

[MS-ODBCSTR-Diff]:

ODBC Connection String Structure

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might 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 that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology,

there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Comments
5/22/2019	Released Preview document.

PREVIEW

Table of Contents

1	Introduction	4
1.1	Glossary	4
1.2	References	5
1.2.1	Normative References	5
1.2.2	Informative References	6
1.3	Overview	7
1.4	Relationship to Other Protocols	7
1.5	Applicability Statement	7
1.6	Versioning and Capability Negotiation	7
1.7	Vendor-Extensible Fields	7
2	Structures	8
2.1	ABNF Rules	8
2.1.1	Common ABNF Rules	8
2.1.2	ODBC Connection String Format	8
2.1.2.1	KeyValuePair	8
2.1.2.2	Key	8
2.1.2.3	Value	8
2.1.2.4	ValueFormat1	8
2.1.2.5	ValueContent1	9
2.1.2.6	ValueContent2	9
2.2	Generic Keys	9
2.2.1	Default Values for Generic Keys	9
2.2.2	Case-sensitivity	10
2.2.3	Multiple Occurrences of the Same Generic Key	10
2.3	Driver Conflict Resolution	10
2.3.1	Determining Which Driver Is Used	10
2.3.2	Conflicts between the Content of a File DSN and Connection String	10
3	Structure Examples	11
3.1	Trusted Connection	11
3.2	Standard Security Connection	11
3.3	Named Instance	11
3.4	Network	11
3.5	Escaped Right Brace	11
3.6	Leading and Trailing Spaces	11
3.7	Values Enclosed by Braces	11
3.8	Driver Conflict Resolution	12
3.9	Multiple Instances of a Generic Key	12
3.10	Multiple Instances of Driver-Specific Key	12
4	Security Considerations	13
4.1	Security Considerations for Implementers	13
4.2	Index of Security Parameters	13
5	Appendix A: Product Behavior	14

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. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

American National Standards Institute (ANSI) character set: A character set defined by a **code page** approved by the American National Standards Institute (ANSI). The term "ANSI" as used to signify Windows code pages is a historical reference and a misnomer that persists in the Windows community. The source of this misnomer stems from the fact that the Windows code page 1252 was originally based on an ANSI draft, which became International Organization for Standardization (ISO) Standard 8859-1 [ISO/IEC-8859-1]. In Windows, the ANSI character set can be any of the following code pages: 1252, 1250, 1251, 1253, 1254, 1255, 1256, 1257, 1258, 874, 932, 936, 949, or 950. For example, "ANSI application" is usually a reference to a non-**Unicode** or code-page-based application. Therefore, "ANSI character set" is often misused to refer to one of the character sets defined by a Windows code page that can be used as an active system code page; for example, character sets defined by code page 1252 or character sets defined by code page 950. Windows is now based on **Unicode**, so the use of ANSI character sets is strongly discouraged unless they are used to interoperate with legacy applications or legacy data.

code page: An ordered set of characters of a specific script in which a numerical index (code-point value) is associated with each character. Code pages are a means of providing support for character sets and keyboard layouts used in different countries. Devices such as the display and keyboard can be configured to use a specific code page and to switch from one code page (such as the United States) to another (such as Portugal) at the user's request.

connection string: A character string expression that uniquely identifies the data store to use for a particular query or set of queries and the methods, including authentication information and configuration options, for connecting to that data store.

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.

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

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.

encryption: In cryptography, the process of obscuring information to make it unreadable without special knowledge.

File DSN: A text file that contains **Data Source Name (DSN)** information.

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

named pipe: A named, one-way, or duplex pipe for communication between a pipe server and one or more pipe clients.

ODBC application: An application that 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 [ISO/IEC9075-3:2008] and provides extensions to that standard.

original equipment manufacturer (OEM) character: An 8-bit encoding used in MS-DOS and Windows operating systems to associate a sequence of bits with specific characters. The ASCII character set maps the letters, numerals, and specified punctuation and control characters to the numbers from 0 to 127. The term "code page" is used to refer to extensions of the ASCII character set that map specified characters and symbols to the numbers from 128 to 255. These code pages are referred to as OEM character sets. For more information, see [MSCHARSET].

registry: A local system-defined database in which applications and system components store and retrieve configuration data. It is a hierarchical data store with lightly typed elements that are logically stored in tree format. Applications use the registry API to retrieve, modify, or delete registry data. The data stored in the registry varies according to the version of the operating system.

Unicode: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The **Unicode** standard [UNICODE5.0.0/2007] provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

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.

[ISO/IEC9075-3:2008] ISO/IEC, "Information technology — Database languages — SQL — Part 3: Call-Level Interface (SQL/CLI)", ISO/IEC 9075-3:2008, July 2008, <https://webstore.iec.ch/publication/11780>

Note There is a charge to download the specification.

[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.rfc-editor.org/rfc/rfc1002.txt>

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

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

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

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

[RFC791] Postel, J., Ed., "Internet Protocol: DARPA Internet Program Protocol Specification", RFC 791, September 1981, <http://www.rfc-editor.org/rfc/rfc791.txt>

[RFC793] Postel, J., Ed., "Transmission Control Protocol: DARPA Internet Program Protocol Specification", RFC 793, September 1981, <http://www.rfc-editor.org/rfc/rfc793.txt>

1.2.2 Informative References

[MSDN-CUFDS] Microsoft Corporation, "Connecting Using File Data Sources", <https://docs.microsoft.com/en-us/sql/odbc/reference/develop-app/connecting-using-file-data-sources>

[MSDN-DAD] Microsoft Corporation, "Database Detach and Attach (SQL Server)", <https://docs.microsoft.com/en-us/sql/relational-databases/databases/database-detach-and-attach-sql-server>

[MSDN-DLO] Microsoft Corporation, "default language Option", in SQL Server 2000 Retired Technical documentation, p. 710, <https://www.microsoft.com/en-us/download/confirmation.aspx?id=51958>

[MSDN-FILE] Microsoft Corporation, "Naming Files, Paths, and Namespaces", <https://docs.microsoft.com/en-us/windows/desktop/FileIO/naming-a-file>

[MSDN-NP] Microsoft Corporation, "Named Pipes", <https://docs.microsoft.com/en-us/windows/desktop/ipc/named-pipes>

[MSDN-NTLM] Microsoft Corporation, "Microsoft NTLM", <https://docs.microsoft.com/en-us/windows/desktop/SecAuthN/microsoft-ntlm>

[MSDN-SD] Microsoft Corporation, "Selecting a Database", [https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms180770\(v=sql.105\)](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms180770(v=sql.105))

[MSDN-UDTD-ODTF] Microsoft Corporation, "Using Date and Time Data -- ODBC Date-time Format", [https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/ms180878\(v=sql.100\)#ODBCdatetimeFormat](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/ms180878(v=sql.100)#ODBCdatetimeFormat)

[MSDN-UNI] Microsoft Corporation, "Using Named Instances", [https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2005/ms165614\(v=sql.90\)](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2005/ms165614(v=sql.90))

[MSDN-UOMSS] Kumar, A. and Brewer, A., "Using ODBC with Microsoft SQL Server", September 1997, [https://docs.microsoft.com/en-us/previous-versions/ms811006\(v=msdn.10\)](https://docs.microsoft.com/en-us/previous-versions/ms811006(v=msdn.10))

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

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

1.3 Overview

The **ODBC connection string** structure is a method for an **Open Database Connectivity (ODBC) application** to specify the parameters that are 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. In Appendix A, 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 ABNF 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

Note: Some of the information in this section is subject to change because it applies to a preliminary product version, and thus may differ from the final version of the software when released. All behavior notes that pertain to the preliminary product version contain specific references to it in the Product Behavior appendix.

A key is a **generic key** if the **KeyName** is **Driver**, **DSN**, **FileDSN**, **PWD**, **SaveFile**, or **UID**. Otherwise, a key is a **driver-specific key**.<1>

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 to the File DSN . For more information about file path formats, see [MSDN-FILE]. 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. For more information about file path formats, see [MSDN-FILE].
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 any previous occurrences. 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, that is, **Driver**, **DSN**, or **FileDSN**, the connection string 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.

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 <2> 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 =DBMSSOCN;
```

"Network=DBMSSOCN" specifies that the name of the network component used to communicate with the data source is DBMSSOCN. 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=srv1;  
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. It is advised that applications 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 and indicates where they are discussed in this document.

Security parameter	Section
<i>Encrypt</i>	Section 5
<i>PWD</i>	Section 2.2
<i>Trusted_Connection</i>	Section 5
<i>UID</i>	Section 2.2

5 Appendix A: Product Behavior

Note: Some of the information in this section is subject to change because it applies to a preliminary product version, and thus may differ from the final version of the software when released. All behavior notes that pertain to the preliminary product version contain specific references to it as an aid to the reader.

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- 2007 Microsoft Office system
- Microsoft Office 2010 system
- Microsoft Office 2013 system
- Microsoft Office 2016
- Microsoft SQL Server 2005
- Microsoft SQL Server 2008
- Microsoft SQL Server 2008 R2
- Microsoft SQL Server 2012
- Microsoft SQL Server 2014
- Microsoft SQL Server 2016
- Microsoft SQL Server 2017
- Microsoft SQL Server 2019 Community Technology Preview 3.0 (CTP 3.0)
- Windows Vista operating system
- Windows Server 2008 operating system
- Windows 7 operating system
- Windows Server 2008 R2 operating system
- Windows 8 operating system
- Windows Server 2012 operating system
- Windows 8.1 operating system
- Windows Server 2012 R2 operating system
- Windows 10 operating system
- Windows Server 2016 operating system
- Windows Server operating system
- Windows Server 2019 operating system

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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 2.2: The **SqlConnectionString** structure is the connection string that is used by the SQL Server ODBC **Driver** and is an implementation of **ODBCConnectionString** that puts additional restrictions on the **generic keys** and defines **driver-specific keys**.

SqlConnectionString Additional Restrictions

The additional restrictions on the generic keys that apply to **SqlConnectionString** are the following:

- The length of a value of a generic key or a driver-specific key has to be less than or equal to 260 characters. Values longer than 260 characters have to 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 that is specified in section 2.2.3.
- If the **DSN** key precedes the **FileDSN** and **Driver** keys, the connection information is obtained from the settings as specified by the value of the **DSN** key. The **DSN** should have been created and exist in the system. If there is no **DSN** key in the system, it is equivalent to having no **DSN** key specified. **KeyValuePairs** of driver-specific keys that are specified in the **connection string** have to 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.

Driver-Specific Keys

Only the following driver-specific keys are valid in the **SqlConnectionString** structure. Any other driver-specific **KeyValuePairs** have to be ignored. The table lists the driver-specific keys that are accepted by **SqlConnectionString** 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 has to be one of the following two formats:</p> <ul style="list-style-type: none"> ▪ TCP Format <p>tcp: <host name>\<instance name> tcp: <host name>,<TCP/IP port number></p> <p>The TCP format has to 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> has to be specified in one of three ways:</p> <ul style="list-style-type: none"> ▪ NetBIOSName [RFC1002] ▪ IPv4Address [RFC791] ▪ IPv6Address [RFC2460]

Key	Meaning
	<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> <ul style="list-style-type: none"> ▪ NP Format <p>np:\\<host name>\pipe\<pipe name></p> <p>The NP format has to start with the prefix "np:" and is followed by a named pipe name.</p> <p>The <host name> has to be specified in one of three ways:</p> <ul style="list-style-type: none"> ▪ NetBIOSName [RFC1002] ▪ IPv4Address [RFC791] ▪ IPv6Address [RFC2460] <p>The <pipe name> is used to identify the database instance to which to be connected. Only when the value of the Network key is specified as "DBNETLIB" can the protocol prefixes "tcp:" and "np:" be used. If any other value of the Network key is specified, the prefixes "tcp:" and "np:" cannot be used.</p> <p>For more information about the format of the Address key, see [MSKB-313295]. For more information about named pipes, see [MSDN-NP].</p>
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 [ISO/IEC9075-3:2008]. 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 has to 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	<p>Sets the application identifier as specified by the ibAppName and cchAppName fields in [MS-TDS] section 2.2.6.4. The default value is the name of the current ODBC application.</p>
AttachDBFileName	<p>Sets the name of the primary file of an attachable database as specified by the ibAtchDBFile and cchAtchDBFile fields in [MS-TDS] section 2.2.6.4. The file path has to conform to the format specified in [MSDN-FILE].</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 is used as the alias. But if the name is already used in the other attached database, the connection has to fail.</p> <p>If the AttachDBFileName key is not specified in the connection string, the value has to 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]. 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. 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 has to be</p>

Key	Meaning
	<p>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>In Microsoft implementations, 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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
<u>ClientCertificate</u>	<p><u>Specifies the certificate to be used for client authentication.</u></p> <p><u>Supported certificate file formats are the following:</u></p> <ul style="list-style-type: none"> ▪ <u>DER (Distinguished Encoding Rules)</u> ▪ <u>PFX (Personal Information Exchange)</u> ▪ <u>PEM (Privacy Enhanced Mail)</u> <p><u>For certificates in DER and PEM formats, an accompanying ClientKey MUST specify the location of the certificate's corresponding private key file. If file location is used for the authentication of a certificate in PFX format, an optional password can be specified if the private key inside the certificate is encrypted. The syntax options available for client-certificate authentication are the following:</u></p> <ul style="list-style-type: none"> ▪ <u>ClientCertificate=sha1:<hash value></u> <u>The ODBC driver uses a SHA1 hash value to locate a certificate in the Windows certificate store.</u> ▪ <u>ClientCertificate=subject:<subject></u> <u>The ODBC driver uses a subject value to locate a certificate in the Windows certificate store.</u> ▪ <u>ClientCertificate=file:<path to cert file></u> <u>The ODBC driver uses a file location to locate a certificate. This option is applicable to certificates when the private key is unencrypted.</u> ▪ <u>ClientCertificate=file:<path to cert file>,password:<password></u> <u>The ODBC driver uses a file location to locate a certificate and the optional password to decrypt the private key inside the certificate. This option is applicable to PFX-formatted certificates only.</u> <u>If the password contains any "," characters, an extra "," character is added immediately after each one. For example, if the password is "a,b,c", the escaped password present in the connection string is "a,,b,,c".</u> <p>Note <u>The ClientCertificate key is not supported by SQL Server 2005, SQL Server 2008.</u></p>

Key	Meaning
	SQL Server 2008 R2, SQL Server 2012, SQL Server 2014, SQL Server 2016, and SQL Server 2017.
ClientKey	<p>Specifies a file location of the private key that is used with a DER- or PEM-formatted certificate that is specified by ClientCertificate. Along with the location of the key file, an optional password can be specified if the key is encrypted.</p> <p>Supported private key file formats are the following:</p> <ul style="list-style-type: none"> ▪ PVK (Private Key) ▪ PEM (Privacy Enhanced Mail) <p>The syntax options available for using the private key are the following:</p> <ul style="list-style-type: none"> ▪ ClientKey=file:<path to key file> <p>The ODBC driver uses a file location to locate the unencrypted private key.</p> <ul style="list-style-type: none"> ▪ ClientKey =file:<path to cert file>,password:<password> <p>The ODBC driver uses a file location to locate the encrypted key and the optional password to decrypt the key.</p> <p>If the password contains any "," characters, an extra "," character is added immediately after each one. For example, if the password is "a,b,c", the escaped password present in the connection string is "a,,b,,c".</p> <p>Note The ClientKey key is not supported by SQL Server 2005, SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, SQL Server 2014, SQL Server 2016, and SQL Server 2017.</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 [MS-TDS] section 2.2.6.4.</p> <p>If the Database key is not specified in the connection string, the value has to 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 [MS-TDS] section 2.2.6.5. 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 has to 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>
Language	<p>Sets the language used for database server warning or error messages as specified by the ibLanguage and cchLanguage fields in [MS-TDS] section 2.2.6.4.</p> <p>If the Language key is not specified in the connection string, the value has to 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 is 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 Network key.
Network	<p>Specifies the network component that is used in communication between the client and the data source. The behavior is platform-dependent.</p> <p>If the Network key is specified and both the Server and Address keys are present, the</p>

Key	Meaning
	<p>value of the Address key takes precedence over the value of the Server key.</p> <p>If the Network key is not specified and both the Server and Address keys are present, the value of the Server key takes precedence over the value of the Address key.</p> <p>For Microsoft implementations, the values and their behaviors for the various components are described here.</p> <ul style="list-style-type: none"> ▪ DBNMPNTW -- The component DBNMPNTW implements the named pipes protocol [MSDN-NP]. ▪ 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. <p>This key does not have a default value. If the value that is specified is not listed above or if 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].</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 has to 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 [ISO/IEC9075-3:2008]. 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 has to 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 are used when converting currency, date, and time data to character data. In Microsoft implementations, client locale settings are stored in the registry.</p> <p>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 have to be used to represent currency, date, and time data.</p> <p>The Regional key applies only on result set data and cannot 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 has to be used.</p> <p>If the Regional key is not specified in the connection string, the value is 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
	For more information about ODBC standard strings, see [MSDN-UDTD-ODTF].
Server	<p>Specifies the name of a database server to which to connect. 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.</p> <p>If the Server key is not specified in the connection string, the value has to be obtained from the contents of the settings in the DSN key. If the settings in the 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 also has to be present.</p> <p>When multiple instances of Microsoft SQL Server are running on the same computer, the server name can be specified as ServerName\InstanceName. "(local)" is also a valid name that represents a copy of SQL Server that is running on the same computer. If InstanceName is omitted, the ODBC driver connects to the default database instance. Which database instance is the default is defined when SQL Server is installed on the machine.</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 has to 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 by using either Kerberos [RFC4120] or another platform-specific authentication as specified by the fIntSecurity field (for details, see [MS-TDS] section 2.2.6.4).</p> <p>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 have to be used to establish a connection with the data source.</p> <p>If the DSN key and the UID key are not included in the connection string or if the value of the UID key is an empty string, the value of the Trusted_Connection key has to be "Yes". If the Trusted_Connection key is not specified in the connection string, the value has to be obtained from the contents of the settings in the DSN key. If the Trusted_Connection key is not specified in DSN or if 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 are ignored. Otherwise, the UID key has to be specified.</p> <p>In Microsoft implementations, this user account is a Windows user account and NTLM authentication [MSDN-NTLM] is used when the value of the Trusted_Connection key is "Yes".</p>
WSID	<p>Sets the workstation identifier as specified by the ibHostName and cchHostName fields in [MS-TDS] section 2.2.6.4. The default value is the name of the workstation that is running the ODBC application.</p>

<2> Section 3.1: In Microsoft implementations, this user account is a Windows user account.