[MS-TDS]: Tabular Data Stream Protocol

This topic lists the Errata found in [MS-TDS] since it was last published. Since this topic is updated frequently, we recommend that you subscribe to these RSS or Atom feeds to receive update notifications.

Errata are subject to the same terms as the Open Specifications documentation referenced.

RSS
Atom

Errata below are for Protocol Document Version V22.0 – 2017/09/15.

Errata Published*	Description
2017/11/13	In Section 2.2.3.1.5, PacketID, a new product behavior note was added. The second sentence was changed from: Each time packet data is sent, the value of PacketID is incremented by 1, modulo 256. Changed to: Each time packet data is sent, the value of PacketID is incremented by 1, modulo 256.<14> <14> Section 2.2.3.1.5: Depending on the message type and provider, such as Microsoft SQL Server Native Client or Microsoft .NET Framework Data Provider for SQL Server, PacketID values start with either 0 or 1, which is an implementation choice. .NET Framework Data Provider for SQL Server uses 1.
2017/11/13	In Section 2.2.5.1.2, Collation Rule Definition, the first paragraph after the definition was changed from: A SQL (SortId==1) collation is one of a predefined set of sort orders. It is identified by having SortId with values as described by [MSDN-SQLCollation]. Changed to: A SQL collation is one of a predefined set of sort orders. The sort orders are identified with non-zero SortId values described by [MSDN-SQLCollation].

2017/11/13	In Section 2.2.5.3.3, Trace Activity Header, the stream-specific rules were changed from:
	
	Stream-Specific Rules:
	
	`ActivityId = 20BYTE ; client Activity ID`
	`; for debugging purposes`
	
	Changed to:
	
	Stream-Specific Rules:

Errata Published*	Description
	GUID_ActivityID = 16 bytes ; client application activity id ; used for debugging purposes ActivitySequence = ULONG ; client application activity sequence ; used for debugging purposes In Section 2.2.6.5, PRELOGIN, the stream-specific rules were changed in part from: Stream-Specific Rules: ... ACTIVITYID = 20BYTE ; client application activity id ; used for debugging purposes ... Changed to: Stream-Specific Rules: . . . GUID_ActivityID = 16 bytes ; client application activity id ; used for debugging purposes ActivitySequence = ULONG ; client application activity sequence ; used for debugging purposes . . .
2017/11/13	In Section 2.2.5.5.3, XML Values, a new product behavior note was added. The last sentence was changed from: Note The actual data value format associated with a XML data type definition stream uses the Microsoft SQL Server Binary XML structure [MS-BINXML] format. Changed to: Note The actual data value format associated with a XML data type definition stream uses the Microsoft SQL Server Binary XML structure [MS-BINXML] format.<22> <22> Section 2.2.5.5.3: When a .NET Framework Data Provider for SQL Server accesses an XML field, the returned data value is encoded in [MS-BINXML] format. For other providers, the value is sent in Unicode text format.

Errata Published*	Description
	In Section 2.2.6.4, LOGIN7, a new product behavior note was added regarding the fSendYukonBinaryXML bit. In the OptionFlags3 row of the Stream Parameter Details table, the third bullet was changed from: ● fSendYukonBinaryXML: 1 if XML data type instances are returned as binary XML. Changed to: ● fSendYukonBinaryXML: 1 if XML data type instances are returned as binary XML.<30> <30> Section 2.2.6.4: SQL Server implementations do not inspect the fSendYukonBinaryXML bit. When using the .NET Framework Data Provider for SQL Server, the server sends binary XML if the TDS version is 7.2 or later.
2017/11/13	In Section 2.2.6.4, LOGIN7, the last paragraph was changed from: Before submitting a password from the client to the server, for every byte in the password buffer starting with the position pointed to by IbPassword, the client SHOULD first swap the four high bits with the four low bits and then do a bit-XOR with 0xA5 (10100101). After reading a submitted password, for every byte in the password buffer starting with the position pointed to by IbPassword, the server SHOULD first do a bit-XOR with 0xA5 (10100101) and then swap the four high bits with the four low bits. Changed to: Before submitting a password from the client to the server, for every byte in the password buffer starting with the position pointed to by ibPassword or ibChangePassword, the client MUST first swap the four high bits with the four low bits and then do a bit-XOR with 0xA5 (10100101). After reading a submitted password, for every byte in the password buffer starting with the position pointed to by ibPassword or ibChangePassword, the server MUST first do a bit-XOR with 0xA5 (10100101) and then swap the four high bits with the four low bits.
2017/11/13	In Section 2.2.7.4, COLMETADATA, a new product behavior note was added regarding the fHidden bit flag. In the Flags row of the Token Stream Parameter Details table, the tenth bullet was changed from: ● fHidden is a bit flag. Its value is 1 if the column is part of a hidden primary key created to support a T-SQL SELECT statement containing FOR BROWSE. Changed to: ● fHidden is a bit flag. Its value is 1 if the column is part of a hidden primary key created to support a T-SQL SELECT statement containing FOR BROWSE.<41> <41> Section 2.2.7.4: SQL Server 2016 supports the fHidden flag only through a many-to-many result and by connecting via ODBC.
2017/11/13	In Section 3.3.5.1, Initial State, the two bullets were changed from:

Errata Published*	Description
	• Return to the client a PRELOGIN structure wrapped in a table response (0x04) packet with Encryption and enter "TLS/SSL Negotiation" state if encryption is negotiated. • Return to the client a PRELOGIN structure wrapped in a table response (0x04) packet without Encryption and enter unencrypted "Login Ready" state if encryption is not negotiated. Changed to: • Return to the client a PRELOGIN structure wrapped in a table response (0x04) packet and enter "TLS/SSL Negotiation" state if encryption is negotiated. • Return to the client a PRELOGIN structure wrapped in a table response (0x04) packet and enter unencrypted "Login Ready" state if encryption is not negotiated.

*Date format: YYYY/MM/DD