` option to obtain a list of available collations for an SAP Sybase IQ database.

Where a character set encoding or label must be specified, use the value from the Character set label column or one of the labels listed by `iqinit -le`. Where a collation must be specified, use the value from the Collation or Alternate collation column or one of the labels listed by `iqinit -l`.

Note: For languages not found in the tables below, the UTF-8 encoding should be used with collations UCA or UTF8BIN.

Windows platforms

Language	Windows Code Page	Character set label	Collation	Alternate collation
Arabic	1256	Windows-1256	1256ARA	
Central and Eastern European	1250	Windows-1250	1250LATIN2	
Danish	1252	Windows-1252	1252LATIN1	
Dutch	1252	Windows-1252	1252LATIN1	
English	1252	Windows-1252	1252LATIN1	
Finnish	1252	Windows-1252	1252SWEFIN	
French	1252	Windows-1252	1252LATIN1	
German	1252	Windows-1252	1252LATIN1	
Greek	1253	Windows-1253	1253ELL	
Hebrew	1253	Windows-1253	1255HEB	
Italian	1252	Windows-1252	1252LATIN1	
Japanese	932	Windows-31J	932JPN	
Korean	949	IBM949	949KOR	
Lithuanian	1257	Windows-1257	1257LIT	
Norwegian	1252	Windows-1252	1252NOR	
Polish	1250	Windows-1250	1250POL	
Portuguese	1252	Windows-1252	1252LATIN1	
Russian	1251	Windows-1251	1251CYR	
Simplified Chinese	936	GBK	936ZHO	
Spanish	1252	Windows-1252	1252SPA	
Swedish	1252	Windows-1252	1252SWEFIN	
Thai	874	TIS-620	874THAIBIN	

About International Language Data

Language	Windows Code Page	Character set label	Collation	Alternate collation
Traditional Chinese - Hong Kong	950	Big5-HKSCS	950ZHO_HK	
Traditional Chinese - Taiwan	950	Big5	950ZHO_TW	
Turkish	1254	Windows-1254	1254TRK	1254TRKALT
Ukrainian	1251	Windows-1251	1251CYR	
Western European	1252	Windows-1252	1252LATIN1	

Unix platforms

Language	Character set label	Collation	Alternate collation
Arabic	ISO_8859-6:1987	UCA	
Central and Eastern European	ISO_8859-2:1987	UCA	
Danish	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
Dutch	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
English	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
Finnish	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
French	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
German	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
Greek	ISO_8859-7:1987	UCA	
Hebrew	ISO_8859-8:1988	UCA	
Italian	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
Japanese	EUC-JP ¹	EUC_JAPAN	
Korean	EUC-KR	EUC_KOREA	
Lithuanian	(use UTF-8)	UCA or UTF8BIN	
Norwegian	ISO-8859-15	ISO9LATIN1	ISO1LATIN1

Language	Character set label	Collation	Alternate collation
Polish	ISO_8859-2:1987	UCA	
Portuguese	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
Russian	ISO_8859-5:1988	UCA	
Simplified Chinese	GB2312	EUC_CHINA	
Spanish	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
Swedish	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
Thai	(use UTF-8)	UCA or UTF8BIN	
Traditional Chinese - Hong Kong	Big5-HKSCS	950ZHO_HK	950TWN
Traditional Chinese - Taiwan	EUC-TW	EUC_TAIWAN	
Traditional Chinese - Taiwan	Big5	950ZHO_TW	
Turkish	ISO_8859-9:1989	920TRK	
Ukrainian	ISO_8859-5:1988	UCA	
Western European	ISO-8859-15	ISO9LATIN1	ISO1LATIN1

¹ EUC-JP is an alternate label for the SAP Sybase IQ label Extended_UNIX_Code_Packed_Format_for_Japanese.

How the Collation Sequence Sorts Characters

The database collation sequence includes alphabetic ordering of letters, which extends to all characters in the character set, including digits and space characters.

More than one character can be associated with each sort position. This is useful, for example, to treat an accented character the same as the character without an accent.

Two characters with the same sort position are considered by the database to be identical in all ways. Therefore, if a collation assigned the characters 'a' and 'e' to the same sort position, then this query:

```
WHERE coll = 'want'
```

is satisfied by a row for which **col1** contains the entry **went**.

At each sort position, lowercase and uppercase forms of a character can be indicated. For case-sensitive databases (the default for SAP Sybase IQ databases), lowercase and uppercase

characters are not treated as equivalent. For case-insensitive databases, lowercase and uppercase versions of the same character are considered equivalent.

Note: Any code that selects a default collation for a German system should select 1252LATIN1, not 1252DEU. 1252DEU differentiates between characters with and without an umlaut, while 1252LATIN1 does not. 1252LATIN1 considers Muller and Müller equal, but 1252DEU does not consider them equal. Because 1252DEU views characters with umlauts as separate characters, it uses this alphabetic ordering: ob, öa.

First-Byte Collation Orderings for Multibyte Character Sets

A sorting order for characters in a multibyte character set can be specified only for the first byte.

Characters that have the same first byte are sorted according to the hexadecimal value of the following bytes.

Collations

A collation describes how to sort and compare characters from a particular character set or encoding. SAP Sybase IQ supports two collation algorithms: the SAP Sybase IQ Collation Algorithm (SACA), and the Unicode Collation Algorithm (UCA). SACA provides fast, compact, and reasonable sorting at the expense of linguistic correctness. UCA provides linguistic correctness, but with a small expense in storage requirements and execution time.

SAP Sybase IQ Collation Algorithm (SACA)

The SAP Sybase IQ Collation Algorithm (SACA) provides reasonable comparison, ordering, and case conversion of single-byte and multibyte character sets. The algorithm is space efficient and fast. The mapped form of a string, such as an index, is the same length as the original string. The mappings for comparison, ordering, and case conversion use a simple table lookup of each byte value of the string.

Single-byte character sets

In a typical collation for a single-byte character set, all accented and unaccented forms of a character are mapped to the same value, making the collation accent insensitive. Accented and unaccented forms of the same letter compare as exactly equal and sort near each other.

The collation also provides conversion between uppercase and lowercase letters, preserving accents.

Multibyte character sets

In multibyte character sets, the lead-bytes are mapped into the 256 distinct values. Follow bytes are compared using their binary value.

For most collations for multibyte character sets, this mapping technique provides a reasonable ordering because the character set encoding groups characters into 256-byte pages identified

by the lead byte. The pages, and the characters within each page, are in a reasonable order in the character set. The collations typically preserve the ordering of the pages (lead bytes) within the character set. Some pages may be ordered by other characteristics. For example, the 932JPN collation provided for Japanese code page 932 groups the full-width (Kanji) and half-width (katakana) characters.

Case conversion is provided only for the 7-bit English characters.

UTF-8 character sets

UTF-8 is a multibyte character set. Each character contains from one to four bytes. SAP Sybase IQ provides the UTF8BIN collation for sorting UTF-8 characters.

In UTF8BIN, lead bytes are mapped into 256 distinct values, and follow bytes are compared using their binary values. Because of the representation of characters in UTF-8 and the limitation of 256 distinct mapping values, it is not possible to group related characters such as accented and unaccented forms of the same letter. The ordering is essentially binary.

Case conversion is supported only for the 7-bit English characters.

Unicode Collation Algorithm (UCA)

The Unicode Collation Algorithm (UCA) is an algorithm for sorting the entire Unicode character set. It provides linguistically correct comparison, ordering, and case conversion. The UCA was developed as part of the Unicode standard. SAP Sybase IQ implements the UCA using the International Components for Unicode (ICU) open source library, developed and maintained by IBM.

Note: The default UCA ordering sorts most characters in most languages into an appropriate order. However, because of the sorting and comparison variations between languages sharing characters, the UCA cannot provide proper sorting for all languages. For this purpose, ICU provides a syntax for tailoring the UCA.

The UCA provides advanced comparison, ordering, and case conversion at a small cost in space and time.

The mapped form of a string is longer than the original string. The algorithm provides sophisticated handling of more complex characters.

Unlike the SAP Sybase IQ Collation Algorithm (SACA) the Unicode Collation Algorithm (UCA) is only for use with single-byte and UTF-8 character sets, and it separates each character into one or more attributes. For letters, these attributes are base character, accent, and case.

Non-letters typically have only one attribute, the base character.

UCA compares character strings as follows:

- Compare the base characters. If one string of base characters differs from the other, then the comparison is complete. Accent and case are not considered.

About International Language Data

- If the database is accent sensitive, compare the accents. If the accents differ, then the comparison is complete. Case is not considered.
- If the database is case sensitive, compare the case of each character.

The original string values are equal if and only if the base characters, accents, and case are the same for both strings.

Example

Suppose UCA is used to compare the strings in the first column of the table below. The subsequent columns describe the three attributes for each string. Notice that the base characters are identical; the words differ only in accents and case.

String	Base characters	Accents	Case
noel	noel	none, none, none, none	lower, lower, lower, lower
noël	noel	none, none, accent, none	lower, lower, lower, lower
Noel	noel	none, none, none, none	upper, lower, lower, lower
Noël	noel	none, none, accent, none	upper, lower, lower, lower

The following table shows the ordering that would occur in the four possible combinations of accent- and case-sensitivity using UCA:

Accent sensitive	Case sensitive	ORDER BY result	Explanation
N	N	Noel, Noël, Noël, noel in any order	<ul style="list-style-type: none"> • Accents ignored • Case ignored • All values considered equal • Random order within set of four
Y	N	Noel, noel in any order, followed by Noël, Noël in any order	<ul style="list-style-type: none"> • No-accent before accents, so e before ë • Case ignored, N and n are in random order within each set of two

Accent sensitive	Case sensitive	ORDER BY result	Explanation
N	Y	Noel, Noël in any order, followed by noël, noel in any order	<ul style="list-style-type: none"> • Uppercase before lowercase, so N before n • Accents ignored, e and ë are in random order within each set of two
Y	Y	Noel noel Noël noël	<ul style="list-style-type: none"> • No-accents before accents, so e before ë • Uppercase before lowercase, so N before n

Collations in an SAP Sybase IQ database

CHAR collation

CHAR data types, including CHAR, VARCHAR, and LONG VARCHAR, can use a collation that uses the SAP Sybase IQ Collation Algorithm (SACA) or they can use the Unicode Collation Algorithm (UCA). In either case, the collation used is referred to as the CHAR collation.

NCHAR collation

NCHAR data types, including NCHAR, NVARCHAR, and LONG NVARCHAR, can use the Unicode Collation Algorithm (UCA) or can use the UTF8BIN collation, which uses the SAP Sybase IQ Collation Algorithm (SACA).

Choosing case and accent sensitivity

When an SAP Sybase IQ database is created, if case sensitivity is not specified, then it is case insensitive. It can be made case sensitive by specifying the appropriate option. It is not possible to change the case sensitivity after the database has been created without rebuilding the database.

The case sensitivity for the database determines the case sensitivity for both the SACA and UCA collations, and so it also determines the case sensitivity of both the CHAR and NCHAR collations.

When an SAP Sybase IQ database is created, if accent sensitivity is not specified, then it is accent insensitive. It can be made accent sensitive by specifying the appropriate option. It is

About International Language Data

not possible to change the accent sensitivity after the database has been created without rebuilding the database.

The accent sensitivity for the database affects only the UCA collation, whether it is used for the CHAR or NCHAR collations or both. If you choose SACA collations for both CHAR and NCHAR collations, then the options for accent sensitivity have no effect. Accent sensitivity is an attribute of SACA collations and cannot be specified using the options provided when creating the database.

Alternate collations

The following table lists compatibility collations that can be used with the SORTKEY and COMPARE functions.

Collation label	Description
874THAIBIN	Code Page 874, Windows Thai, ISO8859-11, binary ordering
932JPN	Code Page 932, Japanese Shift-JIS with Microsoft extensions
936ZHO	Code Page 936, Simplified Chinese, PRC GBK 2312-80 8-bit encoding
949KOR	Code Page 949, Korean KS C 5601-1987 encoding, Wansung
950ZHO_HK	Code Page 950, Traditional Chinese, Big 5 encoding with HKSCS
950ZHO_TW	Code Page 950, Traditional Chinese, Big 5 encoding
1250LATIN2	Code Page 1250, Windows Latin 2, Central/Eastern European
1250POL	Code Page 1250, Windows Latin 2, Polish
1251CYR	Code Page 1251, Cyrillic
1252BIN	Code Page 1252, Windows Latin 1, binary ordering
1252LATIN1	Code Page 1252, Windows Latin 1, Western
1252NOR	Code Page 1252, Windows Latin 1, Norwegian
1252SPA	Code Page 1252, Windows Latin 1, Spanish

Collation label	Description
1252SWEFIN	Code Page 1252, Windows Latin 1, Swedish/Finnish
1253ELL	Code Page 1253, Windows Greek, ISO8859-7 with extensions
1254TRK	Code Page 1254, Windows Latin 5, Turkish, ISO 8859-9 with extensions
1254TRKALT	Code Page 1254, Windows Turkish, ISO8859-9 with extensions, I-dot equals I-no-dot
1255HEB	Code Page 1255, Windows Hebrew, ISO8859-8 with extensions
1256ARA	Code Page 1256, Windows Arabic, ISO8859-6 with extensions
1257LIT	Code Page 1257, Lithuanian
CESU8BIN	CESU-8, 8-bit multibyte encoding for Unicode, binary ordering
EUC_CHINA	Simplified Chinese GB 2312-80 Encoding
EUC_JAPAN	Japanese EUC JIS X 0208-1990 and JIS X 0212-1990 Encoding
EUC_KOREA	Korean KS C 5601-1992 Encoding, Johab
EUC_TAIWAN	Taiwanese Big 5 Encoding
ISO1LATIN1	ISO8859-1, ISO Latin 1, Western, Latin 1 Ordering
ISO9LATIN1	ISO8859-15, ISO Latin 9, Western, Latin 1 Ordering
ISO_1	ISO8859-1, Latin 1, Western
ISO_BINENG	Binary ordering, English ISO/ASCII 7-bit letter case mappings
UCA	Standard default UCA collation
UTF8BIN	UTF-8, 8-bit multibyte encoding for Unicode, binary ordering

Alternate collations

Alternate collations are available for compatibility with older versions of SAP Sybase IQ, or for special purposes. To see the full list of supported alternate collations, run the following command:

```
iqinit -l+
```

Adaptive Server Enterprise collations

The following table lists the supported Adaptive Server Enterprise collations for use with features such as the SORTKEY function.

Description	Collation name	Collation ID
Default Unicode multilingual	default	0
CP 850 Alternative: no accent	altnoacc	39
CP 850 Alternative: lowercase first	altdict	45
CP 850 Western European: no case, preference	altnocsp	46
CP 850 Scandinavian dictionary	scandict	47
CP 850 Scandinavian: no case, preference	scannocp	48
GB Pinyin	gbpinyin	n/a
Binary sort	binary	50
Latin-1 English, French, German dictionary	dict	51
Latin-1 English, French, German no case	nocase	52
Latin-1 English, French, German no case, preference	nocasep	53
Latin-1 English, French, German no accent	noaccent	54
Latin-1 Spanish dictionary	espdict	55
Latin-1 Spanish no case	espnoacs	56
Latin-1 Spanish no accent	espnoac	57
ISO 8859-5 Russian dictionary	rusdict	58

Description	Collation name	Collation ID
ISO 8859-5 Russian no case	rusnoc	59
ISO 8859-5 Cyrillic dictionary	cyrdict	63
ISO 8859-5 Cyrillic no case	cyrnoc	64
ISO 8859-7 Greek dictionary	elldict	65
ISO 8859-2 Hungarian dictionary	hundict	69
ISO 8859-2 Hungarian no accents	hunnoac	70
ISO 8859-2 Hungarian no case	hunnocs	71
ISO 8859-5 Turkish dictionary	turdict	72
ISO 8859-5 Turkish no accents	turnoac	73
ISO 8859-5 Turkish no case	turnocs	74
CP 874 (TIS 620) Royal Thai dictionary	thaidict	1
ISO 14651 ordering standard	14651	22
Shift-JIS binary order	sjisbin	179
Unicode UTF-8 binary sort	utf8bin	24
EUC JIS binary order	eucjisbn	192
GB2312 binary order	gb2312bn	137
CP932 MS binary order	cp932bin	129
Big5 binary order	big5bin	194
EUC KSC binary order	euckscbn	161

Turkish character sets and collations

The Turkish language has two forms of what appears to be the letter İ. One form, referred to as İ-dot, appears as the following:

ı, İ

The second form, referred to as İ-no-dot, appears as the following:

ı, İ

About International Language Data

Even though these letters appear as variations of the same letter, in the Turkish alphabet they are considered to be distinct letters. SAP Sybase IQ provides the Turkish collation 1254TRK to support these variations.

Turkish rules for case conversion of these characters are incompatible with ANSI SQL standard rules for case conversion. For example, Turkish says that the lowercase equivalent of **İ** is:

ı

However, the ANSI standard says that it is:

i

For this reason, correct case-insensitive matching is dependent on whether the text being matched is Turkish or English/ANSI. In many contexts, there is not enough information to make such a distinction, which leads to some non-standard behaviors in such databases.

For example, consider the following statements, executed against a database using the 1254TRK collation:

```
SELECT * FROM syshistory //actual table name is SYSHISTORY  
SELECT * FROM fig //actual name is FİG
```

The first statement references a system object, and ANSI SQL conversion rules are required to match the name. The second statement references a user object, and Turkish conversion rules are required to match the name. However, the database server cannot tell which conversion rules to use until it knows what the object is, and it cannot know what the object is until it knows what conversion rules to use. The situation cannot be resolved properly for both system and user objects. In this example, since the database server is using the Turkish collation 1254TRK, the first statement fails because lowercase I is not considered equivalent to uppercase I. The second statement succeeds.

The incompatibility of Turkish and ANSI standards requires that system object references in Turkish databases specify the object name in the correct case, that is, the case used to create the object. The first statement above should be written as follows:

```
SELECT * FROM SYSHISTORY
```

In fact, only the letter I must be in the correct case.

As an alternative, it is acceptable, although unusual, to write the statement as follows:

```
SELECT * FROM syshistory //I-no-dot
```

Keywords such as INSERT are case-insensitive, even in Turkish databases. SAP Sybase IQ knows that all keywords use only English letters, so it uses ANSI case conversion rules when matching keywords. SAP Sybase IQ also applies this knowledge for certain other identifiers, such as built-in functions. However, objects whose names are stored in the catalog must be specified using the correct case or letter, as described above. To determine the correct case, look in the system view that defines the system object.

Data in case-insensitive Turkish databases

Similar rules govern data in case-insensitive Turkish databases. For example if a data value is:

FIG

then a lowercase reference to that data should be:

fig

Then, the same I-dot character is used in both forms.

Alternative Turkish collation 1254TRKALT

For some application developers, the Turkish letter I problem can cause significant problems. The correct solution is to ensure that all object references are in the proper case or that the proper form of the letter I is used. However, it may be more expedient to violate the Turkish rules in favor of the ANSI rules.

SAP Sybase IQ provides the collation 1254TRKALT, which is identical to 1254TRK, except that it makes I-dot and I-no-dot equivalent characters.

It is important to understand the consequences of this change. In a 1254TRKALT database, the following strings are equal:

fig
fig

This is not correct for a Turkish user, but may be acceptable for other users.

The second issue appears when using ORDER BY. Consider the following strings:

ia
ia
is
is

In a 1254TRK database, an ORDER BY of the strings would produce the following:

ia
is
ia
is

because I-no-dot is less than I-dot. In a 1254TRKALT database, the order would be:

ia
ia
is
is

because I-no-dot is equal to I-dot.

Locales

Both the database server and the client library recognize their language and character set environment using a **locale definition**.

The application locale, or client locale, is used by the client or client library when making requests to the database server to determine the character set in which results should be returned, and the language of error messages, warnings, and other messages. The database server compares its own locale with the application locale to determine whether character set conversion is needed. Different databases on a server may have different locale definitions, and each client may have its own locale.

The locale consists of the following components:

- **Language** – The language is a two-character string using the ISO 639-1 standard values (for example, DE for German). Both the database server and the client have language values for their locale.

The database server uses the locale language to determine the language libraries to load. When creating a database, if no collation is specified, the database server also uses the language, together with the character set, to determine which collation to use.

The client library uses the locale language to determine the language libraries to load, and the language to request from the database.

- **Character set** – The character set is the code page, or encoding, in use. The client and server both have character set values, and they may differ. If they differ, character set conversion is used to enable interoperability.

Locale language

The locale language is the language being used by the user of the client application, or expected to be used by users of the database server.

The client library, and the database server, both determine the language component of the locale in the same manner:

1. Use the value of the SALANG environment variable, if it exists.
2. For Windows, if the SALANG environment variable doesn't exist, check the SAP Sybase IQ language registry entry.
3. Check the operating system language setting.
4. If the language still cannot be determined by the above settings, default to English.

Language label values

The following table displays the valid language label values, together with the equivalent ISO 639 language codes:

Language	ISO 639-1 language code	Language label	Alternative label
Arabic	AR	arabic	N/A
Czech	CS	czech	N/A
Danish	DA	danish	N/A
Dutch	NL	dutch	N/A
English	EN	us_english	english
Finnish	FI	finnish	N/A
French	FR	french	N/A
German	DE	german	N/A
Greek	EL	greek	N/A
Hebrew	HE	hebrew	N/A
Hungarian	HU	hungarian	N/A
Italian	IT	italian	N/A
Japanese	JA	japanese	N/A
Korean	KO	korean	N/A
Lithuanian	LT	lithuanian	N/A
Norwegian	NO	norwegian	norweg
Polish	PL	polish	N/A
Portuguese	PT	portuguese	portugue
Russian	RU	russian	N/A
Simplified Chinese	ZH	chinese	simpchin
Spanish	ES	spanish	N/A
Swedish	SV	swedish	N/A
Thai	TH	thai	N/A
Traditional Chinese	TW	tchinese	tradchin
Turkish	TR	turkish	N/A
Ukrainian	UK	ukrainian	N/A

Locale character set

Both application and server locale definitions have a character set. The application uses its character set when requesting character strings from the database server. The database server compares the database character set with that of the application to determine whether character set conversion is needed. If the database server cannot convert to and from the client character set, the connection fails.

1. If the SACHARSET environment variable is set, its value is used to determine the character set.

The database server uses SACHARSET only when creating new databases, and then only if no collation is specified.

2. If the connection string specifies a character set, it is used.
3. Sybase Open Client applications check the `locales.dat` file in the `locales` subdirectory of the Sybase release directory.
4. Character set information from the operating system is used to determine the locale:
 - On Windows operating systems, the current Windows ANSI code page is used.
 - On Unix platforms, the following Locale Environment Variables are examined, in the specified order: LC_ALL, LC_MESSAGES, LC_CTYPE, LANG. For the first of these environment variables found to be set, its value is used to determine the character set. If the character set cannot be determined from the operating system, the default of iso_1 (also referred to as Windows code page 28591, ISO 8859-1 Latin I, ISO 8859-1 Latin-1, or iso_8859-1:1987) is used.
5. On any other platform, a default code page cp1252 is used.

Understanding Character Set Translation

SAP Sybase IQ can perform character set translation among character sets that represent the same characters, but are located at different positions in the character set or code page.

A degree of compatibility between the character sets is required for this translation. For example, character set translation is possible between EUC-JIS and cp932 character sets, but not between EUC-JIS and cp1252.

These topics describe how SAP Sybase IQ carries out character set translation. This information is for users who may be deploying applications or databases in a multicharacter set environment.

Character Translation for Database Messages

Error and other messages from the database software are held in a language resource library. Localized versions of this library are provided with localized versions of SAP Sybase IQ.

Client application users may see both data and messages from the database. Some database messages, which are strings from the language library, may include placeholders that are filled

by characters from the database. For example, if you execute a query with a column that does not exist, the returned error messages is:

```
Column column-name not found
```

where the database provides the value for *column-name*.

To present these kinds of information to the client application in a consistent manner, even if the database is in a different character set from the language library, the database server automatically translates the characters of the messages to match the character set used in the database collation.

Messages are always translated into the database collation character set, regardless of whether character set conversion is turned on or off.

A further character set translation is carried out if character set translation is turned on (the default) for the database server, and if the client character set is different from that of the database collation.

Using Character Translation for Database Messages

Perform checks before reading database messages.

1. Ensure that the collation for your database is compatible with the character set used on your computer, and with the character set used in the SAP Sybase IQ language resource library. The language resource library differs among different localized versions of SAP Sybase IQ.
2. Verify that all characters of interest to you exist in each character set.

Connection Strings and Character Sets

Connection strings present a special case for character set translation.

The connection string is parsed by the client library, to locate or start a database server as follows:

- It is broken down into its *keyword= value* components. This can be done independently of character set, as long as you do not use the curly braces { } around CommLinks parameters. Instead, use the recommended parentheses (). Curly braces are valid following bytes (bytes other than the first byte) in some multibyte character sets.
- The server is located, its name is interpreted according to the character set of the client machine. For the Windows operating systems, the ANSI character set is used. Extended characters can be used unless they cause character set conversion issues between client and server machine.

For maximum compatibility among different machines, use server names that use alphabetic ASCII characters 1 to 127 (or 33 to 126) and the underscore, without any punctuation characters. Server names are truncated at 40 characters.

- The DatabaseName (DBN) or DatabaseFile (DBF) parameter is interpreted in the database server character set.

About International Language Data

- Once the database is located, the remaining connection parameters are interpreted according to the database character set.

Avoiding Character Set Translation

There is a performance cost associated with character set translation. If you can set up an environment for which no character set translation is required, you need not consider this cost, and your setup is simpler to maintain.

If you work with a single-byte character set and are concerned only with seven-bit ASCII characters (values 1 through 127), then you do not need character set translation. Even if the code pages differ between the database and the client operating system, they are compatible over this range of characters. Many English-language installations meet these requirements. Character set translation is turned on by default.

Even if you do require extended characters,

- If the code page on your client machine operating system matches that used in the database, no character set translation is needed for data in the database.
- If you can use a version of SAP Sybase IQ built for your language, and if you use the code page on your operating system, no character set translation is needed for database messages. The character set used in the SAP Sybase IQ message strings is as follows:

Language	Character set
English	cp1252
French	cp1252
German	cp1252
Japanese	cp932 (Shift-JIS)

Character set translation is disabled if you specify **CharSet=none** in the connection string.

ANSI and OEM Code Pages

SAP Sybase IQ collations are based on code pages that are designated as either ANSI or OEM.

Use an ANSI code page in most cases. If you use OEM, choose a code page that matches the OEM code pages on your users' client machines.

Note: Do not use a separate translation driver. Translation drivers interfere with the server's character set translation, and are likely to result in data corruption.

For Interactive SQL, the iAnywhere JDBC driver handles character set translation.

ANSI ISO_1 Collation

ISO_1 is provided for compatibility with the Adaptive Server Enterprise default ISO_1 collation.

The differences are:

- ß, the lowercase letter sharp s (\xDF), sorts after lowercase **s** in SAP Sybase IQ and SQL Anywhere, but after **ss** in Adaptive Server Enterprise.
- The ligatures corresponding to **Æ** and **æ** (\xC6 and \xE6) sort after **A** and **a** respectively in SAP Sybase IQ and SQL Anywhere (**A**, **a**, **æ**, **Æ**). In Adaptive Server Enterprise the sort order is **A**, **a**, **Æ**, **æ**.

ANSI 1252LATIN1 Collation

This collation includes the Euro currency symbol and several other characters such as Z-with-caron (Ž) and z-with-caron (ž).

For single-byte Windows operating systems, this is the recommended collation.

For Windows users using English, or Western European languages, this is the recommended collation.

ANSI ISO1LATIN1 Collation

This collation is the same as ISO_1, but includes sorting for values in the range A0 - BF.

For compatibility with Adaptive Server Enterprise, the ISO_1 collation includes no characters for 0xA0 - 0xBF. However, the ISO Latin 1 character set on which it is based does include

ANSI and OEM Code Pages

characters in these positions. The ISO1LATIN1 collation reflects the characters in these positions.

If you are not concerned with Adaptive Server Enterprise compatibility, ISO1LATIN1 is generally recommended instead of ISO_1.

ISO1LATIN1 is recommended for UNIX users using English or Western European languages, if you are willing to sacrifice some of the optimal performance of the default collation, ISO_BINENG.

ANSI ISO9LATIN1 Collation

This collation is the same as ISO1LATIN1, but includes the Euro currency symbol and the other new characters included in the 1252LATIN1 collation.

If your machine uses the ISO Latin 9 character set, and you are willing to sacrifice some of the optimal performance of ISO_BINENG, use ISO9LATIN1.

Displaying Collations

Obtain the currently used collation.

At the command prompt, enter:

```
SELECT * FROM SYS.SYSCOLLATION
```

Displaying Collations

Multibyte Collations

SAP Sybase IQ provides collations using multibyte character sets.

SAP Sybase IQ supports variable-width character sets. In these sets, some characters are represented by one byte, and some by more than one, to a maximum of four bytes. The value of the first byte in any character indicates the number of bytes used for that character, and also indicates whether the character is a space character, a digit, or an alphabetic (alpha) character.

For the UTF8 collation, UTF-8 characters are represented by one to four bytes. For other multibyte collations, one or two bytes are used. For all provided multibyte collations, characters comprising two or more bytes are considered to be “alphabetic”, and can be used in identifiers without requiring double quotes.

SAP Sybase IQ does not support 16-bit or 32-bit character sets such as UTF-16 or UTF-32.

All client libraries other than embedded SQL are Unicode-enabled, using the UTF-16 encoding. Translation occurs between the client and the server.

Japanese Language Support

Use collation 932JPN for Japanese Windows applications, and EUC_JAPAN for UNIX applications.

Collation 932JPN supports 32-bit multibyte characters that cannot be loaded into SJIS or SJIS2, which are older collations. SJIS is still supported (without 32-bit multibyte characters); however, SJIS2 is no longer supported.

Thai Language Support

SAP Sybase IQ provides the **CP874toUTF8** utility to convert data files in CP874 format into UTF8, which is a collation supported by SAP Sybase IQ for the Thai language.

You can also use the utility to load data in the CP874 character set without converting it to UTF8.

The SORTKEY() function returns values in the sort order **thaidict** (Thai dictionary), the Thai character set in UTF8 form. The following statements each generate the same result:

```
SELECT c1, SORTKEY(c1) from T1 where rid=3
SELECT c1, SORTKEY(c1, 'thaidict') from T1 where rid=3
SELECT
'\340\270\201\340\271\207', SORTKEY('\340\279\201\340\271\207') from
T1 where rid=3
```


Locale information

You can determine locale information using functions such as `PROPERTY`, `DB_PROPERTY`, and `CONNECTION_PROPERTY`. The following table shows how to use these functions to return locale information about the client connection, database, and database server.

System function and parameter	Return value
<code>SELECT PROPERTY('CharSet');</code>	Character set of the database server. Usually the character set of the computer hosting the database server.
<code>SELECT PROPERTY('DefaultCollation');</code>	Default CHAR collation used by the database server for creating databases.
<code>SELECT PROPERTY('DefaultNcharCollation');</code>	Default NCHAR collation used by the database server for creating databases.
<code>SELECT PROPERTY('Language');</code>	The locale language for the database server.
<code>SELECT DB_PROPERTY('CharSet');</code>	Character set used to store CHAR data in the database.
<code>SELECT DB_PROPERTY('NcharCharSet');</code>	Character set used to store NCHAR data in the database.
<code>SELECT DB_PROPERTY('MultiByteCharSet');</code>	Whether CHAR data uses a multibyte character set (On=yes, Off=no).
<code>SELECT DB_PROPERTY('Language');</code>	Comma-separated list of two-letter codes representing the languages supported by database CHAR collation.
<code>SELECT DB_PROPERTY('Collation');</code>	CHAR collation name in use by the database server.
<code>SELECT DB_PROPERTY('NcharCollation');</code>	NCHAR collation name in use by the database server.
<code>SELECT CONNECTION_PROPERTY('CharSet');</code>	Client's CHAR data character set.
<code>SELECT CONNECTION_PROPERTY('NcharCharSet');</code>	Character set of NCHAR data for the connection.
<code>SELECT CONNECTION_PROPERTY('Language');</code>	Client language for the connection.

Locale information

Selecting a Collation to Support a Specific Locale

Address character set issues properly by using the collation which best supports the locale.

1. Determine the default locale of each computing platform in your environment. The default locale is the character set and language of each computer. On Windows operating systems, the character set is the ANSI code page.
2. Determine whether locale settings are appropriate for your environment.
3. If the default settings are inappropriate, choose a character set, language, and database collation that match your data and avoid character set translation.
4. Set locales on each machine in the environment to these values.
5. Create your database using the default collation, or, if the collation does not match your needs, use a named collation.
6. When choosing the collation for your database,
 - a) Choose a collation that uses a character set and sort order the appropriate for the data. Several collations (some OEM and some ANSI) may meet this requirement.
 - b) Choose a collation that avoids the need for character set translation, which carries a performance cost, as well as extra complexity in system configuration. You can avoid character set translation by using a collation sequence in the database that matches the character set in use on your client machine operating system.

Selecting a Collation to Support a Specific Locale

Choosing Character Sets and Collation Sequences to Maximize Performance

Performance for character data is better with a binary character set and collation sequence than with a nonbinary character set and collation sequence.

To maximize performance, create a database with these default option settings, which result in a binary character set and collation sequence:

```
CREATE DATABASE dbname  
COLLATION 'ISO_BINENG'  
CASE RESPECT
```

Other settings for these options result in a nonbinary character set and collation sequence.

The disadvantage of these settings is that uppercase characters are always sorted before lowercase ones. For example, “BANANA” sorts before “apple.” If you prefer a more natural sort order, but still need a case-sensitive database, and if you are willing to sacrifice some degree of performance, use the collation ISO_1 instead of the default, ISO_BINENG.

Note: Passwords in newly-created databases are case-sensitive, regardless of the case-sensitivity of the database. When you rebuild an existing database, the password follows the case-sensitivity of the original database.

Setting a Locale

You can use the default locale on your operating system, or explicitly set a locale for use by the SAP Sybase IQ components on your machine.

1. Show the IQ engine character set:

```
select * from sa_eng_properties() where proptime like '%CharSet%'
```

2. Set either or both of the IQLANG and IQCHARSET environment variables:

```
IQCHARSET=charset;IQLANG=language_code
```

where *charset* is a valid character set label and *language_code* is a language code from the list of language label values.

SAP Sybase IQ loads the localization information when it executes the **INSERT...LOCATION** statement.

List of Language Label Values

Valid language label values and their equivalent ISO 639 language codes. Set the two-letter ISO_639 language code in the IQ_LANG environment variable.

Language	ISO_639 Language Code	Language Label	Alternative Label
Arabic	AR	arabic	N/A
Czech	CS	czech	N/A
Danish	DA	danish	N/A
Dutch	NL	dutch	N/A
English	EN	us_english	english
Finnish	FI	finnish	N/A
French	FR	french	N/A
German	DE	german	N/A
Greek	EL	greek	N/A
Hebrew	HE	hebrew	N/A
Hungarian	HU	hungarian	N/A

Language	ISO_639 Language Code	Language Label	Alternative Label
Italian	IT	italian	N/A
Japanese	JA	japanese	N/A
Korean	KO	korean	N/A
Lithuanian	LT	lithuanian	N/A
Norwegian	NO	norwegian	norweg
Polish	PL	polish	N/A
Portuguese	PT	portuguese	portugue
Russian	RU	russian	N/A
Simplified Chinese	ZH	chinese	simpchin
Spanish	ES	spanish	N/A
Swedish	SV	swedish	N/A
Thai	TH	thai	N/A
Traditional Chinese	TW	tchinese	tradchin
Turkish	TR	turkish	N/A
Ukrainian	UK	ukrainian	N/A

Setting the Locale for an INSERT...LOCATION Statement

When the database uses a non-default locale for your platform, you must set an environment variable on the local client for SAP Sybase IQ to load the correct information for language, collation sequence, character set, and date/time format.

When determining the locale name, SAP Sybase IQ first checks for the value of the LC_ALL environment variable. If LC_ALL is not set, SAP Sybase IQ uses the value of the LANG environment variable. If neither variable is set, SAP Sybase IQ uses the “default” entry in the locales file.

1. In a text editor, open the \$SYBASE/locales/locales.dat file.

For example:

```
locale = default, us_english, roman8
locale = C, us_english, roman8
```



```
locale = american, us_english, roman8  
locale = english.iso88591, us_english, iso_1
```

2. Set the LC_ALL or LANG environment variable to the correct value. If on the platform in step 1, your database's collation is iso_1 and you are using English, then set the value of the environment variable LC_ALL or LANG to "american.iso88591". Otherwise, SAP Sybase IQ uses the locale name "default" which has collation "roman8".

For example, in the sh or ksh shells, enter:

```
LC_ALL= american.iso88591;export LC_ALL
```

In the csh or tsch shell, enter:

```
setenv LC_ALL american.iso88591
```

Setting a Locale

Disabling Character Set Translation on a Database Server

You can turn character set conversion on and off explicitly on the database server command line.

By default, character set translation is enabled in SAP Sybase IQ, and takes place if the client and server locales differ.

To turn off character set translation, enter `CharSet=none` on the connection string.

Disabling Character Set Translation on a Database Server

Creating a Database with the Default Collation

If you use the default actions when creating an SAP Sybase IQ database, your database has the collation ISO_BINENG.

ISO_BINENG collation provides optimal performance for IQ databases, but not necessarily the most natural sort order.

This differs from SQL Anywhere, which infers the default collation for new databases from the character set in use by the operating system on which the database is created.

If you cannot set up your system in this default manner, decide which collation to use in your database, and whether to use character set translation to ensure that data is exchanged consistently between the pieces of your database system.

Creating a Database with the Default Collation

Changing the database collation

You can change your database collation from one collation to another by performing an unload and reload into the collation of choice. Collations are chosen at database creation time and cannot be changed without rebuilding the database.

Prerequisites

You must have the `SELECT ANY TABLE` and `SERVER OPERATOR` system privileges.

By default, you must have the `SELECT ANY TABLE` system privilege to execute an `UNLOAD` statement. The required privileges can be changed by using the `-gl` database server option.

Task

1. Start the database:

```
START databasefile
```

2. Determine the character set for the existing database:

```
SELECT DB_PROPERTY( 'CharSet' );
```

For early versions of SAP Sybase IQ, this property may not exist. The character set is also implied by the collation name. For example, collation `1252LATIN1` uses code page 1252.

The character set in the existing database should be the same as, or compatible with, the operating system and client character sets. If it is not, it is an excellent reason to rebuild the database, but requires great care in the rebuild process.

In particular, if you have been using a database with collation `850LATIN1` with earlier versions of SAP Sybase IQ that either did not support character set conversion (versions 5 and earlier) or disabled it by default (versions 6 and 7), and if your client applications were normal Windows applications, you may have code page 1252 character data in your database that is interpreting data to be in code page 850.

3. Determine the character set for the data in the existing database:

```
UNLOAD TABLE mytable TO 'mytable-data-in-utf8.dat' ENCODING 'UTF-8';
```

View the results in a text editor.

If accented data is correct, then the character data in the database matches the Windows ANSI code page, which for English and other Western European languages is code page 1252. If the data appears correct in a DOS-based editor, then the character data matches the Windows OEM code page, which is likely 437 or 850.

4. Unload the database.

Changing the database collation

If the data character set is incompatible with the database character set, it is critical that the data is unloaded without character set conversion. Depending on the version of SAP Sybase IQ being used, you can use the internal unload feature of dbunload, or manually unload the data using the UNLOAD TABLE statement.

5. Create the new database, specifying the collations and character sets you want to use, and set the DBA user ID and password to DBA and sql:

```
iqinit -dba DBA,sql -z 1252LATIN1 c:\newdatabase.db
```

6. Stop the old database server and start the new database server:

```
iqsrv16 -n new-server c:\newdatabase.db
```

7. Load the data into the new database.

If the unloaded data and schema (`reload.sql`) match the character set of the computer used to perform the reload, you can use the external reload option of dbunload. The data is automatically converted to the correct character set for the database. For example:

```
dbunload -ix c:\databasefile
```

If the data's encoding does not match the character set of the database, and you are loading the data using LOAD TABLE statements (internal reload), you must use the ENCODING clause; the database server does not, by default, perform character set conversion for data loaded using LOAD TABLE statements.

If the data's encoding does not match the code page of the computer on which you are working, and you are loading using INPUT statements (external reload), you must use the ENCODING clause; otherwise, the database server assumes that the data is in the computer's native character set.

The database is recreated with the new collation.

Index

A

- ANSI code pages
 - choosing 23
- ASCII
 - character set 4

C

- case-sensitivity
 - passwords 33
- character set
 - showing 35
- character set translation
 - disabling 39
 - error messages 20
 - starting a server 39
 - turning on and off 39
- character sets
 - about 1
 - avoiding translation 22
 - choosing 31
 - encoding 1
 - ISO_BINENG 41
 - multibyte 27
 - single-byte 4
 - translation 39
 - Unicode 1, 27
 - UTF8 41
- code pages
 - about 4
 - supported 4
- collations
 - about 1, 7
 - alternate 25
 - ANSI 23
 - choosing 31
 - CP874toUTF8 utility 1
 - CREATE DATABASE statement 1
 - creating a database 33
 - custom 1
 - default 33, 41
 - deprecated 1
 - displaying 25
 - ISO_1 23
 - migrating deprecated 1

- migration 1
 - multibyte 27
 - performance issues 33
 - recommended 25
 - recommended for Japanese 27
 - SORT_COLLATION option 1
 - sorting 7
 - SORTKEY function 1
 - supported 25
- connecting
 - character sets 21
- connection strings
 - character sets 21
- CREATE DATABASE statement
 - collation 1
- custom collations 1

D

- databases
 - character set 20

E

- environment variables
 - LANG 37
 - LC_ALL 37
- error messages
 - character set translation 20
- euro symbol
 - 1252LATIN1 collation 23, 24

I

- ICU library 1
- IQCHARSET environment variable 35
- IQLANG 35
- IQLANG environment variable 35
- ISO_1 collation
 - about 23
- ISO_BINENG
 - default collation 33, 41
- ISO_BINENG collation 41

Index

L

- LANG environment variable 37
- language support
 - about 1
 - collations 31
 - multibyte character sets 27
 - overview 1
- language_code 35
- LC_ALL environment variable 37
- locales
 - about 1
 - character sets 20
 - INSERT...LOCATION 36
 - setting 35

M

- migration
 - collations 1
 - deprecated collations 1
 - sort keys 1
- multibyte character sets
 - using 27

N

- national language support
 - about 1
 - collations 31
 - multibyte character sets 27
 - overview 1

O

- OEM code pages
 - choosing 23

P

- passwords
 - case-sensitivity 33

S

- single-byte character sets
 - about 4
- sort keys
 - migration 1
- sort order
 - collations 1
- SORT_COLLATION option 1
- sorting
 - collations 7
- SORTKEY function 1

U

- Unicode
 - character sets 1, 27
- Unilib library 1
- utilities
 - CP874toUTF8 1