

[MS-ODBCSTR]: ODBC Connection String Structure

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

| Date | Revision History | Revision Class | Comments |
|------------|------------------|----------------|--|
| 06/27/2008 | 1.0 | Major | First release. |
| 10/06/2008 | 1.01 | Editorial | Revised and edited the technical content. |
| 12/12/2008 | 1.02 | Editorial | Revised and edited the technical content. |
| 08/07/2009 | 1.1 | Minor | Updated the technical content. |
| 11/06/2009 | 1.1.2 | Editorial | Revised and edited the technical content. |
| 03/05/2010 | 1.1.3 | Editorial | Revised and edited the technical content. |
| 04/21/2010 | 1.1.4 | Editorial | Revised and edited the technical content. |
| 06/04/2010 | 1.1.5 | Editorial | Revised and edited the technical content. |
| 09/03/2010 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 02/09/2011 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 07/07/2011 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 11/03/2011 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 01/19/2012 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 02/23/2012 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 03/27/2012 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 05/24/2012 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 06/29/2012 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 07/16/2012 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 10/08/2012 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 10/23/2012 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |

| Date | Revision History | Revision Class | Comments |
|-------------|-------------------------|-----------------------|--|
| 03/26/2013 | 1.1.5 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 06/11/2013 | 2.0 | Major | Significantly changed the technical content. |
| 08/08/2013 | 3.0 | Major | Significantly changed the technical content. |

Contents

| | |
|---|-----------|
| 1 Introduction | 6 |
| 1.1 Glossary | 6 |
| 1.2 References | 7 |
| 1.2.1 Normative References | 7 |
| 1.2.2 Informative References | 8 |
| 1.3 Overview | 8 |
| 1.4 Relationship to Other Protocols | 9 |
| 1.5 Applicability Statement | 9 |
| 1.6 Versioning and Capability Negotiation | 9 |
| 1.7 Vendor-Extensible Fields | 9 |
| 2 Structures | 10 |
| 2.1 ABNF Rules | 10 |
| 2.1.1 Common ABNF Rules | 10 |
| 2.1.2 ODBC Connection String Format | 10 |
| 2.1.2.1 KeyValuePair | 10 |
| 2.1.2.2 Key | 10 |
| 2.1.2.3 Value | 10 |
| 2.1.2.4 ValueFormat1 | 11 |
| 2.1.2.5 ValueContent1 | 11 |
| 2.1.2.6 ValueContent2 | 11 |
| 2.2 Generic Keys | 11 |
| 2.2.1 Default Values for Generic Keys | 11 |
| 2.2.2 Case-sensitivity | 11 |
| 2.2.3 Multiple Occurrences of the Same Generic Key | 12 |
| 2.3 Driver Conflict Resolution | 12 |
| 2.3.1 Determining Which Driver Is Used | 12 |
| 2.3.2 Conflicts between the Content of a File DSN and Connection String | 12 |
| 2.4 SqlConnectionString | 12 |
| 2.4.1 SqlConnectionString Additional Restrictions | 12 |
| 2.4.2 Driver-Specific Keys | 13 |
| 3 Structure Examples | 17 |
| 3.1 Trusted Connection | 17 |
| 3.2 Standard Security Connection | 17 |
| 3.3 Named Instance | 17 |
| 3.4 Network | 17 |
| 3.5 Escaped Right Brace | 17 |
| 3.6 Leading and Trailing Spaces | 18 |
| 3.7 Values Enclosed by Braces | 18 |
| 3.8 Driver Conflict Resolution | 18 |
| 3.9 Multiple Instances of a Generic Key | 18 |
| 3.10 Multiple Instances of Driver-Specific Key | 18 |
| 4 Security Considerations | 19 |
| 4.1 Security Considerations for Implementers | 19 |
| 4.2 Index of Security Parameters | 19 |
| 5 Appendix A: Product Behavior | 20 |
| 6 Change Tracking | 23 |

7 Index 25

1 Introduction

The ODBC Connection String Structure is the format that describes the connection strings that are used by **Open Database Connectivity (ODBC) applications**.

A connection string is a string that specifies information about a data source and the means of connecting to it. The **ODBC application** determines how to read the connection string to initiate a connection to a data source.

Sections 1.7 and 2 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

ANSI character set
code page
connection
database
encryption
named pipe
OEM character
registry
Unicode

The following terms are defined in [\[MS-OFCGLOS\]](#):

application
connection string
data source

The following terms are specific to this document:

database instance: A database that has a unique set of services that can have unique settings.

Data Source Name (DSN): A logical name residing in the client system that applications use to request a connection to an ODBC data source. The DSN stores the driver and other connection details.

default database: The current database just after the connection is made.

driver: A library that implements the ODBC APIs against a specific data source to provide data source specific operations. Each driver is specific to a particular data source.

driver-specific key: A keyword in a connection string that is interpreted by an individual driver. Drivers can have different interpretations on the meaning of a value for a keyword.

File DSN: A text file that contains DSN information.

generic key: A keyword in a connection string, the meaning of which is the same across all drivers.

ODBC application: An application which uses Open Database Connectivity (ODBC) to access data sources.

Open Database Connectivity (ODBC): A standard software API method for accessing data that is stored in a variety of proprietary personal computer, minicomputer, and mainframe databases. It is an implementation of [\[TSQL-CLI-2003\]](#) and provides extensions to that standard. <1>

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MS-TDS] Microsoft Corporation, "[Tabular Data Stream Protocol](#)".

[RFC1002] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications", STD 19, RFC 1002, March 1987, <http://www.ietf.org/rfc/rfc1002.txt>

[RFC2460] Deering, S., and Hinden, R., "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998, <http://www.ietf.org/rfc/rfc2460.txt>

[RFC4120] Neuman, C., Yu, T., Hartman, S., and Raeburn, K., "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005, <http://www.ietf.org/rfc/rfc4120.txt>

[RFC4234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005, <http://www.ietf.org/rfc/rfc4234.txt>

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981, <http://www.ietf.org/rfc/rfc791.txt>

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981, <http://www.ietf.org/rfc/rfc0793.txt>

[TSQL] Digital Equipment Corporation, "Information Technology - Database Language SQL", ISO/IEC 9075:1992, July 1992, <http://www.andrew.cmu.edu/user/shadow/sql/sql1992.txt>

[TSQL-CLI-2003] International Organization for Standardization and International Electrotechnical Commission, "Information Technology – Database Languages – SQL – Part 3: Call-Level Interface", ISO/IEC 9075-3:2003, 2003, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=34134

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

[MS-OFCGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)".

[MSDN-CUFDS] Microsoft Corporation, "Connecting Using File Data Sources", [http://msdn.microsoft.com/en-us/library/ms710900\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms710900(VS.85).aspx)

[MSDN-DAD] Microsoft Corporation, "Database Detach and Attach (SQL Server)", <http://msdn.microsoft.com/en-us/library/ms190794.aspx>

[MSDN-DLO] Microsoft Corporation, "default language Option", Administering SQL Server (SQL Server 2000), [http://msdn.microsoft.com/en-us/library/aa196707\(SQL.80\).aspx](http://msdn.microsoft.com/en-us/library/aa196707(SQL.80).aspx)

[MSDN-FILE] Microsoft Corporation, "Naming Files, Paths, and Namespaces", <http://msdn.microsoft.com/en-us/library/aa365247.aspx>

[MSDN-SD] Microsoft Corporation, "Selecting a Database", <http://msdn.microsoft.com/en-us/library/ms180770.aspx>

[MSDN-UDTD-ODTF] Microsoft Corporation, "Using Date and Time Data -- ODBC Date-Time Format", [http://msdn.microsoft.com/en-us/library/ms180878\(SQL.100\).aspx#ODBCDatetimeFormat](http://msdn.microsoft.com/en-us/library/ms180878(SQL.100).aspx#ODBCDatetimeFormat)

[MSDN-UNI] Microsoft Corporation, "Using Named Instances", <http://msdn.microsoft.com/en-us/library/ms165614.aspx>

[MSDN-UOMSS] Kumar, A., and Brewer, A., "Using ODBC with Microsoft SQL Server", September 1997, <http://msdn.microsoft.com/en-us/library/ms811006.aspx>

[MSFT-KB313295] See [MSKB-313295].

[MSFT-KB328383] See [MSKB-328383].

[MSKB-313295] Microsoft Corporation, "How to use the server name parameter in a connection string to specify the client network library", <http://support.microsoft.com/kb/313295>

[MSKB-328383] Microsoft Corporation, "SQL Server clients may change protocols when the client computers try to connect to an instance of SQL Server", <http://support.microsoft.com/kb/328383>

[NTLM] Microsoft Corporation, "Microsoft NTLM", <http://msdn.microsoft.com/en-us/library/aa378749.aspx>

If you have any trouble finding [NTLM], please check [here](#).

[PIPE] Microsoft Corporation, "Named Pipes", <http://msdn.microsoft.com/en-us/library/aa365590.aspx>

1.3 Overview

The ODBC Connection String Structure is a method for an Open Database Connectivity (ODBC) application to specify the parameters used to connect to a **data source**. A **connection string** specifies a set of properties as keys with their associated values. The connection string can include one or more key/value pairs to specify information that includes the **driver** name, the user identification, the password, and/or driver-specific information.

1.4 Relationship to Other Protocols

None.

1.5 Applicability Statement

This document specifies a persistence format for Open Database Connectivity (ODBC) connection strings. The connection strings are used to help establish a **connection** between an ODBC application and a data source in scenarios where network or local connectivity is available. This document further specifies the format of a connection string that is used to establish a connection between an ODBC application and Microsoft SQL Server.

This persistence format provides interoperability with ODBC applications that create or use portions of documents conforming to this structure.

1.6 Versioning and Capability Negotiation

None.

1.7 Vendor-Extensible Fields

Vendors can define **driver-specific keys** and specify their meanings and the corresponding valid values. The name of a driver-specific key MUST conform to the naming rules for a key as specified in section [2.1.2](#) and MUST NOT be the same as the name of any **generic key** specified in section [2.2](#).

2 Structures

An ODBC connection string MUST conform to the Augmented Backus-Naur Form (ABNF) [\[RFC4234\]](#) grammar specified in section [2.1.2](#).

2.1 ABNF Rules

2.1.1 Common ABNF Rules

The following ABNF syntax rules, as specified in [\[RFC4234\]](#), are used in section [2.1.2](#) and are included for reference.

```
SC           = %x3B           ; Semicolon
LCB          = %x7B           ; Left curly brackets
RCB          = %x7D           ; Right curly brackets
EQ           = %x3D           ; Equal sign
ESCAPEDRCB  = 2RCB          ; Double right curly brackets
SpaceStr     = *(SP)         ; Any number (including 0) spaces
```

2.1.2 ODBC Connection String Format

The **ODBCConnectionString** structure specifies a set of keys and their associated values that MUST conform to the following ANBF syntax:

```
ODBCConnectionString = *(KeyValuePair SC) KeyValuePair [SC]
KeyValuePair = (Key EQ Value / SpaceStr)
Key = SpaceStr KeyName
KeyName = (nonSP-SC-EQ *nonEQ)
Value = (SpaceStr ValueFormat1 SpaceStr) / (ValueContent2)
ValueFormat1 = LCB ValueContent1 RCB
ValueContent1 = *(nonRCB / ESCAPEDRCB)
ValueContent2 = SpaceStr / SpaceStr (nonSP-LCB-SC) *nonSC
nonRCB = %x01-7C / %x7E- FFFF ; not "]"
nonSP-LCB-SC = %x01-1F / %x21-3A / %x3C-7A / %x7C- FFFF ; not space, "{" or ";"
nonSP-SC-EQ = %x01-1F / %x21-3A / %x3C / %x3E- FFFF ; not space, ";" or "="
nonEQ = %x01-3C / %x3E- FFFF ; not "="
nonSC = %x01-003A / %x3C- FFFF ; not ";"
```

2.1.2.1 KeyValuePair

If there are only spaces inside a **KeyValuePair**, the **KeyValuePair** MUST be ignored. Otherwise, the **KeyValuePair** MUST contain a **Key** and a **Value** separated by **EQ**. Each **KeyValuePair** specifies a piece of information in a connection string.

2.1.2.2 Key

Any spaces preceding the **Key** MUST be ignored. Any spaces before **EQ** MUST be regarded as a part of the **KeyName**.

2.1.2.3 Value

Value MUST be either [ValueFormat1](#) or [ValueContent2](#).

2.1.2.4 ValueFormat1

ValueFormat1 is recommended to use when there is a need for **Value** to contain **LCB**, **RCB**, or **EQ**. **ValueFormat1** MUST be used when the **Value** contains **SC** or starts with **LCB**.

ValueContent1 MUST be enclosed by **LCB** and **RCB**. Spaces before the enclosing **LCB** and after the enclosing **RCB** MUST be ignored.

2.1.2.5 ValueContent1

ValueContent1 MUST be contained in **ValueFormat1**. If there is an **RCB** in the **ValueContent1**, it MUST use the two-character sequence **ESCAPEDRCB** to represent the one-character value **RCB**.

2.1.2.6 ValueContent2

ValueContent2 MUST not start with **LCB**. **SC** MUST NOT appear in **ValueContent2**. The preceding space MUST be ignored.

ValueContent2 MUST NOT be enclosed by **LCB** and **RCB**.

2.2 Generic Keys

A key is a generic key if the **KeyName** is **Driver**, **DSN**, **FileDSN**, **PWD**, **SaveFile**, or **UID**. Otherwise, it is a driver-driver-specific key. This section specifies the meaning of each generic key, as shown in the following table. All Open Database Connectivity (ODBC) drivers MUST process generic keys as described in this section.

Generic keys MUST NOT be used as driver-specific keys.

| Key | Meaning |
|----------|---|
| Driver | Specifies the name of the ODBC driver. |
| DSN | Specifies the name of the Data Source Name (DSN) . The length of its value MUST be less than or equal to 32 characters |
| FileDSN | Specifies the absolute path or relative path<2> to the File DSN . For more information about File DSN, see [MSDN-CUFDS] . |
| PWD | Specifies the password associated with the specified UID. |
| SaveFile | Specifies the name of a file into which the current connection information is saved, as a File DSN upon a successful connection. If no connection is established, no file is written. This can be a file located either on a remote machine or on the local machine.<3> |
| UID | Specifies the user identification to be used when connecting to the data source. |

2.2.1 Default Values for Generic Keys

None of the generic keys have a default value. A key with a value of an empty string MUST NOT be treated as a missing key.

2.2.2 Case-sensitivity

The names of all generic keys are case-insensitive. The values of the **Driver** and **DSN** generic keys are case-insensitive. The case-sensitivity of the value of all the other generic keys is determined by

the driver. The case-sensitivity of the name and value of driver-specific keys are determined by the driver.

2.2.3 Multiple Occurrences of the Same Generic Key

If there are multiple occurrences of the same generic key, the value of the last [KeyValuePair](#) MUST override the previous occurrence. This rule applies only to generic keys. The resolution of multiple occurrences of a driver-specific key is driver-specific.

2.3 Driver Conflict Resolution

2.3.1 Determining Which Driver Is Used

The **Driver**, **DSN** and **FileDSN** keys specify which driver is used to establish a connection to a data source. A connection string MUST contain at least one of these three **KeyValuePair**s.

If a connection string contains more than one instance of the three different generic keys, **DSN**, **FileDSN**, or **Driver**, it MUST use the key that occurs first.

2.3.2 Conflicts between the Content of a File DSN and Connection String

After the rules specified in section [2.2.3](#) and section [2.3.1](#) are applied, it is possible for the driver information specified in the content of the file specified in the **FileDSN** value to differ from the driver information specified in the connection string. The following conditions indicate a possible conflict:

- If the **FileDSN** key precedes the **DSN** and **Driver** keys in the connection string, and
- If the connection string specifies the **Driver** key, and
- If the value of the **Driver** key is different in the connection string than the value of the **Driver** key in the content of the file specified in the **FileDSN** value

If all of these conditions are true, all information from the file specified in the **FileDSN** value MUST be ignored and all other **KeyValuePair**s specified in the connection string MUST be used.

If at least one of the conditions is false, generic key information specified in the connection string MUST take precedence over information specified in the content of the file specified in the **FileDSN** value. Driver-specific **KeyValuePair**s from the connection string are inserted before driver-specific **KeyValuePair**s from the file specified in the **FileDSN** value in the process of combining information.

2.4 SqlConnectionString

The **SqlConnectionString**[<4>](#) structure is an implementation of **ODBCConnectionString** that has additional restrictions on the generic key. The structure also defines driver-specific keys as specified in section [2.4.2](#).

2.4.1 SqlConnectionString Additional Restrictions

Only the driver-specific keys specified in section [2.4.2](#) are valid in the **SqlConnectionString** structure. Any other **KeyValuePair**s MUST be ignored.

The length of a value of a generic key or a driver-specific key SHOULD be less than or equal to 260 characters. Values longer than 260 characters MUST be truncated.

If there are multiple occurrences of a single driver-specific key, the value from the first **KeyValuePair** takes precedence. This is opposite of the rule for generic keys specified in section [2.2.3](#).

If the **DSN** key precedes the **FileDSN** and **Driver** keys, the connection information SHOULD be obtained from the settings as specified by the value of the **DSN** key. The **DSN** SHOULD be created and exist in the system; otherwise, it is equivalent to having no **DSN** key specified. **KeyValuePairs** of driver-specific keys that are specified in the connection string MUST take precedence over the **KeyValuePairs** specified in the **DSN**.

Both the key and value of generic keys and driver-specific keys are case-insensitive, with the exception of the value of the **PWD** key.

There are no default values for driver-specific keys if the **KeyValuePair** is missing, unless otherwise specified.

2.4.2 Driver-Specific Keys

This section discusses the meaning of each driver-specific key accepted by **SqlConnectionString** and the meaning of their possible values. The following table provides a list of the driver-specific keys and a brief description of each.

| Key | Meaning |
|---------|--|
| Addr | Synonym for the key Address . |
| Address | <p>Specifies the network address of an instance of the database server.</p> <p>If the value of the Address key is not specified, the default value is the value of the Server key.</p> <p>Address MUST be one of the following formats; TCP Format or NP Format.</p> <p>TCP Format</p> <p>tcp:<host name>\<instance name> tcp:<host name>,<TCP/IP port number></p> <p>The TCP Format MUST start with the prefix "tcp:" and is followed by the database instance, specified by a <host name> and an <instance name>.</p> <p>The <host name> MUST be specified in one of three ways: NetBIOSName [RFC1002], IPv4Address [RFC791] or IPv6Address [RFC2460].</p> <p>The <instance name> is used to resolve to a particular TCP/IP port number [RFC793], on which a database instance is hosted. Alternatively, specifying a <TCP/IP port number> directly is also allowed. If both <instance name> and <port number> are not present, the default database instance is used. For more information about instance name, see [MSDN-UNI].</p> <p>NP Format</p> <p>np:\\<host name>\pipe\<pipe name></p> <p>The NP Format MUST start with the prefix "np:" and is followed by a named pipe name.</p> <p>The <host name> MUST be specified in one of three ways: NetBIOSName [RFC1002], IPv4Address [RFC791] or IPv6Address [RFC2460].</p> <p>The <pipe name> is used to identify the database instance to which to be connected.</p> <p>If the value of the Network key is specified, the prefixes "tcp:" and "np:" SHOULD NOT be specified. <5></p> <p>For more information about the format of the Address key, see [MSKB-313295].</p> |

| Key | Meaning |
|------------------|--|
| | For more information about named pipes, see [PIPE] . |
| AnsiNPW | <p>Specifies whether American National Standards Institute (ANSI)-defined behavior is used for handling NULL comparisons, character data padding, warnings, and NULL concatenation as specified in [TSQL]. The valid values are "Yes" and "No." If the value "Yes" is not specified, the value "No" is used.</p> <p>If the AnsiNPW key is not specified in the connection string, the value MUST be obtained from the contents of the settings of the DSN key. If the key is not specified in a DSN or the given DSN does not exist, the default value is "Yes."</p> |
| APP | Sets the application identifier as specified by the ibAppName and cchAppName fields in section 2.2.6.3 of [MS-TDS] . The default value is the name of the current ODBC application. |
| AttachDBFileName | <p>Sets the name of the primary file of an attachable database<6> as specified by the ibAtchDBFile and cchAtchDBFile fields in section 2.2.6.3 of [MS-TDS].</p> <p>If the value of the AttachDBFileName key is specified in the connection string, the database is attached and becomes the default database for the connection.</p> <p>If the Database key is not specified and if the database was previously attached it will not be reattached. The previously attached database will be used as the default database for the connection.</p> <p>If the Database key is specified together with the AttachDBFileName key, the value of the Database key SHOULD be used as the alias. But if the name is already used in the other attached database, the connection MUST fail.</p> <p>If the AttachDBFileName key is not specified in the connection string, the value MUST be obtained from the contents of the settings of the DSN key. If the key is not specified in a DSN or the given DSN does not exist, the default value is an empty string.</p> <p>For more information about attachable databases, see [MSDN-DAD].</p> <p>For more information about default databases, see [MSDN-SD].</p> |
| AutoTranslate | <p>Specifies whether OEM characters or characters in the ANSI character set are translated between the code page of the client and the code page of the database when characters are retrieved from, or sent to, the database.<7> The valid values are "Yes" or "No." If the value "Yes" is not specified, the value "No" is used.</p> <p>If the AutoTranslate key is not specified in the connection string, the value MUST be obtained from the contents of the settings of the DSN key. If the key is not specified in a DSN or the given DSN does not exist, the default Value is "Yes."</p> |
| Database | <p>Sets the name of the initial or default database of a data source as specified by the ibDatabase and cchDatabase fields in section 2.2.6.3 of [MS-TDS].</p> <p>If the Database key is not specified in the connection string, the value MUST be obtained from the contents of the settings of the DSN key. If the key is not specified in a DSN or the given DSN does not exist, the default value is "(default)," which means the default database.</p> <p>For more information about default databases, see [MSDN-SD].</p> |
| Encrypt | <p>Specifies whether encryption is used as specified by section 2.2.6.4 of [MS-TDS]. The valid values are "Yes" and "No." If the value "Yes" is not specified, the value "No" is used.</p> <p>If the Encrypt key is not specified in the connection string, the value MUST be obtained from the contents of the settings of the DSN key. If the key is not specified in a DSN or the given DSN does not exist, the default Value is "No."</p> |

| Key | Meaning |
|-------------|--|
| Language | <p>Sets the language used for database server warning or error messages as specified by the ibLanguage and cchLanguage fields in section 2.2.6.3 of [MS-TDS].</p> <p>If the Language key is not specified in the connection string, the value MUST be obtained from the contents of the settings of the DSN key. If the key is not included in the connection string, or the given DSN does not exist, the default value is "(default)," which means the default language.</p> <p>If the language specified by the value of the Language key is not installed on the server, the key MUST be ignored by the server, and, error or warning messages will be sent to the ODBC application in English.</p> <p>For more information about default language, see [MSDN-DLO].</p> |
| Net | Synonym for the key Network . |
| Network | <p>Specifies the network component used in communication between the client and the data source. The behavior is platform-dependent.<8></p> <p>If the Network key is specified, the value of the Address key takes precedence over the value of the Server key if both the Server and Address keys are present.</p> <p>If the Network key is not specified, the value of the Server key takes precedence over the value of the Address key if both the Server and Address keys are present.</p> |
| QueryLog_On | <p>Specifies whether query profiling of the data source is enabled. The valid values are "Yes" or "1" which are equivalent, or "No." If the value "Yes" or "1" is not specified, the value "No" is used.</p> <p>If the QueryLog_On key is not specified in the connection string, the value MUST be obtained from the contents of the settings of the DSN key. If the key is not specified in a DSN or the given DSN does not exist, the default Value is "No."</p> |
| QuotedId | <p>Specifies whether SQL-92 rules are used regarding the use of quotation marks in SQL statements as specified in [TSQL]. The valid values are "Yes" or "No." If the value "Yes" is not specified, the value "No" is used.</p> <p>If the QuotedId key is not specified in the connection string, the value MUST be obtained from the contents of the settings of the DSN key. If the key is not specified in a DSN or the given DSN does not exist, the default Value is "Yes."</p> <p>For more information about legacy Transact-SQL rules used prior to SQL-92, see [MSDN-UOMSS].</p> |
| Regional | <p>Specifies whether client locale settings<9> are used when converting currency, date, and time data to character data. The valid values are "Yes" or "No." If the value "Yes" is not specified, the value "No" is used.</p> <p>If the value is "No," ODBC standard strings MUST be used to represent currency, date, and time data.</p> <p>This only applies on result set data and MUST NOT be applied on parameter data.</p> <p>The Regional key does not control the conversion from character data into currency, date, and time data, in which ODBC standard string MUST be used.</p> <p>If the Regional key is not specified in the connection string, the value MUST be obtained from the contents of the settings of the DSN key. If the key is not specified in a DSN or the given DSN does not exist, the default Value is "No".</p> <p>For more information about ODBC standard strings, see [MSDN-UDTD-ODTF].</p> |
| Server | <p>Specifies the name of a database server to which to connect.<10></p> <p>If the Server key is not specified in the connection string, the value MUST be obtained from the contents of the settings in the DSN key. If the settings in the</p> |

| Key | Meaning |
|--------------------|--|
| | <p>DSN key do not define the Server key, the default value is the Value of the DSN key. If the Driver key is present, the Server key MUST also be present.</p> |
| StatsLog_On | <p>Specifies whether driver performance statistics are captured. The valid values are "Yes" or "1" which are equivalent, or "No." If the value "Yes" or "1" is not specified, the value "No" is used.</p> <p>If the StatsLog_On key is not specified in the connection string, the value MUST be obtained from the contents of the settings of the DSN key. If the key is not specified in a DSN or the given DSN does not exist, the default Value is "No."</p> |
| Trusted_Connection | <p>Specifies whether a user connects through a user account<11> by using either Kerberos [RFC4120] or another platform-specific authentication<12> as specified by the fIntSecurity field in section 2.2.6.3 of [MS-TDS]. The valid values are "Yes," "1," or empty string, which are equivalent, or "No." If the value "No" is not specified, the value "Yes" is used.</p> <p>If the value is "No," the UID and PWD keys MUST be used to establish a connection with the data source.</p> <p>If the DSN key and the UID key is not included in the connection string, or the value of the UID key is an empty string, the value of the Trusted_Connection key MUST be "Yes." If the Trusted_Connection key is not specified in the connection string, the value MUST be obtained from the contents of the settings in the DSN key. If the Trusted_Connection key is not specified in DSN or the given DSN does not exist, the default value is "No."</p> <p>If the value of the Trusted_Connection key is "Yes," both the UID and PWD keys MUST be ignored. Otherwise, the UID key MUST be specified.</p> |
| WSID | <p>Sets the workstation identifier as specified by the ibHostName and cchHostName fields in section 2.2.6.3 of [MS-TDS]. The default value is the name of the workstation that is running the ODBC application.</p> |

3 Structure Examples

The following section contains connection string examples based on the **SqlConnection** structure.

3.1 Trusted Connection

```
Driver=SQL Server;Server=ServerName;Database=DatabaseName; Trusted_Connection=Yes;
```

"Driver=SQL Server" specifies that Microsoft SQL Server is the driver for this connection.

"Server=ServerName" specifies that ServerName is the name of the server to which the connection is established.

"Database=DatabaseName" specifies that DatabaseName is the name of the data source.

"Trusted_Connection=Yes" specifies that a user account<[13](#)> is used to establish this connection.

3.2 Standard Security Connection

```
Driver=SQL Server;Server=ServerName;Database=DatabaseName; UID=UserName;PWD=UserPassword;
```

"UID=UserName" specifies that UserName is the name of the user that establishes the connection.

"PWD=UserPassword" specifies that UserPassword is the password of the user that establishes the connection.

3.3 Named Instance

```
Driver=SQL Server;Server=ServerName\InstanceName;Database=DatabaseName;  
Trusted_Connection=Yes;
```

"Server=ServerName\InstanceName" specifies that the connection is being established to the named instance InstanceName on the server whose name is ServerName.

3.4 Network

```
Driver=SQL Server;Server=ServerName;Trusted_Connection=Yes;Network =DBMSSOEN;
```

"Network=DBMSSOEN" specifies that the name of the network component used to communicate with the data source is DBMSSOEN. The TCP/IP protocol is used to communicate with the data source.

3.5 Escaped Right Brace

```
DSN=testDSN; UID=sa; PWD={abc;}def}
```

"PWD={abc;}def}" specifies that the password is 8 characters in length, "abc;}def".

3.6 Leading and Trailing Spaces

```
DSN=testDSN; UID={ sa }; PWD=myPwd
```

"UID={ sa }" specifies that the UID is 4 characters, " sa ".

3.7 Values Enclosed by Braces

```
UID=sa; PWD={myPwd}; DATABASE=TestingDB; DSN={testDSN};
```

This example illustrates the use of values that are enclosed within braces. Because the value of the **PWD** key is enclosed within braces, the value is "myPwd".

3.8 Driver Conflict Resolution

```
FileDSN=C:\dsn\file.dsn; DSN=testDSN; UID=sa; PWD=myPwd;
```

This example illustrates conflict resolution between the driver and the generic keys **DSN**, **FileDSN**, **UID**, and **PWD**. Because the **FileDSN** key precedes the **DSN** key, the connection information is read from the "C:\dsn\file.dsn" file.

3.9 Multiple Instances of a Generic Key

```
UID=sa2; PWD=myPwd; DATABASE=TestingDB; DSN=testDSN; UID=sa;
```

This example illustrates multiple instances of the same generic key. Because the **UID** key appears twice in the connection string, the value of the key is "sa".

3.10 Multiple Instances of Driver-Specific Key

```
Trusted_Connection=Yes; Driver=SQL Server; Database=tempdb; Server=svr1;  
Trusted_Connection=No
```

This example illustrates multiple instances of the same driver-specific key. Because the **Trusted_Connection** key appears twice in the connection string, the value of the key is "Yes".

4 Security Considerations

4.1 Security Considerations for Implementers

A connection string can contain credential information in clear text. [Applications](#) should take special care when accessing it and avoid passing the credential information in the connection string whenever possible. Instead, it is recommended that applications use a driver-specific key such as the **Trusted_Connection** key in the **SqlConnection** structure.

4.2 Index of Security Parameters

The following table lists the security parameters for this protocol, along with the numbers of the document sections where they are discussed.

| Security parameter | Section |
|---------------------------|-------------------------------|
| <i>Encrypt</i> | Section 2.4.2 |
| <i>PWD</i> | Section 2.2 |
| <i>Trusted_Connection</i> | Section 2.4.2 |
| <i>UID</i> | Section 2.2 |

5 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft SQL Server 2005
- 2007 Microsoft Office system
- Windows Vista operating system
- Windows Server 2008 operating system
- Microsoft SQL Server 2008
- Windows 7 operating system
- Windows Server 2008 R2 operating system
- Microsoft SQL Server 2008 R2
- Microsoft Office 2010
- Microsoft SQL Server 2012
- Windows Server 2012 operating system
- Windows 8 operating system
- Windows 8.1 operating system
- Windows Server 2012 R2 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

[<1> Section 1.1:](#) Microsoft provides an implementation of ODBC according to [\[TSQL-CLI-2003\]](#) and provides extension to the standard. For example, the `SQLDriverConnect` API is an extension to the standard, and it takes a connection string as an input argument to provide connection information for connecting to a data source.

[<2> Section 2.2:](#) A file path that conforms to the format specified in [\[MSDN-FILE\]](#).

[<3> Section 2.2:](#) A file path that conforms to the format specified in [\[MSDN-FILE\]](#).

[<4> Section 2.4:](#) **SqlConnection**String is the connection string used by the SQL Server ODBC Driver, and it is an implementation of **ODBCConnection**String with driver-specific behaviors.

[<5> Section 2.4.2:](#) In Windows Vista, if the value of the Network key is specified as "DBNETLIB," the protocol prefix ("tcp:" and "np:") can still be used.

<6> [Section 2.4.2](#): A file path that conforms to the format specified in [\[MSDN-FILE\]](#).

<7> [Section 2.4.2](#): If the Value is "Yes," OEM/ANSI character translation is enabled. Otherwise, OEM/ANSI character translation is not performed. When AutoTranslate is set to "Yes," the ODBC driver uses **Unicode** to convert data moved between character variables on the client and character columns, variables, or parameters in a SQL Server database.

When data is sent from a character variable on the client to a character column, variable, or parameter in a SQL Server database, the ODBC driver first converts from SQL_C_CHAR to Unicode by using the active code page of the client, then from Unicode back to character by using the code page of the server. When data is sent from a character column, variable, or parameter in a SQL Server database to a character variable on the client, the ODBC driver first converts from character to Unicode by using the code page of the server, then from Unicode back to character by using the active code page of the client.

Because all of these conversions are done by the ODBC driver executing on the client, the server code page needs to be one of the code pages installed on the client computer.

The AutoTranslate setting has no effect on these conversions: moving data between character client variables and Unicode columns, variables, or parameters in SQL Server databases, and moving data between Unicode client variables and character columns, variables, or parameters in SQL Server databases.

<8> [Section 2.4.2](#): For Windows Vista, the behaviors of the values for the various components are described in the following table.

| Value | Meaning |
|----------|---|
| DBNMPNTW | The component DBNMPNTW implements the named pipes protocol [PIPE] . |
| DBMSSOCN | The component DBMSSOCN implements the TCP/IP protocol. |
| DBMSSPXN | The component DBMSSPXN implements the NWLink IPX/SPX protocol. |
| DBMSRPCN | The component DBMSRPCN implements the Multi-Protocol protocol. |
| DBMSVINN | The component DBMSVINN implements the Banyan Vines protocol. |
| DBMSADSN | The component DBMSADSN implements the ADSP protocol. |
| DBMSSHRN | The component DBMSSHRN implements the Shared Memory protocol. |
| DBMSLPCN | The component DBMSLPCN implements the Shared Memory protocol. |
| DBNETLIB | The default search order of network component can be used. |

If the value specified is not listed above or the Network key is not specified, the default search order of network component can be used. For more information about the default search order of network component, see [\[MSKB-328383\]](#).

<9> [Section 2.4.2](#): On Windows Vista, client locale settings are stored in the **registry**.

<10> [Section 2.4.2](#): The value of the **Server** key can be either the name of a server on the network, or the name of a SQL Server Configuration Manager advanced server entry. "(local)" is also a valid name representing a copy of SQL Server that is running on the same computer. SQL Server 2005 supports multiple instances of SQL Server that is running on the same computer. To specify a named instance of SQL Server, the server name can be specified as

ServerName\InstanceName. Otherwise, if InstanceName is omitted, it connects to the default database instance. Which database instance is the default is defined when SQL Server is installed on the machine.

[<11> Section 2.4.2:](#) In Windows Vista, this is a Windows user account.

[<12> Section 2.4.2:](#) On Windows Vista, NTLM authentication [\[NTLM\]](#) will be used when the value of the Trusted_Connection key is "Yes."

[<13> Section 3.1:](#) On Windows Vista, this is a Microsoft Windows user account.

6 Change Tracking

This section identifies changes that were made to the [MS-ODBCSTR] protocol document between the June 2013 and August 2013 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

| Section | Tracking number (if applicable) and description | Major change (Y or N) | Change type |
|--|--|-----------------------|------------------|
| 1.2.2 Informative References | Removed [MSDN-ODBCPR]. | N | Content updated. |
| 5 Appendix A: Product Behavior | Added Windows 8.1 operating system and Windows Server 2012 R2 operating system to the list of applicable products. | Y | Content updated. |

7 Index

A

[Applicability](#) 9

C

[Capability negotiation](#) 9

[Change tracking](#) 23

E

Examples

[overview](#) 17

F

[Fields - vendor-extensible](#) 9

G

[Glossary](#) 6

I

Implementer

[security considerations](#) 19

[Index of security parameters](#) 19

[Informative references](#) 8

[Introduction](#) 6

N

[Normative references](#) 7

O

[Overview](#) 8

P

Parameters

[security index](#) 19

[Product behavior](#) 20

R

[References](#) 7

[informative](#) 8

[normative](#) 7

[Relationship to other protocols](#) 9

S

Security

[overview](#) 19

[Security: implementer considerations; overview;](#)

[parameter index](#) 19

Structures

[overview](#) 10

T

[Tracking changes](#) 23

V

[Vendor-extensible fields](#) 9

[Versioning](#) 9