# logo-sql08.gif

Working with the New SQL Server 2008 Collations and Earlier Versions of the Native Data Provider

SQL Server Technical Article

**Writer:** Fernando Caro

**Technical Reviewers:** Jimmy Wu, Sanjay Mishra, Qingsong Yao, Juergen Thomas, Burzin Patel, Wanda He, Tres London

**Published:** October 2009

**Applies to:** SQL Server 2008

**Summary:** Collations are essential in setting up a database and for database operations. They are used at the server and database level, but they can also be used at the column or expression level. With SQL Server 2008, new Windows collations (denoted by \*\_100 in the collation name) were added that are in full alignment with the Windows Vista and Windows Server 2008 operating systems. Some of these new collations are not supported with earlier versions of the native data provider during connection to a SQL Server 2008 database, if both the client and the server environment are not in Unicode. It is therefore important to understand these collation scenarios and corresponding solutions.

Copyright

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, MSDN, SQL Server, Windows, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

[Introduction 4](#_Toc242160521)

[General Client-to-SQL Server Collation Interaction 4](#_Toc242160522)

[Client-to-SQL Server Data Flow and Data Conversions 5](#_Toc242160523)

[Collations in SQL Server 2008 7](#_Toc242160524)

[What’s New in SQL Server 2008 7](#_Toc242160525)

[Earlier Versions of the Native Data Provider 8](#_Toc242160526)

[Collations Not Recognized by Earlier Version of the Native Data Provider 8](#_Toc242160527)

[Native Data Provider’s Specific Behavior Scenarios 8](#_Toc242160528)

[Generic Error Message 9](#_Toc242160529)

[Server-Side Error Conditions 10](#_Toc242160530)

[Earlier Versions of the Native Data Providers and Error Message Scenarios 10](#_Toc242160531)

[Conclusion 13](#_Toc242160532)

[Appendix 14](#_Toc242160533)

# Introduction

As databases continue to expand and support a growing global market, users need to be able to work with character data in meaningful ways and use the most up-to-date linguistic collations to meet language market expectations.

As such, collations are a powerful way for users to sort and compare strings according to their own cultural conventions. They play a critical part in database creation and in data operations.

The Microsoft® SQL Server® 2008 data management software introduced new Windows® collations that are in full alignment with the Windows Vista® and Windows Server® 2008 operating systems. These new collations are denoted by \*\_100 in the collation name and represent unique sort semantics that correspond with the Windows locales, though it is not required to be running on Windows Vista or Windows Server 2008 to use these new collations.

The new SQL Server Windows collations include support for the following:

* New East Asian government standards such as JIS2004 for Japan
* Chinese minority scripts (Yi, Uighur, Mongolian, Tibetan)
* Unicode 5.0 case table
* Weights to previously nonweighted characters that would have had equal value in comparisons

They are also linguistically correct for supplementary characters.

It should be noted that while SQL Server 2008 has introduced new Windows collations to reflect these changes, no modifications have been made to the collations supported in SQL Server 2000 and SQL Server 2005 (though this may change in future versions in case of deprecation). In addition, the SQL\_\* named collations are maintained to keep compatibility with previous versions of SQL Server and for applications that use these collations. For more information about collations, see “Working with Collations” (<http://msdn.microsoft.com/en-us/library/ms187582.aspx>). For more information about Unicode, see “Unicode Basics” (<http://msdn.microsoft.com/en-us/library/ms187828.aspx>). Both topics are part of SQL Server Books Online.

This paper describes the set of scenarios and solutions when trying to connect to a SQL Server 2008 database (and both the client and the server environment are not in Unicode) and when using one of the earlier versions of the native data provider that are interacting with an unsupported set of the SQL Server 2008 \*\_100 named collations. The paper also provides solutions for working in these scenarios.

# General Client-to-SQL Server Collation Interaction

In SQL Server, collations constitute two key semantics:

* Sort order as defined by Windows. Sort order applies to character (**char**/**varchar**) and Unicode data types (**nchar**/**nvarchar**).
* The database collation, which determines which code page is used to store the character data used by the **char** and **varchar** character data types. Both these data types are non-Unicode.

In a client-to-SQL Server environment, collations are used by the native data providers to handle the data conversions if the client-to-SQL Server environment is not entirely in Unicode. This is to avoid data being improperly converted (characters would be represented as “?”).

## Client-to-SQL Server Data Flow and Data Conversions

The following figures illustrate scenarios in which the native data providers must convert the data into the corresponding code page used by SQL Server **char** and **varchar** data types or into Unicode. In figures 1 and 2, data conversions occur when the Windows active code page on the client and the database collation that is used to store the character data used by the **char** and **varchar** data types are in different code pages.

**Figure 1**: Client-to-SQL Server data flow, client and SQL Server using different code pages



**Figure 2**: SQL Server-to-client data flow, client and SQL Server using different code pages

In figures 3 and 4, the client data is not in Unicode, but the SQL Server data is.



**Figure 3:** Client-to-SQL Server data flow, client data using a code page, SQL Server using Unicode data types



**Figure 4:** SQL Server-to-client data flow, client data using a code page, SQL Server using Unicode data types

In figures 5 and 6, the client data is in Unicode, and SQL Server is not.



**Figure 5:** Client-to-SQL Server data flow, client using Unicode, SQL Server using a code page



**Figure 6:** SQL Server-to-client to data flow, client using Unicode, SQL Server using a code page

# Collations in SQL Server 2008

## What’s New in SQL Server 2008

Figure 7 illustrates the collation additions in SQL Server 2000, SQL Server 2005, and SQL Server 2008.

Albanian\_

Arabic\_

Chinese\_PRC\_

Chinese\_PRC\_Stroke\_

Chinese\_Taiwan\_Bopomofo\_

Chinese\_Taiwan\_Stroke\_

Croatian\_

Cyrillic\_General\_

Czech\_

Danish\_Norwegian\_

Estonian\_

Finnish\_Swedish\_

French\_

Georgian\_Modern\_Sort\_

German\_PhoneBook\_

Greek\_

Hebrew\_

Hungarian\_

Hungarian\_Technical\_

Icelandic\_

Japanese\_

Japanese\_Unicode\_

Korean\_Wansung\_

Latin1\_General\_

Latvian\_

Lithuanian\_

Modern\_Spanish\_

Polish\_

Romanian\_

Slovenian\_

Thai\_

Traditional\_Spanish\_

Turkish\_

Ukrainian\_

Vietnamese\_

Assamese\_100\_

Azeri\_Cyrillic\_100\_

Azeri\_Latin\_100\_

***Bashkir\_100\_***

Bengali\_100\_

Bosnian\_Cyrillic\_100\_

Bosnian\_Latin\_100\_

Breton\_100\_

Chinese\_Simplified\_Stroke\_Order\_100\_

Chinese\_Traditional\_Pinyin\_100\_

Chinese\_Traditional\_Stroke\_Order\_100\_

***Corsican\_100\_***

Danish\_Greenlandic\_100\_

***Dari\_100\_***

Divehi\_90\_

***Frisian\_100\_***

***Japanese\_Bushu\_Kakusu\_100\_***

Khmer\_100\_

Lao\_100\_

Latin1\_General\_100

Japanese\_XJIS\_100\_

***Maltese\_100\_***

***Maori\_100\_***

***Mapudungan\_100\_***

***Mohawk\_100\_***

Nepali\_100\_

***Norwegian\_100\_***

Pashto\_100\_

***Persian\_100\_***

***Romansh\_100\_***

***Sami\_Norway\_100\_***

***Sami\_Sweden\_Finland\_100\_***

***Serbian\_Cyrillic\_100\_***

***Serbian\_Latin\_100\_***

Slovak\_

Tamazight\_100\_

Tibetan\_100\_

***Turkmen\_100\_***

***Uighur\_100\_***

***Upper\_Sorbian\_100\_***

***Urdu\_100\_***

Welsh\_100\_

***Yakut\_100\_***

New in SQL Server 2008

SQL Server 2000

New in SQL Server 2005

Chinese\_Hong\_Kong\_Stroke\_90\_

Chinese\_PRC\_90\_

Chinese\_PRC\_Stroke\_90\_

Chinese\_Taiwan\_Bopomofo\_90\_

Chinese\_Taiwan\_Stroke\_90\_

Indic\_General\_90\_

Japanese\_90\_

Korean\_90\_

Kazakh\_90\_

Macedonian\_FYROM\_90\_

Syriac\_90\_

Tatar\_90\_

Uzbek\_Latin\_90\_

**Figure 7:** Collations in SQL Server

# Earlier Versions of the Native Data Provider

## Collations Not Recognized by Earlier Versions of the Native Data Provider

Of the new Windows collations introduced in SQL Server 2008, the collations listed in figure 7 in ***black bold italic type*** are not recognized by earlier versions of the native data provider during connection to a SQL Server 2008 database if both the client and the server environment are not in Unicode. There are several possible result cases; for example, consider the following scenario.

**Note**: This can happen with Windows Server 2003 and earlier, as stated previously. It can also happen with Windows Server 2008 and Windows Server 2008 R2.

For instance, consider a situation in which SQL Server 2005 and SQL Server 2008 are installed side-by-side on Windows Server 2008. A SQL Server 2005 data provider, such as Microsoft SQL Server Native Client 9.0 (SNAC9) or Windows Data Access Components (Windows DAC), is used to connect to an instance of SQL Server 2008 that uses one of the new unsupported SQL Server 2008 collations.

As a result of this scenario, one of the following things can happen:

* A down-level client application can lose its connection with SQL Server. Specifically, if an error message cannot be provided during the result set, the connection will be lost. For more information, see “Generic Error Message” later in this paper.
* The client connection is not lost; however, an error message will indicate that there is a collation mismatch between the client and server data, and no data conversion of the server data will occur.

In both cases, the connection is lost or the error message is provided if one of the new Windows collations in SQL Server 2008 is being used, if the collation is not recognized by the earlier version of the native data provider during connection to a SQL Server 2008 database and both the client and server environment are not in Unicode.

This is to prevent data conversions that can produce data loss from occurring.

## Native Data Provider’s Specific Behavior Scenarios

The following table illustrates the current behavior of the earlier version of the native data provider and lists the solution.

|  |  |  |
| --- | --- | --- |
| **Data provider version** | **Existing behavior** | **Solution** |
| SQL Native Client 10.0(SQL Server 2008) | Instead of SQL Native Client 9.0 existing behavior:* Uses a Windows code page if it does not recognize the collation.
* Fails on the client if it cannot determine the correct code page for the data conversion.
 | None (no solution required). |
| SQL Native Client 9.0,(Earlier than SQL Server 2008) | * The data provider attempts to derive the locale and map it to the corresponding client code page.
* If the locale identifier is not recognized by the data provider, the active code page from the client machine system locale is used for the data conversion.
* If the characters are incorrectly mapped from the code page to another code page, the data conversion may result in data loss.
 | Apply the hotfix located [here](http://support.microsoft.com/kb/944929).  |
| Windows DAC | * The data provider attempts to derive the locale identifier from Windows.
* If the locale identifier is not recognized by the data provider, the active code page from the client machine system locale is used for the data conversion.
* If the characters are incorrectly mapped from the code page to another code page, the data conversion may result in data loss.
 | Apply the update located [here](http://support.microsoft.com/default.aspx/kb/954920/en-us).  |

Third-party providers such as FreeTDS, DataDirect, EasySoft, Open Link, and so on can either continue to reverse engineer based on available published Knowledge Base articles or by using the publically documented network protocol, [[MS-TDS]: Tabular Data Stream Protocol Specification](http://msdn.microsoft.com/en-us/library/dd304523%28PROT.13%29.aspx), in the [Windows Server Protocols (WSPP)](http://msdn.microsoft.com/en-us/library/cc964399%28PROT.13%29.aspx) section of the MSDN® Library.

## Generic Error Message

The following error message is provided from the server to the client:

“The statement failed because the database collation CollationName is not recognized by older client drivers. Try upgrading the client operating system or applying a service update to the database client software, or use a different collation. See SQL Server Books Online for more information on changing collations.”

The statement referenced is a USE or ALTER DATABASE statement. The message appears during retrieval of result sets that use one of the new collations not recognized downlevel and that contain one of the following: a **sql\_variant** data type, a column, an output parameter, or an output parameter using **sql\_variant**.

This message appears because one of the new SQL Server 2008 Windows collations has been passed to a downlevel client application that relies on an earlier version of the data provider such as SNAC9 or Windows DAC. This server message also helps third-party applications that rely on these data providers and downlevel clients that have not applied the appropriate solution.

However, some clients may be able to provide a client-side message or properly handle the data when working with one of the new collations. When this happens, the client sets a new login packet bit. This bit tells SQL Server not to send the error message. This means that earlier versions of the native data provider can provide a client-side error message or properly handle the data instead of relying on SQL Server to provide the error message.

## Server-Side Error Conditions

The following table explains what happens when an earlier version of the data provider is used and the client application cannot provide a client-side error.

|  |  |
| --- | --- |
| **Case** | **Result** |
| Result set in server-to-client downlevel for string type (**char**, **varchar**) | Fails with an error message sent to the client or disconnects |
| SQL\_VARIANT | Disconnects client and writes error message to the SQL Server error log |
| RPC non-SQL\_VARIANT output parameters | Fails with an error message sent to the client |
| RPC SQL\_VARIANT parameters | Disconnects from the client and writes an error message to the SQL Server error log |
| ENVCHANGE | Fails at login with an error message sent to the client\* |
| RPC input parameters (input/output) | Client data accepted\*\* |
| TVP/BCP input |
| Input from old client (before version 7.1) that does not send a collation with input parameters |
| Old client sending a multibyte character sets (MBCS) event in a new collation |
| Connection reset where the client cannot set the bit  | Fails the connection or the statement with an error message sent to the client |

\*Occurs only for downlevel clients running on downlevel operating systems.

\*\*For cases where the client data is accepted, this is specifically when the server fails the new collations being sent to the downlevel client.

## Earlier Versions of the Native Data Provider and Error Message Scenarios

This section lists scenarios in which error messages would be sent.

Scenario 1: SQL Server sends to the client a result set containing:

* A **sql\_variant** or **char**/**varchar** type with new collations (COLMETADATA and ROW tokens).
* A computed column of **sql\_variant** or **char**/**varchar** type with new collations (ALTMETADATA and ALTROW tokens).

Scenario 2: The client invokes a remote procedure call (RPC) taking an input parameter of **sql\_variant** or **char**/**varchar** type with new collations (RPCRequest token).

Scenario 3: The client and SQL Server exchange RPCs with an output parameter of **sql\_variant** or **char**/**varchar** type with new collations (RPCRequest and RETURNVALUE tokens).

Scenario 4: An ENVCHANGE occurs during:

* A login statement that contains a new collation. The client requests a connection to a database that uses a new collation during login.
* A USE DB statement that contains a new collation. The client issues a USE DB statement to a database that uses a new collation.

Scenario 5: Cached queries are encountered. A query is executed first by a client that is aware of new collations and then by a client that is unaware of new collations.

Scenario 6: Cached RPCs are encountered. An RPC is executed first by a client that is aware of new collations and then by a client that is unaware of new collations.

Scenario 7: An attempt is made to fetch columns of **sql\_variant** or **char**/**varchar** types with new collations.

Here are more explicit examples of such cases, when one of the new collations is used.

Example 1: An application that uses an earlier version of the data provider selects from a table that contains one or more columns of **sql\_variant** or **char**/**varchar** data types with new collations.

Example 2: An application that uses an earlier version of the data provider selects from a table that contains one or more columns of **sql\_variant** or **char**/**varchar** data types with new collations.

Example 3: An application that uses an earlier version of the data provider calls an RPC with one or more input parameters of **sql\_variant** or **char**/**varchar** data types with new collations.

Example 4: An application that uses an earlier version of the data provider calls an RPC with one or more output parameters of **sql\_variant** or **char**/**varchar** data types with new collations.

Example 5: An application that uses an earlier version of the data provider calls an RPC with one or more input parameters of **sql\_variant** or **char**/**varchar** data types with new collations.

Example 6: An application that uses an earlier version of the data provider calls an RPC with one or more output parameters of **sql\_variant** or **char**/**varchar** data types with new collations.

Example 7: An application that uses an earlier version of the data provider selects from a table that contains one or more columns of **sql\_variant** or **char**/**varchar** data types with new collations. (The data might potentially be cached because it was recently requested by another client.)

Example 8: An application that uses an earlier version of the data provider calls an RPC with one or more output parameters of **sql\_variant** or **char**/**varchar** data types with new collations. (The data might potentially be cached because it was recently requested by another client.)

Example 9: An application that uses an earlier version of the data provider opens a cursor on a table with one or more columns of **sql\_variant** or **char**/**varchar** data types with new collations and then fetches rows from this cursor.

Example 10: An application that uses an earlier version of the data provider opens a cursor on a table with one or more columns of **sql\_variant** or **char**/**varchar** data types with new collations and then fetches rows from this cursor.

Query example:

CREATE TABLE tvarchar (cvc varchar(100) COLLATE Norwegian\_100\_CI\_AS)

GO

INSERT INTO tvarchar VALUES ("MyValue1")

GO

INSERT INTO tvarchar VALUES ("MyValue2")

GO

CREATE TABLE tvariant (cvar sql\_variant)

GO

SELECT \* FROM tvarchar

-- For high-level (for example, a low-level client is one that uses SNAC 9, and a high-level client is one that uses SNAC10; if you use SNAC9 to connect to SQL Server 2008 [downlevel] and it is not patched, it will fail if you are accessing data with an unsupported new SQL Server 2008 Windows-based collation) or patched client: output

cvc

---

MyValue1

MyValue2

-- For low-level unpatched client: output

The statement failed because column 'cvc' (ID=0) uses collation Norwegian\_100\_CI\_AS, which is not recognized by older client drivers Try upgrading the client operating system or applying a service update to the database client software, or use a different collation See SQL Server Books Online for more information on changing collations.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \* FROM tvariant

-- For high-level or patched client: output

cvc

---

MyValue1

MyValue2

-- For low-level unpatched client: output

Client is disconnected. An error is logged in the SQL Server error log that the client is not capable of handling data with collation Norwegian\_100\_CI\_AS.

# Conclusion

With SQL Server 2008, new Windows collations were added to be in full alignment with Windows Vista and Windows Server 2008 operating systems and to provide users with the latest linguistic collations.

Because collations are central to the database and data operations, it is important to understand where use of the new collations with an earlier version of the native data provider may result in problems during attempts to connect to SQL Server 2008 in a client/server environment that is not end-to-end Unicode.

Taking into account these scenarios and applying the appropriate patch or the correct data provider version enables the client application to connect to the database (though if the result set contains one of the unsupported collations in a downlevel connection to a SQL Server 2008 database, an error message will be raised).

**For more information:**

<http://www.microsoft.com/sqlserver/>: SQL Server Web site

<http://technet.microsoft.com/en-us/sqlserver/>: SQL Server TechCenter

<http://msdn.microsoft.com/en-us/sqlserver/>: SQL Server DevCenter

[http://msdn.microsoft.com/en-us/library/dd317752(VS.85).aspx](http://msdn.microsoft.com/en-us/library/dd317752%28VS.85%29.aspx): Code pages

[http://msdn.microsoft.com/en-us/library/dd303449(PROT.13).aspx](http://msdn.microsoft.com/en-us/library/dd303449%28PROT.13%29.aspx): ENVCHANGE

<http://msdn.microsoft.com/en-us/data/aa937733.aspx>: Microsoft SQL Server Native Client (SNAC)

<http://www.unicode.org/glossary/#S> : Supplementary Character

[http://msdn.microsoft.com/en-us/library/aa968814(VS.85).aspx](http://msdn.microsoft.com/en-us/library/aa968814%28VS.85%29.aspx): Windows Data Access Components (Windows DAC)

<http://msdn.microsoft.com/en-us/goglobal/bb688114.aspx>: Windows Code Pages

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

* Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
* Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release. Send feedback.