Windows Protocols Errata

This topic lists the Errata found in the Windows Protocols Technical Specifications, Overview Documents, and Reference documents since they were 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 are content issues in published versions of protocols documents that could impact an **implementation**. Examples of errata are errors or missing information in the normative sections of the Technical Specifications or in the use cases (examples) in the Technical Specifications and Overview Documents.

Content issues that don't impact an implementation, for example, editorial updates due to typos, formatting updates, and rewrites for readability and clarity, are **not** included in Errata.

The sections below list the Windows Protocols documents that contain active Errata (i.e., Errata not yet released with the documents on Docs.Microsoft.Com [DMC]) and provide links to archived Errata (i.e., Errata already released with the documents on DMC).

Protocols Documents with Active Errata

[MC-NBFX]: .NET Binary Format XML Data Structure

[MC-NMF]: .NET Message Framing Protocol

[MS-ADDM]: Active Directory Web Services: Data Model and Common Elements

[MS-ADFSPIP]: Active Directory Federation Services and Proxy Integration Protocol

[MS-ADFSWAP]: Active Directory Federation Service (AD FS) Web Agent Protocol

[MS-ADSC]: Active Directory Schema Classes

[MS-ADTS]: Active Directory Technical Specification

[MS-CIFS]: Common Internet File System (CIFS) Protocol

[MS-CMRP]: Failover Cluster: Management API (ClusAPI) Protocol

[MS-CRTD]: Certificate Templates Structure

[MS-CSRA]: Certificate Services Remote Administration Protocol

[MS-CSVP]: Failover Cluster: Setup and Validation Protocol (ClusPrep)

[MS-DFSC]: Distributed File System (DFS) Referral Protocol

[MS-DHCPE]: Dynamic Host Configuration Protocol (DHCP) Extensions

[MS-DNSP]: Domain Name Service (DNS) Server Management Protocol

[MS-DRSR]: Directory Replication Service (DRS) Remote Protocol

[MS-DTYP]: Windows Data Types

[MS-ECS]: Enterprise Client Synchronization Protocol

[MS-EMFPLUS]: Enhanced Metafile Format Plus Extensions

[MS-ERREF]: Windows Error Codes

[MS-EVEN]: EventLog Remoting Protocol

[MS-FRS2]: Distributed File System Replication Protocol

[MS-FSA]: File System Algorithms

[MS-FSCC]: File System Control Codes

[MS-GPOL]: Group Policy: Core Protocol

[MS-IKEE]: Internet Key Exchange Protocol Extensions

[MS-KILE]: Kerberos Protocol Extensions

[MS-LSAD]: Local Security Authority (Domain Policy) Remote Protocol

[MS-NRBF]: .NET Remoting: Binary Format Data Structure

[MS-NCNBI]: Network Controller Northbound Interface Specification

[MS-NNS]: .NET NegotiateStream Protocol

[MS-PAC]: Privilege Attribute Certificate Data Structure

[MS-PAR]: Print System Asynchronous Remote Protocol

[MS-RAI]: Remote Assistance Initiation Protocol

[MS-RDPBCGR]: Remote Desktop Protocol: Basic Connectivity and Graphics Remoting

[MS-RDPECAM]: Remote Desktop Protocol: Video Capture Virtual Channel Extension

[MS-RDPEDISP]: Remote Desktop Protocol: Display Update Virtual Channel Extension

[MS-RDPEGFX]: Remote Desktop Protocol: Graphics Pipeline Extension

[MS-RDPELE]: Remote Desktop Protocol: Licensing Extension

[MS-RDPEMT]: Remote Desktop Protocol: Multitransport Extension

[MS-RDPEPC]: Remote Desktop Protocol: Print Virtual Channel Extension

[MS-RDPERP]: Remote Desktop Protocol: Remote Programs Virtual Channel Extension

[MS-RDPRFX]: Remote Desktop Protocol: RemoteFX Codec Extension

[MS-RMPR]: Rights Management Services (RMS): Client-to-Server Protocol

[MS-RPRN]: Print System Remote Protocol

[MS-RRASM]: Routing and Remote Access Server (RRAS) Management Protocol

[MS-RRP]: Windows Remote Registry Protocol

[MS-SAMR]: Security Account Manager (SAM) Remote Protocol (Client-to-Server)

[MS-SFU]: Kerberos Protocol Extensions Service for User and Constrained Delegation Protocol

[MS-SMB2]: Server Message Block (SMB) Protocol Versions 2 and 3

[MS-SMBD]: SMB2 Remote Direct Memory Access (RDMA) Transport Protocol

[MS-SWN]: Service Witness Protocol

[MS-TDS]: Tabular Data Stream Protocol

[MS-TSTS]: Terminal Services Terminal Server Runtime Interface Protocol

[MS-WCCE]: Windows Client Certificate Enrollment Protocol

[MS-WKST]: Workstation Service Remote Protocol

[MS-WMIO]: Windows Management Instrumentation Encoding Version 1.0 Protocol

[MS-WSP]: Windows Search Protocol

[MS-WSUSAR]: Windows Server Update Services: Administrative API Remoting Protocol

[MS-WUSP]: Windows Update Services: Client-Server Protocol

[MS-XCA]: Xpress Compression Algorithm

Errata Archives

June 30, 2015 - Download

October 16, 2015 - Download

March 2, 2016 - Download

July 18, 2016 - Download

September 26, 2016 - Download

March 20, 2017 - Download

June 1, 2017 - Download

August 21, 2017 - Download

September 15, 2017 - Download

December 1, 2017 - Download

March 16, 2018 - Download

September 12, 2018 - Download

March 13, 2019 - Download

June 24, 2019 – Download

September 23, 2019 - Download

October 14, 2019 - Download

[MC-DTCXA]: MSDTC Connection Manager OleTx XA Protocol

This topic lists the Errata found in [MC-DTCXA] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MC-NBFX]: .NET Binary Format XML Data Structure

This topic lists the Errata found in [MC-NBFX] 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 V8.0 – 2019/03/13.

Errata Published*	Description
2019/12/09	In Section 2.2.3.30, QNameDictionaryTextRecord(0xBC), the length of the Name field was changed from 3 bytes to variable: Changed from: Name (3 bytes) Changed to: Name (variable) The packet diagram for the message was also changed to reflect the length.

*Date format: YYYY/MM/DD

[MC-NMF]: .NET Message Framing Protocol

This topic lists the Errata found in the MC-NMF document 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 V9.0 – 2018/03/16.

Errata Published*	Description
2018/07/02	In Section 2.2.6, Preamble Message, the field descriptions have been modified as follows and have been moved to follow the packet diagram. Changed from: The VersionRecord MUST be formatted as specified in section 2.2.3.1. The ModeRecord MUST be formatted as specified in section 2.2.3.2. The ViaRecord MUST be formatted as specified in section 2.2.3.3. The EnvelopeEncodingRecord MUST be formatted as specified in section 2.2.3.4 Changed to: VersionRecord (3 bytes): This field MUST be formatted as specified in section 2.2.3.1. ModeRecord (2 bytes): This field MUST be formatted as specified in section 2.2.3.2. ViaRecord (variable): This field MUST be formatted as specified in section 2.2.3.3. EnvelopeEncodingRecord (variable): This field MUST be formatted as specified in section 2.2.3.4

*Date format: YYYY/MM/DD

[MC-PRCR]: Peer Channel Custom Resolver Protocol

This topic lists the Errata found in [MC-PRCR] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 15, 2017 - Download

[MS-ABTP]: Automatic Bluetooth Pairing Protocol

This topic lists the Errata found in [MS-ABTP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-ADA2]: Active Directory Schema Attributes M

This topic lists the Errata found in the MS-ADA2 document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

[MS-ADA3]: Active Directory Schema Attributes N-Z

This topic lists the Errata found in the MS-ADA3 document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-ADDM]: Active Directory Web Services: Data Model and Common Elements

This topic lists the Errata found in [MS-ADDM] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

Errata below are for Protocol Document Version V15.0 – 2018/09/12.

Errata Published*	Description
2018/12/17	In Section 1.2.1, Normative References, the following reference has been deleted: [RFC4346] Dierks, T., and Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006, http://www.ietf.org/rfc/rfc4346.txt In Section 2.1, Endpoints, changed from: The ADWS protocol set uses two types of authentication. Each endpoint (except for the "mex" endpoint) supports one or the other. The forms of authentication are: • Windows Integrated: These endpoints use Transport Layer Security (TLS) [RFC4346] to protect the TCP transport. Integrated Windows authentication using the .Net Negotiate Stream protocol [MS-NNS] is used to authenticate the client to the server at the transport layer and to negotiate the session key used for TLS. Changed to: The ADWS protocol set uses two types of authentication. Each endpoint (except for the "mex" endpoint) supports one or the other. The forms of authentication are: • Windows Integrated: These endpoints use integrated Windows authentication with the .Net Negotiate Stream protocol [MS-NNS] to authenticate the client and provide message security at the transport layer.

* Date format: YYYY/MM/DD

[MS-ADFSOAL]: Active Directory Federation Services OAuth Authorization Code Lookup Protocol

This topic lists the Errata found in [MS-ADFSOAL] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-ADFSPIP]: Active Directory Federation Services and Proxy Integration Protocol

This topic lists the Errata found in the MS-ADFSPIP document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 30, 2015 - Download

July 18, 2016 - Download

September 15, 2017 - Download

September 12, 2018 - Download

Errata below are for Protocol Document Version V7.0 – 2018/09/12.

Errata Published*	Description
2019/05/27	In the sections listed below, the enum Certificate Type values have been changed from string to integer: Section 2.2.2.12, Port Type Section 2.2.2.14, TLS Query Behavior Section 2.2.2.15, Certificate Validation Section 2.2.2.16, Certificate Type Section 2.2.2.17, Error Type Section 3.10.5.1.1.3, Processing Details Section 3.10.5.1.1.3, Processing Details Section 3.11.5, Message Processing Events and Sequencing Rules Section 3.11.5.1, End-user X509 Certificate Success Processing Section 3.11.5.2, End-user X509 Certificate Common Processing Section 6, Appendix A: Full JSON Schema For details on the above changes, see the PDF doc here.
2019/05/27	In Section 3.10.5.1.1.2, Response Body, changed from: No response body is returned. Changed to: The response from the server MUST be returned to the client.

*Date format: YYYY/MM/DD

[MS-ADFSWAP]: Active Directory Federation Service (AD FS) Web Agent Protocol

This topic lists the Errata found in [MS-ADFSWAP] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

Errata below are for Protocol Document Version V11.0 – 2018/09/12.

Errata Published*	Description

2019/11/25	In Section 3.1.4.1.1.3, GetFsTrustInformationSoapOut, and Section 6, Appendix: Full WSDL, the value of minOccurs was changed from 1 to 0.
	
	Changed from:
	
	`<s:complexType name="VersionInformation">`
	`  <s:sequence>`
	`  <s:element minOccurs="1" maxOccurs="1" name="SoftwareVersion" type="s:long" />`
	`  <s:element minOccurs="1" maxOccurs="1" name="Guid" type="s1:guid" />`
	`  <s:element minOccurs="1" maxOccurs="1" name="Version" type="s:long" />`
	`  </s:sequence>`
	
	Changed to:
	
	`<s:complexType name="VersionInformation">`
	`  <s:sequence>`
	`  <s:element minOccurs="0" maxOccurs="1" name="SoftwareVersion" type="s:long" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="Guid" type="s1:guid" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="Version" type="s:long" />`
	`  </s:sequence>`

*Date format: YYYY/MM/DD

[MS-ADLS]: Active Directory Lightweight Directory Services Schema

This topic lists the Errata found in the MS-ADLS document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-ADSC]: Active Directory Schema Classes

This topic lists the Errata found in the MS-ADSC document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

Errata below are for Protocol Document Version V23.0 – 2018/03/16.

Errata Published*	Description

2019/09/16	In Section 2.243, Class samDomain, changed from:
	`(OA;CIOI;RPWP;3f78c3e5-f79a-46bd-a0b8-9d18116ddc79;;PS)`
	`S:(AU;SA;WDWOWP;;;WD)(AU;SA;CR;;;BA)(AU;SA;CR;;;DU)`
	
	Changed to:
	
	`(OA;CIOI;RPWP;3f78c3e5-f79a-46bd-a0b8-9d18116ddc79;;PS)`
	`(OA;CIIO;SW;9b026da6-0d3c-465c-8bee-5199d7165cba;bf967a86-0de6-11d0-a285-00aa003049e2;PS)`
	`(OA;CIIO;SW;9b026da6-0d3c-465c-8bee-5199d7165cba;bf967a86-0de6-11d0-a285-00aa003049e2;CO)`
	`S:(AU;SA;WDWOWP;;;WD)(AU;SA;CR;;;BA)(AU;SA;CR;;;DU)`

*Date format: YYYY/MM/DD

[MS-ADTS]: Active Directory Technical Specification

This topic lists the Errata found in the MS-ADTS document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

March 20, 2017 - Download

September 15, 2017 - Download

December 1, 2017 - Download

March 16, 2018 - Download

September 12, 2018 - Download

March 13, 2019 - Download

Errata below are for Protocol Document Version V9.0 – 2019/03/13.

Errata Published*	Description
2020/02/03	In Section 2.1, Transport, clarified that the value of the controlValue field of the Control structure is an OctetString of length zero as the LDAP_SERVER_DOMAIN_SCOPE_OID control is sent to the DC. Changed from: LDAP transport is specified in section 3.1.1.3, and in [RFC2251] section 5 (for LDAPv3), in [RFC1777] section 3 (for LDAPv2), and in [RFC1798] section 3.1 (for both LDAPv2 and LDAPv3). Changed to: LDAP transport is specified in section 3.1.1.3, and in [RFC2251] section 5 (for LDAPv3), in [RFC1777] section 3 (for LDAPv2), and in [RFC1798] section 3.1 (for both LDAPv2 and LDAPv3). When sending any control to the DC which does not require a controlValue field, the client sets the controlValue field of the Control structure to an OctetString of length zero and explicitly encodes this rather than omitting the controlValue field as indicated in [RFC2251] section 4.12.1. The server MUST ignore any controlValue provided in such requests. In Section 3.1.1.3.4.1.4, LDAP_SERVER_DOMAIN_SCOPE_OID, clarified that the value of the controlValue field of the Control structure is an OctetString of length zero as the LDAP_SERVER_DOMAIN_SCOPE_OID control is sent to the DC.

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>Changed from:

When sending this control to the DC, the controlValue field of the Control structure is omitted.

Changed to:

When sending this control to the DC, the controlValue field of the Control structure is set to an OctetString of length zero as described in section 2.1. The server MUST ignore any controlValue provided in the request. Sending this control to the DC does not cause the server to include any controls in its response.</td></tr>
<tr><td>2019/10/16</td><td>In Section 6.1.6.7.9, trustAttributes, the 'TANC' attribute description has been updated.

Changed from:

Only supported on Windows Server 2012 and later.

Changed to:

Initially supported on Windows Server 2012 and later. After [MSKB-4490425] is installed, this bit is superseded by the TRUST_ATTRIBUTE_CROSS_ORGANIZATION_ENABLE_TGT_DELEGATION bit.

In the same section a new TAEC attribute and description have been added:

<table><tr><td>TAEC

(TRUST_ATTRIBUTE_CROSS_ORGANIZATION_ENABLE_TGT_DELEGATION)
0x00000800</td><td>If this bit is set, tickets granted under this trust MUST be trusted for delegation. The behavior controlled by this bit is as specified in [MS-KILE] section 3.3.5.7.5.Only supported on Windows Server 2008 and later after [MSKB-4490425] updates are installed.</td></tr></table></td></tr>
<tr><td>2019/09/16</td><td>In Section 2.2.20.5.2, KEY_USAGE_FIDO, changed from:

authData: A base64-encoded public key.

Changed to:</td></tr>
</table>

Errata Published*	Description
	authData: A base64-encoded Authenticator Data structure, as described in section 6.1 of [W3C-WebAuthPKC1].
2019/04/29	In Section 3.1.1.2.5, Schema Modifications, information about the error 'unwillingToPerform / ERROR_DS_CANT_CREATE_UNDER_SCHEMA', which occurs when attempting to add any object other than a schema object in the schema NC, has been added. Changed from: A Delete of an attributeSchema or classSchema object (5) fails, with error unwillingToPerform / ERROR_DS_CANT_DELETE. There is no constraint on the amount of time between when an object (5) in the schema NC is successfully added or modified and when the DC enforces the updated schema (1). … Changed to: A Delete of an attributeSchema or classSchema object (5) fails, with error unwillingToPerform / ERROR_DS_CANT_DELETE. An attempt to add any object other than a schema object in the schema NC fails with the error unwillingToPerform / ERROR_DS_CANT_CREATE_UNDER_SCHEMA. There is no constraint on the amount of time between when an object (5) in the schema NC is successfully added or modified and when the DC enforces the updated schema (1). …

*Date format: YYYY/MM/DD

[MS-AIPS]: Authenticated Internet Protocol

This topic lists the Errata found in the MS-AIPS document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

[MS-APDS]: Authentication Protocol Domain Support

This topic lists the Errata found in the MS-APDS document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-AZOD]: Authorization Protocols Overview

This topic lists the Errata found in the MS-AZOD document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-BKRP]: BackupKey Remote Protocol

This topic lists the Errata found in the MS-BKRP document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-CAPR]: Central Access Policy Identifier (ID) Retrieval Protocol

This topic lists the Errata found in the MS-CAPR document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-CDP]: Connected Devices Platform Protocol Version 3

This topic lists the Errata found in the MS-CDP document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-CHAP]: Extensible Authentication Protocol Method for Microsoft Challenge Handshake Authentication Protocol (CHAP)

This topic lists the Errata found in the MS-CHAP document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-CFB]: Compound File Binary File Format

This topic lists the Errata found in the MS-CFB document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

March 16, 2018 - Download

[MS-CIFS]: Common Internet File System (CIFS) Protocol

This topic lists the Errata found in the MS-CIFS document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

June 1, 2017 - Download

December 1, 2017 - Download

September 12, 2018 - Download

Errata below are for Protocol Document Version V28.0 – 2018/09/12.

Errata Published*	Description
2020/01/20	In Section 3.3.5.3, Receiving an SMB_COM_CREATE_DIRECTORY Request, the following was changed from: If these conditions are met, the server MUST attempt to create the directory.<241> If directory creation fails, the server MUST provide an error response to the client (see section 2.2.4.1.2 for the list of expected error codes). Otherwise, the server the server MUST increase Server.Statistics.sts0_fopens by 1 and MUST return Success in the Status field. A new Open object MUST be allocated and inserted into Server.Connection.FileOpenTable with the following default values: ● A new FID MUST be created to uniquely identify this Open request in Server.Connection.FileOpenTable. ● Server.Open.TreeConnect MUST be set to the TreeConnect on which the open request was performed, and Server.Open.TreeConnect.OpenCount MUST be incremented by 1. The server MUST register the Open request by invoking the Server Registers a New Open event ([MS-SRVS] section 3.1.6.4) and MUST assign the return value to Server.OpenFileGlobalId. Changed to: If these conditions are met, the server MUST attempt to create the directory.<241> If directory creation fails, the server MUST provide an error response to the client (see section 2.2.4.1.2 for the list of expected error codes). Otherwise, the server MUST return Success in the Status field.
2019/09/02	In Section 2.2.4.13.2, Response, the following error code has been added to the NT status code table: STATUS_LOCK_NOT_GRANTED (0xC0000055) In Section 3.3.1.7, Per Unique Open, the following ADM element has been added:

Errata Published*	Description
	Server.Open.LastFailedLockOffset: A 32-bit signed integer indicating the lock offset specified in SMB_COM_LOCK_BYTE_RANGE request, which the server failed. In Section 3.3.5.15, Receiving an SMB_COM_LOCK_BYTE_RANGE Request, the following has been changed from: In the event of an error, including failure to grant the lock on the byte range, the server MUST send an error response message. If the server cannot immediately grant the lock, the server SHOULD<265> reattempt the lock request for a brief interval, returning an error response with a Status of STATUS_FILE_LOCK_CONFLICT (ERRDOS/ERRlock) to the client if the lock cannot be granted. If the lock is successful, the server MUST construct an SMB_COM_LOCK_BYTE_RANGE Response (section 2.2.4.13.2) message. The response MUST be sent to the client as specified in section 3.3.4.1. An entry for the newly-granted byte-range lock MUST be added to Server.Open.Locks. The type of the lock MUST be exclusive, and the entry MUST be formatted with a 32-bit offset (LOCKING_ANDX_RANGE32). <264> Section 3.3.5.15: Windows-based servers request a byte-range lock from the underlying object store as described in [MS-FSA] section 2.1.5.7, with the following mapping of input elements: ● Open is the Open indicated by the SMB_Parameters.Words.FID field of the request. ● FileOffset is the SMB_Parameters.Words.LockOffsetInBytes field of the request. ● Length is the SMB_Parameters.Words.CountOfBytesToLock field of the request. ● ExclusiveLock – TRUE ● FailImmediately – TRUE ● LockKey is set to ((Open.FID << 16) | Open.PID.PIDLow). The returned Status is copied into the SMB_Header.Status field of the response. Changed to: If the server cannot immediately grant the lock, the server SHOULD<265> reattempt the lock request for a brief interval. In the event of an error, including failure to grant the lock on the byte range, the server MUST send an error response message. If the underlying object store returns STATUS_CANCELLED, the server MUST set SMB_Header.Status field of the response to STATUS_FILE_LOCK_CONFLICT (ERRDOS/ERRlock). For any other error, status returned MUST be copied into SMB_Header.Status field of the response. The server MUST set Server.Open.LastFailedLockOffset to LockOffsetInBytes field of the request. If the lock is successful, the server MUST construct an SMB_COM_LOCK_BYTE_RANGE Response (section 2.2.4.13.2) message. The response MUST be sent to the client as specified in section 3.3.4.1. An entry for the newly-granted byte-range lock MUST be added to Server.Open.Locks. The type of the lock MUST be exclusive, and the entry MUST be formatted with a 32-bit offset (LOCKING_ANDX_RANGE32). The server MUST set Server.Open.LastFailedLockOffset to -1.

Errata Published*	Description
	<264> Section 3.3.5.15: Windows-based servers request a byte-range lock from the underlying object store as described in [MS-FSA] section 2.1.5.7, with the following mapping of input elements: ● Open is the Open indicated by the SMB_Parameters.Words.FID field of the request. ● FileOffset is the SMB_Parameters.Words.LockOffsetInBytes field of the request. ● Length is the SMB_Parameters.Words.CountOfBytesToLock field of the request. ● ExclusiveLock – TRUE ● FailImmediately – FALSE, if Server.Open.LastFailedLockOffset is equal to LockOffsetInBytes field of the request. Otherwise - TRUE LockKey is set to ((Open.FID << 16) | Open.PID.PIDLow).
2018/10/29	In Section 3.2.4.44, Application Requests Querying DFS Referrals, the following has been changed from: An input buffer containing the application-provided REQ_GET_DFS_REFERRAL structure. Changed to: An input buffer containing the application-provided REQ_GET_DFS_REFERRAL structure specifiedin [MS-DFSC] section 2.2.2. In Section 3.4.4.9, A Local Client Application Queries DFS Referrals, the following has been changed from: An input buffer containing the application-provided REQ_GET_DFS_REFERRAL or REQ_GET_DFS_REFERRAL_EX structure. Changed to: An input buffer containing the application-provided structure REQ_GET_DFS_REFERRAL specified in [MS-DFSC] section 2.2.2 or REQ_GET_DFS_REFERRAL_EX specified in [MS-DFSC] section 2.2.3.

*Date format: YYYY/MM/DD

[MS-CMRP]: Failover Cluster: Management API (ClusAPI) Protocol

This topic lists the Errata found in the MS-CMRP document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

June 1, 2017 - Download

March 16, 2018 - Download

September 12, 2018 - Download

Errata below are for Protocol Document Version V36.0 – 2019/03/13.

Errata Published*	Description

2019/08/19	A new section has been added to define the STORAGE_MEDIA_TYPE enumeration.
	Added:
	
	2.2.2.27 STORAGE_MEDIA_TYPE
	
	The STORAGE_MEDIA_TYPE enumeration defines the possible values of media type.
	typedef enum STORAGE_MEDIA_TYPE
	
	`{`
	`  UNKNOWN = 0x00000000,`
	`  DISK = 0x00000003,`
	`  SSD = 0x00000004,`
	`  SCM = 0x00000005`
	`};`
	
	In Section 2.2.3.51, CLUS_PHYSICAL_DISK_INFO, a reference has been added to the new section 2.2.2.27, STORAGE_MEDIA_TYPE, where the STORAGE_MEDIA_TYPE enumeration is defined.
	
	Changed from:
	...
	
	MediaType (4 bytes): A media type enumerated by STORAGE_MEDIA_TYPE.
	
	...
	
	Changed to:

Errata Published*	Description
	... MediaType (4 bytes): A media type enumerated by STORAGE_MEDIA_TYPE, as specified in section 2.2.2.27. ... In Section 3.1.4.2.141, ApiExecuteReadBatch (Opnum 145), the description of the successful completion of the ApiExecuteReadBatch method has been revised to more clearly indicate that CLUSREG_READ_VALUE is a command type, which is defined previously in the section. Changed from: ... If the read operation is successful, a CLUS_REG_READ_VALUE BATCH_UPDATE_COMMAND is returned with its Data and ValueType fields filled out. ... Changed to: ... If the read operation is successful, a BATCH_UPDATE_COMMAND of command type CLUSREG_READ_VALUE is returned with its Data and ValueType fields filled out. ...
2019/08/19	In Section 2.2.3.37, SR_RESOURCE_TYPE_ELIGIBLE_DISKS_RESULT, the missing underscore in the structure name has been added. Changed from: ... The SR_RESOURCE_TYPE_ELIGIBLE DISKS_RESULT structure SHOULD<33> be used to return a list of disks for storage replication. It is a custom-marshalled structure that contains the following fields. ... - Changed to:

Errata Published*	Description
	... The SR_RESOURCE_TYPE_ELIGIBLE_DISKS_RESULT structure SHOULD<33> be used to return a list of disks for storage replication. It is a custom-marshalled structure that contains the following fields. ... In Section 2.2.3.41, SR_RESOURCE_TYPE_REPLICATED_DISKS_RESULT, the description for ReplicatedDisks was revised to include a descriptor. Changed from: ... ReplicatedDisks (variable): An array of SR_RESOURCE_TYPE_REPLICATED_DISKs, each representing a replicated disk in the cluster state. ... Changed to: ... ReplicatedDisks (variable): An array of SR_RESOURCE_TYPE_REPLICATED_DISK elements, each representing a replicated disk in the cluster state. ... In Section 3.1.1.12, Cluster Version, the missing "_CLEAR" in the name of the second section referenced has been added. Changed from: ... Upgrade: Upgrades the cluster to a higher supported version (Protocol Version 3 only). For more information, see CLUSCTL_CLUSTER_UPGRADE_CLUSTER_VERSION (section 3.1.4.3.7.18), CLUSCTL_CLUSTER_UPGRADE_IN_PROGRESS (section 3.1.4.3.7.19), and CLUSCTL_CLUSTER_IS_READY_FOR_UPGRADE (section 3.1.4.3.7.20). ... Changed to: ... Upgrade: Upgrades the cluster to a higher supported version (Protocol Version 3 only). For more information, see CLUSCTL_CLUSTER_UPGRADE_CLUSTER_VERSION (section 3.1.4.3.7.18), CLUSCTL_CLUSTER_CLEAR_UPGRADE_IN_PROGRESS (section 3.1.4.3.7.19), and CLUSCTL_CLUSTER_IS_READY_FOR_UPGRADE (section 3.1.4.3.7.20). ... In Section 7, Appendix B: Product Behavior, the missing underscore in the name of the structure has been added.

Errata Published*	Description
	Changed from: ... <33> Section 2.2.3.37: The SR_RESOURCE_TYPE_ELIGIBLE_DISKS RESULT structure is not implemented in Windows NT 4.0 SP3, Windows NT 4.0 SP4, Windows 2000, Windows XP, Windows Server 2003, Windows Vista SP1, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2. ... Changed to: ... <33> Section 2.2.3.37: The SR_RESOURCE_TYPE_ELIGIBLE_DISKS_RESULT structure is not implemented in Windows NT 4.0 SP3, Windows NT 4.0 SP4, Windows 2000, Windows XP, Windows Server 2003, Windows Vista SP1, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2. ...

*Date format: YYYY/MM/DD

[MS-COMA]: Component Object Model Plus (COMplus) Remote Administration Protocol

This topic lists the Errata found in the MS-COMA document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-CRTD]: Certificate Templates Structure

This topic lists the Errata found in [MS-CRTD] 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 V24.0 – 2018/09/12.

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td>2019/12/16</td><td>In Section 2.26, msPKI-Enrollment-Flag Attribute, added missing 'CT_FLAG_SKIP_AUTO_RENEWAL' flag and description to the enrollment flags table.

Changed from:

<table>
<tr><td>0x00020000
CT_FLAG_ISSUANCE_POLICIES_FROM_REQUEST</td><td>This flag indicates that the certificate issuance policies to be included in the issued certificate come from the request rather than from the template. The template contains a list of all of the issuance policies that the request is allowed to specify; if the request contains policies that are not listed in the template, then the request is rejected. For the processing rules of this flag, see [MS-WCCE] section 3.2.2.6.2.1.4.5.8.<33></td></tr>
</table>

Changed to:

<table>
<tr><td>0x00020000
CT_FLAG_ISSUANCE_POLICIES_FROM_REQUEST</td><td>This flag indicates that the certificate issuance policies to be included in the issued certificate come from the request rather than from the template. The template contains a list of all of the issuance policies that the request is allowed to specify; if the request contains policies that are not listed in the template, then the request is rejected. For the processing rules of this flag, see [MS-WCCE] section 3.2.2.6.2.1.4.5.8.<33></td></tr>
<tr><td>0x00040000
CT_FLAG_SKIP_AUTO_RENEWAL</td><td>This flag indicates that the certificate should not be auto-renewed, although it has a valid template.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>
In Section 2.27, msPKI-Private-Key-Flag Attribute, added missing 'CT_FLAG_HELLO_LOGON_KEY' flag and description to the private key flags table.

Changed from:

<table>
<tr><td>0x00000800 *
CT_FLAG_EK_VALIDATE_KEY</td><td>This flag indicates that attestation based on the hardware key of the TPM is to be performed. For more details, see [MS-WCCE] section 3.2.2.6.2.1.4.5.7.</td></tr>
</table>

Changed to:

<table>
<tr><td>0x00000800 *
CT_FLAG_EK_VALIDATE_KEY</td><td>This flag indicates that attestation based on the hardware key of the TPM is to be performed. For more details, see [MS-WCCE] section 3.2.2.6.2.1.4.5.7.</td></tr>
<tr><td>0x00200000 *
CT_FLAG_HELLO_LOGON_KEY</td><td>This flag indicates that the key is used for Windows Hello logon. For more details, see [MS-WCCE] section 3.2.2.6.2.1.4.5.7.</td></tr>
</table>
</td></tr>
</table>

*Date format: YYYY/MM/DD

[MS-CSRA]: Certificate Services Remote Administration Protocol

This topic lists the Errata found in the MS-CSRA document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

September 12, 2018 - Download

Errata below are for Protocol Document Version V38.0 – 2018/09/12.

Errata Published*	Description
2019/03/18	In this document, changed the default value of the CA for Windows Server 2019. In Section 3.1.4.2.14, ICertAdminD2::GetConfigEntry (Opnum 44), changed from: … 8. For each input in the left column of the table below, the CA MUST perform the processing rules in the corresponding cell in the right column. Input Parameters | Processing rule for pVariant … | … pwszNodePath is EMPTY and pwszEntry is "Version" | The CA MUST return the value of the OnNextRestart_Config_Product_Version ADM element as a VARIANT. The vt member of the VARIANT MUST be set to VT_I4 and the lVal member MUST be set to the one of the following values: 0x00010001 – Server is Windows 2000 Server operating system 0x00020002 – Server is Windows Server 2003 operating system 0x00030001 – Server is Windows Server 2008 operating system 0x00040001 – Server is Windows Server 2008 R2 operating system 0x00050001 – Server is Windows Server 2012 operating system 0x00060001 – Server is Windows Server 2012 R2 operating system <72>

Errata Published*	Description

	0x00070001 – Server is Windows Server 2016 operating system 0x00080001 – Server is Windows Server 2019 operating system
…	…

Changed to:

…

8. For each input in the left column of the table below, the CA MUST perform the processing rules in the corresponding cell in the right column.

Input Parameters	Processing rule for pVariant
…	…
pwszNodePath is EMPTY and pwszEntry is "Version"	The CA MUST return the value of the OnNextRestart_Config_Product_Version ADM element as a VARIANT. The vt member of the VARIANT MUST be set to VT_I4 and the lVal member MUST be set to the one of the following values: 0x00010001 – Server is Windows 2000 Server operating system 0x00020002 – Server is Windows Server 2003 operating system 0x00030001 – Server is Windows Server 2008 operating system 0x00040001 – Server is Windows Server 2008 R2 operating system 0x00050001 – Server is Windows Server 2012 operating system 0x00060001 – Server is Windows Server 2012 R2 operating system <72> 0x00070001 – Server is Windows Server 2016 operating system or Windows Server 2019 operating system
…	…

In Section 7, Appendix B: Product Behavior, changed from:

<11> Section 3.1.1.10: Microsoft CAs persist only a subset of the configuration data. They store the configuration data in the registry in the following locations:

…

Version

ADM Datum: Config_Product_Version and OnNextRestart_Config_Product_Version

Registry Value Type: REG_DWORD

Errata Published*	Description
	Default Value: By default, the value depends on the Windows version: … 0x00080001: Windows Server 2019 Changed to: <11> Section 3.1.1.10: Microsoft CAs persist only a subset of the configuration data. They store the configuration data in the registry in the following locations: … Version ADM Datum: Config_Product_Version and OnNextRestart_Config_Product_Version Registry Value Type: REG_DWORD Default Value: By default, the value depends on the Windows version: … 0x00070001: Windows Server 2016 or Windows Server 2019

*Date format: YYYY/MM/DD

[MS-CSSP]: Credential Security Support Provider (CredSSP) Protocol

This topic lists the Errata found in the MS-CSSP document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

June 1, 2017 - Download

March 16, 2018 - Download

September 12, 2018 - Download

[MS-CSVP]: Failover Cluster: Setup and Validation Protocol (ClusPrep)

This topic lists the Errata found in the MS-CSVP document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

Errata below are for Protocol Document Version V27.0 – 2019/03/13.

Errata Published*	Description

2019/08/05	In Section 3.2.4.4, CprepPrepareNodePhase2 (Opnum 6), more detail has been added to clarify server processing steps for this call.
	
	Changed from:
	...
	
	When processing this call, the server MUST do the following:
	• Determine the number of disks accessible to the system in an implementation-specific way.
	• For each disk:
	  • Create a ClusPrepDisk object.
	  • Initialize ClusPrepDisk.AttachedState to Not Attached.
	  • Initialize ClusPrepDisk.OnlineState to Not Online.
	  • Initialize ClusPrepDisk.OwnedState to Not Owned.
	  • Add the disk to ClusPrepDiskList.
	• Set pulNumDisks to that number.
	• Set the server Prepare State to Online.
	The server returns the following information to the client:
	• The number of disks attached to the system
	
	Changed to:
	...
	
	When processing this call, the server MUST do the following:
	• Determine the number of disks accessible to the system in an implementation-specific way.
	• If the Flags field includes ForceOfflineNonClusteredDisks but does not include SkipNonClusteredPools, detach spaces using non-clustered pools before including them in disks eligible for validation.
	• If the Flags field includes SkipNonClusteredPools, skip non-clustered pools for validation.
	• If the Flags field includes neither ForceOfflineNonClusteredDisks nor SkipNonClusteredPools, skip non-clustered pools with attached spaces for validation.

Errata Published*	Description

	• For each disk:
	  • Create a ClusPrepDisk object.
	  • Initialize ClusPrepDisk.AttachedState to Not Attached.
	  • Initialize ClusPrepDisk.OnlineState to Not Online.
	  • Initialize ClusPrepDisk.OwnedState to Not Owned.
	  • Add the disk to ClusPrepDiskList.
	• Set pulNumDisks to the number of disks in ClusPrepDiskList.
	• Set the server Prepare State to Online.
	The server returns the following information to the client:
	• pulNumDisks

*Date format: YYYY/MM/DD

[MS-DCOM]: Distributed Component Object Model (DCOM) Remote Protocol

This topic lists the Errata found in the MS-DCOM document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-DFSC]: Distributed File System (DFS) Referral Protocol

This topic lists the Errata found in [MS-DFSC] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

September 12, 2018 - Download

Errata below are for Protocol Document Version V28.0 – 2018/09/12.

Errata Published*	Description
2018/10/29	In Section 3.1.4.2, Sending a DFS Referral Request to the Server, the following has been changed from: The client MUST query the DFS referral, as specified in [MS-CIFS] section 3.4.4.9, by passing ClientGenericContext, HostName, UserCredentials, MaxOutputSize, the REQ_GET_DFS_REFERRAL_EX or REQ_GET_DFS_REFERRAL structure as the input buffer, and the FSCTL code set to FSCTL_DFS_GET_REFERRALS or FSCTL_DFS_GET_REFERRALS_EX based on the input buffer. Changed to: The client MUST query the DFS referral, as specified in [MS-CIFS] section 3.4.4.9, by passing ClientGenericContext, HostName, UserCredentials, MaxOutputSize, the REQ_GET_DFS_REFERRAL_EX or REQ_GET_DFS_REFERRAL structure as the input buffer, and the FSCTL code set to FSCTL_DFS_GET_REFERRALS, if the input buffer is an REQ_GET_DFS_REFERRAL, or FSCTL_DFS_GET_REFERRALS_EX, if the input buffer is an REQ_GET_DFS_REFERRAL_EX.

*Date format: YYYY/MM/DD

[MS-DHCPE]: Dynamic Host Configuration Protocol (DHCP) Extensions

This topic lists the Errata found in [MS-DHCPE] 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 V24.0 – 2018/09/12.

Errata Published*	Description
2019/12/16	In Section 3.1.5.2, Receiving a DHCPACK, added alternate processing for none or two route options. Changed from: If it contains a Microsoft Classless Static Route Option, the client MUST first check whether the option conforms to the syntax specified in section 2.2.8. If any of the parameters in this DHCPv4 option are invalid or incomplete, the DHCPv4 client MUST silently discard the complete DHCPv4 message and start the initialization process again. Otherwise, the specified routes MUST be inserted into the routing table in the TCP/IP stack. Changed to: If it contains a Microsoft Classless Static Route Option, the client MUST first check whether the option conforms to the syntax specified in section 2.2.8. If any of the parameters in this DHCPv4 option are invalid or incomplete, the DHCPv4 client MUST silently discard the complete DHCPv4 message and start the initialization process again. Otherwise, if the DHCPACK does not contain a Classless Static Route Option (121), the specified routes MUST be inserted into the routing table in the TCP/IP stack. If it contains both a Microsoft Classless Static Route Option (249) and a Classless Static Route Option (121) then the client MUST select either (in any implementation-specific way[27]) set of routes as the routes to be added into the routing table in the TCP/IP stack. <27> Section 3.1.5.2: All versions of Windows Vista and Windows Server 2008 and later will insert the last option in the message.

*Date format: YYYY/MM/DD

[MS-DHCPM]: Microsoft Dynamic Host Configuration Protocol (DHCP) Server Management Protocol

This topic lists the Errata found in [MS-DHCPM] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

September 15, 2017 - Download

[MS-DNSP]: Domain Name Service (DNS) Server Management Protocol

This topic lists the Errata found in the MS-DNSP document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

September 12, 2018 - Download

Errata below are for Protocol Document Version V35.0 – 2019/03/13.

Errata Published*	Description
2019/04/29	In Section 2.2.15.1.1.6, DNS_RPC_CRITERIA_ENUM, a product behavior note has been updated to indicate a change in the product versions that support the DnsPolicyCriteriaEDNSSubnet policy. Changed from: DnsPolicyCriteriaEDNSSubnet: Usage of this enum constant will fail the request with Win32 Error-9974 (DNS_ERROR_POLICY_INVALID_SETTINGS).<90> <90> Section 2.2.15.1.1.6: DnsPolicyCriteriaEDNSSubnet is not implemented in Windows Server v1809 operating system and earlier. Changed to: DnsPolicyCriteriaEDNSSubnet: Usage of this enum constant will fail the request with Win32 Error-9974 (DNS_ERROR_POLICY_INVALID_SETTINGS).<90> <90> Section 2.2.15.1.1.6: DnsPolicyCriteriaEDNSSubnet is implemented in Windows Server v1809 operating system with [MSKB-4497934] and later. Also, in Section 1.2.1, Normative References, the following reference has been added: [MSKB-4497934] Microsoft Corporation, "May 20, 2019 - KB4497934", https://support.microsoft.com/en-us/help/4497934

*Date format: YYYY/MM/DD

[MS-DPWSSN]: Devices Profile for Web Services (DPWS) Size Negotiation Extension

This topic lists the Errata found in [MS-DPWSSN] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

June 1, 2017 - Download

[MS-DRSR]: Directory Replication Service (DRS) Remote Protocol

This topic lists the Errata found in the MS-DRSR document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

September 15, 2017 - Download

December 1, 2017 - Download

Errata below are for Protocol Document Version V40.0 – 2019/09/12.

Errata Published*	Description

2019/10/16	In Section 4.1.1.2.3, CreateNtdsData, the pseudocode for creating an nTDSDSA object has been updated by including a check if attributes meet correct order for creating the NtdsDsa object, and if not, setting the ERROR_DS_NO_CROSSREF_FOR_NC error and returning 'false'.
	
	Changed from:
	
	.. if not accessAllowed then
	SetErrorData(SV_PROBLEM_DIR_ERROR, serviceError,
	ERROR_ACCESS_DENIED, pmsgOut, ver)
	return false
	endif
	
	/* Check for the functional level compliance. The functional level.."
	
	Changed to:
	.. if not accessAllowed then
	SetErrorData(SV_PROBLEM_DIR_ERROR, serviceError,
	ERROR_ACCESS_DENIED, pmsgOut, ver)
	return false
	endif
	
	correctOrder := DoAttributesSatisfyPreCheckForCreateNtdsDsa (entList)
	
	if not correctOrder then

Errata Published*	Description
	SetErrorData(SV_PROBLEM_DIR_ERROR, serviceError, ERROR_DS_NO_CROSSREF_FOR_NC, pmsgOut, ver) return false endif /* Check for the functional level compliance. The functional level.." Also, in Section 4.1.1.2.11, DoAttributesSatisfyPreCheckForCreateNtdsDsa, new content has been added to describe the new procedure above added in section 4.1.1.2.3.

*Date format: YYYY/MM/DD

[MS-DTCO]: MSDTC Connection Manager: OleTx Transaction Protocol

This topic lists the Errata found in the MS-DTCO document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

December 1, 2017 - Download

[MS-DSCPM]: Desired State Configuration Pull Model Protocol

This topic lists the Errata found in the MS-DSCPM document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

December 1, 2017 - Download

September 12, 2018 - Download

[MS-DTYP]: Windows Data Types

This topic lists the Errata found in the MS-DTYP document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

September 15, 2017 - Download

December 1, 2017 - Download

September 12, 2018 - Download

Errata below are for Protocol Document Version V35.0 – 2018/09/12

Errata Published*	Description
2020/03/02	In Section 2.5.3.2, Access Check Algorithms Pseudocode, the pseudocode confirming that the object owner is always granted READ_CONTROL and WRITE_DAC has been corrected as follows: Changed from: Set GrantedAccess to GrantedAccess or READ_CONTROL or WRITE_OWNER Changed to: Set GrantedAccess to GrantedAccess or READ_CONTROL or WRITE_DAC
2019/11/11	In Section 2.4.2.4, Well-Known SID Structures, the description of the table entry for AUTHENTICATED_USERS has been updated for clarity, and an associated behavior note added: Changed from: A group that includes all users whose identities were authenticated when they logged on. Changed to: A group that includes all users whose identities were authenticated when they logged on. Users authenticated as Guest or Anonymous are not members of this group.<11>

Errata Published*	Description
	<11> Windows server versions earlier than Windows Server 2003 and client versions earlier than Windows XP SP2 included the Guest account in the Authenticated Users group.

*Date format: YYYY/MM/DD

[MS-DVRD]: Device Registration Discovery Protocol

This topic lists the Errata found in [MS-DVRD] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-DVRE]: Device Registration Enrollment Protocol

This topic lists the Errata found in the MS-DVRE document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-DVRJ]: Device Registration Join Protocol

This topic lists the Errata found in the MS-DVRJ document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-ECS]: Enterprise Client Synchronization Protocol

This topic lists the Errata found in the MS-ECS document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

Errata below are for Protocol Document Version V9.0 – 2018/09/12.

Errata Published*	Description
2020/03/02	In Section 3.6.5.1, Upload Scenario, the following was changed from: For each UploadFile in UploadFileList, the client MUST update UploadFile.CommitStatus to the Status entry returned in the FILE_STATUS structure. Changed to: For each UploadFile in UploadFileList, the client MUST update UploadFile.CommitStatus to the Status entry returned in the FILE_STATUS_ENTRY structure.
2019/09/30	In this document, numerous editorial fixes have been made, e.g., changed instances of "ID" to "Id" or instances of "Id" to "ID"; changed instances of "FileMetaDataTable" to "FileMetadataTable"; and removed whitespaces. Sections updated: 2.2.1 2.2.1.5 2.2.2.5 2.2.2.17 2.2.2.18 3.2.5.1.1 3.2.5.1.1.3 3.3.5.1.1 3.3.5.1.1.2 3.3.5.1.1.3 3.4.5.1.1 3.4.5.1.1.1 3.4.5.1.1.3 3.4.5.2.1

Errata Published*	Description
	3.4.5.2.2 3.4.5.2.2.3 3.4.5.3.1 3.4.5.3.1.3 3.4.5.4.1 3.4.5.4.1.3 3.4.5.5.1 3.4.5.6.1 3.5.5.1.1 3.5.5.2.1 3.6.1.1 3.6.3 3.6.5.1 3.6.5.2 4.1 For details on the above changes, see the PDF doc here.
2019/09/16	In Section 3.4.5.3.1.3, Processing Details, the following was changed from: Otherwise, if FileMetadata.RemoteStreamId is not equal to StreamId, and FileSize is greater than the space available for a user, the server MUST set ProtocolType to 0x00 and MUST set PrepareResult to ERROR_DISK_FULL, as specified in [MS-ERREF] section 2.1.1. Changed to: Otherwise, if FileMetadata.FileStreamId is not equal to FILE_INFO_INPUT_ENTRY.StreamId, and FILE_INFO_INPUT_ENTRY.FileSize is greater than the space available for a user, the server MUST set ProtocolType to 0x00 and MUST set PrepareResult to ERROR_DISK_FULL, as specified in [MS-ERREF] section 2.1.1.

*Date format: YYYY/MM/DD

[MS-EFSR]: Encrypting File System Remote (EFSRPC) Protocol

This topic lists the Errata found in the MS-EFSR document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-EMF]: Enhanced Metafile Format

This topic lists the Errata found in the MS-EMF document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

June 1, 2017 - Download

[MS-EMFPLUS]: Enhanced Metafile Format Plus Extensions

This topic lists the Errata found in the MS-EMFPLUS document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

Errata below are for Protocol Document Version V16.0 – 2018/09/12.

Errata Published*	Description
2019/12/09	In Section 2.3.6.6, EmfPlusSetRenderingOrigin Record, changed from: x (4 bytes): An unsigned integer that defines the horizontal coordinate value of the rendering origin. y (4 bytes): An unsigned integer that defines the vertical coordinate value of the rendering origin. Changed to: x (4 bytes): A signed integer that defines the horizontal coordinate value of the rendering origin. y (4 bytes): A signed integer that defines the vertical coordinate value of the rendering origin.
2018/12/10	In this document several sections have been modified to reference [MS-LCID], the Windows Language Code Identifier (LCID) Reference. In Section 1.6, Versioning and Localization, changed from: Localization: EMF+ structures contain the following locale-specific data: • Language identifiers that correspond to natural languages in locales, including countries, geographical regions, and administrative districts. For details, see the LanguageIdentifier enumeration. Changed to: Localization: EMF+ structures contain the following locale-specific data: • Language identifiers that correspond to natural languages in locales, including countries, geographical regions, and administrative districts. For details, see [MS-LCID] section 2.1. In Section 2.1.1, Enumeration Constant Types, changed from:

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Name</th><th>Section</th><th>Description</th></tr>
<tr><td>LineCapTypeLanguageIdentifier</td><td>2.1.1.17</td><td>Defines identifiers for natural languages in locales, including countries, geographical regions, and administrative districts. Defines types of line caps to use at the ends of lines that are drawn with graphics pens.</td></tr>
</table>

Changed to:

<table>
<tr><th>Name</th><th>Section</th><th>Description</th></tr>
<tr><td>LineCapType</td><td>2.1.1.17</td><td>Defines types of line caps to use at the ends of lines that are drawn with graphics pens.</td></tr>
</table>

In Section 2.1.1.17, LanguageIdentifier Enumeration, the section title and introduction have been changed.

Changed from:
2.1.1.17 LanguageIdentifier Enumeration
The LanguageIdentifier enumeration defines identifiers for natural languages in locales, including countries, geographical regions, and administrative districts.

Changed to:
2.1.1.17 LineCapType Enumeration
The LineCapType enumeration defines types of line caps to use at the ends of lines that are drawn with graphics pens.

In Section 2.2.2.23, EmfPlusLanguageIdentifier Object, changed from:
…
The encoded language identifier values are defined in the LanguageIdentifier enumeration.
…

Changed to:
…
The encoded LCID values are defined in [MS-LCID] section 2.2.
…

Section 2.1.3.2, Language Identifiers, has been removed.
</td></tr>
<tr><td>2018/11/26</td><td>In Section 2.1.1, Enumeration Constant Types, the “WrapMode” enumeration has</td></tr>
</table>

Errata Published*	Description
	been added to the list of defined enumerations. Added: <table><tr><th>Name</th><th>Section</th><th>Description</th></tr><tr><td>…</td><td>…</td><td>…</td></tr><tr><td>WrapMode</td><td>2.1.1.34</td><td>Defines how the pattern from a texture or gradient brush is tiled across a shape or at shape boundaries.</td></tr></table> In Section 2.1.2, Bit Flag Constant Types, the “PathPointType” enumeration has been added to the list of defined flags. Added: <table><tr><th>Name</th><th>Section</th><th>Description</th></tr><tr><td>…</td><td>…</td><td>…</td></tr><tr><td>PathPointType</td><td>2.1.2.6</td><td>Specifies the type properties of points on graphics paths.</td></tr><tr><td>…</td><td>…</td><td>…</td></tr></table> In Section 2.3.8.1, EmfPlusSetTSClip, the name of the “Rects” field has been changed to “rects” throughout the section. For example, changed from: rects (variable): An array of NumRects rectangles that define clipping areas. The format of this data is determined by the C bit in the Flags field. The compression scheme for data in this record uses the following algorithm. Each point of each rectangle is encoded in either a single byte or 2 bytes. If the point is encoded in a single byte, the high bit (0x80) of the byte MUST be set, and the value is a signed number represented by the lower 7 bits. If the high bit is not set, then the value is encoded in 2 bytes, with the high-order byte encoded in the 7 lower bits of the first byte, and the low-order byte value encoded in the second byte. Each point is encoded as the difference between the point in the current rect and the point in the previous rect. The bottom point of the rect is encoded as the difference between the bottom coordinate and the top coordinate on the current rect. See section 2.3.8 for the specification of additional terminal server record types. Changed to: Rects (variable): An array of NumRects rectangles that define clipping areas. The format of this data is determined by the C bit in the Flags field. The compression scheme for data in this record uses the following algorithm. Each point of each rectangle is encoded in either a single byte or 2 bytes. If the point is encoded in a single byte, the high bit (0x80) of the byte MUST be set, and the value is a signed number represented by the lower 7 bits. If the high bit is not set, then the value is encoded in 2 bytes, with the high-order byte encoded in the 7 lower bits

Errata Published*	Description
	of the first byte, and the low-order byte value encoded in the second byte. Each point is encoded as the difference between the point in the current rectangle and the point in the previous rectangle. The bottom point of the rectangle is encoded as the difference between the bottom coordinate and the top coordinate on the current rectangle. See section 2.3.8 for the specification of additional terminal server record types.

*Date format: YYYY/MM/DD

[MS-ERREF]: Windows Error Codes

This topic lists the Errata found in the MS-ERREF document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

June 1, 2017 - Download

September 15, 2017 - Download

Errata below are for Protocol Document Version V19.0 – 2018/09/12.

Errata Published*	Description
2019/08/05	In the Section 1.1, Glossary, the entry for the term message identifier, which is at odds with the definition in Section 2.2, has been removed.

*Date format: YYYY/MM/DD

[MS-EVEN]: EventLog Remoting Protocol

This topic lists the Errata found in the MS-EVEN document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

Errata below are for Protocol Document Version V22.0 – 2018/09/12.

Errata Published*	Description
2019/09/02	In Section 2.2.6, Handles, the section name has been changed to reflect the name of the type it describes. Changed from: 2.2.6 Handles Changed to: 2.2.6 IELF_HANDLE In Section 3.1.4.7, ElfrReadELW (Opnum 10), the name of the EVENTLOG_BACKWARDS_READ flag contained a misspelling in one place. Changed from: … If neither of the two flags are set, the server will treat it as if the EVENTLOG_BACKWORDS_READ flag is set. … Changed to: … If neither of the two flags are set, the server will treat it as if the EVENTLOG_BACKWARDS_READ flag is set. …

*Date format: YYYY/MM/DD

[MS-EVEN6]: EventLog Remoting Protocol Version 6.0

This topic lists the Errata found in the MS-EVEN6 document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 12, 2018 - Download

[MS-FASP]: Firewall and Advanced Security Protocol

This topic lists the Errata found in the MS-FASP document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

March 13, 2019 - Download

[MS-FAX]: Fax Server and Client Remote Protocol

This topic lists the Errata found in the MS-FAX document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 12, 2018 - Download

[MS-FRS2]: Distributed File System Replication Protocol

This topic lists the Errata found in the MS-FRS2 document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

Errata below are for Protocol Document Version V28.0 – 2018/09/12.

Errata Published*	Description
2019/02/19	In Section 1.2.1, Normative References, the following reference has been added: [MS-XCA] Microsoft Corporation, "Xpress Compression Algorithm". In Section 2.2.1.4.15, XPRESS Block, the Block Data field has been changed from: If the value of the Block Compressed Size field is less than the value of the Block Uncompressed Size field, then the data has been compressed. For more information about decompressing compressed data, see section 3.1.1.1.3.9. Changed to: If the value of the Block Compressed Size field is less than the value of the Block Uncompressed Size field, then the data has been compressed. For more information about decompressing compressed data, see section 3.1.1.2. In Section 3.1.1.1, Compression, the following was changed from: Many of the FrsTransport methods use compression to reduce the amount of data that is returned to the client. This section describes algorithms and a conceptual model of possible data organization that an implementation maintains in order to decompress compressed data. The described organization is provided to facilitate the explanation of how the algorithm behaves. Error checking and handling has been omitted from all algorithms in the interests of clarity. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with what is described in this document. Changed to: Many of the FrsTransport methods use the LZ77+Huffman Compression algorithm, specified in [MS-XCA] section 2.1, to compress data. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with what is described in this document.

Errata Published*	Description
	The following sections have been removed and replaced with links to MS-XCA: 3.1.1.1.1 Pseudocode Conventions 3.1.1.1.2 Data Structures 3.1.1.1.2.1 PREFIX_CODE_NODE 3.1.1.1.2.2 PREFIX_CODE_SYMBOL 3.1.1.1.2.3 BITSTRING 3.1.1.1.3 Procedures 3.1.1.1.3.1 PrefixCodeTreeRebuild 3.1.1.1.3.2 PrefixCodeTreeAddLeaf 3.1.1.1.3.3 SortSymbols 3.1.1.1.3.4 CompareSymbols 3.1.1.1.3.5 BitstringInit 3.1.1.1.3.6 BitstringLookup 3.1.1.1.3.7 BitstreamSkip 3.1.1.1.3.8 PrefixCodeTreeDecodeSymbol A new section, 3.1.1.2, Decompression, has been added: FrsTransport methods that compress data will always return information specifying the size of the original data. It is the caller's responsibility to determine whether the returned data is compressed. If the size of the compressed data buffer that is returned by the server in bytes is equal to the size in bytes of the original uncompressed data, then the buffer returned by the server contains uncompressed data. In Section 3.2.4.1.7, RequestRecords (Opnum 6), the description of the compressedRecords field has been changed from: compressedRecords: The data records, compressed using the DFS-R compression algorithm specified in section 3.1.1.1. The compressedRecords bytes correspond to an array of FRS_ID_GVSN entries. DFS-R uses custom marshaling in this RPC call to compress the set of transmitted records. The size of the FRS_ID_GVSN array is given by the numRecords parameter. The decompression algorithm specified in section 3.1.1.1.3.9 can be used to decompress the received data into a buffer of sizeof(FRS_ID_GVSN)*numRecords bytes, which can be re-interpreted as an array of FRS_ID_GVSN entries. Changed to: compressedRecords: The data records, compressed using the algorithm specified in section 3.1.1.1. The compressedRecords bytes correspond to an array of FRS_ID_GVSN entries. DFS-R uses custom marshaling in this RPC call to compress the set of transmitted records. The size of the FRS_ID_GVSN array is given by the numRecords parameter. The decompression algorithm specified in section 3.1.1.1 can be used to decompress the received data into a buffer of sizeof(FRS_ID_GVSN)*numRecords bytes, which can be re-interpreted as an array of FRS_ID_GVSN entries. In Section 3.2.4.1.14, InitializeFileTransferAsync (Opnum 13), changed from: 2. An encapsulation of the marshaled file data stream using the compressed data

Errata Published*	Description
	format (as specified in section 3.2.4.1.14.2) generated by the DFS-R compression algorithm specified in section 3.1.1.1. Even if the marshaled file data stream is not compressed by the server, it is still encapsulated using the compressed data format. Changed to: 2. An encapsulation of the marshaled file data stream using the compressed data format (as specified in section 3.2.4.1.14.2) generated by the compression algorithm specified in section 3.1.1.1. Even if the marshaled file data stream is not compressed by the server, it is still encapsulated using the compressed data format.

*Date format: YYYY/MM/DD

[MS-FSA]: File System Algorithms

This topic lists the Errata found in the MS-FSA document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

March 2, 2016 - Download

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

September 15, 2017 - Download

December 1, 2017 - Download

September 12, 2018 - Download

June 24, 2019 – Download

Errata below are for Protocol Document Version V29.0 – 2019/05/30.

Errata Published*	Description
2019/12/16	In Section 2.1.5.1.1, Creation of a New File, described when to initialize UsnReason to USN_REASON_FILE_CREATE and when to set UsnReason to USN_REASON_OBJECT_ID_CHANGE and USN_REASON_STREAM_CHANGE. Also, clarified how the object store posts a USN change. Changed from: Pseudocode for the operation is as follows: • If StreamTypeToOpen is DirectoryStream and DesiredFileAttributes.FILE_ATTRIBUTE_TEMPORARY is set, the operation MUST be failed with STATUS_INVALID_PARAMETER. • If DesiredFileAttributes.FILE_ATTRIBUTE_READONLY and CreateOptions.FILE_DELETE_ON_CLOSE are both set, the operation MUST be failed with STATUS_CANNOT_DELETE. • If Open.RemainingDesiredAccess.ACCESS_SYSTEM_SECURITY is set and Open.GrantedAccess.ACCESS_SYSTEM_SECURITY is not set and SecurityContext.PrivilegeSet does not contain "SeSecurityPrivilege", the operation MUST be failed with STATUS_ACCESS_DENIED. • If StreamTypeToOpen is DataStream and Open.GrantedAccess.FILE_ADD_FILE is not set and AccessCheck(SecurityContext,Open.Link.ParentFile.SecurityDescriptor, FILE_ADD_FILE) returns FALSE and Open.HasRestoreAccess is FALSE, the operation MUST be failed with STATUS_ACCESS_DENIED. • If StreamTypeToOpen is DirectoryStream and Open.GrantedAccess.FILE_ADD_SUBDIRECTORY is not set and AccessCheck(SecurityContext,Open.Link.ParentFile.SecurityDescriptor, FILE_ADD_SUBDIRECTORY) returns FALSE and Open.HasRestoreAccess is FALSE, the operation

Errata Published*	Description
	MUST be failed with STATUS_ACCESS_DENIED. ● If the object store implements encryption and DesiredFileAttributes.FILE_ATTRIBUTE_ENCRYPTED is TRUE: ● If UserCertificate is empty, the operation MUST be failed with STATUS_CS_ENCRYPTION_NEW_ENCRYPTED_FILE. ● EndIf ● The object store MUST build a new File object with fields initialized as follows:... ● If TunnelCacheEntry.ObjectIdInfo.ObjectId is not empty: ● If TunnelCacheEntry.ObjectIdInfo.ObjectId is not unique on File.Volume: ● The object store MUST construct a FILE_OBJECTID_INFORMATION structure (as specified in [MS-FSCC] section 2.4.28.1) ObjectIdInfo as follows: ● ObjectIdInfo.FileReference set to File.FileId64. ● ObjectIdInfo.ObjectId set to TunnelCacheEntry.ObjectIdInfo.ObjectId. ● ObjectIdInfo.BirthVolumeId set to TunnelCacheEntry.ObjectIdInfo.BirthVolumeId. ● ObjectIdInfo.BirthObjectId set to TunnelCacheEntry.ObjectIdInfo.BirthObjectId. ● ObjectIdInfo.DomainId set to TunnelCacheEntry.ObjectIdInfo.DomainId. ● Send directory change notification as specified in section 2.1.4.1, with Volume equal to File.Volume, Action equal to FILE_ACTION_ID_NOT_TUNNELLED, FilterMatch equal to FILE_NOTIFY_CHANGE_FILE_NAME, FileName equal to "\$Extend\$ObjId", NotifyData equal to ObjectIdInfo, and NotifyDataLength equal to sizeof(FILE_OBJECTID_INFORMATION). ● Else: ● Set File.ObjectId to TunnelCacheEntry.ObjectIdInfo.ObjectId. ● Set File.BirthVolumeId to TunnelCacheEntry.ObjectIdInfo.BirthVolumeId. ● Set File.BirthObjectId to TunnelCacheEntry.ObjectIdInfo.BirthObjectId. ● Set File.DomainId to TunnelCacheEntry.ObjectIdInfo.DomainId. ● EndIf... ● If StreamTypeToOpen is DataStream, then the object store MUST create a new data stream for the file as follows:<53> ● Build a new Stream object with all fields initially set to zero. ● Set Stream.StreamType to DataStream. ● Set Stream.Name to StreamNameToOpen. ● Set Stream.File to File. ● Add Stream to File.StreamList. ● Set Open.Stream to Stream. ● Else the object store MUST create a new directory stream as follows: ● Build a new Stream object with all fields initially set to zero. ● Set Stream.StreamType to DirectoryStream.

Errata Published*	Description
	● Set Stream.File to File. ● Add Stream to File.StreamList. ● Set Open.Stream to Stream. ● EndIf ● If the object store implements encryption and File.FileAttributes.FILE_ATTRIBUTE_ENCRYPTED is TRUE: ● If File.FileType is DataFile, set Stream.IsEncrypted to TRUE. ● EndIf ● The object store MUST update the duplicated information as specified in section 2.1.4.18 with Link equal to Link. ● The object store MUST set Open.File to File. ● The object store MUST set Open.Link to Link. ● The object store MUST insert Link into File.LinkList. ● The object store MUST insert Link into Link.ParentFile.DirectoryList. ● The object store MUST update Link.ParentFile.LastModificationTime, Link.ParentFile.LastChangeTime, and Link.ParentFile.LastAccessTime to the current system time. ● If the Oplock member of the DirectoryStream in Link.ParentFile.StreamList (hereinafter referred to as ParentOplock) is not empty, the object store MUST check for an oplock break on the parent according to the algorithm in section 2.1.4.12, with input values as follows:…. Changed to: Pseudocode for the operation is as follows: ● If StreamTypeToOpen is DirectoryStream and DesiredFileAttributes.FILE_ATTRIBUTE_TEMPORARY is set, the operation MUST be failed with STATUS_INVALID_PARAMETER. ● If DesiredFileAttributes.FILE_ATTRIBUTE_READONLY and CreateOptions.FILE_DELETE_ON_CLOSE are both set, the operation MUST be failed with STATUS_CANNOT_DELETE. ● If Open.RemainingDesiredAccess.ACCESS_SYSTEM_SECURITY is set and Open.GrantedAccess.ACCESS_SYSTEM_SECURITY is not set and SecurityContext.PrivilegeSet does not contain "SeSecurityPrivilege", the operation MUST be failed with STATUS_ACCESS_DENIED. ● If StreamTypeToOpen is DataStream and Open.GrantedAccess.FILE_ADD_FILE is not set and AccessCheck(SecurityContext,Open.Link.ParentFile.SecurityDescriptor, FILE_ADD_FILE) returns FALSE and Open.HasRestoreAccess is FALSE, the operation MUST be failed with STATUS_ACCESS_DENIED. ● If StreamTypeToOpen is DirectoryStream and Open.GrantedAccess.FILE_ADD_SUBDIRECTORY is not set and AccessCheck(SecurityContext,Open.Link.ParentFile.SecurityDescriptor, FILE_ADD_SUBDIRECTORY) returns FALSE and Open.HasRestoreAccess is FALSE, the operation MUST be failed with STATUS_ACCESS_DENIED. ● If the object store implements encryption and DesiredFileAttributes.FILE_ATTRIBUTE_ENCRYPTED is TRUE: ● If UserCertificate is empty, the operation MUST be failed with STATUS_CS_ENCRYPTION_NEW_ENCRYPTED_FILE. ● EndIf ● Initialize UsnReason to zero. ● Set UsnReason.USN_REASON_FILE_CREATE to TRUE. ● The object store MUST build a new File object with fields initialized as follows:... ● If TunnelCacheEntry.ObjectIdInfo.ObjectId is not empty: ● If TunnelCacheEntry.ObjectIdInfo.ObjectId is not unique on

Errata Published*	Description
	File.Volume: ● The object store MUST construct a FILE_OBJECTID_INFORMATION structure (as specified in [MS-FSCC] section 2.4.28.1) ObjectIdInfo as follows: ● ObjectIdInfo.FileReference set to File.FileId64. ● ObjectIdInfo.ObjectId set to TunnelCacheEntry.ObjectIdInfo.ObjectId. ● ObjectIdInfo.BirthVolumeId set to TunnelCacheEntry.ObjectIdInfo.BirthVolumeId. ● ObjectIdInfo.BirthObjectId set to TunnelCacheEntry.ObjectIdInfo.BirthObjectId. ● ObjectIdInfo.DomainId set to TunnelCacheEntry.ObjectIdInfo.DomainId. ● Send directory change notification as specified in section 2.1.4.1, with Volume equal to File.Volume, Action equal to FILE_ACTION_ID_NOT_TUNNELLED, FilterMatch equal to FILE_NOTIFY_CHANGE_FILE_NAME, FileName equal to "\$Extend\$ObjId", NotifyData equal to ObjectIdInfo, and NotifyDataLength equal to sizeof(FILE_OBJECTID_INFORMATION). ● Else: ● Set File.ObjectId to TunnelCacheEntry.ObjectIdInfo.ObjectId. ● Set File.BirthVolumeId to TunnelCacheEntry.ObjectIdInfo.BirthVolumeId. ● Set File.BirthObjectId to TunnelCacheEntry.ObjectIdInfo.BirthObjectId. ● Set File.DomainId to TunnelCacheEntry.ObjectIdInfo.DomainId. ● Set UsnReason.USN_REASON_OBJECT_ID_CHANGE to TRUE. ● EndIf... ● If StreamTypeToOpen is DataStream, then the object store MUST create a new data stream for the file as follows:<53> ● Build a new Stream object with all fields initially set to zero. ● Set Stream.StreamType to DataStream. ● Set Stream.Name to StreamNameToOpen. ● Set Stream.File to File. ● Add Stream to File.StreamList. ● Set Open.Stream to Stream. ● If Stream.Name is not empty, set UsnReason.USN_REASON_STREAM_CHANGE to TRUE. ● Else the object store MUST create a new directory stream as follows: ● Build a new Stream object with all fields initially set to zero. ● Set Stream.StreamType to DirectoryStream. ● Set Stream.File to File. ● Add Stream to File.StreamList. ● Set Open.Stream to Stream. ● EndIf ● If the object store implements encryption and File.FileAttributes.FILE_ATTRIBUTE_ENCRYPTED is TRUE: ● If File.FileType is DataFile, set Stream.IsEncrypted to TRUE.

Errata Published*	Description
	● EndIf ● The object store MUST update the duplicated information as specified in section 2.1.4.18 with Link equal to Link. ● The object store MUST set Open.File to File. ● The object store MUST set Open.Link to Link. ● The object store MUST insert Link into File.LinkList. ● The object store MUST insert Link into Link.ParentFile.DirectoryList. ● The object store MUST post a USN change as specified in section 2.1.4.11 with File equal to File, Reason equal to UsnReason, and FileName equal to Link.Name. ● The object store MUST update Link.ParentFile.LastModificationTime, Link.ParentFile.LastChangeTime, and Link.ParentFile.LastAccessTime to the current system time. ● If the Oplock member of the DirectoryStream in Link.ParentFile.StreamList (hereinafter referred to as ParentOplock) is not empty, the object store MUST check for an oplock break on the parent according to the algorithm in section 2.1.4.12, with input values as follows:...

2019/12/16	In Section 2.1.4.17, Algorithm for Noting That a File Has Been Modified, added product behavior note <42> with information about when file systems choose to defer processing for a file that has been modified to a later time.
	
	Changed from:
	
	The inputs for this algorithm are as follows:
	● Open: The Open through which the file was modified.
	The pseudocode for the algorithm is as follows:
	● If Open.UserSetModificationTime is FALSE, set Open.File.LastModificationTime to the current system time.
	● If Open.UserSetChangeTime is FALSE, set Open.File.LastChangeTime to the current system time.
	● If Open.UserSetAccessTime is FALSE, set Open.File.LastAccessTime to the current system time.
	● Set Open.File.FileAttributes.FILE_ATTRIBUTE_ARCHIVE to TRUE.
	
	Changed to:
	
	The inputs for this algorithm are as follows:
	● Open: The Open through which the file was modified.
	The pseudocode for the algorithm is as follows:
	● The object store SHOULD<42>:
	  ● If Open.UserSetModificationTime is FALSE, set Open.File.LastModificationTime to the current system time.
	  ● If Open.UserSetChangeTime is FALSE, set Open.File.LastChangeTime to the current system time.
	  ● If Open.UserSetAccessTime is FALSE, set Open.File.LastAccessTime to the current system time.
	  ● Set Open.File.FileAttributes.FILE_ATTRIBUTE_ARCHIVE to TRUE.
	
	<42> Section 2.1.4.17: File systems may choose to defer processing for a file that has been modified to a later time, favoring performance over accuracy. The NTFS file system on versions prior to Windows 10 v1809 operating system, Windows Server v1809 operating system, and Windows Server 2019, and non-NTFS file systems on all versions of Windows, defer this processing until the Open gets closed.
	
	Added new Section 2.1.4.19, Algorithm for Noting That a File Has Been Accessed:

Errata Published*	Description
	2.1.4.19 Algorithm for Noting That a File Has Been Accessed The inputs for this algorithm are as follows: ● Open: The Open through which the file was accessed. The pseudocode for the algorithm is as follows: ● The object store SHOULD<43>: ● If Open.UserSetAccessTime is FALSE, set Open.File.LastAccessTime to the current system time. <43> Section 2.1.4.19: File systems may choose to defer processing for a file that has been accessed to a later time, favoring performance over accuracy. The NTFS file system on versions prior to Windows 10 v1809, Windows Server v1809, and Windows Server 2019, and non-NTFS file systems on all versions of Windows, defer this processing until the Open gets closed. In Section 2.1.5.2, Server Requests a Read, clarified the object store behavior when the file has been accessed. Changed from: ... Pseudocode for the operation is as follows: ... ● If (ByteOffset >= Open.Stream.ValidDataLength): ● If Open.Mode.FILE_SYNCHRONOUS_IO_ALERT is TRUE or Open.Mode.FILE_SYNCHRONOUS_IO_NONALERT is TRUE, the object store MUST set Open.CurrentByteOffset to (ByteOffset + ByteCount). ● If Open.File.UserSetAccessTime is FALSE, the object store MUST update Open.File.LastAccessTime to the current system time. ● The object store MUST return: ● BytesRead set to ByteCount. ● OutputBuffer filled with ByteCount zero(s). ● Status set to STATUS_SUCCESS. ● EndIf ● If ((ByteOffset + ByteCount) >= Open.Stream.ValidDataLength), truncate ByteCount to (Open.Stream.ValidDataLength - ByteOffset). ● Set BytesToRead to BlockAlign(ByteCount,Open.File.Volume.LogicalBytesPerSector). ● Read BytesToRead bytes from the disk at offset ByteOffset for this stream into OutputBuffer. If the read from the disk failed, the operation MUST be failed with the same error status. ● If RequestedByteCount > ByteCount, zero out OutputBuffer between ByteCount and RequestedByteCount. ● If Open.Mode.FILE_SYNCHRONOUS_IO_ALERT is TRUE or Open.Mode.FILE_SYNCHRONOUS_IO_NONALERT is TRUE, the object store MUST set Open.CurrentByteOffset to (ByteOffset + RequestedByteCount). ● If Open.File.UserSetAccessTime is FALSE, the object store MUST update Open.File.LastAccessTime to the current system time. ● Upon successful completion of the operation, the object store MUST return: ● BytesRead set to RequestedByteCount. ● Status set to STATUS_SUCCESS.

Errata Published*	Description
	● Else ● Read ByteCount bytes at offset ByteOffset from the cache for this stream into OutputBuffer. ● If Open.Mode.FILE_SYNCHRONOUS_IO_ALERT is TRUE or Open.Mode.FILE_SYNCHRONOUS_IO_NONALERT is TRUE, the object store MUST set Open.CurrentByteOffset to (ByteOffset + ByteCount). ● If Open.File.UserSetAccessTime is FALSE, the object store MUST update Open.File.LastAccessTime to the current system time. ● Upon successful completion of the operation, the object store MUST return: ● BytesRead set to ByteCount. ● Status set to STATUS_SUCCESS. ● EndIf Changed to: ... Pseudocode for the operation is as follows: ... ● If (ByteOffset >= Open.Stream.ValidDataLength): ● If Open.Mode.FILE_SYNCHRONOUS_IO_ALERT is TRUE or Open.Mode.FILE_SYNCHRONOUS_IO_NONALERT is TRUE, the object store MUST set Open.CurrentByteOffset to (ByteOffset + ByteCount). ● The object store MUST note that the file has been accessed as specified in section 2.1.4.19 with Open equal to Open. ● The object store MUST return: ● BytesRead set to ByteCount. ● OutputBuffer filled with ByteCount zero(s). ● Status set to STATUS_SUCCESS. ● EndIf ● If ((ByteOffset + ByteCount) >= Open.Stream.ValidDataLength), truncate ByteCount to (Open.Stream.ValidDataLength - ByteOffset). ● Set BytesToRead to BlockAlign(ByteCount,Open.File.Volume.LogicalBytesPerSector). ● Read BytesToRead bytes from the disk at offset ByteOffset for this stream into OutputBuffer. If the read from the disk failed, the operation MUST be failed with the same error status. ● If RequestedByteCount > ByteCount, zero out OutputBuffer between ByteCount and RequestedByteCount. ● If Open.Mode.FILE_SYNCHRONOUS_IO_ALERT is TRUE or Open.Mode.FILE_SYNCHRONOUS_IO_NONALERT is TRUE, the object store MUST set Open.CurrentByteOffset to (ByteOffset + RequestedByteCount). ● The object store MUST note that the file has been accessed as specified in section 2.1.4.19 with Open equal to Open. ● Upon successful completion of the operation, the object store MUST return: ● BytesRead set to RequestedByteCount. ● Status set to STATUS_SUCCESS. ● Else ● Read ByteCount bytes at offset ByteOffset from the cache for this stream into OutputBuffer. ● If Open.Mode.FILE_SYNCHRONOUS_IO_ALERT is TRUE or Open.Mode.FILE_SYNCHRONOUS_IO_NONALERT is TRUE, the object store MUST set

Errata Published*	Description
	Open.CurrentByteOffset to (ByteOffset + ByteCount). ● The object store MUST note that the file has been accessed as specified in section 2.1.4.19 with Open equal to Open. ● Upon successful completion of the operation, the object store MUST return: ● BytesRead set to ByteCount. ● Status set to STATUS_SUCCESS. ● EndIf In Section 2.1.5.5.3, Directory Information Queries, clarified the object store behavior when the file has been accessed. Changed from: ... Pseudocode for the algorithm is as follows: ... ● If Open.File.UserSetAccessTime is FALSE, the object store MUST update Open.File.LastAccessTime to the current system time. ● The object store MUST return: ● Status set to StatusToReturn. ● OutputBuffer containing an array of as many entries that match the query as will fit in OutputBufferSize. ● BytesReturned containing the number of bytes filled in OutputBuffer. Changed to: ... Pseudocode for the algorithm is as follows: ... ● The object store MUST note that the file has been accessed as specified in section 2.1.4.19 with Open equal to Open. ● The object store MUST return: ● Status set to StatusToReturn. ● OutputBuffer containing an array of as many entries that match the query as will fit in OutputBufferSize. ● BytesReturned containing the number of bytes filled in OutputBuffer.
2019/11/25	In Section 2.1.5.11.30, FileNormalizedNameInformation, changed the pseudocode from BuildNormalizedRelativeName to BuildRelativeName. Changed from: ... Pseudocode for the operation is as follows: ... ● Set FileName to BuildNormalizedRelativeName(Open.Link, Open.File.Volume.RootDirectory).

Errata Published*	Description

	Changed to:
	...
	Pseudocode for the operation is as follows:
	...
	
	● Set FileName to BuildRelativeName(Open.Link, Open.File.Volume.RootDirectory).
	
	Removed Section 2.1.4.19, BuildNormalizedRelativeName -- Algorithm for Building the Normalized Relative Path Name for a Link.
	
	<Deleted content>
	The inputs for this algorithm are:
	
	● Link: A Link whose relative path name is being created.
	● RootDirectory: A DirectoryFile indicating how far to walk up the directory hierarchy when creating the relative path name.
	This algorithm returns a Unicode string representing the portion of a Link's path name from the RootDirectory to the Link itself, inclusive. The returned string starts with a backslash and uses backslashes as path separators. If Link is not a descendant of RootDirectory, the algorithm returns an empty string to indicate the error.
	Pseudocode for the algorithm is as follows:
	● If Link.File equals RootDirectory:
	  ● Return "\".
	● Else If Link.File equals Link.File.Volume.RootDirectory:
	  ● Return an empty string.
	● Else If Link.ParentFile equals RootDirectory:
	  ● Return "\" + Link.Name.
	● Else
	  ● Set ParentRelativeName to BuildRelativeName(Link.ParentFile, RootDirectory).
	  ● If ParentRelativeName is empty:
	  ● Return an empty string.
	  ● Else
	  ● Return ParentRelativeName + "\" + Link.Name.
	  ● EndIf
	● EndIf

2019/10/16	In Section 2.1.5.14.2, FileBasicInformation, the timestamp behavior has been clarified by updating the pseudocode for the operation.
	
	Changed from:
	...
	Pseudocode for the operation is as follows:
	● If InputBufferSize is less than sizeof(FILE_BASIC_INFORMATION), the operation MUST be failed with STATUS_INFO_LENGTH_MISMATCH.
	● The operation MUST be failed with STATUS_INVALID_PARAMETER under any of the following conditions:
	  ● If InputBuffer.CreationTime is less than -1.
	  ● If InputBuffer.LastAccessTime is less than -1.
	  ● If InputBuffer.LastWriteTime is less than -1.

Errata Published*	Description
	● If InputBuffer.ChangeTime is less than -1.<148> ... ● If InputBuffer.ChangeTime != 0: ● The object store MUST set Open.UserSetChangeTime to TRUE. ● If InputBuffer.ChangeTime != -1: ● Set BreakParentOplock to TRUE. ● If InputBuffer.ChangeTime !=Open.File.LastChangeTime, the object store MUST set UsnReason.USN_REASON_BASIC_INFO_CHANGE to TRUE. ● The object store MUST set Open.File.LastChangeTime to InputBuffer.ChangeTime. ● EndIf ● EndIf ● If InputBuffer.CreationTime != 0 and InputBuffer.CreationTime != -1:... ● If InputBuffer.LastAccessTime != 0: ● The object store MUST set Open.UserSetAccessTime to TRUE. ● If InputBuffer.LastAccessTime != -1: ● Set BreakParentOplock to TRUE. ● If InputBuffer. LastAccessTime != Open.File.LastAccessTime, the object store MUST set UsnReason.USN_REASON_BASIC_INFO_CHANGE to TRUE. ● The object store MUST set Open.File.LastAccessTime to InputBuffer. LastAccessTime. ● The object store MUST set Open.File.PendingNotifications.FILE_NOTIFY_CHANGE_LAST_ACCESS to TRUE. ● If Open.UserSetChangeTime is FALSE and InputBuffer.ChangeTime != -1, the object store MUST set Open.File.LastChangeTime to CurrentTime. ● EndIf ● EndIf ● If InputBuffer.LastWriteTime != 0: ● The object store MUST set Open.UserSetModificationTime to TRUE. ● If InputBuffer.LastWriteTime != -1: ● Set BreakParentOplock to TRUE. ● If InputBuffer. LastWriteTime != Open.File.LastModificationTime, the object store MUST set UsnReason.USN_REASON_BASIC_INFO_CHANGE to TRUE. ● The object store MUST set Open.File.LastModificationTime to InputBuffer. LastWriteTime. ● The object store MUST set Open.File.PendingNotifications.FILE_NOTIFY_CHANGE_LAST_WRITE to TRUE. ● If Open.UserSetChangeTime is FALSE and InputBuffer.ChangeTime != -1, the object store MUST set Open.File.LastChangeTime to CurrentTime. ● EndIf ● EndIf... Changed to: ... Pseudocode for the operation is as follows: ● If InputBufferSize is less than sizeof(FILE_BASIC_INFORMATION), the operation MUST be failed

Errata Published*	Description
	with STATUS_INFO_LENGTH_MISMATCH. ● The operation MUST be failed with STATUS_INVALID_PARAMETER under any of the following conditions: ● If InputBuffer.CreationTime is less than -2. ● If InputBuffer.LastAccessTime is less than -2. ● If InputBuffer.LastWriteTime is less than -2. ● If InputBuffer.ChangeTime is less than -2.<148>... ● If InputBuffer.ChangeTime != 0: ● If InputBuffer.ChangeTime != -2: ● The object store MUST set Open.UserSetChangeTime to TRUE. ● If InputBuffer.ChangeTime != -1: ● Set BreakParentOplock to TRUE. ● If InputBuffer.ChangeTime !=Open.File.LastChangeTime, the object store MUST set UsnReason.USN_REASON_BASIC_INFO_CHANGE to TRUE. ● The object store MUST set Open.File.LastChangeTime to InputBuffer.ChangeTime. ● EndIf ● Else ● The object store MUST set Open.UserSetChangeTime to FALSE. ● EndIf ● EndIf ● If InputBuffer.CreationTime != 0 and InputBuffer.CreationTime != -1 and InputBuffer.CreationTime != -2: ... ● If InputBuffer.LastAccessTime != 0: ● If InputBuffer.LastAccessTime != -2: ● The object store MUST set Open.UserSetAccessTime to TRUE. ● If InputBuffer.LastAccessTime != -1: ● Set BreakParentOplock to TRUE. ● If InputBuffer. LastAccessTime != Open.File.LastAccessTime, the object store MUST set UsnReason.USN_REASON_BASIC_INFO_CHANGE to TRUE. ● The object store MUST set Open.File.LastAccessTime to InputBuffer. LastAccessTime. ● The object store MUST set Open.File.PendingNotifications.FILE_NOTIFY_CHANGE_LAST_ACCESS to TRUE. ● If Open.UserSetChangeTime is FALSE and InputBuffer.ChangeTime != -1, the object store MUST set Open.File.LastChangeTime to CurrentTime. ● EndIf ● Else: ● The object store MUST set Open.UserSetAccessTime to FALSE. ● EndIf ● EndIf ● If InputBuffer.LastWriteTime != 0: ● If InputBuffer.LastWriteTime != -2: ● The object store MUST set Open.UserSetModificationTime to TRUE. ● If InputBuffer.LastWriteTime != -1: ● Set BreakParentOplock to TRUE.

Errata Published*	Description
	● If InputBuffer. LastWriteTime != Open.File.LastModificationTime, the object store MUST set UsnReason.USN_REASON_BASIC_INFO_CHANGE to TRUE. ● The object store MUST set Open.File.LastModificationTime to InputBuffer. LastWriteTime. ● The object store MUST set Open.File.PendingNotifications.FILE_NOTIFY_CHANGE_LAST_WRITE to TRUE. ● If Open.UserSetChangeTime is FALSE and InputBuffer.ChangeTime != -1, the object store MUST set Open.File.LastChangeTime to CurrentTime. ● EndIf ● Else: ● The object store MUST set Open.UserSetModificationTime to FALSE. ● EndIf ● EndIf ...
2019/09/02	In Section 2.1.1.1, Per Volume, a new ADM element has been added: ● ReservedSpace: A 64-bit unsigned integer specifying the amount of free space of the volume in bytes that is reserved for implementation specific use and not available to callers. This value MUST be a multiple of ClusterSize and MUST be less than or equal to Volume.FreeSpace. In Section 2.1.5.9.11, FSCTL_GET_NTFS_VOLUME_DATA, the following bullet point has been changed from: OutputBuffer.TotalReserved set to an implementation-specific value. Changed to: OutputBuffer.TotalReserved set to Open.File.Volume.ReservedSpace / Open.File.Volume.ClusterSize. In Section 2.1.5.9.12, FSCTL_GET_REFS_VOLUME_DATA, the following bullet point has been changed from: OutputBuffer.TotalReserved set to an implementation-specific value. Changed to: OutputBuffer.TotalReserved set Open.File.Volume.ReservedSpace / Open.File.Volume.ClusterSize. In Section 2.1.5.12.3, FileFsSizeInformation, the following bullet points have been changed from: OutputBuffer.AvailableAllocationUnits set to Open.File.Volume.FreeSpace / Open.File.Volume.ClusterSize. If RemainingQuota < Open.File.Volume.FreeSpace: Changed to:

Errata Published*	Description
	OutputBuffer.AvailableAllocationUnits set to (Open.File.Volume.FreeSpace - Open.File.Volume.ReservedSpace) / Open.File.Volume.ClusterSize. If RemainingQuota < (Open.File.Volume.FreeSpace - Open.File.Volume.ReservedSpace): In Section 2.1.5.12.7, FileFsFullSizeInformation, the following bullet points have been changed from: OutputBuffer.CallerAvailableAllocationUnits set to Open.File.Volume.FreeSpace / Open.File.Volume.ClusterSize. OutputBuffer.ActualAvailableAllocationUnits set to Open.File.Volume.FreeSpace / Open.File.Volume.ClusterSize. If RemainingQuota < Open.File.Volume.FreeSpace: Changed to: OutputBuffer.CallerAvailableAllocationUnits set to (Open.File.Volume.FreeSpace - Open.File.Volume.ReservedSpace) / Open.File.Volume.ClusterSize. OutputBuffer.ActualAvailableAllocationUnits set to (Open.File.Volume.FreeSpace - Open.File.Volume.ReservedSpace) / Open.File.Volume.ClusterSize. If RemainingQuota < (Open.File.Volume.FreeSpace - Open.File.Volume.ReservedSpace):
2019/07/08	In Section 2.1.5.11.28, FileStandardLinkInformation, the error code was changed from: This operation is not supported and MUST be failed with STATUS_INVALID_INFO_CLASS. Changed to: This operation is not supported and MUST be failed with STATUS_NOT_SUPPORTED.

*Date format: YYYY/MM/DD

[MS-FSCC]: File System Control Codes

This topic lists the Errata found in the MS-FSCC document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

September 15, 2017 - Download

December 1, 2017 - Download

September 12, 2018 - Download

September 23, 2019 - Download

Errata below are for Protocol Document Version V29.0 – 2019/05/30.

Errata Published*	Description
2020/01/20	In Section 2.4.8, FileBothDirectoryInformation, clarified Windows behavior when using ReFS or NTFS and the position of a file within the parent directory is not fixed and can be changed at any time by updating product behavior note <96> in the FileIndex field description. Changed from: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0x00000000 and MUST be ignored.<96> <96> Section 2.4.8: When using ReFS or NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 set this value to zero for files on NTFS file systems. Changed to: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0 and MUST be ignored.<96> <96> Section 2.4.8: When using ReFS or NTFS, the position of a file within the parent directory

Errata Published*	Description
	is not fixed and can be changed at any time. Windows sets this value to zero for files on ReFS and NTFS file systems. In Section 2.4.10, FileDirectoryInformation, clarified Windows behavior when using ReFS or NTFS and the position of a file within the parent directory is not fixed and can be changed at any time by updating product behavior note <101> in the FileIndex field description. Changed from: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0 and MUST be ignored.<101> <101> Section 2.4.10: When using NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 set this value to zero for files on NTFS file systems. Changed to: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0 and MUST be ignored.<101> <101> Section 2.4.10: When using ReFS or NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows sets this value to zero for files on ReFS and NTFS file systems. In Section 2.4.14, FileFullDirectoryInformation, clarified Windows behavior when using ReFS or NTFS and the position of a file within the parent directory is not fixed and can be changed at any time by updating product behavior note <103> in the FileIndex field description. Changed from: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems such as NTFS, in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0, and MUST be ignored.<103> <103> Section 2.4.14: When using NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 set this value to zero for files on NTFS file systems. Changed to: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0 and

Errata Published*	Description
	MUST be ignored.<103> <103> Section 2.4.14: When using ReFS or NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows sets this value to zero for files on ReFS and NTFS file systems. In Section 2.4.17, FileIdBothDirectoryInformation, clarified Windows behavior when using ReFS or NTFS and the position of a file within the parent directory is not fixed and can be changed at any time by updating product behavior note <105> in the FileIndex field description. Changed from: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0, and MUST be ignored.<105> <105> Section 2.4.17: When using NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 set this value to zero for files on NTFS file systems. Changed to: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0 and MUST be ignored.<105> <105> Section 2.4.17: When using ReFS or NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows sets this value to zero for files on ReFS and NTFS file systems. In Section 2.4.18, FileIdFullDirectoryInformation, clarified Windows behavior when using ReFS or NTFS and the position of a file within the parent directory is not fixed and can be changed at any time by updating product behavior note <108> in the FileIndex field description. Changed from: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0 and MUST be ignored.<108> <108> Section 2.4.18: When using NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 set this value to zero for files on NTFS file systems. Changed to:

Errata Published*	Description
	... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0 and MUST be ignored.<108> <108> Section 2.4.18: When using ReFS or NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows sets this value to zero for files on ReFS and NTFS file systems. In Section 2.4.19, FileIdGlobalTxDirectoryInformation, clarified Windows behavior when using ReFS or NTFS and the position of a file within the parent directory is not fixed and can be changed at any time by adding product behavior note <110> in the FileIndex field description. Changed from: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0 and MUST be ignored. Changed to: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0 and MUST be ignored.<110> <110> Section 2.4.19: When using ReFS or NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows sets this value to zero for files on ReFS and NTFS file systems. In Section 2.4.26, FileNamesInformation, clarified Windows behavior when using ReFS or NTFS and the position of a file within the parent directory is not fixed and can be changed at any time by updating product behavior note <116> in the FileIndex field description. Changed from: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0, and MUST be ignored.<115> <115> Section 2.4.26: When using NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 set this value to zero for files on NTFS file systems. Changed to: ... FileIndex (4 bytes): A 32-bit unsigned integer that contains the byte offset of the file within the

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>parent directory. For file systems in which the position of a file within the parent directory is not fixed and can be changed at any time to maintain sort order, this field SHOULD be set to 0 and MUST be ignored.<116>

<116> Section 2.4.26: When using ReFS or NTFS, the position of a file within the parent directory is not fixed and can be changed at any time. Windows sets this value to zero for files on ReFS and NTFS file systems.</td></tr>
<tr><td>2019/11/25</td><td>In Section 2.4.42, FileNormalizedNameInformation, clarified what a normalized name means.

Changed from:
This information class is used to query the normalized name of a file. This information class returns a FILE_NAME_INFORMATION data element containing an absolute pathname, as specified in section 2.1.7. <126>
...

Changed to:
This information class is used to query the normalized name of a file. A normalized name is an absolute pathname where each short name component has been replaced with the corresponding long name component, and each name component uses the exact letter casing stored on disk. This information class returns a FILE_NAME_INFORMATION data element containing an absolute pathname, as specified in section 2.1.7. <126>
...</td></tr>
<tr><td>2019/11/25</td><td>In Section 2.1.2.1, Reparse Tags, added multiple Microsoft reparse tags to the table.

Changed from:
...
The following reparse tags, with the exception of IO_REPARSE_TAG_SYMLINK, are processed on the server and are not processed by a client after transmission over the wire. Clients SHOULD treat associated reparse data as opaque data.<2>

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>IO_REPARSE_TAG_RESERVED_ZERO
0x00000000</td><td>Reserved reparse tag value.</td></tr>
<tr><td>IO_REPARSE_TAG_RESERVED_ONE
0x00000001</td><td>Reserved reparse tag value.</td></tr>
<tr><td>IO_REPARSE_TAG_MOUNT_POINT
0xA0000003</td><td>Used for mount point support, specified in section 2.1.2.5.</td></tr>
<tr><td>IO_REPARSE_TAG_HSM
0xC0000004</td><td>Obsolete. Used by legacy Hierarchical Storage Manager Product.</td></tr>
<tr><td>IO_REPARSE_TAG_DRIVER_EXTENDER
0x80000005</td><td>Home server drive extender.<3></td></tr>
<tr><td>IO_REPARSE_TAG_HSM2
0x80000006</td><td>Obsolete. Used by legacy Hierarchical Storage Manager Product.</td></tr>
<tr><td>IO_REPARSE_TAG_SIS
0x80000007</td><td>Used by single-instance storage (SIS) filter driver. Server-side interpretation only, not meaningful over the wire.</td></tr>
</table></td></tr>
</table>

Errata Published*	Description

IO_REPARSE_TAG_DFS 0x8000000A	Used by the DFS filter. The DFS is described in the Distributed File System (DFS): Referral Protocol Specification [MS-DFSC]. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_FILTER_MANAGER 0x8000000B	Used by filter manager test harness.<4>
IO_REPARSE_TAG_SYMLINK 0xA000000C	Used for symbolic link support. See section 2.1.2.4.
IO_REPARSE_TAG_DFSR 0x80000012	Used by the DFS filter. The DFS is described in [MS-DFSC]. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_NFS 0x80000014	Used by the Network File System (NFS) component. Server-side interpretation only, not meaningful over the wire.

Changed to:

...

The following reparse tags, with the exception of IO_REPARSE_TAG_SYMLINK, are processed on the server and are not processed by a client after transmission over the wire. Clients SHOULD treat associated reparse data as opaque data.<2>

Value	Meaning
IO_REPARSE_TAG_RESERVED_ZERO 0x00000000	Reserved reparse tag value.
IO_REPARSE_TAG_RESERVED_ONE 0x00000001	Reserved reparse tag value.
IO_REPARSE_TAG_RESERVED_TWO 0x00000002	Reserved reparse tag value.
IO_REPARSE_TAG_MOUNT_POINT 0xA0000003	Used for mount point support, specified in section 2.1.2.5.
IO_REPARSE_TAG_HSM 0xC0000004	Obsolete. Used by legacy Hierarchical Storage Manager Product.
IO_REPARSE_TAG_DRIVE_EXTENDER 0x80000005	Home server drive extender.<3>
IO_REPARSE_TAG_HSM2 0x80000006	Obsolete. Used by legacy Hierarchical Storage Manager Product.
IO_REPARSE_TAG_SIS 0x80000007	Used by single-instance storage (SIS) filter driver. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_WIM	Used by the WIM Mount filter. Server-side

Errata Published*	Description
	0x80000008 — interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_CSV 0x80000009 — Obsolete. Used by Clustered Shared Volumes (CSV) version 1 in Windows Server 2008 R2 operating system. Server-side interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_DFS 0x8000000A — Used by the DFS filter. The DFS is described in the Distributed File System (DFS): Referral Protocol Specification [MS-DFSC]. Server-side interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_FILTER_MANAGER 0x8000000B — Used by filter manager test harness.<4>
	IO_REPARSE_TAG_SYMLINK 0xA000000C — Used for symbolic link support. See section 2.1.2.4.
	IO_REPARSE_TAG_IIS_CACHE 0xA0000010 — Used by Microsoft Internet Information Services (IIS) caching. Server-side interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_DFSR 0x80000012 — Used by the DFS filter. The DFS is described in [MS-DFSC]. Server-side interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_DEDUP 0x80000013 — Used by the Data Deduplication (Dedup) filter. Server-side interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_APPXSTRM 0xC0000014 — Not used.
	IO_REPARSE_TAG_NFS 0x80000014 — Used by the Network File System (NFS) component. Server-side interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_FILE_PLACEHOLDER 0x80000015 — Obsolete. Used by Windows Shell for legacy placeholder files in Windows 8.1. Server-side interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_DFM 0x80000016 — Used by the Dynamic File filter. Server-side interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_WOF 0x80000017 — Used by the Windows Overlay filter, for either WIMBoot or single-file compression. Server-side interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_WCI 0x80000018 — Used by the Windows Container Isolation filter. Server-side interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_WCI_1 0x90001018 — Used by the Windows Container Isolation filter. Server-side interpretation only, not meaningful over the wire.
	IO_REPARSE_TAG_GLOBAL_REPARSE 0xA0000019 — Used by NPFS to indicate a named pipe symbolic link from a server silo into the host silo. Server-side interpretation only, not meaningful over the

Errata Published*	Description

	wire.
IO_REPARSE_TAG_CLOUD 0x9000001A	Used by the Cloud Files filter, for files managed by a sync engine such as Microsoft OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_1 0x9000101A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_2 0x9000201A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_3 0x9000301A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_4 0x9000401A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_5 0x9000501A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_6 0x9000601A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_7 0x9000701A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_8 0x9000801A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_9 0x9000901A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_A 0x9000A01A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_B 0x9000B01A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.

Errata Published*	Description

IO_REPARSE_TAG_CLOUD_C 0x9000C01A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_D 0x9000D01A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_E 0x9000E01A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_CLOUD_F 0x9000F01A	Used by the Cloud Files filter, for files managed by a sync engine such as OneDrive. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_APPEXECLINK 0x8000001B	Used by Universal Windows Platform (UWP) packages to encode information that allows the application to be launched by CreateProcess. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_PROJFS 0x9000001C	Used by the Windows Projected File System filter, for files managed by a user mode provider such as VFS for Git. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_LX_SYMLINK 0xA000001D	Used by the Windows Subsystem for Linux (WSL) to represent a UNIX symbolic link. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_STORAGE_SYNC 0x8000001E	Used by the Azure File Sync (AFS) filter. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_WCI_TOMBSTONE 0xA000001F	Used by the Windows Container Isolation filter. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_UNHANDLED 0x80000020	Used by the Windows Container Isolation filter. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_ONEDRIVE 0x80000021	Not used.
IO_REPARSE_TAG_PROJFS_TOMBSTONE 0xA0000022	Used by the Windows Projected File System filter, for files managed by a user mode provider such as VFS for Git. Server-side interpretation only, not meaningful over the wire.
IO_REPARSE_TAG_AF_UNIX 0x80000023	Used by the Windows Subsystem for Linux (WSL) to represent a UNIX domain socket. Server-side interpretation only, not meaningful over the wire.

Errata Published*	Description
	<table><tr><td>IO_REPARSE_TAG_LX_FIFO 0x80000024</td><td>Used by the Windows Subsystem for Linux (WSL) to represent a UNIX FIFO (named pipe). Server-side interpretation only, not meaningful over the wire.</td></tr><tr><td>IO_REPARSE_TAG_LX_CHR 0x80000025</td><td>Used by the Windows Subsystem for Linux (WSL) to represent a UNIX character special file. Server-side interpretation only, not meaningful over the wire.</td></tr><tr><td>IO_REPARSE_TAG_LX_BLK 0x80000026</td><td>Used by the Windows Subsystem for Linux (WSL) to represent a UNIX block special file. Server-side interpretation only, not meaningful over the wire.</td></tr><tr><td>IO_REPARSE_TAG_WCI_LINK 0xA0000027</td><td>Used by the Windows Container Isolation filter. Server-side interpretation only, not meaningful over the wire.</td></tr><tr><td>IO_REPARSE_TAG_WCI_LINK_1 0xA0001027</td><td>Used by the Windows Container Isolation filter. Server-side interpretation only, not meaningful over the wire.</td></tr></table>
2019/11/25	In Section 2.4.7, FileBasicInformation, clarified when a timestamp value of -2 is set in the CreationTime, LastAccessTime, LastWriteTime, and ChangeTime field descriptions. Also added a product behavior note to the CreationTime field description and clarified Windows behavior in the existing product behavior notes. Changed from: ... CreationTime (8 bytes): The time when the file was created; see section 2.1.1. A valid time for this field is an integer greater than or equal to 0. When setting file attributes, a value of 0 indicates to the server that it MUST NOT change this attribute. When setting file attributes, a value of -1 indicates to the server that it MUST NOT change this attribute for all subsequent operations on the same file handle. This field MUST NOT be set to a value less than -1. LastAccessTime (8 bytes): The last time the file was accessed; see section 2.1.1. A valid time for this field is an integer greater than or equal to 0. When setting file attributes, a value of 0 indicates to the server that it MUST NOT change this attribute. When setting file attributes, a value of -1 indicates to the server that it MUST NOT change this attribute for all subsequent operations on the same file handle. This field MUST NOT be set to a value less than -1.<92> LastWriteTime (8 bytes): The last time information was written to the file; see section 2.1.1. A valid time for this field is an integer greater than or equal to 0. When setting file attributes, a value of 0 indicates to the server that it MUST NOT change this attribute. When setting file attributes, a value of -1 indicates to the server that it MUST NOT change this attribute for all subsequent operations on the same file handle. This field MUST NOT be set to a value less than -1.<93> ChangeTime (8 bytes): The last time the file was changed; see section 2.1.1. A valid time for this field is an integer greater than or equal to 0. When setting file attributes, a value of 0 indicates to the server that it MUST NOT change this attribute. When setting file attributes, a value of -1 indicates to the server that it MUST NOT change this attribute for all subsequent operations on the same file handle. This field MUST NOT be set to a value less than -1.<94> Changed to:

Errata Published*	Description
	... CreationTime (8 bytes): The time when the file was created; see section 2.1.1. A valid time for this field is an integer greater than or equal to 0. When setting file attributes, a value of 0 indicates to the server that it MUST NOT change this attribute. When setting file attributes, a value of -1 indicates to the server that it MUST NOT change this attribute for all subsequent operations on the same file handle. When setting file attributes, a value of -2 indicates to the server that it MUST change this attribute for all subsequent operations on the same file handle. This field MUST NOT be set to a value less than -2.<92> LastAccessTime (8 bytes): The last time the file was accessed; see section 2.1.1. A valid time for this field is an integer greater than or equal to 0. When setting file attributes, a value of 0 indicates to the server that it MUST NOT change this attribute. When setting file attributes, a value of -1 indicates to the server that it MUST NOT change this attribute for all subsequent operations on the same file handle. When setting file attributes, a value of -2 indicates to the server that it MUST change this attribute for all subsequent operations on the same file handle. This field MUST NOT be set to a value less than -2.<93> LastWriteTime (8 bytes): The last time information was written to the file; see section 2.1.1. A valid time for this field is an integer greater than or equal to 0. When setting file attributes, a value of 0 indicates to the server that it MUST NOT change this attribute. When setting file attributes, a value of -1 indicates to the server that it MUST NOT change this attribute for all subsequent operations on the same file handle. When setting file attributes, a value of -2 indicates to the server that it MUST change this attribute for all subsequent operations on the same file handle. This field MUST NOT be set to a value less than -2.<94> ChangeTime (8 bytes): The last time the file was changed; see section 2.1.1. A valid time for this field is an integer greater than or equal to 0. When setting file attributes, a value of 0 indicates to the server that it MUST NOT change this attribute. When setting file attributes, a value of -1 indicates to the server that it MUST NOT change this attribute for all subsequent operations on the same file handle. When setting file attributes, a value of -2 indicates to the server that it MUST change this attribute for all subsequent operations on the same file handle. This field MUST NOT be set to a value less than -2.<95> In Section 6, Appendix B: Product Behavior, added product behavior note <92> and updated product behavior notes <93>, <94>, and <95>. Changed from: ... <92> Section 2.4.7: The file system updates the values of the LastAccessTime, LastWriteTime, and ChangeTime members as appropriate after an I/O operation is performed on a file. However, a driver or application can request that the file system not update one or more of these members for I/O operations that are performed on the caller's file handle by setting the appropriate members to -1. The caller can set one, all, or any other combination of these three members to -1. Only the members that are set to -1 will be unaffected by I/O operations on the file handle; the other members will be updated as appropriate. This behavior is consistent across all file system types. Note that even though -1 can be used with the CreationTime field, it has no effect because file creation time is never updated in response to file system calls such as read and write. <93> Section 2.4.7: The file system updates the values of the LastAccessTime, LastWriteTime, and ChangeTime members as appropriate after an I/O operation is performed on a file. However, a driver or application can request that the file system not update one or more of these members for I/O operations that are performed on the caller's file handle by setting the appropriate members to -1. The caller can set one, all, or any other combination of these three members to -1. Only the members that are set to -1 will be unaffected by I/O operations on the file handle; the other members will be updated as appropriate. This behavior is consistent across all file

Errata Published*	Description
	system types. Note that even though -1 can be used with the CreationTime field, it has no effect because file creation time is never updated in response to file system calls such as read and write. <94> Section 2.4.7: The file system updates the values of the LastAccessTime, LastWriteTime, and ChangeTime members as appropriate after an I/O operation is performed on a file. However, a driver or application can request that the file system not update one or more of these members for I/O operations that are performed on the caller's file handle by setting the appropriate members to -1. The caller can set one, all, or any other combination of these three members to -1. Only the members that are set to -1 will be unaffected by I/O operations on the file handle; the other members will be updated as appropriate. This behavior is consistent across all file system types. Note that even though -1 can be used with the CreationTime field, it has no effect because file creation time is never updated in response to file system calls such as read and write. ... Changed to: ... <92> Section 2.4.7: The file system updates the values of the LastAccessTime, LastWriteTime, and ChangeTime members as appropriate after an I/O operation is performed on a file. However, a driver or application can request that the file system not update one or more of these members for I/O operations that are performed on the caller's file handle by setting the appropriate members to -1. A driver or application can subsequently request that the file system resume updating one or more of these members for I/O operations that are performed on the caller's file handle by setting the appropriate members to -2. The caller can set one, all, or any other combination of these three members to -1 and/or -2. Only the members that are set to -1 will be unaffected by I/O operations on the file handle; the other members will be updated as appropriate. This behavior is consistent across all file system types. Note that even though -1 and -2 can be used with the CreationTime field, they have no effect because file creation time is never updated in response to file system calls such as read and write. File system | Support value of -2 FAT | No EXFAT | No FAT32 | No Cdfs | No UDFS | No NTFS | Windows 8.1 and later, Windows Server 2012 R2 and later and Windows Server v1709 operating system and later ReFS | Windows 10 v1507 operating system and later, Windows Server 2016 and later, and Windows Server v1709 and later <93> Section 2.4.7: The file system updates the values of the LastAccessTime, LastWriteTime, and ChangeTime members as appropriate after an I/O operation is performed on a file. However, a driver or application can request that the file system not update one or more of these members for I/O operations that are performed on the caller's file handle by setting the appropriate members to -1. A driver or application can subsequently request that the file system resume

Errata Published*	Description
	updating one or more of these members for I/O operations that are performed on the caller's file handle by setting the appropriate members to -2. The caller can set one, all, or any other combination of these three members to -1 and/or -2. Only the members that are set to -1 will be unaffected by I/O operations on the file handle; the other members will be updated as appropriate. This behavior is consistent across all file system types. Note that even though -1 and -2 can be used with the CreationTime field, they have no effect because file creation time is never updated in response to file system calls such as read and write. File system: Support value of -2 FAT: No EXFAT: No FAT32: No Cdfs: No UDFS: No NTFS: Windows 8.1 and later, Windows Server 2012 R2 and later and Windows Server v1709 operating system and later ReFS: Windows 10 v1507 operating system and later, Windows Server 2016 and later, and Windows Server v1709 and later <94> Section 2.4.7: The file system updates the values of the LastAccessTime, LastWriteTime, and ChangeTime members as appropriate after an I/O operation is performed on a file. However, a driver or application can request that the file system not update one or more of these members for I/O operations that are performed on the caller's file handle by setting the appropriate members to -1. A driver or application can subsequently request that the file system resume updating one or more of these members for I/O operations that are performed on the caller's file handle by setting the appropriate members to -2. The caller can set one, all, or any other combination of these three members to -1 and/or -2. Only the members that are set to -1 will be unaffected by I/O operations on the file handle; the other members will be updated as appropriate. This behavior is consistent across all file system types. Note that even though -1 and -2 can be used with the CreationTime field, they have no effect because file creation time is never updated in response to file system calls such as read and write. File system: Support value of -2 FAT: No EXFAT: No FAT32: No Cdfs: No UDFS: No NTFS: Windows 8.1 and later, Windows Server 2012 R2 and later and Windows Server v1709 operating system and later

Errata Published*	Description

ReFS	Windows 10 v1507 operating system and later, Windows Server 2016 and later, and Windows Server v1709 and later

<95> Section 2.4.7: The file system updates the values of the LastAccessTime, LastWriteTime, and ChangeTime members as appropriate after an I/O operation is performed on a file. However, a driver or application can request that the file system not update one or more of these members for I/O operations that are performed on the caller's file handle by setting the appropriate members to -1. A driver or application can subsequently request that the file system resume updating one or more of these members for I/O operations that are performed on the caller's file handle by setting the appropriate members to -2. The caller can set one, all, or any other combination of these three members to -1 and/or -2. Only the members that are set to -1 will be unaffected by I/O operations on the file handle; the other members will be updated as appropriate. This behavior is consistent across all file system types. Note that even though -1 and -2 can be used with the CreationTime field, they have no effect because file creation time is never updated in response to file system calls such as read and write.

File system	Support value of -2
FAT	No
EXFAT	No
FAT32	No
Cdfs	No
UDFS	No
NTFS	Windows 8.1 and later, Windows Server 2012 R2 and later and Windows Server v1709 operating system and later
ReFS	Windows 10 v1507 operating system and later, Windows Server 2016 and later, and Windows Server v1709 and later

...

*Date format: YYYY/MM/DD

[MS-FSRVP]: File Server Remote VSS Protocol

This topic lists the Errata found in the MS-FSRVP document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

[MS-FSVCA]: File Set Version Comparison Algorithms

This topic lists the Errata found in the MS-FSVCA document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-GPPREF]: Group Policy: Preferences Extension Data Structure

This topic lists the Errata found in [MS-GPPREF] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

September 12, 2018 - Download

[MS-GPSB]: Group Policy: Security Protocol Extension

This topic lists the Errata found in [MS-GPSB] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

September 12, 2018 - Download

[MS-GPOL]: Group Policy: Core Protocol

This topic lists the Errata found in [MS-GPOL] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

June 1, 2017 - Download

Errata below are for Protocol Document Version V36.1 – 2019/03/15.

Errata Published*	Description
2019/05/27	In Section 2.2.4, GPO Search, changed from: The gpt.ini file MUST be encoded in UTF-8 and is described with the following Augmented Backus-Naur Form (ABNF), as specified in [RFC4234]. Changed to: The gpt.ini file MUST be encoded in ANSI and is described with the following Augmented Backus-Naur Form (ABNF), as specified in [RFC4234].

*Date format: YYYY/MM/DD

[MS-GSSA]: Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG) Protocol Extension

This topic lists the Errata found in the MS-GSSA document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-HGSA]: Host Guardian Service: Attestation Protocol

This topic lists the Errata found in the MS-HGSA document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

September 15, 2017 - Download

September 12, 2018 - Download

June 24, 2019 – Download

[MS-HTTPE]: Hypertext Transfer Protocol (HTTP) Extensions

This topic lists the Errata found in [MS-HTTPE] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-HVRS]: Hyper-V Remote Storage Profile

This topic lists the Errata found in [MS-HVRS] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

March 20, 2017 - Download

[MS-ICPR]: ICertPassage Remote Protocol

This topic lists the Errata found in the MS-ICPR document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-IKEE]: Internet Key Exchange Protocol Extensions

This topic lists the Errata found in the MS-IKEE document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

September 15, 2017 - Download

Errata below are for Protocol Document Version V27.0 – 2018/09/12.

Errata Published*	Description
2019/10/28	In Section 2.2.8, Configuration Attribute (IKEv2) Packet, changed from: Length (2 bytes): The length of the data in the value field. Changed to: Length (2 bytes): The length of the data in the Value field. In Section 2.2.11.2, Encrypted Fragment Payload, changed from: Next_Payload (1 byte): In the very first fragment (with Fragment Number equal to 1), this field MUST be set to the payload type of the first inner payload. In the remainder of the Fragment messages (with Fragment Number greater than 1), this field MUST be set to zero. Changed to: Next_Payload (1 byte): In the very first fragment (with Fragment_Number equal to 1), this field MUST be set to the payload type of the first inner payload. In the remainder of the Fragment messages (with Fragment_Number greater than 1), this field MUST be set to zero. In Section 3.3.1, Abstract Data Model, references have been added o disambiguate which fields in section 2.2.3.1 set the values of the ADM elements: Fragment ID, Fragment Number, Flag, and Fragment Data. Changed from: Fragment queue: A queue holding the fragments that correspond to incomplete IKE messages, indexed by the Fragment ID. Each entry in the queue MUST contain: -- The Fragment ID -- The Fragment Number -- A Flag that indicates whether this fragment is the last one (that is, the LAST_FRAGMENT

Errata Published*	Description
	bit is set in the Fragment payload). -- The Fragment Data For definitions of the previous values, see section 2.2.3.1. Flow state table: The following information MUST be maintained. Changed to: Fragment queue: A queue holding the fragments that correspond to incomplete IKE messages, indexed by the Fragment ID. Each entry in the queue MUST contain: -- The Fragment ID, which is set to the Fragment_ID field in section 2.2.3.1. -- The Fragment Number, which is set to the Fragment_Number field in section 2.2.3.1. -- A Flag that is set to the Flags field in section 2.2.3.1 to indicates whether this fragment is the last one (that is, the LAST_FRAGMENT bit is set in the Fragment payload). -- The Fragment Data, which is set to the Fragment_Data field in section 2.2.3.1. Flow state table: The following information MUST be maintained. In Section 3.3.2, Timers, the second bullet point has been changed from: When the fragmentation reassembly timer fires, the delay MUST NOT exceed 90 seconds.<17> Changed to: When the fragment reassembly timer fires, the delay MUST NOT exceed 90 seconds.<17> In Section 3.3.5.3, Receiving Other IKE Messages, the action taken by the Receiver upon receipt of an IKE message (to discard such a message when a Fragment payload is present and it is not the only payload in the message) has been clarified. Changed from: On receipt of an IKE message, the host MUST check if the message contains a Fragment payload. If a Fragment payload is present, this payload MUST be the only payload in the message. If not, the host MUST silently discard the message. Changed to: On receipt of an IKE message, the host MUST check if the message contains a Fragment payload. If a Fragment payload is present, and the payload is not the only payload in the message, the host MUST silently discard the message' In Section 3.3.5.3, Receiving Other IKE Messages, text has been changed to clarify from where to retrieve the Fragment ID. Changed from: Retrieve the Fragment ID from the Fragment payload.

Errata Published*	Description
	Changed to: Retrieve the Fragment ID from the Fragment_ID field in the Fragment payload. In Section 3.3.5.3, Receiving Other IKE Messages, text has been changed to clarify how fragments not of the same Fragment Number are added to the Fragment queue in the corresponding entry of the MMSAD. Changed from: If the queue for this Fragment ID already contains a fragment with the same Fragment Number, the host MUST silently discard the message. If not, the host MUST queue the Fragment payload's fields in the corresponding entry of the MMSAD, indexed by the Fragment Id Changed to: If the queue for this Fragment ID already contains a fragment with the same Fragment Number, the host MUST silently discard the message. If not, the host MUST add an entry to the Fragment queue in the corresponding entry of the MMSAD, with the queue entry fields initialized based on the associated fields of the Fragment payload. In Section 3.3.5.3, Receiving Other IKE Messages, changed from: The host MUST then check whether all Fragment payloads for this Fragment ID have been received (that is, whether Fragment payloads that have a Fragment number from 1 to n.. Changed to: The host MUST then check whether all Fragment payloads for this Fragment ID have been received (that is, whether Fragment payloads that have a Fragment Number from 1 to n.. In Section 3.3.5.3, Receiving Other IKE Messages, text has been changed to clarify the error condition where the host MUST discard all Fragment payloads for a specific Fragment ID. Changed from: A Fragment payload has been received with a Fragment number greater than the Fragment number of the fragment with the Flags field set to LAST_FRAGMENT.' Changed to: A Fragment payload has been received with a Fragment Number greater than the Fragment Number of an entry in the Fragment queue with the Flags field set to LAST_FRAGMENT. In Section 3.3.5.3, Receiving Other IKE Messages, changed from: Fragment payloads (without the Fragment payload header) in the order of their Fragment number. Changed to:

Errata Published*	Description
	Fragment payloads (without the Fragment payload header) in the order of their Fragment Number. In Section 3.15.1, Abstract Data Model, references have been added to disambiguate which fields in section 2.2.3.1 set the values of the ADM elements: Fragment ID, Fragment Number, and Fragment Data. Changed from: Fragment queue: A queue holding the fragments that correspond to incomplete IKE messages, indexed by the Fragment ID. Each entry in the queue MUST contain the following: Fragment ID, which is the Message ID Fragment Number Total Fragments Fragment Data Flow state table: The following information MUST be maintained. Changed to: Fragment queue: A queue holding the fragments that correspond to incomplete IKE messages, indexed by the Fragment ID. Each entry in the queue MUST contain the following: Fragment ID, which is the Message ID, is set to the Fragment_ID field in section 2.2.3.1. Fragment Number, which is set to the Fragment_Number field in section 2.2.3.1. Total Fragments Fragment Data, which is set to the Fragment_Data field in section 2.2.3.1. Flow state table: The following information MUST be maintained.

*Date format: YYYY/MM/DD

[MS-IPAMM2]: IP Address Management (IPAM) Management Protocol Version 2

This topic lists the Errata found in [MS-IPAMM2] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

June 1, 2017 - Download

[MS-IPHTTPS]: IP over HTTPS (IP-HTTPS) Tunneling Protocol

This topic lists the Errata found in the MS-IPHTTPS document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 12, 2018 - Download

[MS-IRP]: Internet Information Services (IIS) Inetinfo Remote Protocol

This topic lists the Errata found in [MS-IRP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-KILE]: Kerberos Protocol Extensions

This topic lists the Errata found in [MS-KILE] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

Errata below are for Protocol Document Version V34.0 – 2018/09/12.

Errata Published*	Description
2020/03/02 (correction to Errata published on 2019/10/28)	In Section 3.3.5.7.5, Cross-Domain Trust and Referrals, changed from: If the TRUST_ATTRIBUTE_CROSS_ORGANIZATION_NOENABLE_TGT_DELEGATION flag is set in the trustAttributes field ([MS-ADTS] section 6.1.6.7.9), the KDC MUST<63> return a ticket with the ok-as-delegate flag notset in TicketFlags. Changed to: If the TRUST_ATTRIBUTE_CROSS_ORGANIZATION_ENABLE_TGT_DELEGATION flag is set in the trustAttributes field ([MS-ADTS] section 6.1.6.7.9), the KDC MUST<66> return a ticket with the ok-as-delegate flag set in TicketFlags.
2019/11/25	In Section 2.2.1, KERB-EXT-ERROR, added that other bit values SHOULD be ignored. Changed from: Flags: Set to 0x00000001. Changed to: Flags: Set to 0x00000001. Other bit values SHOULD be ignored on receipt.
2019/10/28	in Section 1.2.2 Informative References, added a reference to [MSKB-4490425]. Changed from: [MS-SFU] Microsoft Corporation, "Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol". [RFC1510] Kohl, J., and Neuman, C., "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993, http://www.ietf.org/rfc/rfc1510.txt

Errata Published*	Description
	Changed to: [MS-SFU] Microsoft Corporation, "Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol". [MSKB-4490425] Microsoft Corporation, "Updates to TGT delegation across incoming trusts in Windows Server", https://support.microsoft.com/en-us/help/4490425/updates-to-tgt-delegation-across-incoming-trusts-in-windows-server [RFC1510] Kohl, J., and Neuman, C., "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993, http://www.ietf.org/rfc/rfc1510.txt In Section 3.3.5.7.5 Cross-Domain Trust and Referrals, added ticket return directives for various TRUST_ATTRIBUTE_CROSS_ORGANIZATION type flags. Changed from: If there is a failure in the check, the KDC MUST reject the authentication request with KDC_ERROR_POLICY. If the TRUST_ATTRIBUTE_CROSS_ORGANIZATION_NO_TGT_DELEGATION flag is set in the trustAttributes field ([MS-ADTS] section 6.1.6.7.9), the KDC MUST return a ticket with the ok-as-delegate flag not set in TicketFlags. Changed to: If there is a failure in the check, the KDC MUST reject the authentication request with KDC_ERROR_POLICY. The KDC MUST NOT return a ticket with the ok-as-delegate flag set in TicketFlags. If the TRUST_ATTRIBUTE_CROSS_ORGANIZATION_NOENABLE_TGT_DELEGATION flag is set in the trustAttributes field ([MS-ADTS] section 6.1.6.7.9), the KDC MUST<63> return a ticket with the ok-as-delegate flag notset in TicketFlags. If the TRUST_ATTRIBUTE_CROSS_ORGANIZATION_NO_TGT_DELEGATION is set in the trustedAttributes field ([MS-ADTS] section 6.1.6.7.9) the KDC MUST NOT return a ticket with the ok-as-delegate flag set in TicketFlags. In Section 6 Appendix A: Product Behavior, added a behavior note to state applicable products for the TRUST_ATTRIBUTE_CROSS_ORGANIZATION_ENABLE_TGT_DELEGATION flag. Changed from: <63> Section 3.3.5.7.6: Not supported in Windows 2000 and Windows Server 2003. Changed to:

Errata Published*	Description
	<63> Section 3.3.5.7.5: The TRUST_ATTRIBUTE_CROSS_ORGANIZATION_ENABLE_TGT_DELEGATION flag is supported on Windows Server 2003 and later when [MSKB-4490425] is installed. <64> Section 3.3.5.7.6: Not supported in Windows 2000 and Windows Server 2003.
2019/09/30	In Section 3.2.5.7, TGS Exchange, behavior note <32> has been updated to document the way in which current versions of Windows Server diverge from the [RFC6806] standard. Changed from: <32> Section 3.2.5.7: Compound Identity and FAST are not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, or Windows Server 2008 R2. Changed to: <32> Section 3.2.5.7: Compound Identity and FAST are not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, or Windows Server 2008 R2. Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server, and Windows Server 2019 do not completely conform to RFC6806, in that they will set the Enc-Pa-Rep flag in the Ticket flags, despite not supporting encrypted PA data in TGS-REP messages, if they have FAST enabled.

*Date format: YYYY/MM/DD

[MS-KPP]: Key Provisioning Protocol

This topic lists the Errata found in the MS-KPP document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-KPS]: Key Protection Service Protocol

This topic lists the Errata found in the MS-KPP document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

September 15, 2017 - Download

[MS-LCID]: Windows Language Code Identifier (LCID) Reference

This topic lists the Errata found in [MS-LCID] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 15, 2017 - Download

December 1, 2017 - Download

September 12, 2018 - Download

March 13, 2019 - Download

[MS-LSAD]: Local Security Authority (Domain Policy) Remote Protocol

This topic lists the Errata found in [MS-LSAD] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

March 16, 2018 - Download

Errata below are for Protocol Document Version V43.0 – 2019/09/12.

Errata Published*	Description
2019/10/16	In Section 2.2.4.4, LSAPR_POLICY_AUDIT_EVENTS_INFO: Changed from: MaximumAuditingEventCount Changed to: MaximumAuditEventCount

*Date format: YYYY/MM/DD

[MS-LSAT]: Local Security Authority (Translation Methods) Remote Protocol

This topic lists the Errata found in [MS-LSAT] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

June 1, 2017 - Download

June 1, 2017 - Download

September 15, 2017 - Download

[MS-MDE]: Mobile Device Enrollment Protocol

This topic lists the Errata found in [MS-MDE] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

September 15, 2017 - Download

[MS-MDE2]: Mobile Device Enrollment Protocol Version 2

This topic lists the Errata found in [MS-MDE2] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

September 15, 2017 - Download

December 1, 2017 - Download

September 12, 2018 - Download

[MS-MDM]: Mobile Device Management Protocol

This topic lists the Errata found in [MS-MDM] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

December 1, 2017 - Download

[MS-MICE]: Miracast over infrastructure Connection Establishment Protocol

This topic lists the Errata found in [MS-MICE] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 12, 2018 - Download

[MS-MSSOD]: Media Streaming Server Protocols Overview

This topic lists the Errata found in [MS-MSSOD] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

December 1, 2017 - Download

[MS-MWBE]: Microsoft Web Browser Federated Sign-On Protocol Extensions

This topic lists the Errata found in [MS-MWBE] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous version of this document, see the following ERRATA archive:

June 30, 2015 - Download

[MS-MWBF]: Microsoft Web Browser Federated Sign-On Protocol

This topic lists the Errata found in [MS-MWBF] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

June 1, 2017 - Download

September 12, 2018 - Download

[MS-NCNBI]: Network Controller Northbound Interface Specification

This topic lists the Errata found in the MS-NCNBI document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

March 16, 2018 - Download

September 12, 2018 - Download

Errata below are for Protocol Document Version V6.0 – 2018/09/12.

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td>2018/12/17</td><td>
In several sections throughout this document, missing element Type designations have been added to existing element or header tables. For example, in Section 2.2.1.2, Request Headers, the text in bold has been added to the existing table as shown below.

<table>
<tr><th>Header</th><th>Section</th><th>Type</th><th>Description</th></tr>
<tr><td>Content-Type</td><td>2.2.1.1</td><td>Required or Optional Required for PUT, must be "application/json; charset=UTF-8". Optional for GET or Delete</td><td>The content type of the payload.</td></tr>
</table>

In the following sections, the added Type designations are shown in bold.

2.2.2, Common JSON Elements

<table>
<tr><td>resourceId</td><td>Optional or Required When optional for ancestor resource, then required for descendant resource. See section 2.2.3.</td></tr>
<tr><td>resourceRef</td><td>Read-only Optional or Required See section 1.3.3.2.</td></tr>
<tr><td>properties.etag</td><td>Read-only</td></tr>
</table>
</td></tr>
</table>

Errata Published*	Description

properties.provisioningState	**Read-only**

3.1.5.1 accessControlLists

configurationState.id	**Optional Read-only**
virtualNetworkInterfaceErrors	**Optional Read-only**

3.1.5.5.3 frontendIPConfigurations

configurationState.vipEndpointStates	**Read-only**
configurationState.vipEndpointStates.vipEndpoint	**Read-only**
configurationState.vipEndpointStates.dipEndpointStates	**Read-only**
configurationState.vipEndpointStates.dipEndpointStates.dipEndpoint	**Read-only**
configurationState.vipEndpointStates.dipEndpointStates.hostIPAddress	**Read-only**
configurationState.vipEndpointStates.dipEndpointStates.hostId	**Read-only**
configurationState.vipEndpointStates.dipEndpointStates.AdapterId	**Read-only**
configurationState.vipEndpointStates.dipEndpointStates.ProbeRule	**Read-only**

3.1.5.11 networkInterfaces

dnsSettings	**Optional**
dnsSettings.dnsServers	**Optional**
ipConfigurations	**Read-only**
isHostVirtualNetworkInterface	**Optional FALSE is default. Cannot be changed after creation.**
internalDnsNameLabel	**Optional**
isPrimary	**Optional TRUE is default.**
isMultitenantStack	**Optional**
privateMacAddress	**Optional**
privateMacAllocationMethod	**Required**
serviceInsertionElements	**Optional Read-only**

Errata Published*	Description
	3.1.5.14 publicIPAddresses dnsSettings — **Optional** 3.1.5.15 servers connections — **Required** connections.credential — **Required** connections.credentialType — **Required** connections.managementAddresses — **Required** certificate — **Optional or Required Required only if the certificate used by the server is self-signed.** 3.1.5.18 virtualNetworks configurationState.id — **Optional Read-only** configurationState.hostErrors — **Optional Read-only** 3.1.5.18.3 virtualNetworkPeerings remoteVirtualNetwork — **Required** 3.1.5.21 virtualServers connections.credential — **Optional** connections.credentialType — **Optional** connections.managementAddresses — **Optional** In Section 3.1.5.26, changed from: HTTP method — Description

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>PUT</td><td>Create a new virtualNetworkManager resource or update an existing VirtualGateways resource.</td></tr>
<tr><td>GET</td><td>Get one virtualNetworkManager resource</td></tr>
</table>
Changed to:
<table>
<tr><td>HTTP method</td><td>Description</td></tr>
<tr><td>PUT</td><td>Update the virtualNetworkManager singleton resource.</td></tr>
<tr><td>GET</td><td>Get the virtualNetworkManager resource.</td></tr>
</table>
</td></tr>
</table>

* Date format: YYYY/MM/DD

[MS-NCT]: Network Cost Transfer Protocol

This topic lists the Errata found in the MS-NCT document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-NFPB]: Near Field Proximity Bidirectional Services Protocol

This topic lists the Errata found in [MS-NFPB] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-NFPS]: Near Field Proximity Sharing Protocol

This topic lists the Errata found in [MS-NFPS] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-NKPU]: Network Key Protector Unlock Protocol

This topic lists the Errata found in [MS-NKPU] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

December 1, 2017 - Download

[MS-NLMP]: NT LAN Manager (NTLM) Authentication Protocol

This topic lists the Errata found in [MS-NLMP] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

September 23, 2019 - Download

Errata below are for Protocol Document Version V31.0 – 2019/09/23.

Errata Published*	Description
2019/12/16	In Section 3.4, Session Security Details, added ANONYMOUS user with Guest user and section reference. Changed from: For the case of Guest user login, there is no session security. Changed to: For the cases of ANONYMOUS user and Guest user login, there is no session security (see section 3.2.5.1.2). In Section 5.1, Security Considerations for Implementers, added ANONYMOUS user, Guest user, and Guest log in case 2 of 3. Changed from: The use of NullSession results in a SessionBaseKey with all zeroes, which does not provide security. Therefore, applications are generally advised not to use NullSession. The Guest user account is disabled by default in Windows for security reasons. If the Guest user account is enabled, it is strongly recommended to set a password so that logon failures do not result in Guest logins (section 3.2.5.1.2). Changed to: The use of ANONYMOUS user NullSession results in a SessionBaseKey with all zeroes, which does not provide security. Therefore, applications are generally advised not to use NullSession. The use of Guest user GuestSession results in a SessionBaseKey with all zeroes, which does not provide security. The Guest user account is disabled by default in Windows for security reasons. If the Guest user account is enabled, it is strongly recommended to set a password so that logon failures do not result in Guest logins (section 3.2.5.1.2). If a password is set on the Guest account, then there is a guest fallback where logons will be tried with unknown usernames against the Guest password.

*Date format: YYYY/MM/DD

[MS-NMFMB]: .NET Message Framing MSMQ Binding Protocol

This topic lists the Errata found in [MS-NMFMB] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

December 1, 2017 - Download

[MS-NNS]: .NET NegotiateStream Protocol

This topic lists the Errata found in [MS-NNS] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

Errata below are for Protocol Document Version V7.0 – 2017/12/01.

Errata Published*	Description
2019/02/19	In Section 2.2.2, Data Message, the maximum size of the PayloadSize field has been changed from '0x0000FC00' to '0x0000FC30', to accommodate for both the application data size and the size increase that occurs when this protocol signs or encrypts the data to be transferred. Changed from: PayloadSize (4 bytes): The unsigned size, in bytes, of the Payload field. The maximum value for this field is 0x0000FC00 (that is, 63K, or 64,512). Changed to: PayloadSize (4 bytes): The unsigned size, in bytes, of the Payload field. The maximum value for this field is 0x0000FC30 (64,560).

*Date format: YYYY/MM/DD

[MS-NRBF]: .NET Remoting: Binary Format Data Structure

This topic lists the Errata found in [MS-NRBF] 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 V12.0 – 2019/03/13.

Errata Published*	Description
2019/10/28	In Section 3.0, Structure Examples, in the logical Request message for dotNET_Framework 1.1, changed the BinaryMethodCall value from: BinaryMethodCall: RecordTypeEnum: BinaryMethodCall (0x21) MessageEnum: 00000014 Changed to: BinaryMethodCall: RecordTypeEnum: BinaryMethodCall (0x15) MessageEnum: 00000014

*Date format: YYYY/MM/DD

[MS-NRPC]: Netlogon Remote Protocol

This topic lists the Errata found in [MS-NRPC] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

September 23, 2019 - Download

[MS-NSPI]: Name Service Provider Interface (NSPI) Protocol

This topic lists the Errata found in [MS-NSPI] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-OAPX]: OAuth 2.0 Protocol Extensions

This topic lists the Errata found in [MS-OAPX] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

June 1, 2017 - Download

September 15, 2017 - Download

September 12, 2018 - Download

[MS-OAPXBC]: OAuth 2.0 Protocol Extensions for Broker Clients

This topic lists the Errata found in [MS-OAPXBC] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

September 26, 2016 - Download

June 1, 2017 - Download

September 15, 2017 - Download

September 12, 2018 - Download

[MS-OIDCE]: OpenID Connect 1.0 Protocol Extensions

This topic lists the Errata found in [MS-OIDCE] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

December 1, 2017 - Download

March 16, 2018 - Download

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures

This topic lists the Errata found in [MS-OLEDS] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

December 1, 2017 - Download

September 12, 2018 - Download

[MS-OTPCE]: One-Time Password Certificate Enrollment Protocol

This topic lists the Errata found in [MS-OTPCE] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-PAC]: Privilege Attribute Certificate Data Structure

This topic lists the Errata found in [MS-PAC] 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 V19.0 – 2018/09/12.

Errata Published*	Description
2019/09/02	In Section 3.1, Logon Authorization Information, the string format for two SIDs has been changed from: S-1-5-397955417-626881126-188441444 Changed to: S-1-5-21-397955417-626881126-188441444 Changed from: S-1-5-397955417-626881126-188441444-3392609 Changed to: S-1-5-21-397955417-626881126-188441444-3392609

*Date format: YYYY/MM/DD

[MS-PAR]: Print System Asynchronous Remote Protocol

This topic lists the Errata found in [MS-PAR] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

Errata below are for Protocol Document Version V15.0 – 2018/09/12.

Errata Published*	Description
2018/12/10	In Section 3.1.4.2.7, RpcAsyncInstallPrinterDriverFromPackage (Opnum 62), changed from: The print server SHOULD<10> perform the following additional validation steps: … • Validate that, if the printer driver specified by the client is a derived printer driver, either the class printer driver on which the derived printer driver depends is already installed on the print server, or a driver package containing the class printer driver is installed in the print server's driver store, or the print server can locate a driver package containing the class printer driver through some other implementation-specific mechanism;<11> otherwise, the server returns ERROR_UNKNOWN_PRINTER_DRIVER. Changed to: The print server SHOULD<10> perform the following additional validation steps: … • Validate that, if the printer driver specified by the client is a derived printer driver, either the class printer driver on which the derived printer driver depends is already installed on the print server, or a driver package containing the class printer driver is installed in the print server's driver store, or the print server can locate a driver package containing the class printer driver through some other implementation-specific mechanism;<11> otherwise, the server returns ERROR_UNKNOWN_PRINTER_DRIVER. This HRESULT error code is constructed by using the HRESULT From WIN32 Error Code Macro ([MS-ERREF] section 2.1.2) on the 16-bit Win32 value for this error ([MS-ERREF] section 2.2).

*Date format: YYYY/MM/DD

[MS-PEAP]: Protected Extensible Authentication Protocol (PEAP)

This topic lists the Errata found in [MS-PEAP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-PKAP]: Public Key Authentication Protocol

This topic lists the Errata found in the MS-PKAP document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-PSRDP]: PowerShell Remote Debugging Protocol

This topic lists the Errata found in [MS-PSRDP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-PSRP]: PowerShell Remoting Protocol

This topic lists the Errata found in [MS-PSRP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

[MS-RA]: Remote Assistance Protocol

This topic lists the Errata found in [MS-RA] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-RAI]: Remote Assistance Initiation Protocol

This topic lists the Errata found in [MS-RAI] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

Errata below are for Protocol Document Version V9.0 – 2018/09/12.

Errata Published*	Description
2019/06/24	In Section 2.2.2, Remote Assistance Connection String 2, details for the URI attribute have been added for the Listener node. Changed from: ... 3. The Transport Node has Listener child Nodes that give information about the Server IP and port. This Listener node <L> has the following attributes. Value | Meaning P | Port: The dynamic port on which the Remote Assistance connection could happen. N | Server Name: The name/IP address of the server, that is, the novice computer. ... Changed to: ... 3. The Transport Node has Listener child Nodes that give information about the Server IP and port. This Listener node <L> has the following attributes. Value | Meaning

Errata Published*	Description
	<table><tr><td>P</td><td>Port: The dynamic port on which the Remote Assistance connection could happen.</td></tr><tr><td>N</td><td>Server Name: The name/IP address of the server, that is, the novice computer.</td></tr><tr><td>U</td><td>URI: The full URI if websocket listener is enabled. The U (URI) is used instead of the P (port) attribute. N (server name) attribute is still included.</td></tr></table> …

*Date format: YYYY/MM/DD

[MS-RDPADRV]: Remote Desktop Protocol Audio Level and Drive Letter Persistence Virtual Channel Extension

This topic lists the Errata found in [MS-RDPADRV] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 12, 2018 - Download

[MS-RDPBCGR]: Remote Desktop Protocol: Basic Connectivity and Graphics Remoting

This topic lists the Errata found in [MS-RDPBCGR] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

March 20, 2017 - Download

June 1, 2017 - Download

December 1, 2017 - Download

March 16, 2018 - Download

September 12, 2018 - Download

March 13, 2019 - Download

September 23, 2019 - Download

Errata below are for Protocol Document Version V51.0 – 2019/09/23.

Errata Published *	Description
2020/02/17	In Section 2.2.7.2.7 Large Pointer Capability Set (TS_LARGE_POINTER_CAPABILITYSET), clarified the minimum values for MaxRequestSize based on which flags are specified in the largePointerSupportFlags field. Changed from: To support large pointer shapes, the client and server MUST support multifragment updates and indicate this support by exchanging the Multifragment Update Capability Set (section 2.2.7.2.6). The MaxRequestSize field of the Multifragment Update Capability Set MUST be set based on the flags included in the largePointerSupportFlags field. If LARGE_POINTER_FLAG_384x384 (0x00000001) is not included, then the MaxRequestSize field MUST be set to at least 38,055 bytes (so that a 96 x 96 pixel 32bpp pointer can be transported). If LARGE_POINTER_FLAG_384x384 (0x00000002) is included, then the MaxRequestSize MUST be set to at least 608,299 bytes (so that a 384 x 384 pixel 32bpp pointer can be transported). Changed to: To support large pointer shapes, the client and server MUST support multifragment updates and indicate this support by exchanging the Multifragment Update Capability Set (section 2.2.7.2.6).

Errata Published *	Description
	The MaxRequestSize field of the Multifragment Update Capability Set MUST be set based on the flags included in the largePointerSupportFlags field. If only the LARGE_POINTER_FLAG_96x96 (0x00000001) flag is specified, then the MaxRequestSize field MUST be set to at least 38,055 bytes (so that a 96 x 96 pixel 32bpp pointer can be transported). If the LARGE_POINTER_FLAG_384x384 (0x00000002) flag is included, then the MaxRequestSize MUST be set to at least 608,299 bytes (so that a 384 x 384 pixel 32bpp pointer can be transported).
2020/01/20	In Section 2.2.7.2.7, Large Pointer Capability Set (TS_LARGE_POINTER_CAPABILITYSET), provided the hexadecimal value when LARGE_POINTER_FLAG_384x384 is included and when it is not included. Changed from: The TS_LARGE_POINTER_CAPABILITYSET structure is used to specify capabilities related to large mouse pointer shape support. This capability is sent by both client and server. To support large pointer shapes, the client and server MUST support multifragment updates and indicate this support by exchanging the Multifragment Update Capability Set (section 2.2.7.2.6). The MaxRequestSize field of the Multifragment Update Capability Set MUST be set based on the flags included in the largePointerSupportFlags field. If the LARGE_POINTER_FLAG_384x384 is not included, then the MaxRequestSize field MUST be set to at least 38,055 bytes (so that a 96 x 96 pixel 32bpp pointer can be transported). If the LARGE_POINTER_FLAG_384x384 is included, then the MaxRequestSize MUST be set to at least 608,299 bytes (so that a 384 x 384 pixel 32bpp pointer can be transported). ... Changed to: The TS_LARGE_POINTER_CAPABILITYSET structure is used to specify capabilities related to large mouse pointer shape support. This capability is sent by both client and server. To support large pointer shapes, the client and server MUST support multifragment updates and indicate this support by exchanging the Multifragment Update Capability Set (section 2.2.7.2.6). The MaxRequestSize field of the Multifragment Update Capability Set MUST be set based on the flags included in the largePointerSupportFlags field. If LARGE_POINTER_FLAG_384x384 (0x00000001) is not included, then the MaxRequestSize field MUST be set to at least 38,055 bytes (so that a 96 x 96 pixel 32bpp pointer can be transported). If LARGE_POINTER_FLAG_384x384 (0x00000002) is included, then the MaxRequestSize MUST be set to at least 608,299 bytes (so that a 384 x 384 pixel 32bpp pointer can be transported). ...
2020/01/06	In Section 2.2.10.1.1.4.1.1, Logon Errors Info (TS_LOGON_ERRORS_INFO), added the ERROR_CODE_ACCESS_DENIED value to the ErrorNotificationType field table. Changed from: ... ErrorNotificationType (4 bytes): A 32-bit, unsigned integer that specifies an NTSTATUS value (see [ERRTRANS] for information about translating NTSTATUS error codes to usable text strings), or one of the following values. Value | Meaning

Errata Published *	Description

LOGON_MSG_DISCONNECT_REFUSED 0xFFFFFFF9	The"Disconnection Refused" dialog is being displayed by Winlogon. The session identifier is specified by the ErrorNotificationData field.
LOGON_MSG_NO_PERMISSION 0xFFFFFFFA	The "No Permission" dialog is being displayed by Winlogon. The session identifier is specified by the ErrorNotificationData field.
LOGON_MSG_BUMP_OPTIONS 0xFFFFFFFB	The "Session Contention" dialog is being displayed by Winlogon. The session identifier is specified by the ErrorNotificationData field.
LOGON_MSG_ RECONNECT_OPTIONS 0xFFFFFFFC	The "Session Reconnection" dialog is being displayed by Winlogon. The session identifier is specified by the ErrorNotificationData field.
LOGON_MSG_SESSION_TERMINATE 0xFFFFFFFD	The session is being terminated. The session identifier is specified by the ErrorNotificationData field.
LOGON_MSG_SESSION_CONTINUE 0xFFFFFFFE	The logon process is continuing. The session identifier is specified by the ErrorNotificationData field.

...

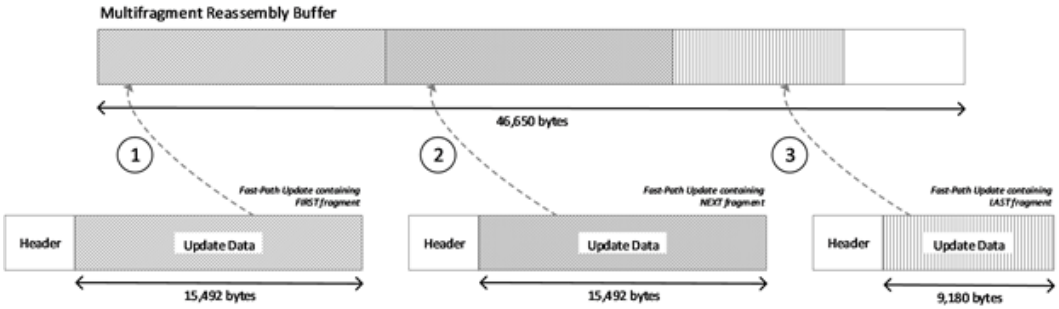
Changed to:

...

ErrorNotificationType (4 bytes): A 32-bit, unsigned integer that specifies an NTSTATUS value (see [ERRTRANS] for information about translating NTSTATUS error codes to usable text strings), or one of the following values.

Value	Meaning
LOGON_MSG_DISCONNECT_REFUSED 0xFFFFFFF9	The "Disconnection Refused" dialog is being displayed by Winlogon. The session identifier is specified by the ErrorNotificationData field.
LOGON_MSG_NO_PERMISSION 0xFFFFFFFA	The "No Permission" dialog is being displayed by Winlogon. The session identifier is specified by the ErrorNotificationData field.
LOGON_MSG_BUMP_OPTIONS 0xFFFFFFFB	The "Session Contention" dialog is being displayed by Winlogon. The session identifier is specified by the ErrorNotificationData field.
LOGON_MSG_ RECONNECT_OPTIONS 0xFFFFFFFC	The "Session Reconnection" dialog is being displayed by Winlogon. The session identifier is specified by the ErrorNotificationData field.
LOGON_MSG_SESSION_TERMINATE 0xFFFFFFFD	The session is being terminated. The session identifier is specified by the ErrorNotificationData field.

Errata Published *	Description
	<table><tr><td>LOGON_MSG_SESSION_CONTINUE 0xFFFFFFFE</td><td>The logon process is continuing. The session identifier is specified by the ErrorNotificationData field.</td></tr><tr><td>ERROR_CODE_ACCESS_DENIED 0xFFFFFFFF</td><td>The logon process failed and cannot proceed. The contents of the ErrorNotificationData field SHOULD be ignored.</td></tr></table> ...
2020/01/06	In Section 2.2.7.2.7, Large Pointer Capability Set (TS_LARGE_POINTER_CAPABILITYSET), described how the MaxRequestSize field is set when the LARGE_POINTER_FLAG_384x384 is included and when it is not included. Changed from: ... To support large pointer shapes, the client and server MUST support multifragment updates and indicate this support by exchanging the Multifragment Update Capability Set (section 2.2.7.2.6). The MaxRequestSize field of the Multifragment Update Capability Set MUST be set to at least 38,055 bytes (so that a 96 x 96 pixel 32bpp pointer can be transported). ... Changed to: ... To support large pointer shapes, the client and server MUST support multifragment updates and indicate this support by exchanging the Multifragment Update Capability Set (section 2.2.7.2.6). The MaxRequestSize field of the Multifragment Update Capability Set MUST be set based on the flags included in the largePointerSupportFlags field. If the LARGE_POINTER_FLAG_384x384 is not included, then the MaxRequestSize field MUST be set to at least 38,055 bytes (so that a 96 x 96 pixel 32bpp pointer can be transported). If the LARGE_POINTER_FLAG_384x384 is included, then the MaxRequestSize MUST be set to at least 608,299 bytes (so that a 384 x 384 pixel 32bpp pointer can be transported)....
2020/01/06	Added the following new section 3.2.5.9.3.1, Processing Fast-Path Update Fragments: 3.2.5.9.3.1 Processing Fast-Path Update Fragments A Fast-Path Update (section 2.2.9.1.2.1) structure contains fragmented data in the updateData field if the fragmentation subfield of the updateHeader field is non-zero: ● FASTPATH_FRAGMENT_FIRST (0x2) ● FASTPATH_FRAGMENT_NEXT (0x3) ● FASTPATH_FRAGMENT_LAST (0x1) Fragments MUST be reassembled in the order in which they arrive from the server. A FASTPATH_FRAGMENT_FIRST fragment MUST start a sequence of fragments. Zero, one, or more FASTPATH_FRAGMENT_NEXT fragments MUST follow a FASTPATH_FRAGMENT_FIRST fragment. The FASTPATH_FRAGMENT_LAST fragment MUST follow a FASTPATH_FRAGMENT_NEXT or a FASTPATH_FRAGMENT_FIRST fragment. Valid fragment sequences can be summarized as: ● FIRST fragment, LAST fragment ● FIRST fragment, multiple NEXT fragments, LAST fragment Any deviation from the set of valid fragment sequences SHOULD trigger a disconnect.

Errata Published *	Description
	As fragments are received from the server, the client SHOULD copy the contents into a reassembly buffer. When the FASTPATH_FRAGMENT_LAST fragment has been received, the reassembly buffer will contain an update that SHOULD be processed. The type of the update is determined by the updateCode subfield in the updateHeader field (all updates MUST have the same updateCode and compression subfield values). An overview of the reassembly process is presented in the figure titled "Reassembly of a fragmented update".  Figure 7: Reassembly of a fragmented update
2019/10/28	In Section 2.2.8.1.2, Client Fast-Path Input Event PDU (TS_FP_INPUT_PDU), referenced the TS_FP_FIPS_INFO structure in the fipsInformation field description. Changed from: ... fipsInformation (4 bytes): Optional FIPS header information, present when the Encryption Method selected by the server (sections 5.3.2 and 2.2.1.4.3) is ENCRYPTION_METHOD_FIPS (0x00000010). The Fast-Path FIPS Information structure is specified in section 2.2.8.1.2.1 ... Changed to: ... fipsInformation (4 bytes): An optional Fast-Path FIPS Information (section 2.2.8.1.2.1) structure, present when the Encryption Method selected by the server (sections 5.3.2 and 2.2.1.4.3) is ENCRYPTION_METHOD_FIPS (0x00000010). ... In Section 2.2.9.1.2, Server Fast-Path Update PDU (TS_FP_UPDATE_PDU), referenced the TS_FP_FIPS_INFO structure in the fipsInformation field description. Changed from: ... fipsInformation (4 bytes): Optional FIPS header information, present when the Encryption Method selected by the server (sections 5.3.2 and 2.2.1.4.3) is ENCRYPTION_METHOD_FIPS (0x00000010). The Fast-Path FIPS Information structure is specified in section 2.2.8.1.2.1. ... Changed to: ... fipsInformation (4 bytes): An optional Fast-Path FIPS Information (section 2.2.8.1.2.1) structure, present when the Encryption Method selected by the server (sections 5.3.2 and 2.2.1.4.3) is

Errata Published *	Description
	ENCRYPTION_METHOD_FIPS (0x00000010). ... In Section 3.2.5.8.1.2, Sending Fast-Path Input Event PDU, added Fast-Path to the fipsInformation field description. Changed from: ... ● length1 and length2: Packet Length (section 2.2.8.1.2) ● fipsInformation: Optional FIPS Information (section 2.2.8.1.2) ● dataSignature: Optional Data Signature (section 2.2.8.1.2) ● numEvents: Optional Number of Events (section 2.2.8.1.2) ● PDU Contents (collection of fast-path input events): ... Changed to: ... ● length1 and length2: Packet length (section 2.2.8.1.2) ● fipsInformation: Optional Fast-Path FIPS Information (section 2.2.8.1.2) ● dataSignature: Optional data signature (section 2.2.8.1.2) ● numEvents: Optional number of events (section 2.2.8.1.2) ● PDU contents (collection of fast-path input events): ... In Section 3.2.5.9.3, Processing Fast-Path Update PDU, added Fast-Path to the fipsInformation field description. Changed from: ... ● length1 and length2: Packet Length (section 2.2.9.1.2) ● fipsInformation: Optional FIPS Information (section 2.2.9.1.2) ● dataSignature: Optional Data Signature (section 2.2.9.1.2) ● PDU Contents (collection of fast-path output updates): ... Changed to: ... ● length1 and length2: Packet length (section 2.2.9.1.2) ● fipsInformation: Optional Fast-Path FIPS Information (section 2.2.9.1.2) ● dataSignature: Optional data signature (section 2.2.9.1.2) ● PDU contents (collection of fast-path output updates): ... In Section 3.3.5.8.1.2, Processing Fast-Path Input Event PDU, added Fast-Path to the fipsInformation field description. Changed from:

Errata Published *	Description
	... ● length1 and length2: Packet Length (section 2.2.8.1.2) ● fipsInformation: Optional FIPS Information (section 2.2.8.1.2) ● dataSignature: Optional Data Signature (section 2.2.8.1.2) ● numEvents: Optional Number of Events (section 2.2.8.1.2) ● PDU Contents (collection of input events): ... Changed to: ... ● length1 and length2: Packet length (section 2.2.8.1.2) ● fipsInformation: Optional Fast-Path FIPS Information (section 2.2.8.1.2) ● dataSignature: Optional data signature (section 2.2.8.1.2) ● numEvents: Optional number of events (section 2.2.8.1.2) ● PDU contents (collection of input events): ... In Section 3.3.5.9.3, Sending Fast-Path Update PDU, added Fast-Path to the fipsInformation field description. Changed from: ... ● length1 and length2: Packet Length (section 2.2.9.1.2) ● fipsInformation: Optional FIPS Information (section 2.2.9.1.2) ● dataSignature: Optional Data Signature (section 2.2.9.1.2) ● PDU Contents (collection of fast-path output updates): ... Changed to: ... ● length1 and length2: Packet length (section 2.2.9.1.2) ● fipsInformation: Optional Fast-Path FIPS Information (section 2.2.9.1.2) ● dataSignature: Optional data signature (section 2.2.9.1.2) ● PDU contents (collection of fast-path output updates): ...
2019/10/28	In Section 2.2.1.17.1, Persistent Key List PDU Data (TS_BITMAPCACHE_PERSISTENT_LIST_PDU), changed PERSIST_FIRST_PDU to PERSIST_PDU_FIRST and PERSIST_LAST_PDU to PERSIST_PDU_LAST in the bBitMask table. Changed from: ... bBitMask (1 byte): An 8-bit, unsigned integer. The sequencing flag.

Errata Published *	Description
	<table><tr><th>Flag</th><th>Meaning</th></tr><tr><td>PERSIST_FIRST_PDU 0x01</td><td>Indicates that the PDU is the first in a sequence of Persistent Key List PDUs.</td></tr><tr><td>PERSIST_LAST_PDU 0x02</td><td>Indicates that the PDU is the last in a sequence of Persistent Key List PDUs.</td></tr></table> If neither PERSIST_FIRST_PDU (0x01) nor PERSIST_LAST_PDU (0x02) are set, then the current PDU is an intermediate packet in a sequence of Persistent Key List PDUs. ... Changed to: ... bBitMask (1 byte): An 8-bit, unsigned integer. The sequencing flag. <table><tr><th>Flag</th><th>Meaning</th></tr><tr><td>PERSIST_PDU_FIRST 0x01</td><td>Indicates that the PDU is the first in a sequence of Persistent Key List PDUs.</td></tr><tr><td>PERSIST_PDU_LAST</td><td>Indicates that the PDU is the last in a sequence of Persistent Key List PDUs.</td></tr></table> <table><tr><th>Flag</th><th>Meaning</th></tr><tr><td>0x02</td><td></td></tr></table> If neither PERSIST_FIRST_PDU (0x01) nor PERSIST_LAST_PDU (0x02) are set, then the current PDU is an intermediate packet in a sequence of Persistent Key List PDUs. ... In Section 2.2.7.1.1, General Capability Set (TS_GENERAL_CAPABILITYSET), changed field names generalCompressionTypes to compressionTypes and generalCompressionLevel to compressionLevel. Changed from: The TS_GENERAL_CAPABILITYSET structure is used to advertise general characteristics and is based on the capability set specified in [T128] section 8.2.3. This capability is sent by both client and server.

Errata Published *	Description

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
capabilitySetType																lengthCapability															
osMajorType																osMinorType															
protocolVersion																pad2octetsA															
generalCompressionTypes																extraFlags															
updateCapabilityFlag																remoteUnshareFlag															

generalCompressionLevel	refreshRectSupport	suppressOutputSupport

...

generalCompressionTypes (2 bytes): A 16-bit, unsigned integer. General compression types. This field MUST be set to zero.

...

generalCompressionLevel (2 bytes): A 16-bit, unsigned integer. General compression level. This field MUST be set to zero.

...

Changed to:

The TS_GENERAL_CAPABILITYSET structure is used to advertise general characteristics and is based on the capability set specified in [T128] section 8.2.3. This capability is sent by both client and server.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
capabilitySetType																lengthCapability															
osMajorType																osMinorType															
protocolVersion																pad2octetsA															
compressionTypes																extraFlags															
updateCapabilityFlag																remoteUnshareFlag															
compressionLevel																refreshRectSupport										suppressOutputSupport					

compressionTypes (2 bytes): A 16-bit, unsigned integer. General compression types. This field MUST be set to zero.

...

compressionLevel (2 bytes): A 16-bit, unsigned integer. General compression level. This field MUST be set to zero.

...

Errata Published *	Description
	In Section 2.2.7.1.11, Sound Capability Set (TS_SOUND_CAPABILITYSET), changed SOUND_BEEPS_FLAG to SOUND_FLAG_BEEPS in the soundFlags table. Changed from: ... soundFlags (2 bytes): A 16-bit, unsigned integer. Support for sound options. **Flag** | **Meaning** SOUND_BEEPS_FLAG 0x0001 | Playing a beep sound is supported. Changed to: ... soundFlags (2 bytes): A 16-bit, unsigned integer. Support for sound options. **Flag** | **Meaning** SOUND_FLAG_BEEPS 0x0001 | Playing a beep sound is supported. In Section 2.2.8.1.1.2.1, Basic (TS_SECURITY_HEADER), changed RDP_SEC_TRANSPORT_RSP to SEC_TRANSPORT_RSP in the flags field table. Changed from: ... flags (2 bytes): A 16-bit, unsigned integer that contains security flags.

Errata Published *	Description

Flag	Meaning
SEC_EXCHANGE_PKT 0x0001	Indicates that the packet is a Security Exchange PDU (section 2.2.1.10). This packet type is sent from client to server only. The client only sends this packet if it will be encrypting further communication and Standard RDP Security mechanisms (section 5.3) are in effect.
SEC_TRANSPORT_REQ 0x0002	Indicates that the packet is an Initiate Multitransport Request PDU (section 2.2.15.1). This flag MUST NOT be present if the PDU containing the security header is being sent from client to server. This flag MUST NOT be present if the PDU containing the security header is not being sent on the MCS message channel. The ID of the message channel is specified in the Server Message Channel Data (section 2.2.1.4.5).
RDP_SEC_TRANSPORT_RSP 0x0004	Indicates that the packet is an Initiate Multitransport Response PDU (section 2.2.15.2). This flag MUST NOT be present if the PDU containing the security header is being sent from server to client. This flag MUST NOT be present if the PDU containing the security header is not being sent on the MCS message channel. The ID of the message channel is specified in the Server Message Channel Data (section 2.2.1.4.5).
SEC_ENCRYPT 0x0008	Indicates that the packet is encrypted.

...

Changed to:

...

flags (2 bytes): A 16-bit, unsigned integer that contains security flags.

Flag	Meaning
SEC_EXCHANGE_PKT 0x0001	Indicates that the packet is a Security Exchange PDU (section 2.2.1.10). This packet type is sent from client to server only. The client only sends this packet if it will be encrypting further communication and Standard RDP Security mechanisms (section 5.3) are in effect.
SEC_TRANSPORT_REQ 0x0002	Indicates that the packet is an Initiate Multitransport Request PDU (section 2.2.15.1). This flag MUST NOT be present if the PDU containing the security header is being sent from client to server. This flag MUST NOT be present if the PDU containing the security header is not being sent on the MCS message channel. The ID of the message channel is specified in the Server Message Channel Data (section 2.2.1.4.5).
SEC_TRANSPORT_RSP 0x0004	Indicates that the packet is an Initiate Multitransport Response PDU (section 2.2.15.2). This flag MUST NOT be present if the PDU containing the security header is being sent from server to client. This flag MUST NOT be present if the PDU containing the security header is not being sent on the MCS message channel. The ID of the message channel is specified in the Server Message Channel Data (section 2.2.1.4.5).
SEC_ENCRYPT 0x0008	Indicates that the packet is encrypted.

...

In Section 2.2.9.1.2.1.3, Fast-Path Synchronize Update (TS_FP_UPDATE_SYNCHRONIZE), changed the referenced structure from TS_UPDATE_SYNCHRONIZE_PDU_DATA to TS_UPDATE_SYNC.

Changed from:

The TS_FP_UPDATE_SYNCHRONIZE structure is the fast-path variant of the

Errata Published *	Description
	TS_UPDATE_SYNCHRONIZE_PDU_DATA (section 2.2.9.1.1.3.1.3) structure. ... Changed to: The TS_FP_UPDATE_SYNCHRONIZE structure is the fast-path variant of the TS_UPDATE_SYNC (section 2.2.9.1.1.3.1.3) structure. ... In Section 3.2.5.10.2, Processing Early User Authorization Result PDU, changed the AUTHZ_ACCESS_DENIED hexidecimal notation from 0x0000052E to 0x00000005. Changed from: The structure and fields of the Early User Authorization Result PDU are specified in section 2.2.10.2. If the authorizationResult field is set to AUTHZ_ACCESS_DENIED (0x0000052E), the client SHOULD drop the connection as user authorization has failed and login to the remote session will not be possible. Changed to: The structure and fields of the Early User Authorization Result PDU are specified in section 2.2.10.2. If the authorizationResult field is set to AUTHZ_ACCESS_DENIED (0x00000005), the client SHOULD drop the connection as user authorization has failed and login to the remote session will not be possible. In Section 4.1.4, Server MCS Connect Response PDU with GCC Conference Create Response, added proprietary server certificates for dwVersion, dwSigAlgId, and dwKeyAlgId. Changed from: ... 01 00 00 00 -> PROPRIETARYSERVERCERTIFICATE::dwVersion = 1 01 00 00 00 -> PROPRIETARYSERVERCERTIFICATE::dwSigAlgId = MD5RSA (1) 01 00 00 00 -> PROPRIETARYSERVERCERTIFICATE::dwKeyAlgId = RSAKEY (1) ... Changed to: ... 01 00 00 00 -> PROPRIETARYSERVERCERTIFICATE::dwVersion = CERT_CHAIN_VERSION_1 (1) 01 00 00 00 -> PROPRIETARYSERVERCERTIFICATE::dwSigAlgId = SIGNATURE_ALG_RSA (1) 01 00 00 00 -> PROPRIETARYSERVERCERTIFICATE::dwKeyAlgId = KEY_EXCHANGE_ALG_RSA (1) ... In Section 4.1.12, Server Demand Active PDU, changed generalCompressionTypes to compressionTypes and generalCompressionLevel to compressionLevel. Also changed TS_VIRTUALCHANNEL_CAPABILITYSET::vccaps1 to TS_VIRTUALCHANNEL_CAPABILITYSET::flags. Changed from:

Errata Published *	Description
	00 00 -> TS_GENERAL_CAPABILITYSET::generalCompressionTypes = 0 ... 00 00 -> TS_GENERAL_CAPABILITYSET::generalCompressionLevel = 0 ... 02 00 00 00 -> TS_VIRTUALCHANNEL_CAPABILITYSET::vccaps1 = 0x00000002 = VCCAPS_COMPR_CS_8K ... Changed to: ... 00 00 -> TS_GENERAL_CAPABILITYSET::compressionTypes = 0 ... 00 00 -> TS_GENERAL_CAPABILITYSET::compressionLevel = 0 ... 02 00 00 00 -> TS_VIRTUALCHANNEL_CAPABILITYSET::flags = 0x00000002 = VCCAPS_COMPR_CS_8K In Section 4.1.13, Client Confirm Active PDU, changed generalCompressionTypes to compressionTypes and generalCompressionLevel to compressionLevel. Updated instances of TS_BITMAPCACHE_CAPABILITYSET_REV2::CellCacheInfo[x] to TS_BITMAPCACHE_CAPABILITYSET_REV2::BitmapCache(x)CellInfo. Also changed TS_VIRTUALCHANNEL_CAPABILITYSET::vccaps1 to TS_VIRTUALCHANNEL_CAPABILITYSET::flags. Changed from: 00 00 -> TS_GENERAL_CAPABILITYSET::generalCompressionTypes = 0 ... 00 00 -> TS_GENERAL_CAPABILITYSET::generalCompressionLevel = 0 ... 78 00 00 00 -> TS_BITMAPCACHE_CAPABILITYSET_REV2::CellCacheInfo[0] = 0x00000078 TS_BITMAPCACHE_CELL_CACHE_INFO::NumEntries = 0x78 = 120 TS_BITMAPCACHE_CELL_CACHE_INFO::k = FALSE 78 00 00 00 -> TS_BITMAPCACHE_CAPABILITYSET_REV2::CellCacheInfo[1] = 0x00000078 TS_BITMAPCACHE_CELL_CACHE_INFO::NumEntries = 0x78 = 120 TS_BITMAPCACHE_CELL_CACHE_INFO::k = FALSE fb 09 00 80 -> TS_BITMAPCACHE_CAPABILITYSET_REV2::CellCacheInfo[2] = 0x800009fb TS_BITMAPCACHE_CELL_CACHE_INFO::NumEntries = 0x9fb = 2555 TS_BITMAPCACHE_CELL_CACHE_INFO::k = TRUE 00 00 00 00 -> TS_BITMAPCACHE_CAPABILITYSET_REV2::CellCacheInfo[3] = 0x00000000

Errata Published *	Description
	00 00 00 00 -> TS_BITMAPCACHE_CAPABILITYSET_REV2::CellCacheInfo[4] = 0x00000000 ... 01 00 00 00 -> TS_VIRTUALCHANNEL_CAPABILITYSET::vccaps1 = 0x00000001 = VCCAPS_COMPR_SC Changed to: ... 00 00 -> TS_GENERAL_CAPABILITYSET::compressionTypes = 0 ... 00 00 -> TS_GENERAL_CAPABILITYSET::compressionLevel = 0 ... 78 00 00 00 -> TS_BITMAPCACHE_CAPABILITYSET_REV2::BitmapCache0CellInfo = 0x00000078 TS_BITMAPCACHE_CELL_CACHE_INFO::NumEntries = 0x78 = 120 TS_BITMAPCACHE_CELL_CACHE_INFO::k = FALSE 78 00 00 00 -> TS_BITMAPCACHE_CAPABILITYSET_REV2::BitmapCache1CellInfo = 0x00000078 TS_BITMAPCACHE_CELL_CACHE_INFO::NumEntries = 0x78 = 120 TS_BITMAPCACHE_CELL_CACHE_INFO::k = FALSE fb 09 00 80 -> TS_BITMAPCACHE_CAPABILITYSET_REV2::BitmapCache2CellInfo = 0x800009fb TS_BITMAPCACHE_CELL_CACHE_INFO::NumEntries = 0x9fb = 2555 TS_BITMAPCACHE_CELL_CACHE_INFO::k = TRUE 00 00 00 00 -> TS_BITMAPCACHE_CAPABILITYSET_REV2::BitmapCache3CellInfo = 0x00000000 00 00 00 00 -> TS_BITMAPCACHE_CAPABILITYSET_REV2::BitmapCache4CellInfo = 0x00000000 ... 01 00 00 00 -> TS_VIRTUALCHANNEL_CAPABILITYSET::flags = 0x00000001 = VCCAPS_COMPR_SC In Section 4.1.14, Client Synchronize PDU, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 ... Changed to: ...

Errata Published *	Description
	00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 ... In Section 4.1.15, Client Control PDU - Cooperate, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 ... Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 ... In Section 4.1.16, Client Control PDU - Request Control, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 ... Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 ... In Section 4.1.17, Client Persistent Key List PDU, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed TS_BITMAPCACHE_PERSISTENT_LIST::numEntries[x] to TS_BITMAPCACHE_PERSISTENT_LIST_PDU::numEntriesCache(x) and TS_BITMAPCACHE_PERSISTENT_LIST::totalEntries[x] to TS_BITMAPCACHE_PERSISTENT_LIST_PDU::totalEntriesCache(x). Also changed TS_BITMAPCACHE_PERSISTENT_LIST to TS_BITMAPCACHE_PERSISTENT_LIST_PDU.

Errata Published *	Description
	Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::numEntries[0] = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::numEntries[1] = 0 19 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::numEntries[2] = 0x19 = 25 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::numEntries[3] = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::numEntries[4] = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::totalEntries[0] = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::totalEntries[1] = 0 19 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::totalEntries[2] = 0x19 = 25 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::totalEntries[3] = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::totalEntries[4] = 0 03 -> TS_BITMAPCACHE_PERSISTENT_LIST::bBitMask = 0x03 0x03 = 0x01 | 0x02 = PERSIST_FIRST_PDU | PERSIST_LAST_PDU 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::Pad2 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST::Pad3 TS_BITMAPCACHE_PERSISTENT_LIST::entries: ... Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::numEntriesCache0 = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::numEntriesCache1 = 0 19 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::numEntriesCache2 = 0x19 = 25 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::numEntriesCache3 = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::numEntriesCache4 = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::totalEntriesCache0 = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::totalEntriesCache1 = 0 19 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::totalEntriesCache2 = 0x19 = 25 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::totalEntriesCache3 = 0 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::totalEntriesCache4 = 0

Errata Published *	Description
	03 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::bBitMask = 0x03 0x03 = 0x01 | 0x02 = PERSIST_FIRST_PDU | PERSIST_LAST_PDU 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::Pad2 00 00 -> TS_BITMAPCACHE_PERSISTENT_LIST_PDU::Pad3 TS_BITMAPCACHE_PERSISTENT_LIST_PDU::entries: ... In Section 4.1.18, Client Font List PDU, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 ... Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 ... In Section 4.1.19, Server Synchronize PDU, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 ... Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 ...

Errata Published *	Description
	In Section 4.1.20, Server Control PDU - Cooperate, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 ... Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 ... In Section 4.1.21, Server Control PDU - Granted Control, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 ... Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 ... In Section 4.1.22, Server Font Map PDU, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Also changed instances of TS_FONT_MAP_PDU_DATA to TS_FONT_MAP_PDU. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 00 00 -> TS_FONT_MAP_PDU_DATA::numberEntries = 0 00 00 -> TS_FONT_MAP_PDU_DATA::totalNumEntries = 0

Errata Published *	Description
	03 00 -> TS_FONT_MAP_PDU_DATA::mapFlags = 0x0003 0x0003 = 0x0002 | 0x0001 = FONTMAP_LAST | FONTMAP_FIRST 04 00 -> TS_FONT_MAP_PDU_DATA::entrySize = 4 bytes ... Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 00 00 -> TS_FONT_MAP_PDU::numberEntries = 0 00 00 -> TS_FONT_MAP_PDU::totalNumEntries = 0 03 00 -> TS_FONT_MAP_PDU::mapFlags = 0x0003 0x0003 = 0x0002 | 0x0001 = FONTMAP_LAST | FONTMAP_FIRST 04 00 -> TS_FONT_MAP_PDU::entrySize = 4 bytes ... In Section 4.2.1, Client Shutdown Request PDU, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 In Section 4.2.2, Server Shutdown Request Denied PDU, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed from: 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0

Errata Published *	Description
	00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 In Section 4.3.1, Logon Info Version 2, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 ... Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 ... In Section 4.3.2, Plain Notify, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to TS_SHAREDATAHEADER::compressedLength. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 ... Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 ... In Section 4.3.3, Logon Info Extended, changed TS_SHAREDATAHEADER::generalCompressedType to TS_SHAREDATAHEADER::compressedType and TS_SHAREDATAHEADER::generalCompressedLength to

Errata Published *	Description
	TS_SHAREDATAHEADER::compressedLength. Changed from: ... 00 -> TS_SHAREDATAHEADER::generalCompressedType = 0 00 00 -> TS_SHAREDATAHEADER::generalCompressedLength = 0 ... Changed to: ... 00 -> TS_SHAREDATAHEADER::compressedType = 0 00 00 -> TS_SHAREDATAHEADER::compressedLength = 0 ... Also in this document, numerous editorial fixes have also been made, e.g., changed instances of "Id" to "ID" such as shareId to shareID, originatorId to originatorID, streamId to streamID, nodeId to nodeID, and MCSChannelId to MCSChannelID; made minor updates to section titles; and changed various name conventions such as totalEntries(x) to totalEntriesCache(x), Requested protocols to requestedProtocols, Selected protocols to selectedProtocol, channel(x) to channelIdArray[x], TIME_ZONE_INFORMATION to TS_TIME_ZONE_INFORMATION, "Cache 2, Key 0, Low 32-bits" to "Low 32-bits of Cache 2, Key 0" or "Cache 2, Key 0, High 32-bits to "High 32-bits of Cache 2, Key 0". Sections updated: 2.2.1.13.1.1 2.2.1.13.2.1 2.2.1.17.1 2.2.3.1.1 2.2.5.1.1 2.2.7.2.4 2.2.8.1.1.1.2 2.2.9.2.1.1 2.2.14.1.5 3.2.1.8 3.2.5.3.13.1 4.1.1 4.1.2 4.1.3 4.1.4 4.1.10 4.1.12 4.1.14 4.1.15 4.1.16 4.1.17 4.1.18 4.1.19

Errata Published *	Description
	4.1.20 4.1.21 4.1.22 4.2.1 4.2.2 4.3.1 4.3.2 4.3.3
2019/10/28	In Section 3.2.5.3.4, Processing MCS Connect Response PDU with GCC Conference Create Response, changed MCS Response Initial PDU to MCS Connect Response PDU when referring to the figure that gives a basic high-level overview of the nested structure for the MCS Connect Response PDU. Changed from: The structure and fields of the MCS Connect Response PDU with GCC Conference Create Response are specified in section 2.2.1.4. A basic high-level overview of the nested structure for the MCS Connect Response PDU is illustrated in section 1.3.1.1, in the figure specifying MCS Response Initial PDU. ... Changed to: The structure and fields of the MCS Connect Response PDU with GCC Conference Create Response are specified in section 2.2.1.4. A basic high-level overview of the nested structure for the MCS Connect Response PDU is illustrated in section 1.3.1.1, in the figure specifying the MCS Connect Response PDU....

*Date format: YYYY/MM/DD

[MS-RDPEA]: Remote Desktop Protocol: Audio Output Virtual Channel Extension

This topic lists the Errata found in [MS-RDPEA] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-RDPEAR]: Remote Desktop Protocol Authentication Redirection Virtual Channel

This topic lists the Errata found in [MS-RDPEAR] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 15, 2017 - Download

[MS-RDPECLIP]: Remote Desktop Protocol: Clipboard Virtual Channel Extension

This topic lists the Errata found in [MS-RDPECLIP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

September 12, 2018 - Download

[MS-RDPECAM]: Remote Desktop Protocol: Video Capture Virtual Channel Extension

This topic lists the Errata found in [MS-RDPECAM] 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 V1.0 – 2018/09/12.

Errata Published*	Description

2019/02/19	In Section 4.6.2, Property List Response, an annotated dump of a Property List Response (section 2.2.3.17) has been added.
	
	Added:
	The following is an annotated dump of a Property List Response (section 2.2.3.17).
	
	`00000000  02 15 01 02 03 00 00 00 00 fa 00 00 00 05 00 00`
	`.....`
	`00000010  00 00 00 00 00 02 02 01 00 00 00 00 ff 00 00 00`
	`.....`
	`00000020  01 00 00 00 80 00 00 00  ........`
	`02->SHARED_MSG_HEADER::Version = 2`
	`15->SHARED_MSG_HEADER::MessageId = PropertyListResponse(21)`
	`01->PropertyDescription[0]::PropertySet = CameraControl(1)`
	`02->PropertyDescription[0]::PropertyId = Focus(2)`
	`03->PropertyDescription[0]::Capabilities = Manual and Auto(1 + 2)`
	`00 00 00 00->PropertyDescription[0]::MinValue = 0`
	`fa 00 00 00->PropertyDescription[0]::MaxValue = 250`
	`05 00 00 00->PropertyDescription[0]::Step = 5`
	`00 00 00 00->PropertyDescription[0]::DefaultValue = 0`
	`02->PropertyDescription[1]::PropertySet = VideoProcAmp(2)`
	`02->PropertyDescription[1]::PropertyId = Brightness(2)`
	`01->PropertyDescription[1]::Capabilities = Manual(1)`
	`00 00 00 00->PropertyDescription[1]::MinValue = 0`
	`ff 00 00 00->PropertyDescription[1]::MaxValue = 255`
	`01 00 00 00->PropertyDescription[1]::Step = 1`
	`80 00 00 00->PropertyDescription[1]::DefaultValue = 128`

2019/02/19	In Section 2.2.1, Shared Message Header (SHARED_MSG_HEADER), updated values to hexadecimal format for consistency in the MessageId field table. Changed from: ... MessageId (1 byte): An 8-bit unsigned integer that specifies the type of the message. Value | Meaning SuccessResponse 1 | A Success Response (section 2.2.3.1) message.

Errata Published*	Description

ErrorResponse 2	An Error Response (section 2.2.3.2) message.
SelectVersionRequest 3	A Select Version Request (section 2.2.2.1) message.
SelectVersionResponse 4	A Select Version Response (section 2.2.2.2) message.
DeviceAddedNotification 5	A Device Added Notification (section 2.2.2.3) message.
DeviceRemovedNotification 6	A Device Removed Notification (section 2.2.2.4) message.
ActivateDeviceRequest 7	An Activate Device Request (section 2.2.3.3) message.
DeactivateDeviceRequest 8	A Deactivate Device Request (section 2.2.3.4) message.
StreamListRequest 9	A Stream List Request (section 2.2.3.5) message.
StreamListResponse 10	A Stream List Response (section 2.2.3.6) message.
MediaTypeListRequest 11	A Media Type List Request (section 2.2.3.7) message.
MediaTypeListResponse 12	A Media Type List Response (section 2.2.3.8) message.
CurrentMediaTypeRequest 13	A Current Media Type Request (section 2.2.3.9) message.
CurrentMediaTypeResponse 14	A Current Media Type Response (section 2.2.3.10) message.
StartStreamsRequest 15	A Start Streams Request (section 2.2.3.11) message.
StopStreamsRequest 16	A Stop Streams Request (section 2.2.3.12) message.
SampleRequest 17	A Sample Request (section 2.2.3.13) message.
SampleResponse 18	A Sample Response (section 2.2.3.14) message.
SampleErrorResponse 19	A Sample Error Response (section 2.2.3.15) message.
PropertyListRequest 20	A Property List Request (section 2.2.3.16) message. This message is supported only by version 2 of the protocol.
PropertyListResponse 21	A Property List Response (section 2.2.3.17) message. This message is supported only by version 2 of the

Errata Published*	Description

	protocol.
PropertyValueRequest 22	A Property Value Request (section 2.2.3.18) message. This message is supported only by version 2 of the protocol.
PropertyValueResponse 23	A Property Value Response (section 2.2.3.19) message. This message is supported only by version 2 of the protocol.
SetPropertyValueRequest 24	A Set Property Value Request (section 2.2.3.20) message. This message is supported only by version 2 of the protocol.

Changed to:

...

MessageId (1 byte): An 8-bit unsigned integer that specifies the type of the message.

Value	Meaning
SuccessResponse 0x01	A Success Response (section 2.2.3.1) message.
ErrorResponse 0x02	An Error Response (section 2.2.3.2) message.
SelectVersionRequest 0x03	A Select Version Request (section 2.2.2.1) message.
SelectVersionResponse 0x04	A Select Version Response (section 2.2.2.2) message.
DeviceAddedNotification 0x05	A Device Added Notification (section 2.2.2.3) message.
DeviceRemovedNotification 0x06	A Device Removed Notification (section 2.2.2.4) message.
ActivateDeviceRequest 0x07	An Activate Device Request (section 2.2.3.3) message.
DeactivateDeviceRequest 0x08	A Deactivate Device Request (section 2.2.3.4) message.
StreamListRequest 0x09	A Stream List Request (section 2.2.3.5) message.
StreamListResponse 0x0A	A Stream List Response (section 2.2.3.6) message.
MediaTypeListRequest 0x0B	A Media Type List Request (section 2.2.3.7) message.
MediaTypeListResponse 0x0C	A Media Type List Response (section 2.2.3.8) message.
CurrentMediaTypeRequest 0x0D	A Current Media Type Request

Errata Published*	Description

	(section 2.2.3.9) message.
CurrentMediaTypeResponse 0x0E	A Current Media Type Response (section 2.2.3.10) message.
StartStreamsRequest 0x0F	A Start Streams Request (section 2.2.3.11) message.
StopStreamsRequest 0x10	A Stop Streams Request (section 2.2.3.12) message.
SampleRequest 0x11	A Sample Request (section 2.2.3.13) message.
SampleResponse 0x12	A Sample Response (section 2.2.3.14) message.
SampleErrorResponse 0x13	A Sample Error Response (section 2.2.3.15) message.
PropertyListRequest 0x14	A Property List Request (section 2.2.3.16) message. This message is supported only by version 2 of the protocol.
PropertyListResponse 0x15	A Property List Response (section 2.2.3.17) message. This message is supported only by version 2 of the protocol.
PropertyValueRequest 0x16	A Property Value Request (section 2.2.3.18) message. This message is supported only by version 2 of the protocol.
PropertyValueResponse 0x17	A Property Value Response (section 2.2.3.19) message. This message is supported only by version 2 of the protocol.
SetPropertyValueRequest 0x18	A Set Property Value Request (section 2.2.3.20) message. This message is supported only by version 2 of the protocol.

In Section 2.2.3.2, Error Response, updated values to hexadecimal format for consistency in the ErrorCode field table.

Changed from:

...

ErrorCode (4 bytes): A 32-bit unsigned integer containing an error code.

Value	Meaning
UnexpectedError 1	An unexpected error occurred.
InvalidMessage 2	An invalid message was received. Either the message is malformed, or

Errata Published*	Description

	the protocol version or message type is unexpected.
NotInitialized 3	The object MUST be initialized before the requested operation can be carried out. This error could be returned, for example, when attempting to communicate with a deactivated camera device.
InvalidRequest 4	The request is invalid in the current state.
InvalidStreamNumber 5	The provided stream number was invalid.
InvalidMediaType 6	The data specified for the stream format is invalid, inconsistent, or not supported.
OutOfMemory 7	The client ran out of memory.
ItemNotFound 8	The device does not support the requested property. This error code is generated only by version 2 of the protocol.
SetNotFound 9	The device does not support the requested property set. This error code is generated only by version 2 of the protocol.
OperationNotSupported 10	The requested operation is not supported. This error code is generated only by version 2 of the protocol.

Changed to:

...

ErrorCode (4 bytes): A 32-bit unsigned integer containing an error code.

Value	Meaning
UnexpectedError 0x00000001	An unexpected error occurred.
InvalidMessage 0x00000002	An invalid message was received. Either the message is malformed, or the protocol version or message type is unexpected.
NotInitialized 0x00000003	The object MUST be initialized before the requested operation can be carried out. This error could be returned, for example, when attempting to communicate with a deactivated camera device.
InvalidRequest 0x00000004	The request is invalid in the current state.
InvalidStreamNumber 0x00000005	The provided stream number was

Errata Published*	Description

	invalid.
InvalidMediaType 0x00000006	The data specified for the stream format is invalid, inconsistent, or not supported.
OutOfMemory 0x00000007	The client ran out of memory.
ItemNotFound 0x00000008	The device does not support the requested property. This error code is generated only by version 2 of the protocol.
SetNotFound 0x00000009	The device does not support the requested property set. This error code is generated only by version 2 of the protocol.
OperationNotSupported 0x0000000A	The requested operation is not supported. This error code is generated only by version 2 of the protocol.

In Section 2.2.3.6.1, STREAM_DESCRIPTION, updated the value to hexadecimal format for consistency in the StreamCategory field table.

Changed from:

...

StreamCategory (1 byte): An 8-bit unsigned integer that specifies the category of the stream.

Value	Meaning
Capture 1	Capture category streams provide a stream of compressed or uncompressed digital video.

Changed to:

...

StreamCategory (1 byte): An 8-bit unsigned integer that specifies the category of the stream.

Value	Meaning
Capture 0x01	Capture category streams provide a stream of compressed or uncompressed digital video.

In Section 2.2.3.8.1, MEDIA_TYPE_DESCRIPTION, updated values to hexadecimal format for consistency in the Format field table.

Changed from:

...

Format (1 byte): An 8-bit unsigned integer that specifies the stream codec.

Errata Published*	Description

Value	Meaning
H264 1	H.264 video as described in [ITU-H.264-201704]. Media samples contain H.264 bitstream data with start codes and interleaved sequence parameter set/picture parameter set (SPS/PPS) packets. Each sample contains one complete picture, either one field or one frame.
MJPG 2	Motion JPEG. Motion JPEG is a video compression format in which each video frame of a digital video sequence is independently compressed as a JPEG image.
YUY2 3	YUY2 video as specified in [MSDN-YUVFormats].
NV12 4	NV12 video as described in [MSDN-YUVFormats].
I420 5	I420 video. Identical to YV12 as described in [MSDN-YUVFormats] except that the order of the U and V planes is reversed.
RGB24 6	RGB, 24 bits per pixel.
RGB32 7	RGB, 32 bits per pixel.

...

Changed to:

...

Format (1 byte): An 8-bit unsigned integer that specifies the stream codec.

Value	Meaning
H264 0x01	H.264 video as described in [ITU-H.264-201704]. Media samples contain H.264 bitstream data with start codes and interleaved sequence parameter set/picture parameter set (SPS/PPS) packets. Each sample contains one complete picture, either one field or one frame.
MJPG 0x02	Motion JPEG. Motion JPEG is a video compression format in which each video frame of a digital video sequence is independently compressed as a JPEG image.
YUY2 0x03	YUY2 video as specified in [MSDN-YUVFormats].
NV12 0x04	NV12 video as described in [MSDN-YUVFormats].

Errata Published*	Description

I420 0x05	I420 video. Identical to YV12 as described in [MSDN-YUVFormats] except that the order of the U and V planes is reversed.
RGB24 0x06	RGB, 24 bits per pixel.
RGB32 0x07	RGB, 32 bits per pixel.

...

In Section 2.2.3.17.1, PROPERTY_DESCRIPTION, updated values to hexadecimal format for consistency in the PropertySet and PropertyId field tables.

Changed from:

...

PropertySet (1 byte): An 8-bit unsigned integer that specifies the property set.

Value	Meaning
CameraControl 1	This property set category controls camera device settings.
VideoProcAmp 2	This property set controls devices that can adjust the image color attributes of analog or digital signals.

PropertyId (1 byte): An 8-bit unsigned integer that contains the identifier of the property within the property set specified by the PropertySet field.

CameraControl properties:

Value	Meaning
Exposure 1	This property controls the exposure time of the device.
Focus 2	This property controls the focus setting of the device.
Pan 3	This property controls the pan setting of the device.
Roll 4	This property controls the roll setting of the device.
Tilt 5	This property controls the tilt setting of the device.
Zoom 6	This property controls the zoom setting of the device.

VideoProcAmp properties:

Value	Meaning
BacklightCompensation 1	This property controls the backlight

Errata Published*	Description

	compensation setting of the device. This value MUST be either 0 or 1. The value 0 indicates that backlight compensation is disabled. The value 1 indicates that backlight compensation is enabled.
Brightness 2	This property controls the brightness setting of the device.
Contrast 3	This property controls the contrast setting of the device.
Hue 4	This property controls the hue setting of the device.
WhiteBalance 5	This property controls the white balance setting of the device.

...

Changed to:

...

PropertySet (1 byte): An 8-bit unsigned integer that specifies the property set.

Value	Meaning
CameraControl 0x01	This property set category controls camera device settings.
VideoProcAmp 0x02	This property set controls devices that can adjust the image color attributes of analog or digital signals.

PropertyId (1 byte): An 8-bit unsigned integer that contains the identifier of the property within the property set specified by the PropertySet field.

CameraControl properties:

Value	Meaning
Exposure 0x01	This property controls the exposure time of the device.
Focus 0x02	This property controls the focus setting of the device.
Pan 0x03	This property controls the pan setting of the device.
Roll 0x04	This property controls the roll setting of the device.
Tilt 0x05	This property controls the tilt setting of the device.
Zoom 0x06	This property controls the zoom setting of the device.

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>VideoProcAmp properties:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>BacklightCompensation 0x01</td><td>This property controls the backlight compensation setting of the device. This value MUST be either 0 or 1. The value 0 indicates that backlight compensation is disabled. The value 1 indicates that backlight compensation is enabled.</td></tr>
<tr><td>Brightness 0x02</td><td>This property controls the brightness setting of the device.</td></tr>
<tr><td>Contrast 0x03</td><td>This property controls the contrast setting of the device.</td></tr>
<tr><td>Hue 0x04</td><td>This property controls the hue setting of the device.</td></tr>
<tr><td>WhiteBalance 0x05</td><td>This property controls the white balance setting of the device.</td></tr>
</table>
...

In Section 2.2.3.19.1, PROPERTY_VALUE, updated values to hexadecimal format for consistency in the Mode field table.

Changed from:
...
Mode (1 byte): An 8-bit unsigned integer that specifies how the property was set.
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>Manual 1</td><td>The value was set manually.</td></tr>
<tr><td>Auto 2</td><td>The value was set automatically.</td></tr>
</table>
...

Changed to:
...
Mode (1 byte): An 8-bit unsigned integer that specifies how the property was set.
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>Manual 0x01</td><td>The value was set manually.</td></tr>
<tr><td>Auto 0x02</td><td>The value was set automatically.</td></tr>
</table>
...</td></tr>
</table>

*Date format: YYYY/MM/DD

[MS-RDPEDISP]: Remote Desktop Protocol: Display Update Virtual Channel Extension

This topic lists the Errata found in the MS-RDPEDISP document 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 V7.0 – 2018/09/12.

Errata Published*	Description
2019/08/19	In Section 1, Introduction, changed the source of the display configuration changes from server to client. Changed from: This document specifies the Remote Desktop Protocol: Display Control Channel Extension to the Remote Desktop Protocol: Basic Connectivity and Graphics Remoting, as specified in [MS-RDPBCGR] sections 1 to 5. This control protocol is used by the server to request display configuration changes in a remote session. Display configuration changes include the addition, removal and repositioning of monitors, resolution updates, and orientation updates. Changed to: This document specifies the Remote Desktop Protocol: Display Control Channel Extension to the Remote Desktop Protocol: Basic Connectivity and Graphics Remoting, as specified in [MS-RDPBCGR] sections 1 to 5. This control protocol is used by the client to request display configuration changes in a remote session. Display configuration changes include the addition, removal and repositioning of monitors, resolution updates, and orientation updates.

*Date format: YYYY/MM/DD

[MS-RDPEDYC]: Remote Desktop Protocol: Dynamic Channel Virtual Channel Extension

This topic lists the Errata found in [MS-RDPEDYC] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

June 1, 2017 - Download

December 1, 2017 - Download

September 12, 2018 - Download

[MS-RDPEFS]: Remote Desktop Protocol: File System Virtual Channel Extension

This topic lists the Errata found in [MS-RDPEFS] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

June 1, 2017 - Download

September 15, 2017 - Download

December 1, 2017 - Download

[MS-RDPEGDI]: Remote Desktop Protocol: Graphics Device Interface (GDI) Acceleration Extensions

This topic lists the Errata found in [MS-RDPEGDI] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

[MS-RDPEGFX]: Remote Desktop Protocol: Graphics Pipeline Extension

This topic lists the Errata found in [MS-RDPEGFX] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

March 20, 2017 - Download

September 15, 2017 - Download

September 12, 2018 - Download

Errata below are for Protocol Document Version V14.0 – 2018/09/12.

Errata Published*	Description
2019/02/19	In Section 2.2.4.5, RFX_AVC444_BITMAP_STREAM, "YUV420 frame" in the cbAvc420EncodedBitstream1 field description has been replaced with "luma frame". Changed from: ... cbAvc420EncodedBitstream1 (30 bits): A 30-bit unsigned integer that specifies the size, in bytes, of the luma frame present in the avc420EncodedBitstream1 field. If no YUV420 frame is present, then this field MUST be set to zero. ... Changed to: ... cbAvc420EncodedBitstream1 (30 bits): A 30-bit unsigned integer that specifies the size, in bytes, of the YUV420 frame present in the avc420EncodedBitstream1 field. If no YUV420 frame is present, then this field MUST be set to zero. ... In Section 2.2.4.6, RFX_AVC444V2_BITMAP_STREAM, "YUV420 frame" in the cbAvc420EncodedBitstream1 field description has been replaced with "luma frame". Changed from: ... cbAvc420EncodedBitstream1 (30 bits): A 30-bit unsigned integer that specifies the size, in bytes, of the luma frame present in the avc420EncodedBitstream1 field. If no YUV420 frame is present, then this field MUST be set to zero. ...

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>Changed to:
...
cbAvc420EncodedBitstream1 (30 bits): A 30-bit unsigned integer that specifies the size, in bytes, of the YUV420 frame present in the avc420EncodedBitstream1 field. If no YUV420 frame is present, then this field MUST be set to zero.
...</td></tr>
<tr><td>2018/12/10</td><td>In Section 2.2.1.6, RDPGFX_CAPSET, the RDPGFX_CAPVERSION_106 value has been changed from 0x000A0601 to 0x000A0600 in the version field description.

Changed from:
...
version (4 bytes): A 32-bit unsigned integer that specifies the version of the capability set.

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>...</td><td>…</td></tr>
<tr><td>RDPGFX_CAPVERSION_106
0x000A0601</td><td>RDPGFX_CAPSET_VERSION106
(section 2.2.3.9)</td></tr>
</table>

Changed to:
...
version (4 bytes): A 32-bit unsigned integer that specifies the version of the capability set.

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>...</td><td>…</td></tr>
<tr><td>RDPGFX_CAPVERSION_106
0x000A0600</td><td>RDPGFX_CAPSET_VERSION106
(section 2.2.3.9)</td></tr>
</table>

In Section 2.2.3.9, RDPGFX_CAPSET_VERSION106, the RDPGFX_CAPVERSION_106 value has been changed from 0x000A0601 to 0x000A0600 in the version field description.

Changed from:
...
version (4 bytes): A 32-bit unsigned integer that specifies the version of the capability set. This field MUST be set to RDPGFX_CAPVERSION_106 (0x000A0601).
...

Changed to:
...
version (4 bytes): A 32-bit unsigned integer that specifies the version of the capability set. This field MUST be set to RDPGFX_CAPVERSION_106 (0x000A0600).
...</td></tr>
</table>

*Date format: YYYY/MM/DD

[MS-RDPEGT]: Remote Desktop Protocol Geometry Tracking Virtual Channel Protocol Extension

This topic lists the Errata found in [MS-RDPEGFT] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-RDPEI]: Remote Desktop Protocol: Input Virtual Channel Extension

This topic lists the Errata found in [MS-RDPEI] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-RDPELE]: Remote Desktop Protocol: Licensing Extension

This topic lists the Errata found in [MS-RDPELE] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 12, 2018 - Download

Errata below are for Protocol Document Version V14.0 – 2018/09/12.

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td>2020/01/06</td><td>
In Section 2.2.2.2, Client New License Request (CLIENT_NEW_LICENSE_REQUEST), modified version range for CLIENT_OS_ID_WINNT_POST_52 value and added new version value, CLIENT_OS_ID_WINNT_POST_100 in product behavior note <9>.

Changed from:
...

PlatformId (4 bytes): A 32-bit unsigned integer. This field is composed of two identifiers: the operating system identifier and the independent software vendor (ISV) identifier. The platform ID is composed of the logical OR of these two values.
The most significant byte of the PlatformId field contains the operating system version of the client.<9>
...
<9> Section 2.2.2.2: On Windows platforms, it is relative to the major version number of the operating system and has the following values.
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>CLIENT_OS_ID_WINNT_351
0x01000000</td><td>The client operating system version is 3.51.</td></tr>
<tr><td>CLIENT_OS_ID_WINNT_40
0x02000000</td><td>The client operating system version is 4.00.</td></tr>
<tr><td>CLIENT_OS_ID_WINNT_50
0x03000000</td><td>The client operating system version is 5.00.</td></tr>
<tr><td>CLIENT_OS_ID_WINNT_POST_52
0x04000000</td><td>The client operating system version is 5.20 or later.</td></tr>
</table>

Changed to:
...

PlatformId (4 bytes): A 32-bit unsigned integer. This field is composed of two identifiers: the operating system identifier and the independent software vendor (ISV) identifier. The platform
</td></tr>
</table>

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>ID is composed of the logical OR of these two values.
The most significant byte of the PlatformId field contains the operating system version of the client.<9>...

<9> Section 2.2.2.2: On Windows platforms, it is relative to the major version number of the operating system and has the following values.

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>CLIENT_OS_ID_WINNT_351
0x01000000</td><td>The client operating system version is 3.51.</td></tr>
<tr><td>CLIENT_OS_ID_WINNT_40
0x02000000</td><td>The client operating system version is 4.0.</td></tr>
<tr><td>CLIENT_OS_ID_WINNT_50
0x03000000</td><td>The client operating system version is 5.0.</td></tr>
<tr><td>CLIENT_OS_ID_WINNT_POST_52
0x04000000</td><td>The client operating system version is between 5.2 and 6.3 (inclusive).</td></tr>
<tr><td>CLIENT_OS_ID_WINNT_POST_100
0x08000000</td><td>The client operating system version is 10.0 or later.</td></tr>
</table>
</td></tr>
<tr><td>2020/01/06</td><td>Moved what was formerly section 2.2.2.7.1, License Error Message (LICENSE_ERROR_MESSAGE), to newly created section 2.2.2.8.

Changed from:
2.2.2.7.1 License Error Message (LICENSE_ERROR_MESSAGE)

The license error message specified in [MS-RDPBCGR] section 2.2.1.12.1.3 can be used by both client and server.

If the client supports extended error, the terminal server includes information relevant to the error code in the bbErrorInfo field of the Licensing Error Message. For more details, see [MS-RDPBCGR] section 2.2.1.12.1.3.

Changed to:
2.2.2.8 License Error Message (LICENSE_ERROR_MESSAGE)
The license error message specified in [MS-RDPBCGR] section 2.2.1.12.1.3 can be used by both client and server.

If the client supports extended error, the terminal server includes information relevant to the error code in the bbErrorInfo field of the Licensing Error Message. For more details, see [MS-RDPBCGR] section 2.2.1.12.1.3.

In this document, numerous editorial fixes have also been made, e.g., removed links from instances of "license error message" or "License Error message" and then referenced newly created section 2.2.2.8.

Sections updated:</td></tr>
</table>

Errata Published*	Description
	2.2.2 2.2.2.1.1 2.2.2.8 3.1.5.1 3.1.5.2 3.1.5.3 3.2.5.3 3.2.5.5 3.2.5.8 3.2.5.9 3.3.5.13.3.5.9 In Section 6, Appendix A: Product Behavior, product behavior note <16> was updated to include a reference to the new section 2.2.2.8.
2018/12/10	In Section 4, Protocol Examples, text has been added to clarify that the sample protocol packets are only examples and should not be considered to have been generated as part of the same protocol run. Changed from: For a complete listing of RDP headers, see [MS-RDPBCGR] section 4. Changed to: For a complete listing of RDP headers, see [MS-RDPBCGR] section 4. The sample protocol packets listed in sections 4.1 through to 4.7 are provided as examples and should not be considered to have been generated as part of the same protocol run.
2018/12/10	In Section 2.2.2.5, Client Platform Challenge Response (CLIENT_PLATFORM_CHALLENGE_RESPONSE), text has been added to clarify that decrypted Client Hardware Identification should follow the Platform Challenge Response Data in the MACData field description. Changed from: MACData (16 bytes): An array of 16 bytes containing an MD5 digest (MAC) generated over the decrypted Client Hardware Identification and Platform Challenge Response Data. ... Changed to: MACData (16 bytes): An array of 16 bytes containing an MD5 digest (MAC) generated over the Platform Challenge Response Data and decrypted Client Hardware Identification. ...

*Date format: YYYY/MM/DD

[MS-RDPEMC]: Remote Desktop Protocol: Multiparty Virtual Channel Extension

This topic lists the Errata found in [MS-RDPEMC] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-RDPEMT]: Remote Desktop Protocol: Multitransport Extension

This topic lists the Errata found in [MS-RDPEMT] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

March 2, 2016 - Download

Errata below are for Protocol Document Version V10.0 – 2018/09/12.

Errata Published*	Description
2019/03/18	In Section 1.3, Overview, clarified that a port number is not specified in an Initiate Multitransport Request PDU. Changed from: The Initiate Multitransport Request PDU contains information that uniquely identifies the multitransport connection; it contains a request ID and a cookie, a protocol identifier that identifies the type of multitransport connection that the client attempts to establish, and a port number that identifies the port on which the server is listening. When the client receives the Initiate Multitransport Request PDU, it attempts to establish a secure multitransport connection with the server. Changed to: The Initiate Multitransport Request PDU contains information that uniquely identifies the multitransport connection; it contains a request ID, a cookie, and a protocol identifier that identifies the type of multitransport connection that the client attempts to establish. When the client receives the Initiate Multitransport Request PDU, it attempts to establish a secure multitransport connection with the server.

*Date format: YYYY/MM/DD

[MS-RDPEPC]: Remote Desktop Protocol: Print Virtual Channel Extension

This topic lists the Errata found in [MS-RDPEPC] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

June 1, 2017 - Download

Errata below are for Protocol Document Version V10.0 – 2018/09/12.

Errata Published*	Description
2019/07/08	In Section 2.2.2.1, Client Device List Announce Request (DR_PRN_DEVICE_ANNOUNCE), added the section number that describes XPS mode to the RDPDR_PRINTER_ANNOUNCE_FLAG_XPSFORMAT value meaning in the Flags field table. Changed from: ... Flags (4 bytes): A 32-bit unsigned integer that indicates the properties of the client printer queue. This bit field MUST be a valid combination of any of the following values. Value / Meaning ... / ,,, RDPDR_PRINTER_ANNOUNCE_FLAG_XPSFORMAT 0x00000010 / This client/printer supports XML Paper Specification (XPS) format. ... / ,,, Changed to: ... Flags (4 bytes): A 32-bit unsigned integer that indicates the properties of the client printer queue. This bit field MUST be a valid combination of any of the following values. Value / Meaning ... / … RDPDR_PRINTER_ANNOUNCE_FLAG_XPSFORMAT 0x00000010 / This client/printer supports XML Paper Specification (XPS) format (section 3.1.1.2).

Errata Published*	Description
	... | ...
	In Section 2.2.2.2, Server Printer Set XPS Mode (DR_PRN_USING_XPS), added that the DR_PRN_USING_XPS message indicates to the client that future printer write request messages will use the XPS format. Changed from: This message is sent from server to client to set the device in XPS mode (see section 3.1.1.2). ... Changed to: This message is sent from server to client to set the device in XPS mode (see section 3.1.1.2) and indicate to the client that future Printer Write Request (section 2.2.2.9) messages will use the XPS format. ... In Section 3.1.1.2, XPS Mode, added the section number that describes the server behavior if it chooses to use the XPS format. Changed from: ... The server MUST notify the client with the message DR_PRN_USING_XPS (section 2.2.2.2) if it chooses to use the XPS format. ... Changed to: ... The server MUST notify the client with the DR_PRN_USING_XPS (section 2.2.2.2) message as described in section 3.3.5.1.2 if it chooses to use the XPS format. ... In Section 3.3.5.1.2, Sending a Printer Set XPS Mode Message, clarified the server Printer Write Request message section number. Changed from: ... If the server chooses to send print data in XPS format, the server MUST send this message to the client prior to sending any data in the write request messages (section 2.2.2.1). Changed to: ... If the server chooses to send print data in XPS format, the server MUST send this message to the client prior to sending any data in the Printer Write Request (section 2.2.2.9) message.

*Date format: YYYY/MM/DD

[MS-RDPEPNP]: Remote Desktop Protocol: Plug and Play Devices Virtual Channel Extension

This topic lists the Errata found in [MS-RDPEPNP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-RDPERP]: Remote Desktop Protocol: Remote Programs Virtual Channel Extension

This topic lists the Errata found in [MS-RDPERP] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

March 2, 2016 - Download

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

September 15, 2017 - Download

September 12, 2018 - Download

September 23, 2019 - Download

Errata below are for Protocol Document Version V23.0 – 2019/09/23.

Errata Published*	Description
2019/10/28	In Section 3.2.5.1.4, Constructing Confirm Active PDU, added a reference to explain the capabilitySets field of the TS_CONFIRM_ACTIVE_PDU structure. Changed from: ... Remote applications integrated locally (RAIL) clients MUST populate this PDU with two RAIL-specific capabilities in the capabilitySets field of the TS_CONFIRM_ACTIVE_PDU structure: the Remote Programs Capability Set, as specified in section 2.2.1.1.1, and the Window List Capability Set, as specified in section 2.2.1.1.2. ... Changed to: ... Remote applications integrated locally (RAIL) clients MUST populate this PDU with two RAIL-specific capabilities in the capabilitySets field of the TS_CONFIRM_ACTIVE_PDU ([MS-RDPBCGR] section 2.2.1.13.2.1) structure: the Remote Programs Capability Set, as specified in section 2.2.1.1.1, and the Window List Capability Set, as specified in section 2.2.1.1.2. ... In Section 3.2.5.1.5, Processing Demand Active PDU, added a reference to explain the capabilitySets field of the TS_DEMAND_ACTIVE_PDU structure.

Errata Published*	Description
	Changed from: ... Remote applications integrated locally (RAIL) clients MUST verify that this PDU contains two RAIL-specific capabilities in the capabilitySets field of the TS_DEMAND_ACTIVE_PDU structure: the Remote Programs Capability Set, as specified in section 2.2.1.1.1, and the Window List Capability Set, as specified in section 2.2.1.1.2. If it does not contain these capability sets, or if the RailSupportLevel of the Remote Programs Capability Set is not set to at least TS_RAIL_LEVEL_SUPPORTED, or the WndSupportLevel of the Window List Capability Set is TS_WINDOW_LEVEL_NOT_SUPPORTED (0), the client MUST drop the connection. ... Changed to: ... Remote applications integrated locally (RAIL) clients MUST verify that this PDU contains two RAIL-specific capabilities in the capabilitySets field of the TS_DEMAND_ACTIVE_PDU ([MS-RDPBCGR] section 2.2.1.13.1.1) structure: the Remote Programs Capability Set, as specified in section 2.2.1.1.1, and the Window List Capability Set, as specified in section 2.2.1.1.2. If it does not contain these capability sets, or if the RailSupportLevel of the Remote Programs Capability Set is not set to at least TS_RAIL_LEVEL_SUPPORTED, or the WndSupportLevel of the Window List Capability Set is TS_WINDOW_LEVEL_NOT_SUPPORTED (0), the client MUST drop the connection. ... In Section 3.2.5.2.8.1, Sending Client Get Application ID PDU, changed Windows Information Order to Window Information Order. Changed from: After being initialized as specified in section 2.2.2.6.5, this PDU MAY be sent from a client to a server after receiving a Windows Information Order containing the WINDOW_ORDER_STATE_NEW (0x10000000) flag. Changed to: After being initialized as specified in section 2.2.2.6.5, this PDU MAY be sent from a client to a server after receiving a Window Information Order containing the WINDOW_ORDER_STATE_NEW (0x10000000) flag. In Section 3.3.5.1.4, Constructing Demand Active PDU, added a reference to explain the capabilitySets field of the TS_DEMAND_ACTIVE_PDU structure. Changed from: ... If the client has requested support for remote applications integrated locally (RAIL) in the Client Info PDU (as specified in [MS-RDPBCGR] section 2.2.1.11), and the server supports RAIL, the server MUST specify two RAIL–specific capabilities in the capabilitySets field of the TS_DEMAND_ACTIVE_PDU structure: the Remote Programs Capability Set (section 2.2.1.1.1) and the Window List Capability Set (section 2.2.1.1.2). ... Changed to:

Errata Published*	Description
	... If the client has requested support for remote applications integrated locally (RAIL) in the Client Info PDU (as specified in [MS-RDPBCGR] section 2.2.1.11), and the server supports RAIL, the server MUST specify two RAIL–specific capabilities in the capabilitySets field of the TS_DEMAND_ACTIVE_PDU ([MS-RDPBCGR] section 2.2.1.13.1.1) structure: the Remote Programs Capability Set (section 2.2.1.1.1) and the Window List Capability Set (section 2.2.1.1.2). ... In Section 3.3.5.1.5, Processing Confirm Active PDU, added a reference to explain the capabilitySets field of the TS_CONFIRM_ACTIVE_PDU structure. Changed from: ... If the client has requested support for remote applications integrated locally (RAIL) in the Client Info PDU (see section 3.2.5.1.3), and the server has indicated support for RAIL in the Demand Active PDU (see section 3.3.5.1.4), the server MUST verify that this PDU contains two RAIL-specific capabilities in the capabilitySets field of the TS_CONFIRM_ACTIVE_PDU structure: the Remote Programs Capability Set (section 2.2.1.1.1) and the Window List Capability Set (section 2.2.1.1.2). If it does not contain these capability sets, or the RailSupportLevel of the Remote Programs Capability Set is not set to at least TS_RAIL_LEVEL_SUPPORTED, or the WndSupportLevel of the Window List Capability Set is TS_WINDOW_LEVEL_NOT_SUPPORTED (0), the server MUST drop the connection. ... Changed to: ... If the client has requested support for remote applications integrated locally (RAIL) in the Client Info PDU (see section 3.2.5.1.3), and the server has indicated support for RAIL in the Demand Active PDU (see section 3.3.5.1.4), the server MUST verify that this PDU contains two RAIL-specific capabilities in the capabilitySets field of the TS_CONFIRM_ACTIVE_PDU ([MS-RDPBCGR] section 2.2.1.13.2.1) structure: the Remote Programs Capability Set (section 2.2.1.1.1) and the Window List Capability Set (section 2.2.1.1.2). If it does not contain these capability sets, or the RailSupportLevel of the Remote Programs Capability Set is not set to at least TS_RAIL_LEVEL_SUPPORTED, or the WndSupportLevel of the Window List Capability Set is TS_WINDOW_LEVEL_NOT_SUPPORTED (0), the server MUST drop the connection. ... In Section 4.3.1, TS_RAIL_ORDER_EXEC, changed ArgumentsLength to ArgumentsLen in the network capture. Changed from: ... Header: 01 00 -> TS_RAIL_PDU_HEADER::orderType = TS_RAIL_ORDER_EXEC (1) (2 Bytes) 5e 00 -> TS_RAIL_PDU_HEADER::orderLength = 94 (2 Bytes) 08 00 -> Flags : TS_RAIL_EXEC_FLAG_EXPAND_ARGUMENTS (2 Bytes) 14 00 -> ExeOrFileLength : 0x14 (2 Bytes) 26 00 -> WorkingDirLength : 0x26 (2 Bytes) 18 00 -> ArgumentsLength : 0x18 (2 Bytes) ...

Errata Published*	Description
	Changed to: ... Header: 01 00 -> TS_RAIL_PDU_HEADER::orderType = TS_RAIL_ORDER_EXEC (1) (2 Bytes) 5e 00 -> TS_RAIL_PDU_HEADER::orderLength = 94 (2 Bytes) 08 00 -> Flags : TS_RAIL_EXEC_FLAG_EXPAND_ARGUMENTS (2 Bytes) 14 00 -> ExeOrFileLength : 0x14 (2 Bytes) 26 00 -> WorkingDirLength : 0x26 (2 Bytes) 18 00 -> ArgumentsLen : 0x18 (2 Bytes) ...
2019/10/28	In Section 4.1.1.1, New or Existing Windows, changed TS_WINDOW_ORDER_HEADER::Flags to TS_WINDOW_ORDER_HEADER::Header in the network capture. Changed from: ... 2e -> TS_WINDOW_ORDER_HEADER::Flags (1 Byte) 81 00 -> TS_WINDOW_ORDER_HEADER::OrderSize (2 Bytes) 9e df 08 19 -> TS_WINDOW_ORDER_HEADER::FieldsPresentFlags (4 Bytes) 58 01 12 00 -> WindowId 00 00 00 00 -> OwnerWindowId 00 00 cf 14 -> Style 00 01 00 00 -> ExtendedStyle 05 -> ShowState ... Changed to: ... 2e -> TS_WINDOW_ORDER_HEADER::Header (1 Byte) 81 00 -> TS_WINDOW_ORDER_HEADER::OrderSize (2 Bytes) 9e df 08 19 -> TS_WINDOW_ORDER_HEADER::FieldsPresentFlags (4 Bytes) 58 01 12 00 -> WindowId 00 00 00 00 -> OwnerWindowId 00 00 cf 14 -> Style 00 01 00 00 -> ExtendedStyle 05 -> ShowState ... In Section 4.1.1.2, Deleted Window, changed TS_WINDOW_ORDER_HEADER::Flags to TS_WINDOW_ORDER_HEADER::Header in the network capture. Changed from: ... 2e -> TS_WINDOW_ORDER_HEADER::Flags (1 Byte) 0b 00 -> TS_WINDOW_ORDER_HEADER::OrderSize (2 Bytes) 00 00 00 21 -> TS_WINDOW_ORDER_HEADER::FieldsPresentFlags (4 Bytes)

Errata Published*	Description
	(WINDOW_ORDER_TYPE_WINDOW | WINDOW_ORDER_STATE_DELETED) 24 00 03 00 -> WindowId Changed to: ... 2e -> TS_WINDOW_ORDER_HEADER::Header (1 Byte) 0b 00 -> TS_WINDOW_ORDER_HEADER::OrderSize (2 Bytes) 00 00 00 21 -> TS_WINDOW_ORDER_HEADER::FieldsPresentFlags (4 Bytes) (WINDOW_ORDER_TYPE_WINDOW | WINDOW_ORDER_STATE_DELETED) 24 00 03 00 -> WindowId ... In Section 4.1.1.3, New or Existing Notification Icons, changed TS_NOTIFYICON_ORDER_HEADER::Flags to TS_NOTIFYICON_ORDER_HEADER::Header in the network capture. Changed from: ... 2e -> TS_NOTIFYICON_ORDER_HEADER::Flags(1 Byte) 9d 04 -> TS_NOTIFYICON_ORDER_HEADER::OrderSize(2 Bytes) 01 00 00 52 -> TS_NOTIFYICON_ORDER_HEADER::FieldsPresentFlags (4 Bytes) WINDOW_ORDER_TYPE_NOTIFY | WINDOW_ORDER_FIELD_NOTIFY_TIP | WINDOW_ORDER_STATE_NEW | WINDOW_ORDER_ICON) 8e 00 01 00 -> TS_NOTIFYICON_ORDER_HEADER::WindowId d2 9c 00 00 -> TS_NOTIFYICON_ORDER_HEADER::NotifyIconId ... Changed to: ... 2e -> TS_NOTIFYICON_ORDER_HEADER::Header(1 Byte) 9d 04 -> TS_NOTIFYICON_ORDER_HEADER::OrderSize(2 Bytes) 01 00 00 52 -> TS_NOTIFYICON_ORDER_HEADER::FieldsPresentFlags (4 Bytes) WINDOW_ORDER_TYPE_NOTIFY | WINDOW_ORDER_FIELD_NOTIFY_TIP | WINDOW_ORDER_STATE_NEW | WINDOW_ORDER_ICON) 8e 00 01 00 -> TS_NOTIFYICON_ORDER_HEADER::WindowId d2 9c 00 00 -> TS_NOTIFYICON_ORDER_HEADER::NotifyIconId ... In Section 4.1.1.4, Deleted Notification Icons, changed TS_NOTIFYICON_ORDER_HEADER::Flags to TS_NOTIFYICON_ORDER_HEADER::Header in the network capture. Changed from: ... 2e -> TS_NOTIFYICON_ORDER_HEADER::Flags(1 Byte) 0f 00 -> TS_NOTIFYICON_ORDER_HEADER::OrderSize(2 Bytes)

Errata Published*	Description
	01 00 00 62 -> TS_NOTIFYICON_ORDER_HEADER::FieldsPresentFlags (4 Bytes) WINDOW_ORDER_TYPE_NOTIFY | WINDOW_ORDER_STATE_DELETED | WINDOW_ORDER_FIELD_NOTIFY_TIP | WINDOW_ORDER_ICON) f4 01 03 00 -> TS_NOTIFYICON_ORDER_HEADER::WindowId 00 00 00 00 -> TS_NOTIFYICON_ORDER_HEADER::NotifyIconId ... Changed to: ... 2e -> TS_NOTIFYICON_ORDER_HEADER::Header(1 Byte) 0f 00 -> TS_NOTIFYICON_ORDER_HEADER::OrderSize(2 Bytes) 01 00 00 62 -> TS_NOTIFYICON_ORDER_HEADER::FieldsPresentFlags (4 Bytes) WINDOW_ORDER_TYPE_NOTIFY | WINDOW_ORDER_STATE_DELETED | WINDOW_ORDER_FIELD_NOTIFY_TIP | WINDOW_ORDER_ICON) f4 01 03 00 -> TS_NOTIFYICON_ORDER_HEADER::WindowId 00 00 00 00 -> TS_NOTIFYICON_ORDER_HEADER::NotifyIconId ... In Section 4.1.1.5, Actively Monitored Desktop, changed TS_DESKTOP_ORDER_HEADER::Flags to TS_DESKTOP_ORDER_HEADER::Header in the network capture. Changed from: ... 2e -> TS_DESKTOP_ORDER_HEADER::Flags 14 00 -> TS_DESKTOP_ORDER_HEADER::OrderSize 30 00 00 04 -> TS_DESKTOP_ORDER_HEADER::FieldsPresentFlags (0x4000030) (WINDOW_ORDER_TYPE_DESKTOP | WINDOW_ORDER_FIELD_DESKTOP_ZORDER WINDOW_ORDER_FIELD_DESKTOP_ACTIVEWND) a0 00 01 00 -> ActiveWindowId 02 -> NumWindowIds 66 00 02 00 a0 00 01 00 -> WindowIds ... Changed to: ... 2e -> TS_DESKTOP_ORDER_HEADER::Header 14 00 -> TS_DESKTOP_ORDER_HEADER::OrderSize 30 00 00 04 -> TS_DESKTOP_ORDER_HEADER::FieldsPresentFlags (0x4000030) (WINDOW_ORDER_TYPE_DESKTOP | WINDOW_ORDER_FIELD_DESKTOP_ZORDER WINDOW_ORDER_FIELD_DESKTOP_ACTIVEWND) a0 00 01 00 -> ActiveWindowId 02 -> NumWindowIds 66 00 02 00 a0 00 01 00 -> WindowIds ...

Errata Published*	Description
	In Section 4.1.1.6 Non-monitored Desktop, changed TS_DESKTOP_ORDER_HEADER::Flags to TS_DESKTOP_ORDER_HEADER::Header in the network capture. Changed from: ... 2e -> TS_DESKTOP_ORDER_HEADER::Flags 07 00 -> TS_DESKTOP_ORDER_HEADER::OrderSize 01 00 00 04 -> TS_DESKTOP_ORDER_HEADER::FieldsPresentFlags (WINDOW_ORDER_TYPE_DESKTOP | WINDOW_ORDER_FIELD_DESKTOP_NONE) ... Changed to: ... 2e -> TS_DESKTOP_ORDER_HEADER::Header 07 00 -> TS_DESKTOP_ORDER_HEADER::OrderSize 01 00 00 04 -> TS_DESKTOP_ORDER_HEADER::FieldsPresentFlags (WINDOW_ORDER_TYPE_DESKTOP | WINDOW_ORDER_FIELD_DESKTOP_NONE) ...

*Date format: YYYY/MM/DD

[MS-RDPESC]: Remote Desktop Protocol: Smart Card Virtual Channel Extension

This topic lists the Errata found in [MS-RDPESC] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

June 1, 2017 - Download

[MS-RDPESP]: Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension

This topic lists the Errata found in [MS-RDPESP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-RDPEUDP]: Remote Desktop Protocol: UDP Transport Extension

This topic lists the Errata found in [MS-RDPEUDP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

March 2, 2016 - Download

October 16, 2015 - Download

June 30, 2015 - Download

June 1, 2017 - Download

September 23, 2019 - Download

[MS-RDPEUDP2]: Remote Desktop Protocol: UDP Transport Extension Version 2

This topic lists the Errata found in [MS-RDPEUDP2] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

March 13, 2019 - Download

September 23, 2019 - Download

[MS-RDPEV]: Remote Desktop Protocol: Video Redirection Virtual Channel Extension

This topic lists the Errata found in [MS-RDPEV] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-RDPEVOR]: Remote Desktop Protocol: Video Optimized Remoting Virtual Channel Extension

This topic lists the Errata found in [MS-RDPEVOR] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

[MS-RDPEXPS]: Remote Desktop Protocol: XML Paper Specification (XPS) Print Virtual Channel Extension

This topic lists the Errata found in [MS-RDPEXPS] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-RDPRFX]: Remote Desktop Protocol: RemoteFX Codec Extension

This topic lists the Errata found in [MS-RDPRFX] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

June 1, 2017 - Download

Errata below are for Protocol Document Version V20.0 – 2018/09/12.

Errata Published*	Description
2019/02/19	In Section 4.2.4.1, Input TS_RFX_TILESET Message, updated the first line of an annotated dump of a TS_RFX_TILESET message containing a single encoded 64x64 tile from "00000000 c7 cc 3e 0b 00 00 01 01 c2 ca 00 00 51 50 01 40" to "00000000 c7 cc 3e 0b 00 00 01 00 c2 ca 00 00 51 50 01 40". Changed from: The following is an annotated dump of a TS_RFX_TILESET (section 2.2.2.3.4) message containing a single encoded 64x64 tile. 00000000 c7 cc 3e 0b 00 00 01 01 c2 ca 00 00 51 50 01 40 ... Changed to: The following is an annotated dump of a TS_RFX_TILESET (section 2.2.2.3.4) message containing a single encoded 64x64 tile. 00000000 c7 cc 3e 0b 00 00 01 00 c2 ca 00 00 51 50 01 40 ...
2019/02/19	In Section 3.1.8.1.6, Linearization, updated the converted value of -10 to 10 after coefficients from LL3 undergo differential encoding. Changed from: ... The coefficients from LL3 also undergo differential encoding. Except for the first coefficient, every raster-scanned LL3 coefficient is subtracted from its previous neighbor. For example, if the raster-scanned LL3 coefficients are [64, 32, 42, 54, 50, 60, 40, 70] Then, after differential encoding, they would get converted to

Errata Published*	Description
	[64, -32, 10, 12, -4, -10, -20, 30] Changed to: ... The coefficients from LL3 also undergo differential encoding. Except for the first coefficient, every raster-scanned LL3 coefficient is subtracted from its previous neighbor. For example, if the raster-scanned LL3 coefficients are [64, 32, 42, 54, 50, 60, 40, 70] Then, after differential encoding, they would get converted to [64, -32, 10, 12, -4, 10, -20, 30]

*Date format: YYYY/MM/DD

[MS-RMPR]: Rights Management Services (RMS): Client-to-Server Protocol

This topic lists the Errata found in [MS-RMPR] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

Errata below are for Protocol Document Version V38.0 – 2018/09/12.

Errata Published*	Description
2019/10/16	In Section 2.2.3.4, string Element, parentheses around the ArrayOfString element have been removed. In Section 3.4.4.3.2.1, AcquireTemplates, parentheses around the ArrayOfString element have been removed. In Section 3.4.4.3.2.2, AcquireTemplatesResponse: Changed from: ArrayOfGuideTemplate Changed to: ArrayOfGuidTemplate In Section 3.5.4.2.3.2, ArrayOfGetClientLicensorCertResponse: Changed from: name="ArrayOfGetClIentLicensorCertResponse"> <xs:sequence> Changed to: name="ArrayOfGetClientLicensorCertResponse"> <xs:sequence>

Errata Published*	Description
	In Section 3.6.4.1, Synchronous Enrollment Operation, and Section 3.6.4.2, Asynchronous Enrollment Operation: Changed from: serverState Changed to: ServerState

*Date format: YYYY/MM/DD

[MS-RMSOD]: Rights Management Services Protocols Overview

This topic lists the Errata found in [MS-RMSOD] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

June 1, 2017 - Download

[MS-RPCE]: Remote Procedure Call Protocol Extensions

This topic lists the Errata found in the MS-RPCE document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

September 15, 2017 - Download

[MS-RPCH]: Remote Procedure Call over HTTP Protocol

This topic lists the Errata found in [MS-RPCH] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-RPRN]: Print System Remote Protocol

This topic lists the Errata found in [MS-RPRN] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

December 1, 2017 - Download

Errata below are for Protocol Document Version V32.0 – 2018/09/12.

Errata Published*	Description
2018/12/10	In Section 1.7, Versioning and Capability Negotiation, changed from: • Capability Negotiation: Functional negotiation … by comparing the value returned by the server in the dwBuildNumber member of OSVERSIONINFO (section 2.2.3.10.1) with well-known version-specific dwBuildNumber values.<2> <2> Section 1.7: The values of the dwBuildNumber member in the OSVERSIONINFO structure (section 2.2.3.10.1) for specific versions of Windows are shown in the table that follows. Version | dwBuildNumber value Windows 10 and Windows Server 2016 | >= 10586 … | … Changed to: • Capability Negotiation: Functional negotiation … by comparing the value returned by the server in the dwBuildNumber member of OSVERSIONINFO (section 2.2.3.10.1) with well-known version-specific dwBuildNumber values.<2> <2> Section 1.7: The values of the dwBuildNumber member in the OSVERSIONINFO structure (section 2.2.3.10.1) for specific versions of Windows are shown in the table that follows. Version | dwBuildNumber value Windows Server operating system | >= 16299 Windows 10 and Windows Server 2016 | >= 10586 … | …

Errata Published*	Description
	In Section 2.2.3.10.1, OSVERSIONINFO, changed from: dwBuildNumber (4 bytes): The build number of the OS. This is a version-specific value.<168> <168> Section 2.2.3.10.1: The values of the dwBuildNumber member in the OSVERSIONINFO structure (section 2.2.3.10.1) for specific versions of Windows is shown in the table that follows. On Windows Vista and later, an error is returned if the value is less than that shown in the table. Changed to: dwBuildNumber (4 bytes): The build number of the OS. This is a version-specific value.<168> <168> Section 2.2.3.10.1: The values of the dwBuildNumber member in the OSVERSIONINFO structure (section 2.2.3.10.1) for specific versions of Windows are listed in the product behavior note for dwBuildNumber in Versioning and Capability Negotiation (section 1.7). In Section 3.1.4.1.8.8, SPLCLIENT_CONTAINER Parameters, changed from: pClientInfo: This parameter is a pointer to an SPLCLIENT_CONTAINER (section 2.2.1.2.14) structure that specifies client information. The Level member of the SPLCLIENT_CONTAINER structure MUST be 0x00000001.<245> The value of the dwBuildNum member is used to verify that the client OS version is valid. It is a version-specific number.<246> <246> Section 3.1.4.1.8.8: The values of the dwBuildNumber member in the OSVERSIONINFO structure (section 2.2.3.10.1) for specific versions of Windows are listed in the product behavior note for dwBuildNumber in Versioning and Capability Negotiation (section 1.7). Changed to: pClientInfo: This parameter is a pointer to an SPLCLIENT_CONTAINER (section 2.2.1.2.14) structure that specifies client information. The Level member of the SPLCLIENT_CONTAINER structure MUST be 0x00000001.<245> The value of the dwBuildNum member is used to verify that the client OS version is valid. It is a version-specific number.<246> <246> Section 3.1.4.1.8.8: The values of the dwBuildNumber member in the OSVERSIONINFO structure (section 2.2.3.10.1) for specific versions of Windows are listed in the product behavior note for dwBuildNumber in Versioning and Capability Negotiation (section 1.7). On Windows Vista and later, an error is returned if the value is less than that shown for the corresponding Windows version in the table.
2018/10/29	In Section 2.2.3.10.1, OSVERSIONINFO, the description of dwBuildNumber has been changed from: dwBuildNumber (4 bytes): The build number of the OS.<168>. Changed to:

Errata Published*	Description
	dwBuildNumber (4 bytes): The build number of the OS. This SHOULD<168> be a version-specific value. In Section 3.1.4.1.8.8, SPLCLIENT_CONTAINER Parameters, the following has been changed from: pClientInfo: This parameter is a pointer to an SPLCLIENT_CONTAINER (section 2.2.1.2.14) structure that specifies client information. The Level member of the SPLCLIENT_CONTAINER structure MUST be 0x00000001.<245> Changed to: pClientInfo: This parameter is a pointer to an SPLCLIENT_CONTAINER (section 2.2.1.2.14) structure that specifies client information. The Level member of the SPLCLIENT_CONTAINER structure MUST be 0x00000001.<245> The dwBuildNum member is used to verify that the client OS version is valid. It SHOULD<246> be a version-specific number. In Section 7, Appendix B: Product Behavior, the following behavior notes have been changed. Changed from: <168> Section 2.2.3.10.1: The dwBuildNumber value for OSVERSIONINFO and OSVERSIONINFOEX for specific versions of Windows is shown in the table that follows. Changed to: <168> Section 2.2.3.10.1: The dwBuildNumber value for OSVERSIONINFO and OSVERSIONINFOEX for specific versions of Windows is shown in the table that follows. On Windows Vista and later, an error is returned if the value is less than that shown in the table. Changed from: <245> Section 3.1.4.1.8.8: Windows does not use the following members: pUserName, dwBuildNum, dwMajorVersion, dwMinorVersion, and wProcessorArchitecture. pMachineName is used only if the server cannot determine the client machine name using remote procedure call (RPC) functions. The pMachineName member can be NULL. Changed to: <245> Section 3.1.4.1.8.8: Windows does not use the following members: pUserName, dwMajorVersion, dwMinorVersion, and wProcessorArchitecture. pMachineName is used only if the server cannot determine the client machine name using remote procedure call (RPC) functions. The pMachineName member can be NULL. In that section a new behavior note 246 has been added:

Errata Published*	Description
	<246> Section 3.1.4.1.8.8: Windows version-specific values are listed in the product behavior note for dwBuildNumber in OSVERSIONINFO structure (section 2.2.3.10.1).

*Date format: YYYY/MM/DD

[MS-RRASM]: Routing and Remote Access Server (RRAS) Management Protocol

This topic lists the Errata found in [MS-RRASM] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

Errata below are for Protocol Document Version V23.0 – 2018/09/12.

Errata Published *	Description
2019/10/28	In Section 2.2.1.2.45, MIB_IPMCAST_OIF_STATS, changed dwIfNextHopIPAddr to dwNextHopAddr in the dwNextHopAddr field description. Changed from: ... dwNextHopAddr: Specifies the address of the next hop that corresponds to dwOutIfIndex. The dwOutIfIndex and dwIfNextHopIPAddr members uniquely identify a next hop on point-to-multipoint interfaces, where one interface connects to multiple networks. Examples of point-to-multipoint interfaces include non-broadcast multiple-access (NBMA) interfaces, and the internal interface on which all dial-up clients connect. For Ethernet and other broadcast interfaces, specify zero (0). Also specify zero (0) for point-to-point interfaces, which are identified by only dwOutIfIndex. ... Changed to: ... dwNextHopAddr: Specifies the address of the next hop that corresponds to dwOutIfIndex. The dwOutIfIndex and dwNextHopAddr members uniquely identify a next hop on point-to-multipoint interfaces, where one interface connects to multiple networks. Examples of point-to-multipoint interfaces include non-broadcast multiple-access (NBMA) interfaces, and the internal interface on which all dial-up clients connect. For Ethernet and other broadcast interfaces, specify zero (0). Also specify zero (0) for point-to-point interfaces, which are identified by only dwOutIfIndex. ... In Section 2.2.1.2.130, PPP_PROJECTION_INFO_1, changed dwAuthenticatedData to dwAuthenticationData in the dwAuthenticationData field description. Changed from: ... dwAuthenticationData: The same as dwAuthenticatedData in PPP_LCP_INFO. ... Changed to:

Errata Published *	Description
	... dwAuthenticationData: The same as dwAuthenticationData in PPP_LCP_INFO (see section 2.2.1.2.71). ... In Section 2.2.1.2.176, IGMP_MIB_GROUP_INFO, changed interface types RAS_SERVER to IGMP_IF_RAS_SERVER and RAS_CLIENT to IGMP_IF_RAS_CLIENT. Changed from: The IGMP_MIB_GROUP_INFO structure is used in the IGMP_MIB_IF_GROUPS_LIST (section 2.2.1.2.176) structure. If the interface is of type RAS_SERVER then the group membership of all the RAS clients is summarized, and the GroupUpTime and GroupExpiryTime is the maximum over all member RAS clients, while the V1HostPresentTimeLeft is set to 0. If the interface is of type RAS_CLIENT, the IpAddr is the next hop IP address of the RAS client. The membership is summarized over the RAS clients unless the IGMP_ENUM_FOR_RAS_CLIENTS_ID flag is set in Flags. ... Changed to: The IGMP_MIB_GROUP_INFO structure is used in the IGMP_MIB_IF_GROUPS_LIST (section 2.2.1.2.175) structure. If the interface is of type IGMP_IF_RAS_SERVER then the group membership of all the RAS clients is summarized, and the GroupUpTime and GroupExpiryTime is the maximum over all member RAS clients, while the V1HostPresentTimeLeft is set to 0. If the interface is of type IGMP_IF_RAS_CLIENT, the IpAddr is the next hop IP address of the RAS client. The membership is summarized over the RAS clients unless the IGMP_ENUM_FOR_RAS_CLIENTS_ID flag is set in Flags. ... In Section 2.2.1.2.181, IP_NAT_MIB_QUERY, changed instances of RMIBGetEntryFirst to RMIBEntryGetFirst. Changed from: The IP_NAT_MIB_QUERY structure is used to retrieve Network Address Translator (NAT) information and is passed to the following methods: ● RMIBEntryGet (section 3.1.4.30) ● RMIBGetEntryFirst (section 3.1.4.31) ● RMIBEntryGetNext (section 3.1.4.32) .... Oid: This is an index of the NAT MIB. It MUST be one of the following values.

Errata Published *	Description

Value	Meaning
IP_NAT_INTERFACE_STATISTICS_OID 0x00000000	NAT interface statistics information is retrieved. When RMIBEntryGet, RMIBGetEntryFirst, and RMIBEntryGetNext return pMibOutEntry or pInfoStruct it MUST be typecast to IP_NAT_INTERFACE_STATISTICS (section 2.2.1.2.185).
IP_NAT_INTERFACE_MAPPING_TABLE_OID 0x00000001	NAT interface mapping table information. When RMIBEntryGet, RMIBGetEntryFirst, and RMIBEntryGetNext return pMibOutEntry or pInfoStruct it MUST be typecast to IP_NAT_ENUMERATE_SESSION_MAPPINGS (section 2.2.1.2.183).
IP_NAT_MAPPING_TABLE_OID 0x00000002	NAT mapping table information. Retrieves the session mappings of an interface. When RMIBEntryGet, RMIBGetEntryFirst, and RMIBEntryGetNext return pMibOutEntry or pInfoStruct it

Changed to:

The IP_NAT_MIB_QUERY structure is used to retrieve Network Address Translator (NAT) information and is passed to the following methods:

- RMIBEntryGet (section 3.1.4.30)
- RMIBEntryGetFirst (section 3.1.4.31)
- RMIBEntryGetNext (section 3.1.4.32)

...

Oid: This is an index of the NAT MIB. It MUST be one of the following values.

Value	Meaning
IP_NAT_INTERFACE_STATISTICS_OID 0x00000000	NAT interface statistics information is retrieved. When RMIBEntryGet, RMIBEntryGetFirst, and RMIBEntryGetNext return pMibOutEntry or pInfoStruct it MUST be typecast to IP_NAT_INTERFACE_STATISTICS (section 2.2.1.2.184).
IP_NAT_INTERFACE_MAPPING_TABLE_OID 0x00000001	NAT interface mapping table information. When RMIBEntryGet, RMIBEntryGetFirst, and RMIBEntryGetNext return pMibOutEntry or pInfoStruct it MUST be typecast to IP_NAT_ENUMERATE_SESSION_MAPPINGS (section 2.2.1.2.182).
IP_NAT_MAPPING_TABLE_OID 0x00000002	NAT mapping table information. Retrieves the session mappings of an interface. When RMIBEntryGet, RMIBEntryGetFirst, and RMIBEntryGetNext return pMibOutEntry or pInfoStruct it MUST be typecast to IP_NAT_ENUMERATE_SESSION_MAPPINGS.

In Section 2.2.1.2.260, BGP_POLICY, changed eType value from MatchMaxPrefix to MatchMaxPrefixes. And changed eAttrType values ModifyLocalPref to NewLocalPref, ModifyNextHop to NewNextHop, and ModifyMed to NewMed.

Changed from:

Errata Published *	Description
	... A BGP policy: ● MUST NOT have more than one Match clause with eType in BGP_POLICY_MATCH set to MatchASNRange (0x3). ● MUST NOT have more than one Match clause with eType in BGP_POLICY_MATCH set to MatchMaxPrefix (0x5). ● MUST NOT have more than one modify Action clause with eAttrType in BGP_POLICY_MODIFY (section 2.2.1.2.259) set to ModifyLocalPref (0x3). ● MUST NOT have more than one modify Action clause with eAttrType in BGP_POLICY_MODIFY set to ModifyNextHop (0x4). ● MUST NOT have more than one modify Action clause with eAttrType in BGP_POLICY_MODIFY set to ModifyMed (0x5). ● MUST have only one Action clause with bDeny in BGP_POLICY_ACTION set to TRUE when a Match clause with eType in BGP_POLICY_MATCH is specified as MatchMaxPrefix (0x5). Changed to: ... A BGP policy: ● MUST NOT have more than one Match clause with eType in BGP_POLICY_MATCH set to MatchASNRange (0x3). ● MUST NOT have more than one Match clause with eType in BGP_POLICY_MATCH set to MatchMaxPrefixes (0x5). ● MUST NOT have more than one modify Action clause with eAttrType in BGP_POLICY_MODIFY (section 2.2.1.2.258) set to NewLocalPref (0x3). ● MUST NOT have more than one modify Action clause with eAttrType in BGP_POLICY_MODIFY set to NewNextHop (0x4). ● MUST NOT have more than one modify Action clause with eAttrType in BGP_POLICY_MODIFY set to NewMed (0x5). ● MUST have only one Action clause with bDeny in BGP_POLICY_ACTION set to TRUE when a Match clause with eType in BGP_POLICY_MATCH is specified as MatchMaxPrefixes (0x5). In Section 3.1.4.44, RMprAdminServerSetInfo (Opnum 43), changed return value ERROR_REBOOT_REQUIRED to ERROR_SUCCESS_REBOOT_REQUIRED when the RRAS server completes the processing successfully. Changed from: ... When processing this call, the RRASM server MUST do the following: ... ● If the RRAS server completes the processing successfully return either ERROR_SUCCESS or ERROR_REBOOT_REQUIRED<316> based on the impact of the configuration change as indicated by the RRAS server. Otherwise return the error status. ... Changed to: ... When processing this call, the RRASM server MUST do the following: ... ● If the RRAS server completes the processing successfully return either ERROR_SUCCESS or ERROR_SUCCESS_REBOOT_REQUIRED<316> based on the impact of the configuration change as indicated by the RRAS server. Otherwise return the error status. ...

Errata Published *	Description
	In Section 3.1.4.48, RMprAdminServerSetInfoEx (Opnum 47), changed return value ERROR_REBOOT_REQUIRED to ERROR_SUCCESS_REBOOT_REQUIRED when the RRAS server completes the processing successfully. Changed from: ... When processing this call, the RRASM server MUST do the following: ... ● If the RRAS server completes the processing successfully, it MUST return either ERROR_SUCCESS, ERROR_REBOOT_REQUIRED<321>, or ERROR_RESTART_REQUIRED<322> based on the impact of the configuration change. Otherwise return the error status. ... Changed to: ... When processing this call, the RRASM server MUST do the following: ... ● If the RRAS server completes the processing successfully, it MUST return either ERROR_SUCCESS, ERROR_SUCCESS_REBOOT_REQUIRED<321>, or ERROR_RESTART_REQUIRED<322> based on the impact of the configuration change. Otherwise return the error status. ... In Section 3.4.4.5 RasRpcSubmitRequest (Opnum 12), changed instances of GetDevConfig to GetDevConfigStruct when describing client behavior for the ReqType REQTYPE_GETDEVCONFIG. Changed from: ... REQTYPE_GETDEVCONFIG Before calling the method, the client MUST set the GetDevConfig.size value to the size of the GetDevConfig.config buffer. If the returned GetDevConfig.retcode is set to ERROR_BUFFER_TOO_SMALL (0x0000025B), the buffer that was passed in was not big enough to hold the device configuration information. The client SHOULD again call the API with GetDevConfig.size set to the size of returned GetDevConfig.size. ... Changed to: ... REQTYPE_GETDEVCONFIG Before calling the method, the client MUST set the GetDevConfigStruct.size value to the size of the GetDevConfigStruct.config buffer. If the returned GetDevConfigStruct.retcode is set to ERROR_BUFFER_TOO_SMALL (0x0000025B), the buffer that was passed in was not big enough to hold the device configuration information. The client SHOULD again call the API with GetDevConfigStruct.size set to the size of returned GetDevConfigStruct.size. ... In Section 7, Appendix B: Product Behavior, changed the return value ERROR_REBOOT_REQUIRED

Errata Published *	Description
	to ERROR_SUCCESS_REBOOT_REQUIRED in product behavior note <316> when the configuration change requires a reboot of the machine for the settings to be applied. Changed from: <316> Section 3.1.4.44: Windows will return the error value ERROR_REBOOT_REQUIRED when the configuration change requires a reboot of the machine for the settings to be applied. One such implementation requirement is when the number of ports configured is more than the maximum number of ports that the tunneling protocols are configured to support initially. Changed to: <316> Section 3.1.4.44: Windows will return the error value ERROR_SUCCESS_REBOOT_REQUIRED when the configuration change requires a reboot of the machine for the settings to be applied. One such implementation requirement is when the number of ports configured is more than the maximum number of ports that the tunneling protocols are configured to support initially. In this document, numerous editorial fixes have also been made, e.g., changed instances of "Ipv6" and "IPv6" to "IPV6"; changed instances of "GetDevConfig" to "GetDevConfigStruct"; updated hexadecimal syntax to USHORT 16-bit format; and also added section numbers to programming elements where applicable. Sections updated: 2.2.1.2.103 2.2.1.2.104 2.2.1.2.134 2.2.1.2.136 2.2.1.2.156 2.2.1.2.158 2.2.2.2.79 2.2.5.1.1 3.1.4.30 3.1.4.31 3.1.4.33 3.1.4.38 3.1.4.44 3.3.4.5 7 - the following product behavior notes were upated: <266> <268> <272> <290> <293> <298> <305>
2019/10/28	In Section 2.2.1.2.37 MIB_IPMCAST_BOUNDARY, added names of dwStatus values in the table.

Errata Published *	Description
	Changed from: dwStatus: A status value that describes the current status of this entry in a mullticast forwarding entry (MFE) boundary table. Value Meaning 0x00000001 The entry has an active status. 0x00000002 The entry has a notInService status. 0x00000003 The entry has a notReady status. 0x00000004 The entry has a createAndGo status. 0x00000005 The entry has a createAndWait status. 0x00000006 The entry has a destroy status. Changed to: dwStatus: A status value that describes the current status of this entry in a multicast forwarding entry (MFE) boundary table. Value Meaning ROWSTATUS_ACTIVE 0x00000001 The entry has an active status. ROWSTATUS_NOTINSERVICE 0x00000002 The entry has a notInService status. ROWSTATUS_NOTREADY 0x00000003 The entry has a notReady status. ROWSTATUS_CREATEANDGO 0x00000004 The entry has a createAndGo status. ROWSTATUS_CREATEANDWAIT 0x00000005 The entry has a createAndWait status. ROWSTATUS_DESTROY 0x00000006 The entry has a destroy status. Section 2.2.1.2.105 IPX_MIB_INDEX, added missing value 3 in the table.

Errata Published *	Description
	Changed from: TableId: Specifies the type of table. Values MUST be one of the following values. Value Meaning IPX_BASE_ENTRY 0x00000000 IPX base. See IPXMIB_BASE (section 2.2.1.2.107). IPX_INTERFACE_TABLE 0x00000001 IPX interface table. See IPX_INTERFACE (section 2.2.1.2.109). IPX_DEST_TABLE 0x00000002 IPX destination table. See IPX_ROUTE (section 2.2.1.2.110). IPX_SERV_TABLE 0x00000004 IPX service table. See IPX_SERVICE (section 2.2.1.2.121). IPX_STATIC_SERV_TABLE 0x00000005 IPX static service table. See IPX_STATIC_SERVICE_INFO (section 2.2.1.2.95). Changed to: TableId: Specifies the type of table. Values MUST be one of the following values. Value Meaning IPX_BASE_ENTRY 0x00000000 IPX base. See IPXMIB_BASE (section 2.2.1.2.106). IPX_INTERFACE_TABLE 0x00000001 IPX interface table. See IPX_INTERFACE (section 2.2.1.2.108). IPX_DEST_TABLE 0x00000002 IPX destination table. See IPX_ROUTE (section 2.2.1.2.109). IPX_STATIC_ROUTE_TABLE 0x00000003 IPX Static Route Table. See IPX_STATIC_ROUTE_INFO (section 2.2.1.2.93). IPX_SERV_TABLE

Errata Published *	Description
	0x00000004 IPX service table. See IPX_SERVICE (section 2.2.1.2.120). IPX_STATIC_SERV_TABLE 0x00000005 IPX static service table. See IPX_STATIC_SERVICE_INFO (section 2.2.1.2.94). Section 2.2.1.2.177 IGMP_MIB_GROUP_INFO, updated names of values in the introduction: RAS_SERVER to IGMP_IF_RAS_SERVER, RAS_CLIENT to IGMP_IF_RAS_CLIENT, and IGMP_ENUM_FOR_RAS_CLIENTS_ID to IGMP_ENUM_FOR_RAS_CLIENTS. Changed from: The IGMP_MIB_GROUP_INFO structure is used in the IGMP_MIB_IF_GROUPS_LIST (section 2.2.1.2.176) structure. If the interface is of type RAS_SERVER then the group membership of all the RAS clients is summarized, and the GroupUpTime and GroupExpiryTime is the maximum over all member RAS clients, while the V1HostPresentTimeLeft is set to 0. If the interface is of type RAS_CLIENT, the IpAddr is the next hop IP address of the RAS client. The membership is summarized over the RAS clients unless the IGMP_ENUM_FOR_RAS_CLIENTS_ID flag is set in Flags. Changed to: The IGMP_MIB_GROUP_INFO structure is used in the IGMP_MIB_IF_GROUPS_LIST (section 2.2.1.2.175) structure. If the interface is of type IGMP_IF_RAS_SERVER then the group membership of all the RAS clients is summarized, and the GroupUpTime and GroupExpiryTime is the maximum over all member RAS clients, while the V1HostPresentTimeLeft is set to 0. If the interface is of type IGMP_IF_RAS_CLIENT, the IpAddr is the next hop IP address of the RAS client. The membership is summarized over the RAS clients unless the IGMP_ENUM_FOR_RAS_CLIENTS_ID flag is set in Flags. Section 2.2.1.2.178 IGMP_MIB_IF_STATS, in the LastQuerierChangeTime description changed member name from igmpInterfaceQuerier to QuerierIpAddr. Changed from: LastQuerierChangeTime: The number of seconds since igmpInterfaceQuerier was last changed. Changed to: LastQuerierChangeTime: The number of seconds since QuerierIpAddr was last changed. Section 2.2.1.2.179 IGMP_MIB_GROUP_SOURCE_INFO_V3, added section. Adjusted references and reference numbers 2.2.1.2.180 to 2.2.1.2.271 throughout to compensate for section number changes. Changed from: (missing section) Changed to: The IGMP_MIB_GROUP_SOURCE_INFO_V3 structure provides information about each source IP endpoint. typedef struct _IGMP_MIB_GROUP_SOURCE_INFO_V3 {

Errata Published *	Description
	DWORD Source; DWORD SourceExpiryTime; DWORD SourceUpTime; DWORD Flags; } IGMP_MIB_GROUP_SOURCE_INFO_V3, *PIGMP_MIB_GROUP_SOURCE_INFO_V3; Source: IP endpoint address of a source. SourceExpiryTime: The time, in seconds, that remains before source expires. Not valid for exclusion mode. SourceUpTime: The time, in seconds since the source was up. Flags: Reserved. This is unused and SHOULD be NULL, or MAY be set to 0x00000000. Section 2.2.1.2.180 IGMP_MIB_GROUP_INFO_V3, for Sources array of IGMP_MIB_GROUP_SOURCE_INFO_V3 added reference to 2.2.1.2.179. Changed from: NumSources: The number of entries of IGMP_MIB_GROUP_SOURCE_INFO_V3. Sources: The IGMP_MIB_GROUP_SOURCE_INFO_V3 structure. Changed to: NumSources: The number of entries of IGMP_MIB_GROUP_SOURCE_INFO_V3. Sources: The IGMP_MIB_GROUP_SOURCE_INFO_V3 structure (section 2.2.1.2.179). 6 Appendix A: Full IDL, moved location of struct IGMP_MIB_GROUP_SOURCE_INFO_V3 to before struct IGMP_MIB_GROUP_INFO_V3. Changed from: typedef struct _IPRIP_PEER_STATS { DWORD PS_LastPeerRouteTag; DWORD PS_LastPeerUpdateTickCount; DWORD PS_LastPeerUpdateVersion; DWORD PS_BadResponsePacketsFromPeer;

Errata Published *	Description

	DWORD PS_BadResponseEntriesFromPeer;
	
	} IPRIP_PEER_STATS, *PIPRIP_PEER_STATS;
	
	typedef struct _IGMP_MIB_GROUP_SOURCE_INFO_V3 {
	
	DWORD Source;
	
	DWORD SourceExpiryTime; //not valid for exclusion mode
	
	DWORD SourceUpTime;
	
	DWORD Flags;
	
	} IGMP_MIB_GROUP_SOURCE_INFO_V3, *PIGMP_MIB_GROUP_SOURCE_INFO_V3;
	
	typedef struct _IGMP_MIB_GET_INPUT_DATA {
	
	DWORD TypeId;
	
	USHORT Flags;
	
	USHORT Signature;
	
	DWORD IfIndex;
	
	DWORD RasClientAddr;
	
	DWORD GroupAddr;
	
	DWORD Count;
	
	} IGMP_MIB_GET_INPUT_DATA, *PIGMP_MIB_GET_INPUT_DATA;
	
	Changed to:
	
	typedef struct _IGMP_MIB_GROUP_IFS_LIST {
	
	DWORD GroupAddr;
	
	DWORD NumInterfaces;
	
	BYTE Buffer[1];
	
	} IGMP_MIB_GROUP_IFS_LIST, *PIGMP_MIB_GROUP_IFS_LIST;

Errata Published *	Description

	typedef struct _IGMP_MIB_GROUP_SOURCE_INFO_V3 {
	
	DWORD Source;
	
	DWORD SourceExpiryTime; //not valid for exclusion mode
	
	DWORD SourceUpTime;
	
	DWORD Flags;
	
	} IGMP_MIB_GROUP_SOURCE_INFO_V3, *PIGMP_MIB_GROUP_SOURCE_INFO_V3;
	
	typedef struct _IGMP_MIB_GROUP_INFO_V3
	
	{
	
	union {
	
	DWORD IfIndex;
	
	DWORD GroupAddr;
	
	};
	
	DWORD IpAddr;
	
	DWORD GroupUpTime;
	
	DWORD GroupExpiryTime;
	
	DWORD LastReporter;
	
	DWORD V1HostPresentTimeLeft;
	
	DWORD Flags;
	
	//v3 additions
	
	DWORD Version; //1/2/3
	
	DWORD Size; //size of this struct
	
	DWORD FilterType;//EXCLUSION/INCLUSION
	
	DWORD V2HostPresentTimeLeft;

Errata Published *	Description
	DWORD NumSources; //IGMP_MIB_GROUP_SOURCE_INFO_V3 Sources[0]; } IGMP_MIB_GROUP_INFO_V3, *PIGMP_MIB_GROUP_INFO_V3;

*Date format: YYYY/MM/DD

[MS-RRP]: Windows Remote Registry Protocol

This topic lists the Errata found in the MS-RRP document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 12, 2018 - Download

Errata below are for Protocol Document Version V34.0 – 2019/03/13.

Errata Published*	Description
2019/08/05	In Sections 3.1.5.4, 3.1.5.28, and 3.1.5.29: Changed all instances of BaseRegEnumValues to BaseRegEnumValue In Section 3.1.5.7: Changed PRRP_UNICODE_STRING to RRP_UNICODE_STRING Changed instances of KEY_CREATE_SUBKEY to KEY_CREATE_SUB_KEY In Section 3.1.5.15: Changed phKeyResult to phKResult In Section 3.1.5.19: Changed RegRestoreKey to BaseRegRestoreKey In Section 3.1.5.20: Removed an extra space in ERROR_INVALID_HANDLE In Sections 3.1.5.26 and 3.1.5.30: Changed lpValueBuf to lpvalueBuf

*Date format: YYYY/MM/DD

[MS-RSMC]: Remote Session Monitoring and Control Protocol

This topic lists the Errata found in [MS-RSMC] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

[MS-RSVD]: Remote Shared Virtual Disk Protocol

This topic lists the Errata found in [MS-RSVD] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

September 26, 2016 - Download

March 20, 2017 - Download

June 1, 2017 - Download

September 15, 2017 - Download

December 1, 2017 - Download

[MS-SAMR]: Security Account Manager (SAM) Remote Protocol (Client-to-Server)

This topic lists the Errata found in [MS-SAMR] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

September 15, 2017 - Download

September 12, 2018 - Download

Errata below are for Protocol Document Version V40.0 – 2018/09/12.

Errata Published*	Description
2019/05/27	In Section 2.1, Transport, changed from: The server SHOULD use this identity to perform method-specific access checks, as specified in the message processing section of each method.<11> The server SHOULD<12> reject calls that do not use an authentication level of either RPC_C_AUTHN_LEVEL_NONE or RPC_C_AUTHN_LEVEL_PKT_PRIVACY (see [MS-RPCE] section 2.2.1.1.8). Changed to: The server SHOULD use this identity to perform method-specific access checks, as specified in the message processing section of each method.<11> RPC clients for this protocol MUST use authentication level RPC_C_AUTHN_LEVEL_NONE when invoking RPC over SMB methods. The server SHOULD<12> reject calls that do not use an authentication level of either RPC_C_AUTHN_LEVEL_NONE or RPC_C_AUTHN_LEVEL_PKT_PRIVACY (see [MS-RPCE] section 2.2.1.1.8).

*Date format: YYYY/MM/DD

[MS-SAMS]: Security Account Manager (SAM) Remote Protocol (Server-to-Server)

This topic lists the Errata found in the MS-KPP document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-SCMR]: Service Control Manager Remote Protocol

This topic lists the Errata found in [MS-SCMR] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 12, 2018 - Download

September 23, 2019 - Download

[MS-SHLLINK]: Shell Link (.LNK) Binary File Format

This topic lists the Errata found in [MS-SHLLINK] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 12, 2018 - Download

[MS-SFMWA]: Server and File Management Web APIs

This topic lists the Errata found in [MS-SFMWA] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

March 20, 2017 - Download

[MS-SFU]: Kerberos Protocol Extensions Service for User and Constrained Delegation Protocol

This topic lists the Errata found in the MS-SFU document 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 V17.0 – 2018/09/12.

Errata Published*	Description
2019/12/09	In Section 3.2.5.2, KDC Receives S4U2proxy KRB_TGS_REQ, moved content to subsequent sections to reorder processing steps. Changed from: When a KDC processes a TGS-REQ ([RFC4120] section 3.3.2) and it is a S4U2proxy KRB_TGS_REQ message, the KDC will perform the following steps. If the service ticket in the additional-tickets field is not set to forwardable<19> and the PA-PAC-OPTIONS [167] ([MS-KILE] section 2.2.10) padata type has the resource-based constrained delegation bit: • Not set, then the KDC MUST return KRB-ERR-BADOPTION with STATUS_NO_MATCH. • Set and the USER_NOT_DELEGATED bit is set in the UserAccountControl field in the KERB_VALIDATION_INFO structure ([MS-PAC] section 2.5), then the KDC MUST return KRB-ERR-BADOPTION with STATUS_ACCOUNT_RESTRICTION ([MS-ERREF] section 2.3.1). Service 1's KDC verifies both server ([MS-PAC] section 2.8.1) and KDC ([MS-PAC] section 2.8.2) signatures of the PAC. If Service 2 is in another domain, then its KDC verifies only the KDC signature of the PAC. If verification fails, the KDC MUST return KRB-AP-ERR-MODIFIED. When a KDC determines that a referral TGT is required ([RFC6806] section 8), then if Service 2 is not in the KDC's realm, the KDC SHOULD<20> reply with referral TGT (section 3.2.5.3.1). Changed to: When a KDC processes a TGS-REQ ([RFC4120] section 3.3.2) and it is a S4U2proxy KRB_TGS_REQ message, the KDC will perform the steps in the following sections. Deleted Section 3.2.5.2.1, KDC Confirms Delegation is Allowed and moved Section 3.2.5.2.1.2, Using ServicesAllowedToSendForwardedTicketsTo. Changed from: 3.2.5.2.1 KDC Confirms Delegation is Allowed If the KDC is for the realm of: • Service 2 only: The KDC uses the ServicesAllowedToReceiveForwardedTicketsFrom parameter

Errata Published*	Description
	to check if Service 1 is allowed to receive a service ticket for the principal.<21> • Service 1 and Service 2: First the KDC uses the ServicesAllowedToReceiveForwardedTicketsFrom parameter to check if Service 1 is allowed to receive a service ticket for the principal. If it fails or the ServicesAllowedToReceiveForwardedTicketsFrom parameter is empty, then the KDC uses the ServicesAllowedToSendForwardedTicketsTo parameter to check if Service 2 is listed on Service 1 as allowed to receive a service ticket for the principal. <22> Changed to: 3.2.5.2.1 Using ServicesAllowedToSendForwardedTicketsTo If the KDC is for the realm of both Service 1 and Service 2, then the KDC checks if the security principal name (SPN) for Service 2, identified in the sname and srealm fields of the KRB_TGS_REQ message, is in the Service 1 account's ServicesAllowedToSendForwardedTicketsTo parameter. If it is, then the delegation policy is satisfied. If not, and the PA-PAC-OPTIONS [167] ([MS-KILE] section 2.2.10) padata type does not have the resource-based constrained delegation bit, then the KDC MUST return KRB-ERR-BADOPTION. If Service 1’s ServicesAllowedToSendForwardedTicketsTo parameter was empty, this is returned with STATUS_NOT_SUPPORTED, else STATUS_NO_MATCH. If the service ticket in the additional-tickets field is not set to forwardable<19> and the PA-PAC-OPTIONS [167] ([MS-KILE] section 2.2.10) padata type has the resource-based constrained delegation bit set, then the KDC MUST return KRB-ERR-BADOPTION with STATUS_NO_MATCH. Added Section 3.2.5.2.2, Verification of the PAC and moved content from Section 3.2.5.2, KDC Receives S4U2proxy KRB_TGS_REQ to this new section. Changed from: --- Changed to: Service 1's KDC verifies both server ([MS-PAC] section 2.8.1) and KDC ([MS-PAC] section 2.8.2) signatures of the PAC. If Service 2 is in another domain, then its KDC verifies only the KDC signature of the PAC. If verification fails, the KDC MUST return KRB-AP-ERR-MODIFIED. Moved Section 3.2.5.2.1.1, Using ServicesAllowedToReceiveForwardedTicketsFrom, to after Section 3.2.5.2.1.2, Using ServicesAllowedToSendForwardedTicketsTo. Changed from: 3.2.5.2.1.1 Using ServicesAllowedToReceiveForwardedTicketsFrom If the Service 2 account's ServicesAllowedToReceiveForwardedTicketsFrom is nonempty and cname in the encrypted part of both TGTs match, the KDC creates a Token/Authorization Context ([MS-DTYP] section 2.5.2) for Service 1 from the PAC data in Service 1's TGT, and performs an access check using the ServicesAllowedToReceiveForwardedTicketsFrom parameter. If the access check succeeds, then the KDC replies with a service ticket for Service 2 (section 5.2.5.4.1).<23>

Errata Published*	Description
	Changed to: 3.2.5.2.3 Using ServicesAllowedToReceiveForwardedTicketsFrom If the delegation policy was not satisfied via ServicesAllowedToSendForwardedTicketsTo, this is the KDC for Service 2, and the Service 2 account's ServicesAllowedToReceiveForwardedTicketsFrom is nonempty and cname in the encrypted part of both TGTs match, the KDC creates a Token/Authorization Context ([MS-DTYP] section 2.5.2) for Service 1 from the PAC data in Service 1's TGT. Then the KDC performs an access check using the ServicesAllowedToReceiveForwardedTicketsFrom parameter.<20> If the access check succeeds, then the KDC replies with a service ticket for Service 2. If the access check fails, the KDC MUST return KRB-ERR-BADOPTION with STATUS_NOT_FOUND. If this is the KDC for Service 1, and the service ticket in the additional-tickets field is not set to forwardable,<21> and the USER_NOT_DELEGATED bit is set in the UserAccountControl field in the KERB_VALIDATION_INFO structure ([MS-PAC] section 2.5), then the KDC MUST return KRB-ERR-BADOPTION with STATUS_ACCOUNT_RESTRICTION ([MS-ERREF] section 2.3.1). When a KDC determines that a referral TGT is required ([RFC6806] section 8), then if Service 2 is not in the KDC's realm, the KDC SHOULD<22> reply with referral TGT (section 3.2.5.1.1). For details on the above changes, see the PDF doc here.
2019/10/16	In Section 3.2.5.2.2, KDC Replies with Service Ticket, the reference to the FORWARDABLE flag being set has been removed. Changed from: The KDC MUST reply with the service ticket where: ● The sname field contains the name of Service 2. ● The realm field contains the realm of Service 2. ● The cname field contains the cname from the service ticket in the additional-tickets field. ● The crealm field contains the crealm from the service ticket in the additional-tickets field. ● The S4U_DELEGATION_INFO structure is in the new PAC. ● If the TrustedToAuthenticationForDelegation parameter on the Service 1 principal is set to TRUE: ● The FORWARDABLE ticket flag is set. Changed to: The KDC MUST reply with the service ticket where: ● The sname field contains the name of Service 2. ● The realm field contains the realm of Service 2. ● The cname field contains the cname from the service ticket in the additional-tickets field. ● The crealm field contains the crealm from the service ticket in the additional-tickets field. ● The S4U_DELEGATION_INFO structure is in the new PAC
2019/09/02	In Section 3.2.5.2, KDC Receives S4U2proxy KRB_TGS_REQ, has been changed from: --Set and the USER_NOT_DELEGATED bit is set in the UserAccountControl field in the KERB_VALIDATION_INFO structure ([MS-PAC] section 2.5), then the KDC MUST return KRB-ERR-BADOPTION with STATUS_NOT_FOUND.

Errata Published*	Description
	Changed to: --Set and the USER_NOT_DELEGATED bit is set in the UserAccountControl field in the KERB_VALIDATION_INFO structure ([MS-PAC] section 2.5), then the KDC MUST return KRB-ERR-BADOPTION with STATUS_ACCOUNT_RESTRICTION ([MS-ERREF] section 2.3.1).

2019/07/22	In Section 2.2.1, PA-FOR-USER, corrected that PA-FOR-USER is not encrypted.
	
	Changed from:
	The following code defines the ASN.1 structure of the PA-FOR-USER padata type.
	
	`padata-type ::= PA-FOR-USER`
	`-- value 129`
	`padata-value ::= EncryptedData`
	`-- PA-FOR-USER-ENC`
	
	`PA-FOR-USER-ENC ::= SEQUENCE {`
	`userName [0] PrincipalName,`
	`userRealm [1] Realm,`
	`cksum [2] Checksum,`
	`auth-package [3] KerberosString`
	`}`
	
	Changed to:
	The following code defines the ASN.1 structure of the PA-FOR-USER padata type.
	
	`PA-FOR-USER ::= SEQUENCE {`
	`-- PA TYPE 129`
	`userName [0] PrincipalName,`
	`userRealm [1] Realm,`
	`cksum [2] Checksum,`
	`auth-package [3] KerberosString`
	`}`

2019/04/29	In this document, changes have been made to clarify the behavior of S4u2Self with x509 certificate regarding use of PA_FOR_USER following a cross-realm referral. In Section 3.1.5.1.1, Service Sends S4U2self KRB_TGS_REQ, has been changed from: The user identification for these cases is carried in a PA-FOR-USER padata type or a PA-S4U-X509-USER padata type, respectively. Changed to: The PA-FOR-USER padata type can be used only in the former case, while a PA-S4U-X509-USER padata type can carry the user identity in both cases." A new section, Section 3.1.5.1.1.1, When to Use Each padata Type, has been added: What padata type Service 1 sends is determined by two factors. First, determine whether the TGT session key is of a newer type, defined here as ciphers that are not DES or RC4 based.

Errata Published*	Description
	Second, determine whether the client username was provided explicitly or was extracted from a certificate. Service 1 SHOULD populate and send a PA-FOR-USER structure when one of the following is true: --No certificate was presented for the user. --No user name was explicitly provided, and instead a certificate was provided that contained the user name in the Subject Alternate Name (SAN) field. Service 1 SHOULD populate and send a PA-S4U-X509-USER structure when one of the following is true: --No PA-FOR-USER is being sent. --The session key of the TGT being used is not a DES or RC4 key type. In Section 3.1.5.1.1.2, the title has been changed from "Using the User's Realm and User Name" to "Identify the User". In Section 3.1.5.1.1.13.1.5.1.1.2, Sending the S4USelf KRB_TGT_REQ, has been changed from: The S4U2self information in the KRB_TGS_REQ consists of: padata-type = PA-FOR-USER (ID129), which consists of four fields: userName, userRealm, cksum, and auth-package. Changed to: If Service 1 sends a PA-FOR-USER (ID129) structure, it consists of four fields: userName, userRealm, cksum, and auth-package. In that same section, the following paragraph has been added: If sending a PA-S4U-X509-USER (ID 130) structure, the cname and crealm should contain the same values as used for userName and userRealm in a PA-FOR-USER. If a client certificate was provided, the subject-certificate field MUST contain the client's X509 certificate encoded in ASN.1, as specified in [RFC3280]." Section 3.1.5.1.1.2, Using the User's Certificate to Identify the User, has been removed. In Section 3.1.5.1.2, Service Receives S4U2self KRB_TGS_REP, the following paragraph has been added: In service tickets from KDCs that support S4U, the cname contains the name of the user. Services can further detect if the KDC supports PA_S4U_X509_USER by checking the reply padata for a PA-S4U-X509-USER preauth data. Furthermore, the KDC uses this reply padata to return a normalized form of the user name. Service 1 MUST take the cname from the reply PA-S4U-X509-USER and use it to replace both the cname from PA-S4U-X509-USER and the userName from PA-FOR-USER in any subsequent KRB_TGS_REQ requests used to chase referrals back to Service 1's realm. Additionally, the certificate is removed from the PA-S4U-X509-USER padata. For details on the above changes, see the PDF doc here.

*Date format: YYYY/MM/DD

[MS-SMB]: Server Message Block (SMB) Protocol

This topic lists the Errata found in [MS-SMB] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

June 1, 2017 - Download

December 1, 2017 - Download

September 12, 2018 - Download

[MS-SMB2]: Server Message Block (SMB) Protocol Versions 2 and 3

This topic lists the Errata found in [MS-SMB2] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

September 26, 2016 - Download

March 20, 2017 - Download

June 1, 2017 - Download

September 15, 2017 - Download

December 1, 2017 - Download

March 16, 2018 - Download

September 12, 2018 - Download

March 13, 2019 - Download

September 23, 2019 - Download

Errata below are for Protocol Document Version V12.0 – 2018/09/12.

Errata Published*	Description
2020/02/17	In Section 2.2.14, SMB2 CREATE Response, the following was removed: <49> Section 2.2.14: Windows-based clients never use exclusive oplocks. Because there are no situations where it would require an exclusive oplock where it would not also require an SMB2_OPLOCK_LEVEL_BATCH, it always requests an SMB2_OPLOCK_LEVEL_BATCH. In Section 2.2.24.1, Oplock Break Acknowledgment, the following was changed from: OplockLevel (1 byte): The client will set this field to the lowered oplock level that the client accepts for this file. This field MUST contain one of the following values.<55>. Value | Meaning SMB2_OPLOCK_LEVEL_NONE 0x00 | The client has lowered its oplock level for this file to none.

Errata Published*	Description

SMB2_OPLOCK_LEVEL_II 0x01	The client has lowered its oplock level for this file to level II.

<55> Section 2.2.24.1: Windows-based clients never use exclusive oplocks. There are no situations where an exclusive oplock would be used instead of using a SMB2_OPLOCK_LEVEL_BATCH.

Changed to:

OplockLevel (1 byte): The client will set this field to the lowered oplock level that the client accepts for this file. This field MUST contain one of the following values.

Value	Meaning
SMB2_OPLOCK_LEVEL_NONE 0x00	The client has lowered its oplock level for this file to none.
SMB2_OPLOCK_LEVEL_II 0x01	The client has lowered its oplock level for this file to level II.
SMB2_OPLOCK_LEVEL_EXCLUSIVE 0x08	The client has lowered its oplock level for this file to level Exclusive.

In Section 2.2.25.1, Oplock Break Response, the following was added:

SMB2_OPLOCK_LEVEL_EXCLUSIVE

0x08 The server has lowered oplock level for this file to level Exclusive.

In Section 3.3.4.6, Object Store Indicates an Oplock Break, the following was changed from:

The underlying object store on the local resource indicates the breaking of an opportunistic lock, specifying the LocalOpen and the new oplock level, a status code of the oplock break, and optionally expects the new oplock level in return. The new oplock level MUST be either SMB2_OPLOCK_LEVEL_NONE or SMB2_OPLOCK_LEVEL_II. The conditions under which each oplock level is to be indicated are described in [MS-FSA] section 2.1.5.17.3.

Changed to:

The underlying object store on the local resource indicates the breaking of an opportunistic lock, specifying the LocalOpen and the new oplock level, a status code of the oplock break, and optionally expects the new oplock level in return. The new oplock level SHOULD<200> be SMB2_OPLOCK_LEVEL_NONE or SMB2_OPLOCK_LEVEL_II or SMB2_OPLOCK_LEVEL_EXCLUSIVE. The conditions under which each oplock level is to be indicated are described in [MS-FSA] section 2.1.5.17.3.

<200> Section 3.3.4.6: In Windows-based SMB2 servers, underlying object store never breaks opportunistic lock to SMB2_OPLOCK_LEVEL_EXCLUSIVE oplock level.

Errata Published*	Description
	In Section 3.3.5.22.1, Processing an Oplock Acknowledgment, the following was changed from: The server MUST locate the session, as specified in section 3.3.5.2.9. The server MUST locate the tree connection, as specified in section 3.3.5.2.11. Next, the server MUST locate the open on which the client is acknowledging an oplock break by performing a lookup in Session.OpenTable using FileId.Volatile of the request as the lookup key. If no open is found, or if Open.DurableFileId is not equal to FileId.Persistent, the server MUST fail the request with STATUS_FILE_CLOSED. Otherwise, the server MUST locate the Request in Connection.RequestList for which Request.MessageId matches the MessageId value in the SMB2 header, and set Request.Open to the Open. If the OplockLevel in the acknowledgment is SMB2_OPLOCK_LEVEL_LEASE, the server MUST do the following: If Open.OplockState is not Breaking, stop processing the acknowledgment, and send an error response with STATUS_INVALID_PARAMETER. ● If Open.OplockState is Breaking, complete the oplock break request received from the object store as described in section 3.3.4.6, with a new level SMB2_OPLOCK_LEVEL_NONE in an implementation-specific manner,<380> and set Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE, and Open.OplockState to None. If Open.OplockLevel is SMB2_OPLOCK_LEVEL_EXCLUSIVE or SMB2_OPLOCK_LEVEL_BATCH, and if OplockLevel is not SMB2_OPLOCK_LEVEL_II or SMB2_OPLOCK_LEVEL_NONE, the server MUST do the following: ● If Open.OplockState is not Breaking, stop processing the acknowledgment, and send an error response with STATUS_INVALID_OPLOCK_PROTOCOL. If Open.OplockState is Breaking, complete the oplock break request received from the object store, as described in section 3.3.4.6, with a new level SMB2_OPLOCK_LEVEL_NONE in an implementation-specific manner,<381> and set Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE and Open.OplockState to None. If Open.OplockLevel is SMB2_OPLOCK_LEVEL_II, and if OplockLevel is not SMB2_OPLOCK_LEVEL_NONE, the server MUST do the following: ● If Open.OplockState is not Breaking, stop processing the acknowledgment, and send an error response with STATUS_INVALID_OPLOCK_PROTOCOL. If Open.OplockState is Breaking, complete the oplock break request received from the object store, as described in section 3.3.4.6, with a new level SMB2_OPLOCK_LEVEL_NONE in an implementation-specific manner,<382> and set Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE and Open.OplockState to None. If OplockLevel is SMB2_OPLOCK_LEVEL_II or SMB2_OPLOCK_LEVEL_NONE, the server MUST do the following: ● If Open.OplockState is not Breaking, stop processing the acknowledgment, and send an error response with STATUS_INVALID_DEVICE_STATE.

Errata Published*	Description
	If Open.OplockState is Breaking, complete the oplock break request received from the object store as described in section 3.3.4.6, with a new level received in OplockLevel in an implementation-specific manner.<384> If the object store indicates an error, set the Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE, the Open.OplockState to None, and send the error response with the error code received. If the object store indicates success, update Open.OplockLevel and Open.OplockState as follows: ● If OplockLevel is SMB2_OPLOCK_LEVEL_II, set Open.OplockLevel to SMB2_OPLOCK_LEVEL_II and Open.OplockState to Held. ● If OplockLevel is SMB2_OPLOCK_LEVEL_NONE, set Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE and the Open.OplockState to None. The server then MUST construct an oplock break response using the syntax specified in section 2.2.25.1 with the following value: ● OplockLevel MUST be set to Open.OplockLevel. This response MUST then be sent to the client. The status code returned by this operation MUST be one of those defined in [MS-ERREF]. Common status codes returned by this operation include: ● STATUS_ACCESS_DENIED ● STATUS_FILE_CLOSED ● STATUS_INVALID_OPLOCK_PROTOCOL ● STATUS_INVALID_PARAMETER ● STATUS_INVALID_DEVICE_STATE ● STATUS_NETWORK_NAME_DELETED ● STATUS_USER_SESSION_DELETED Changed to: The server MUST locate the session, as specified in section 3.3.5.2.9. The server MUST locate the tree connection, as specified in section 3.3.5.2.11. Next, the server MUST locate the open on which the client is acknowledging an oplock break by performing a lookup in Session.OpenTable using FileId.Volatile of the request as the lookup key. If no open is found, or if Open.DurableFileId is not equal to FileId.Persistent, the server MUST fail the request with STATUS_FILE_CLOSED. Otherwise, the server MUST locate the Request in Connection.RequestList for which Request.MessageId matches the MessageId value in the SMB2

Errata Published*	Description
	header, and set Request.Open to the Open. If Open.OplockState is not Breaking, the server MUST stop processing the acknowledgment, and send an error response with STATUS_INVALID_DEVICE_STATE. If the OplockLevel in the acknowledgment is SMB2_OPLOCK_LEVEL_LEASE, the server MUST complete the oplock break request received from the object store as described in section 3.3.4.6, with a new level SMB2_OPLOCK_LEVEL_NONE in an implementation-specific manner,<381> and set Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE, and Open.OplockState to None, send an error response with STATUS_INVALID_PARAMETER and stop processing. If any of the following conditions is TRUE, the server MUST complete the oplock break request received from the object store, as described in section 3.3.4.6, with a new level SMB2_OPLOCK_LEVEL_NONE in an implementation-specific manner<382>, set Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE and Open.OplockState to None, send an error response with STATUS_INVALID_OPLOCK_PROTOCOL, and stop processing: ● If Open.OplockLevel is SMB2_OPLOCK_LEVEL_EXCLUSIVE, and if OplockLevel is not SMB2_OPLOCK_LEVEL_II or SMB2_OPLOCK_LEVEL_NONE. ● If Open.OplockLevel is SMB2_OPLOCK_LEVEL_BATCH and if OplockLevel is not SMB2_OPLOCK_LEVEL_II, or SMB2_OPLOCK_LEVEL_NONE, or SMB2_OPLOCK_LEVEL_EXCLUSIVE. ● If Open.OplockLevel is SMB2_OPLOCK_LEVEL_II, and OplockLevel is not SMB2_OPLOCK_LEVEL_NONE. If OplockLevel is SMB2_OPLOCK_LEVEL_EXCLUSIVE, the server MUST complete the oplock break request received from the object store as described in section 3.3.4.6, with a new level SMB2_OPLOCK_LEVEL_NONE in an implementation-specific manner.<383> If OplockLevel is SMB2_OPLOCK_LEVEL_II or SMB2_OPLOCK_LEVEL_NONE, the server MUST complete the oplock break request received from the object store as described in section 3.3.4.6, with a new level received in OplockLevel in an implementation-specific manner.<384> If the object store indicates an error, the server MUST set the Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE, the Open.OplockState to None, send the error response with the error code received, and stop processing. If the object store indicates success, the server MUST update Open.OplockLevel and Open.OplockState as follows: ● If OplockLevel is SMB2_OPLOCK_LEVEL_EXCLUSIVE, set Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE and Open.OplockState to None. ● If OplockLevel is SMB2_OPLOCK_LEVEL_II, set Open.OplockLevel to SMB2_OPLOCK_LEVEL_II and Open.OplockState to Held. ● If OplockLevel is SMB2_OPLOCK_LEVEL_NONE, set Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE and the Open.OplockState to None. The server then MUST construct an oplock break response using the syntax specified in section 2.2.25.1 with the following value:

Errata Published*	Description
	● OplockLevel MUST be set to Open.OplockLevel. This response MUST then be sent to the client. The status code returned by this operation MUST be one of those defined in [MS-ERREF]. Common status codes returned by this operation include: ● STATUS_ACCESS_DENIED ● STATUS_FILE_CLOSED ● STATUS_INVALID_OPLOCK_PROTOCOL ● STATUS_INVALID_PARAMETER ● STATUS_INVALID_DEVICE_STATE ● STATUS_NETWORK_NAME_DELETED ● STATUS_USER_SESSION_DELETED <381> Section 3.3.5.22.1: Windows-based servers complete the oplock break indication request with the object store by providing the following SMB2 parameters as input parameters, as specified [MS-FSA] section 2.1.5.18: Object Store parameter SMB2 parameter Open Open.LocalOpen Type SMB2_OPLOCK_LEVEL_NONE <383> Section 3.3.5.22.1: Windows-based servers complete the oplock break indication request with the object store by providing the following SMB2 parameters as input parameters, as specified [MS-FSA] section 2.1.5.18: Object Store parameter SMB2 parameter Open Open.LocalOpen Type SMB2_OPLOCK_LEVEL_NONE
2020/02/03	In Section 3.3.5.9, Receiving an SMB2 CREATE Request, the following was added: ● If Connection.Dialect belongs to the SMB 3.x dialect family and Open.LocalOpen is a reparse point, set the SMB2_CREATE_FLAG_REPARSEPOINT bit in the Flags field. In Section 3.3.5.9.4, Handling the SMB2_CREATE_TIMEWARP_TOKEN Create Context, the following was removed:

Errata Published*	Description
	● If Connection.Dialect belongs to the SMB 3.x dialect family, the server MUST set the SMB2_CREATE_FLAG_REPARSEPOINT bit in the Flags field in SMB2 CREATE response.
2020/02/03	In Section 3.2.4.3, Application Requests Opening a File, the following was removed: The RequestedOplockLevel field is set to the oplock level that is requested by the application. If the application does not provide a requested oplock level, the client MUST choose an implementation-specific oplock level.<116> In Section 3.2.4.3, Application Requests Opening a File, the following was added: The RequestedOplockLevel field is set as below: If CreateOptions includes FILE_DIRECTORY_FILE, If Connection.SupportsDirectoryLeasing is TRUE, the client SHOULD set RequestedOplockLevel field to SMB2_OPLOCK_LEVEL_LEASE. Otherwise, RequestedOplockLevel field is set to SMB2_OPLOCK_LEVEL_ NONE. Otherwise, If the filename is stream name as defined in [MS-FSCC] section 2.1.5.3, RequestedOplockLevel field is set to SMB2_OPLOCK_LEVEL_NONE. Otherwise, if Connection.SupportsFileLeasing is TRUE, the client SHOULD set RequestedOplockLevel field to SMB2_OPLOCK_LEVEL_LEASE. Otherwise, if the oplock level requested by the application is SMB2_OPLOCK_LEVEL_NONE or SMB2_OPLOCK_LEVEL_II or SMB2_OPLOCK_LEVEL_BATCH, the client MUST set RequestedOplockLevel field to the oplock level that is requested by the application. Otherwise, RequestedOplockLevel field is set to an implementation-specific oplock level.<116> <116> Section 3.2.4.3: Windows-based clients will request a batch oplock for file creates when application does not provide a requested oplock level, or an exclusive oplock is specified, or a lease is requested.
2020/01/20	In Section 2.2.33, SMB2 QUERY_DIRECTORY Request, clarified the meaning of the Flags field values. Removed product behavior note <64> from the SMB2_INDEX_SPECIFIED Flags field value that stated that Windows-based servers do not support resuming an enumeration at a specified FileIndex. Changed from: ... Flags (1 byte): Flags indicating how the query directory operation MUST be processed. This field MUST be a logical OR of the following values, or zero if none are selected: Value | Meaning SMB2_RESTART_SCANS 0x01 | The server MUST restart the enumeration from the beginning as specified in section 3.3.5.18. SMB2_RETURN_SINGLE_ENTRY 0x02 | The server MUST only return the first entry of the search results. SMB2_INDEX_SPECIFIED 0x04 | The server SHOULD<64> return entries beginning at the byte number specified by FileIndex. SMB2_REOPEN 0x10 | The server MUST restart the enumeration from the beginning, and the search pattern MUST be changed to the provided value. ....

Errata Published*	Description
	... Changed to: ... Flags (1 byte): Flags indicating how the query directory operation MUST be processed. This field MUST be a logical OR of the following values, or zero if none are selected: | Value	Meaning
SMB2_RESTART_SCANS 0x01	The server is requested to restart the enumeration from the beginning as specified in section 3.3.5.18.
SMB2_RETURN_SINGLE_ENTRY 0x02	The server is requested to only return the first entry of the search results.
SMB2_INDEX_SPECIFIED 0x04	The server is requested to return entries beginning at the byte number specified by FileIndex.
SMB2_REOPEN 0x10	The server is requested to restart the enumeration from the beginning, and the search pattern is to be changed to the provided value. | .... (Removed the following product behavior note) <64> Section 2.2.33: Windows-based servers do not support resuming an enumeration at a specified FileIndex. The server will ignore this flag. In Section 3.3.5.18, Receiving an SMB2 QUERY_DIRECTORY Request, clarified product behavior note <351> when Windows-based servers perform query directory requests by updating the FileIndex input parameter. Also updated product behavior note <352> by clarifying that Windows-based servers ignore SMB2_INDEX_SPECIFIED in the Flags field and in the FileIndex value. Changed from: ... The server MUST invoke the query directory procedure from the underlying object store in an implementation-specific manner<352>. An underlying object store MAY<353> choose to support resuming enumerations by index number, if SMB2_INDEX_SPECIFIED is set in the Flags field and an index number is specified in the FileIndex field of the SMB2 QUERY_DIRECTORY Request. ... <352> Section 3.3.5.18: Windows-based servers perform query directory requests, as specified in [MS-FSA] section 2.1.5.5 with the following input parameters:

Errata Published*	Description
	● Open is set to Open.LocalOpen. ● FileInformationClass is set to the InformationClass that is received in the SMB2 QUERY_DIRECTORY Request. ● OutputBufferSize is set to the OutputBufferLength that is received in the SMB2 QUERY_DIRECTORY Request. ● If SMB2_RESTART_SCANS or SMB2_REOPEN is set in the Flags field of the SMB2 QUERY_DIRECTORY Request, RestartScan is set to TRUE. ● If SMB2_RETURN_SINGLE_ENTRY is set in the Flags field of the request, ReturnSingleEntry is set to TRUE. ● FileIndex is set to FileIndex received in the SMB2 QUERY_DIRECTORY Request. ● FileNamePattern is set to the search pattern specified in the SMB2 QUERY_DIRECTORY by FileNameOffset and FileNameLength. <353> Section 3.3.5.18: Windows-based servers do not support resuming an enumeration at a specified FileIndex. The server will ignore this flag. Changed to: ... The server MUST invoke the query directory procedure from the underlying object store in an implementation-specific manner<351>. The server MAY<352> choose to support resuming enumerations by index number, if SMB2_INDEX_SPECIFIED is set in the Flags field and an index number is specified in the FileIndex field of the SMB2 QUERY_DIRECTORY Request. ... <351> Section 3.3.5.18: Windows-based servers perform query directory requests, as specified in [MS-FSA] section 2.1.5.5 with the following input parameters: ● Open is set to Open.LocalOpen. ● FileInformationClass is set to the InformationClass that is received in the SMB2 QUERY_DIRECTORY Request. ● OutputBufferSize is set to the OutputBufferLength that is received in the SMB2 QUERY_DIRECTORY Request. ● If SMB2_RESTART_SCANS or SMB2_REOPEN is set in the Flags field of the SMB2 QUERY_DIRECTORY Request, RestartScan is set to TRUE. ● If SMB2_RETURN_SINGLE_ENTRY is set in the Flags field of the request, ReturnSingleEntry is set to TRUE. ● FileIndex is set to 0. ● FileNamePattern is set to the search pattern specified in the SMB2 QUERY_DIRECTORY by FileNameOffset and FileNameLength. <352> Section 3.3.5.18: Windows-based servers ignore SMB2_INDEX_SPECIFIED in Flags field and FileIndex value.
2020/01/06	In Section 2.2.33, SMB2 QUERY_DIRECTORY Request, clarified the meaning of the Flags field values. Removed product behavior note <64> from the SMB2_INDEX_SPECIFIED Flags field value that stated that Windows-based servers do not support resuming an enumeration at a specified FileIndex. Changed from: ... Flags (1 byte): Flags indicating how the query directory operation MUST be processed. This field MUST be a logical OR of the following values, or zero if none are selected: Value | Meaning

Errata Published*	Description

SMB2_RESTART_SCANS 0x01	The server MUST restart the enumeration from the beginning as specified in section 3.3.5.18.
SMB2_RETURN_SINGLE_ENTRY 0x02	The server MUST only return the first entry of the search results.
SMB2_INDEX_SPECIFIED 0x04	The server SHOULD<64> return entries beginning at the byte number specified by FileIndex.
SMB2_REOPEN 0x10	The server MUST restart the enumeration from the beginning, and the search pattern MUST be changed to the provided value. ....

...

Changed to:

...

Flags (1 byte): Flags indicating how the query directory operation MUST be processed. This field MUST be a logical OR of the following values, or zero if none are selected:

Value	Meaning
SMB2_RESTART_SCANS 0x01	The server is requested to restart the enumeration from the beginning as specified in section 3.3.5.18.
SMB2_RETURN_SINGLE_ENTRY 0x02	The server is requested to only return the first entry of the search results.
SMB2_INDEX_SPECIFIED 0x04	The server is requested to return entries beginning at the byte number specified by FileIndex.
SMB2_REOPEN 0x10	The server is requested to restart the enumeration from the beginning, and the search pattern is to be changed to the provided value.

....

(Removed the following product behavior note)

<64> Section 2.2.33: Windows-based servers do not support resuming an enumeration at a specified FileIndex. The server will ignore this flag.

In Section 3.3.5.18, Receiving an SMB2 QUERY_DIRECTORY Request, clarified product behavior note <351> when Windows-based servers perform query directory requests by updating the FileIndex input parameter. Also updated product behavior note <352> by clarifying that Windows-based servers ignore SMB2_INDEX_SPECIFIED in the Flags field and in the FileIndex value.

Errata Published*	Description
	Changed from: ... The server MUST invoke the query directory procedure from the underlying object store in an implementation-specific manner<352>. An underlying object store MAY<353> choose to support resuming enumerations by index number, if SMB2_INDEX_SPECIFIED is set in the Flags field and an index number is specified in the FileIndex field of the SMB2 QUERY_DIRECTORY Request. ... <352> Section 3.3.5.18: Windows-based servers perform query directory requests, as specified in [MS-FSA] section 2.1.5.5 with the following input parameters: ● Open is set to Open.LocalOpen. ● FileInformationClass is set to the InformationClass that is received in the SMB2 QUERY_DIRECTORY Request. ● OutputBufferSize is set to the OutputBufferLength that is received in the SMB2 QUERY_DIRECTORY Request. ● If SMB2_RESTART_SCANS or SMB2_REOPEN is set in the Flags field of the SMB2 QUERY_DIRECTORY Request, RestartScan is set to TRUE. ● If SMB2_RETURN_SINGLE_ENTRY is set in the Flags field of the request, ReturnSingleEntry is set to TRUE. ● FileIndex is set to FileIndex received in the SMB2 QUERY_DIRECTORY Request. ● FileNamePattern is set to the search pattern specified in the SMB2 QUERY_DIRECTORY by FileNameOffset and FileNameLength. <353> Section 3.3.5.18: Windows-based servers do not support resuming an enumeration at a specified FileIndex. The server will ignore this flag. Changed to: ... The server MUST invoke the query directory procedure from the underlying object store in an implementation-specific manner<351>. The server MAY<352> choose to support resuming enumerations by index number, if SMB2_INDEX_SPECIFIED is set in the Flags field and an index number is specified in the FileIndex field of the SMB2 QUERY_DIRECTORY Request. ... <351> Section 3.3.5.18: Windows-based servers perform query directory requests, as specified in [MS-FSA] section 2.1.5.5 with the following input parameters: ● Open is set to Open.LocalOpen. ● FileInformationClass is set to the InformationClass that is received in the SMB2 QUERY_DIRECTORY Request. ● OutputBufferSize is set to the OutputBufferLength that is received in the SMB2 QUERY_DIRECTORY Request. ● If SMB2_RESTART_SCANS or SMB2_REOPEN is set in the Flags field of the SMB2 QUERY_DIRECTORY Request, RestartScan is set to TRUE. ● If SMB2_RETURN_SINGLE_ENTRY is set in the Flags field of the request, ReturnSingleEntry is set to TRUE. ● FileIndex is set to 0. ● FileNamePattern is set to the search pattern specified in the SMB2 QUERY_DIRECTORY by FileNameOffset and FileNameLength. <352> Section 3.3.5.18: Windows-based servers ignore SMB2_INDEX_SPECIFIED in Flags field and FileIndex value.

Errata Published*	Description
2019/12/16	In Section 3.2.5.1.3, Verifying the Signature, clarified when verification is not required and described under what circumstances the client retrieves SessionId. Also, removed product behavior note <149> that described when Windows-based clients will not disconnect the connection but simply disregard the incorrectly signed response. Changed from: If the client implements the SMB 3.x dialect family and if the decryption in section 3.2.5.1.1.1 succeeds, the client MUST skip the processing in this section. If the MessageId is 0xFFFFFFFFFFFFFFFF, no verification is necessary. If the SMB2 header of the response has SMB2_FLAGS_SIGNED set in the Flags field and the message is not encrypted, the client MUST verify the signature as follows: The client MUST look up the session in the Connection.SessionTable using the SessionId in the SMB2 header of the response. If the session is not found, the response MUST be discarded as invalid. If Connection.Dialect belongs to the SMB 3.x dialect family, and the received message is an SMB2 SESSION_SETUP Response without a status code equal to STATUS_SUCCESS in the header, the client MUST verify the signature of the message as specified in section 3.1.5.1, using Session.SigningKey as the signing key, and passing the response message. For all other messages, the client MUST look up the Channel in Session.ChannelList, where the Channel.Connection matches the connection on which this message is received, and MUST use Channel.SigningKey for verifying the signature as specified in section 3.1.5.1. Otherwise, the client MUST verify the signature of the message as specified in section 3.1.5.1, using Session.SessionKey as the signing key, and passing the response message. If signature verification fails, the client MUST discard the received message and do no further processing for it. The client MAY also choose to disconnect the connection. If signature verification succeeds, the client MUST continue processing the packet, as specified in subsequent sections. If the SMB2 header of the response does not have SMB2_FLAGS_SIGNED set in the Flags field, the client MUST determine if the server failed to sign a packet that required signing. If the message is an interim response or an SMB2 OPLOCK_BREAK notification, signing validation MUST NOT occur. Otherwise, the client MUST look up the session in the Connection.SessionTable using the SessionId in the SMB2 header of the response. If the session is found, the Session.SigningRequired is equal to TRUE, the message is not an interim response, and the message is not an SMB2 OPLOCK_BREAK notification, the client MUST discard the received message and do no further processing for it. The client MAY also choose to disconnect the connection. If there is no SessionId, if the session is not found, or if Session.SigningRequired is FALSE, the client continues processing on the packet, as specified in subsequent sections.<149> Changed to: The client MUST skip the processing in this section if any of the following is TRUE: ● Client implements the SMB 3.x dialect family and decryption in section 3.2.5.1.1.1 succeeds ● MessageId is 0xFFFFFFFFFFFFFFFF ● Status in the SMB2 header is STATUS_PENDING For SMB2 SESSION_SETUP, the client MUST retrieve SessionId from SMB2 header of the response. For all other messages, the client MUST retrieve SessionId from the corresponding Request.Message. The client MUST look up the session in the Connection.SessionTable using the SessionId. If the session is not found, the response MUST be discarded as invalid. Otherwise if Session.SigningRequired is TRUE, the client MUST perform the following: ● If Connection.Dialect belongs to the SMB 3.x dialect family, and the received message is an SMB2 SESSION_SETUP Response without a status code equal to STATUS_SUCCESS in the header, the client MUST verify the signature of the message as specified in section 3.1.5.1, using Session.SigningKey as the signing key, and passing the response message. For all other

Errata Published*	Description
	messages, the client MUST look up the Channel in Session.ChannelList, where the Channel.Connection matches the connection on which this message is received, and MUST use Channel.SigningKey for verifying the signature as specified in section 3.1.5.1. ● Otherwise, the client MUST verify the signature of the message as specified in section 3.1.5.1, using Session.SessionKey as the signing key, and passing the response message. If signature verification fails, the client MUST discard the received message. The client MAY also choose to disconnect the connection. (removed the following product behavior note) <149> Section 3.2.5.1.3: Windows-based clients will not disconnect the connection but simply disregard the incorrectly signed response.
2019/11/25	In Section 3.3.5.14, Receiving an SMB2 LOCK Request, addressed when the server verifies whether the lock/unlock request along with the LockSequence value has been successfully processed. Clarified when lock sequence verification is neither resilient nor persistent in product behavior note <316>. Changed from: ... If the LockSequence value in the SMB2 LOCK Request (section 2.2.26) is not zero, and either one of the following conditions is TRUE, the server SHOULD verify whether the lock/unlock request with that LockSequence value has been successfully processed before: ● Connection.Dialect is "2.1" and Open.IsResilient is TRUE. ● Connection.Dialect belongs to the SMB 3.x dialect family.<316> The server verifies the LockSequence by performing the following steps: ● The server MUST use LockSequenceIndex as an index into Open.LockSequenceArray in order to locate the sequence number entry. If the index exceeds the maximum extent of the Open.LockSequenceArray, or LockSequenceIndex is 0, or if the Open.LockSequenceArray.Valid is FALSE, the server MUST skip step 2 and continue lock/unlock processing. ● The server MUST compare LockSequenceNumber to the SequenceNumber of the entry located in step 1. If the sequence numbers are equal, the server MUST complete the lock/unlock request with success. Otherwise, the server MUST reset the entry by setting Valid to FALSE and continue lock/unlock processing.... <316> Section 3.3.5.14: Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 do not verify the LockSequence value in the SMB2 LOCK Request (section 2.2.26) when both Open.IsResilient and Open.IsPersistent are FALSE. Changed to: ... If Connection.Dialect is not "2.0.2" and LockSequence value in the SMB2 LOCK Request (section 2.2.26) is not zero, the server SHOULD<316> verify whether the lock/unlock request with that LockSequence value has been successfully processed by performing the following steps: ● The server MUST use LockSequenceIndex as an index into Open.LockSequenceArray in order to locate the sequence number entry. If the index exceeds the maximum extent of the Open.LockSequenceArray, or LockSequenceIndex is 0, or if the Open.LockSequenceArray.Valid is FALSE, the server MUST skip step 2 and continue lock/unlock processing. ● The server MUST compare LockSequenceNumber to the SequenceNumber of the entry located in step 1. If the sequence numbers are equal, the server MUST complete the lock/unlock request

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>with success. Otherwise, the server MUST reset the entry by setting Valid to FALSE and continue lock/unlock processing....

<316> Section 3.3.5.14: Windows 7 operating system and Windows Server 2008 R2 operating system do not perform lock sequence verification when Open.IsResilient is FALSE.

Windows 8 operating system through Windows 10 v1909 and Windows Server 2012 operating system through Windows Server v1909 do not perform lock sequence verification when both Open.IsResilient and Open.IsPersistent are FALSE.</td></tr>
<tr><td>2019/11/25</td><td>In Section 2.2.14, SMB2 CREATE Response, changed the value OPLOCK_LEVEL_LEASE to SMB2_OPLOCK_LEVEL_LEASE in the OplockLevel table.

Changed from:
...
OplockLevel (1 byte): The oplock level that is granted to the client for this open. This field MUST contain one of the following values.<49>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>SMB2_OPLOCK_LEVEL_NONE
0x00</td><td>No oplock was granted.</td></tr>
<tr><td>SMB2_OPLOCK_LEVEL_II
0x01</td><td>A level II oplock was granted.</td></tr>
<tr><td>SMB2_OPLOCK_LEVEL_EXCLUSIVE
0x08</td><td>An exclusive oplock was granted.</td></tr>
<tr><td>SMB2_OPLOCK_LEVEL_BATCH
0x09</td><td>A batch oplock was granted.</td></tr>
<tr><td>OPLOCK_LEVEL_LEASE
0xFF</td><td>A lease is requested. If set, the response packet MUST contain an SMB2_CREATE_RESPONSE_LEASE create context.</td></tr>
</table>
...

Changed to:
...
OplockLevel (1 byte): The oplock level that is granted to the client for this open. This field MUST contain one of the following values.<49>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>SMB2_OPLOCK_LEVEL_NONE
0x00</td><td>No oplock was granted.</td></tr>
<tr><td>SMB2_OPLOCK_LEVEL_II
0x01</td><td>A level II oplock was granted.</td></tr>
</table></td></tr>
</table>

Errata Published*	Description

SMB2_OPLOCK_LEVEL_EXCLUSIVE 0x08	An exclusive oplock was granted.
SMB2_OPLOCK_LEVEL_BATCH 0x09	A batch oplock was granted.
SMB2_OPLOCK_LEVEL_LEASE 0xFF	A lease is requested. If set, the response packet MUST contain an SMB2_CREATE_RESPONSE_LEASE create context.

...

In Section 2.2.23.1, Oplock Break Notification, added the value SMB2_OPLOCK_LEVEL_EXCLUSIVE to the OplockLevel table and removed product behavior note <54> from the OplockLevel field description.

Changed from:

OplockLevel (1 byte): The server sets this to the maximum value of the OplockLevel that the server will accept for an acknowledgment from the client. This field MUST contain one of the following values.<54>

Value	Meaning
SMB2_OPLOCK_LEVEL_NONE 0x00	No oplock is available.
SMB2_OPLOCK_LEVEL_II 0x01	A level II oplock is available.

Changed to:

...

OplockLevel (1 byte): The server sets this to the maximum value of the OplockLevel that the server will accept for an acknowledgment from the client. This field MUST contain one of the following values.

Value	Meaning
SMB2_OPLOCK_LEVEL_NONE 0x00	No oplock is available.
SMB2_OPLOCK_LEVEL_II 0x01	A level II oplock is available.
SMB2_OPLOCK_LEVEL_EXCLUSIVE 0x08	Exclusive oplock is available.

...

Errata Published*	Description
	(removed the following product behavior note) <54> Section 2.2.23.1: Windows-based clients never use exclusive oplocks. Because there are no situations where it would require an exclusive oplock where it would not also require an SMB2_OPLOCK_LEVEL_BATCH, it always requests an SMB2_OPLOCK_LEVEL_BATCH. In Section 3.2.4.3, Application Requests Opening a File, clarified when clients will request a batch oplock for file creates in product behavior note <117>. Changed from: …. The SMB2 CREATE Request MUST be initialized as follows: ● The SecurityFlags field is set to 0. ● The RequestedOplockLevel field is set to the oplock level that is requested by the application. If the application does not provide a requested oplock level, the client MUST choose an implementation-specific oplock level.<117> ...<117> Section 3.2.4.3: Windows-based clients will request a batch oplock for file creates. Changed to: …. The SMB2 CREATE Request MUST be initialized as follows: ● The SecurityFlags field is set to 0. ● The RequestedOplockLevel field is set to the oplock level that is requested by the application. If the application does not provide a requested oplock level, the client MUST choose an implementation-specific oplock level.<117> ... <117> Section 3.2.4.3: Windows-based clients will request a batch oplock for file creates, when no oplock level is specified or an exclusive oplock is specified. In Section 3.2.5.19.1, Receiving an Oplock Break Notification, clarified client actions based on the the Open.OplockLevel and the new OplockLevel that is received in the Oplock Break Notification. Removed product behavior note <164>. Changed from: The client MUST locate the open in the Session.OpenTable using the FileId in the Oplock Break Notification following the SMB2 header. If the open is not found, the oplock break indication MUST be discarded, and no further processing is required. If the open is found, the client MUST take action based on the Open.OplockLevel and the new OplockLevel that is received in the Oplock Break Notification. If the Open.OplockLevel is SMB2_OPLOCK_LEVEL_NONE, no action is required, and no further processing is required. If the Open.OplockLevel is SMB2_OPLOCK_LEVEL_II, and the OplockLevel is SMB2_OPLOCK_LEVEL_NONE, the client MUST set Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE and an Oplock Break Acknowledgment MUST NOT be sent. If the Open.OplockLevel is SMB2_OPLOCK_LEVEL_BATCH, and the OplockLevel is SMB2_OPLOCK_LEVEL_NONE, the client MUST set Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE. The client MUST flush any writes or byte range locks that it has cached locally to the server. When that is complete, the client MUST send an oplock break

Errata Published*	Description
	acknowledgment, as specified in the following sections. If the Open.OplockLevel is SMB2_OPLOCK_LEVEL_BATCH, and the OplockLevel is SMB2_OPLOCK_LEVEL_II, the client MUST set Open.OplockLevel to SMB2_OPLOCK_LEVEL_II. The client MUST flush any writes or byte range locks that it has cached locally to the server. When that is complete, the client MUST send an oplock break acknowledgment, specified as follows. The client MAY<164> choose to request and support SMB2_OPLOCK_LEVEL_EXCLUSIVE. If it does, the break operation would match those specified above for SMB2_OPLOCK_LEVEL_BATCH. It MUST NOT break from batch to exclusive. If the client is required to send an oplock break acknowledgment, it MUST construct a request following the syntax that is specified in section 2.2.24.1. The SMB2 header is initialized as follows: ● Command MUST be set to SMB2 OPLOCK_BREAK. ● The MessageId field is set as specified in section 3.2.4.1.3. ● The client MUST set SessionId to Open.TreeConnect.Session.SessionId. ● The client MUST set TreeId to Open.TreeConnect.TreeConnectId. The Oplock Break Acknowledgment request is initialized as follows: ● The FileId MUST be set to Open.FileId. ● The OplockLevel MUST be set to Open.OplockLevel. The request MUST be sent to the server. Changed to: The client MUST locate the open in the Session.OpenTable using the FileId in the Oplock Break Notification following the SMB2 header. If the open is not found, the client MUST stop processing. If the open is found, the client MUST take action based on the Open.OplockLevel and the new OplockLevel that is received in the Oplock Break Notification. ● If Open.OplockLevel is SMB2_OPLOCK_LEVEL_II, and the new OplockLevel is SMB2_OPLOCK_LEVEL_NONE, the client MUST set Open.OplockLevel to SMB2_OPLOCK_LEVEL_NONE and MUST stop processing. ● If Open.OplockLevel is SMB2_OPLOCK_LEVEL_EXCLUSIVE and the new OplockLevel is SMB2_OPLOCK_LEVEL_NONE or SMB2_OPLOCK_LEVEL_II, locate the File in GlobalFileTable using Open.FileName. The client MUST flush any writes or byte range locks that it has cached locally to the server. The client MUST set Open.OplockLevel to new OplockLevel and send an oplock break acknowledgment. ● If Open.OplockLevel is SMB2_OPLOCK_LEVEL_BATCH and the new OplockLevel is SMB2_OPLOCK_LEVEL_EXCLUSIVE, locate the File in GlobalFileTable using Open.FileName. The client MUST process as below: ● Close any cached handles that have already been closed by the application, as specified in section 3.2.4.5. ● If File.OpenTable is empty, stop processing. ● Otherwise, set Open.OplockLevel to new OplockLevel and send an oplock break acknowledgment. ● If Open.OplockLevel is SMB2_OPLOCK_LEVEL_BATCH, and the new OplockLevel is SMB2_OPLOCK_LEVEL_NONE or SMB2_OPLOCK_LEVEL_II, locate the File in GlobalFileTable using Open.FileName. The client MUST process as below: ● For all cached handles in File.OpenTable, ● Flush any writes or byte range locks that it has cached locally to the server. ● Close any cached handles that have already been closed by the application, as specified in section 3.2.4.5. ● If File.OpenTable is empty, stop processing. ● Otherwise, set Open.OplockLevel to new OplockLevel and send an oplock break acknowledgment. ● Otherwise, the client MUST stop processing.

Errata Published*	Description
	The client MUST construct Oplock Break Acknowledgment following the syntax that is specified in section 2.2.24.1. The SMB2 header is initialized as follows: ● Command MUST be set to SMB2 OPLOCK_BREAK. ● The MessageId field is set as specified in section 3.2.4.1.3. ● The client MUST set SessionId to Open.TreeConnect.Session.SessionId. ● The client MUST set TreeId to Open.TreeConnect.TreeConnectId. The Oplock Break Acknowledgment is initialized as follows: ● The FileId MUST be set to Open.FileId. ● The OplockLevel MUST be set to Open.OplockLevel. The Oplock Break Acknowledgment MUST be sent to the server. (removed the following product behavior note) <164> Section 3.2.5.19.1: Windows-based clients will not request exclusive oplocks. In Section 3.3.1.10, Per Open, changed OPLOCK_LEVEL_LEASE to SMB2_OPLOCK_LEVEL_LEASE in the Open.OplockLevel description. Changed from: ... ● Open.OplockLevel: The current oplock level for this open. This value MUST be one of the OplockLevel values defined in section 2.2.14: SMB2_OPLOCK_LEVEL_NONE, SMB2_OPLOCK_LEVEL_II, SMB2_OPLOCK_LEVEL_EXCLUSIVE, SMB2_OPLOCK_LEVEL_BATCH, or OPLOCK_LEVEL_LEASE. ... Changed to: ... ● Open.OplockLevel: The current oplock level for this open. This value MUST be one of the OplockLevel values defined in section 2.2.14: SMB2_OPLOCK_LEVEL_NONE, SMB2_OPLOCK_LEVEL_II, SMB2_OPLOCK_LEVEL_EXCLUSIVE, SMB2_OPLOCK_LEVEL_BATCH, or SMB2_OPLOCK_LEVEL_LEASE. ...
2019/11/11	In Section 2.2.32.5.1.2, SOCKADDR_IN6, added a product behavior note to clarify Windows behavior when the ScopeId field should be set to zero. Changed from: ... ScopeId (4 bytes): The server SHOULD set this field to zero, and the client MUST ignore it on receipt. Changed to: ... ScopeId (4 bytes): The server SHOULD<64> set this field to zero, and the client MUST ignore it on receipt. <64> Section 2.2.32.5.1.2: Windows 10 v1709 operating system through Windows 10 v1909 operating system and Windows Server v1709 operating system through Windows Server v1909

Errata Published*	Description
	operating system set this field to any value.
2019/11/11	In Section 2.2.2, SMB2 ERROR Response, clarified when the ErrorData variable field should match the ErrorData format. Changed from: ... ErrorData (variable): A variable-length data field that contains extended error information. If the ErrorContextCount field in the response is nonzero, this field MUST be formatted as a variable-length array of SMB2 ERROR Context structures as specified in section 2.2.2.1. Each SMB2 ERROR Context MUST start at an 8-byte aligned boundary relative to the start of the SMB2 ERROR Response. Otherwise, it MUST be formatted as specified in section 2.2.2.2. If the ByteCount field is zero then the server MUST supply an ErrorData field that is one byte in length, and SHOULD set that byte to zero; the client MUST ignore it on receipt.<4> Changed to: ... ErrorData (variable): A variable-length data field that contains extended error information. If the ErrorContextCount field in the response is nonzero, this field MUST be formatted as a variable-length array of SMB2 ERROR Context structures as specified in section 2.2.2.1. Each SMB2 ERROR Context MUST start at an 8-byte aligned boundary relative to the start of the SMB2 ERROR Response. Otherwise, it SHOULD<4> be formatted as specified in section 2.2.2.2. In Section 6, Appendix A: Product Behavior, clarified the applicable product versions that set ErrorData to one uninitialized byte when ByteCount is zero in product behavior note <4>. Changed from: <4> Section 2.2.2: Windows–based SMB2 servers leave this one byte of ErrorData uninitialized and it can contain any value. Changed to: <4> Section 2.2.2: Windows 10 v1703 operating system and prior and Windows Server 2016 and prior set ErrorData to one uninitialized byte when ByteCount is zero.
2019/11/11	In Section 3.2.1.2, Per SMB2 Transport Connection, added the Connection.OfferedDialects behavior when the client implements the SMB 3.x dialect family. Changed from: ... If the client implements the SMB 3.x dialect family, it MUST also implement the following: ... ● Connection.Server: A reference to the server entry to which the connection is established. ... Changed to: ... If the client implements the SMB 3.x dialect family, it MUST also implement the following: ... ● Connection.Server: A reference to the server entry to which the connection is established.

Errata Published*	Description
	● Connection.OfferedDialects: An array of dialects sent in the SMB2 NEGOTIATE Request on this connection. ... In Section 3.2.4.2.2.2, SMB2-Only Negotiate, described when the client should set DialectCount to 1 and set Dialects array to Server.DialectRevision. Also added product behavior note <106> to describe the Windows product versions that set Dialects array to all the dialects the client implements and DialectCount to the number of dialects in Dialects array. Changed from: ... If the application has provided SpecifiedDialects, the client MUST do the following: ● Set the DialectCount to number of elements in the SpecifiedDialects. ● Set the value in Dialects array to the values in SpecifiedDialects. Otherwise, ● Set DialectCount to 0. ... ● If RequireMessageSigning is TRUE, the client MUST set the SMB2_NEGOTIATE_SIGNING_REQUIRED bit to TRUE in SecurityMode. If RequireMessageSigning is FALSE, the client MUST set the SMB2_NEGOTIATE_SIGNING_ENABLED bit to TRUE in SecurityMode. The client MUST store the value of the SecurityMode field in Connection.ClientSecurityMode. ... Changed to: ... If the application has provided SpecifiedDialects, the client MUST do the following: ● Set the DialectCount to number of elements in the SpecifiedDialects. ● Set the value in Dialects array to the values in SpecifiedDialects. Otherwise, if the client implements the SMB 3.x dialect family and an alternate connection is being established to an already connected Server, the client SHOULD<106> set DialectCount to 1 and set Dialects array to Server.DialectRevision. Otherwise, ● Set DialectCount to 0.... ● If the client implements SMB 3.x dialect family, Connection.OfferedDialects MUST be set to the values in Dialects array. ● If RequireMessageSigning is TRUE, the client MUST set the SMB2_NEGOTIATE_SIGNING_REQUIRED bit to TRUE in SecurityMode. If RequireMessageSigning is FALSE, the client MUST set the SMB2_NEGOTIATE_SIGNING_ENABLED bit to TRUE in SecurityMode. The client MUST store the value of the SecurityMode field in Connection.ClientSecurityMode. ... <106> Section 3.2.4.2.2.2: Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 set Dialects array to all the dialects the client implements and DialectCount to the number of dialects in Dialects array.

Errata Published*	Description
	In Section 3.2.5.5, Receiving an SMB2 TREE_CONNECT Response, revised the description of the VALIDATE_NEGOTIATE_INFO request structure. Added product behavior note<156> to describe Windows behavior when the client sets Dialects array to Connection.OfferedDialects. Changed from: ... ● The SMB2 IOCTL Request MUST be initialized as specified in section 2.2.31, with the exception of the following values: ... ● The VALIDATE_NEGOTIATE_INFO request structure is constructed as follows and copied into the request at InputOffset bytes from the beginning of the SMB2 header: ● Capabilities is set to Connection.ClientCapabilities. ● Guid is set to the Connection.ClientGuid value. ● SecurityMode is set to Connection.ClientSecurityMode. ● Set DialectCount to 0. ● If the client implements the SMB 2.0.2 dialect, it MUST do the following: ● Increment the DialectCount by 1. ● Set the value in Dialects[DialectCount-1] array to 0x0202. ● If the client implements the SMB 2.1 dialect, it MUST do the following: ● Increment the DialectCount by 1. ● Set the value in Dialects[DialectCount-1] array to 0x0210. ● If the client implements the SMB 3.0 dialect, it MUST do the following: ● Increment the DialectCount by 1. ● Set the value in the Dialects[DialectCount-1] array to 0x0300. ● If the client implements the SMB 3.0.2 dialect, it MUST do the following: ● Increment the DialectCount by 1. ● Set the value in the Dialects[DialectCount-1] array to 0x0302. ● The OutputOffset field offset to the Buffer[], in bytes, from the beginning of the SMB2 header. Changed to: ... ● The SMB2 IOCTL Request MUST be initialized as specified in section 2.2.31, with the exception of the following values: ... ● The VALIDATE_NEGOTIATE_INFO request structure is constructed as follows and copied into the request at InputOffset bytes from the beginning of the SMB2 header: ● Capabilities is set to Connection.ClientCapabilities. ● Guid is set to the Connection.ClientGuid value. ● SecurityMode is set to Connection.ClientSecurityMode. ● Dialects array SHOULD<156> be set to Connection.OfferedDialects. ● Set DialectCount to the number of dialects in Dialects array. ● The OutputOffset field offset to the Buffer[], in bytes, from the beginning of the SMB2 header…..

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td><156> Section 3.2.5.5: Windows 10 v1507 operating system through Windows 10 v1909, Windows Server 2016, Windows Server v1709 through Windows Server v1909, and Windows Server 2019 set Dialects array to all the dialects the client implements.</td></tr>
<tr><td>2019/11/11</td><td>In Section 2.2.21, SMB2 WRITE Request, described the SMB2_CHANNEL_RDMA_V1 and SMB2_CHANNEL_RDMA_V1_INVALIDATE values as it relates to the SMB 3.x dialect family and the Channel field of the request in the RemainingBytes, WriteChannelInfoOffset, and WriteChannelInfoLength field descriptions. Also added RemainingBytes to the value descriptions in the Channel table.

Changed from:
...
Channel (4 bytes): For the SMB 2.0.2 and 2.1 dialects, this field MUST NOT be used and MUST be reserved. The client MUST set this field to 0, and the server MUST ignore it on receipt. For the SMB 3.x dialect family, this field MUST contain exactly one of the following values:

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>SMB2_CHANNEL_NONE
0x00000000</td><td>No channel information is present in the request. The WriteChannelInfoOffset and WriteChannelInfoLength fields MUST be set to zero by the client and MUST be ignored by the server.</td></tr>
</table>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>SMB2_CHANNEL_RDMA_V1
0x00000001</td><td>One or more SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structures as specified in [MS-SMBD] section 2.2.3.1 are present in the channel information specified by WriteChannelInfoOffset and WriteChannelInfoLength fields.</td></tr>
<tr><td>SMB2_CHANNEL_RDMA_V1_INVALIDATE
0x00000002</td><td>This flag is not valid for the SMB 3.0 dialect. One or more SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structures as specified in [MS-SMBD] section 2.2.3.1 are present in the channel information specified by the WriteChannelInfoOffset and WriteChannelInfoLength fields. The server is requested to perform remote invalidation when responding to the request as specified in [MS-SMBD] section 3.1.4.2.</td></tr>
</table>

RemainingBytes (4 bytes): The number of subsequent bytes the client intends to write to the file after this operation completes. This value is provided to facilitate write caching and is not binding on the server.

WriteChannelInfoOffset (2 bytes): For the SMB 2.0.2 and 2.1 dialects, this field MUST NOT be used and MUST be reserved. The client MUST set this field to 0, and the server MUST ignore it on receipt. For the SMB 3.x dialect family, it contains the offset, in bytes, from the beginning of the SMB2 header to the channel data as specified by the Channel field of the request.

WriteChannelInfoLength (2 bytes): For the SMB 2.0.2 and SMB 2.1 dialects, this field MUST NOT be used and MUST be reserved. The client MUST set this field to 0, and the server MUST ignore it on receipt. For the SMB 3.x dialect family, it contains the length, in bytes, of the channel data as specified by the Channel field of the request.
...

Changed to:
...
Channel (4 bytes): For the SMB 2.0.2 and 2.1 dialects, this field MUST NOT be used and MUST be</td></tr>
</table>

Errata Published*	Description
	reserved. The client MUST set this field to 0, and the server MUST ignore it on receipt. For the SMB 3.x dialect family, this field MUST contain exactly one of the following values: **Value** / **Meaning** SMB2_CHANNEL_NONE 0x00000000 — No channel information is present in the request. The **RemainingBytes**, **WriteChannelInfoOffset** and **WriteChannelInfoLength** fields MUST be set to zero by the client and MUST be ignored by the server. **Value** / **Meaning** SMB2_CHANNEL_RDMA_V1 0x00000001 — One or more SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structures as specified in [MS-SMBD] section 2.2.3.1 are present in the channel information specified by **RemainingBytes**, **WriteChannelInfoOffset** and **WriteChannelInfoLength** fields. SMB2_CHANNEL_RDMA_V1_INVALIDATE 0x00000002 — This flag is not valid for the SMB 3.0 dialect. One or more SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structures as specified in [MS-SMBD] section 2.2.3.1 are present in the channel information specified by the **RemainingBytes**, **WriteChannelInfoOffset** and **WriteChannelInfoLength** fields. The server is requested to perform remote invalidation when responding to the request as specified in [MS-SMBD] section 3.1.4.2. RemainingBytes (4 bytes): For the SMB 3.x dialect family and the Channel field of the request contains SMB2_CHANNEL_RDMA_V1 or SMB2_CHANNEL_RDMA_V1_INVALIDATE, this field contains the length, in bytes, of the data being written. WriteChannelInfoOffset (2 bytes): For the SMB 3.x dialect family and the Channel field of the request contains SMB2_CHANNEL_RDMA_V1 or SMB2_CHANNEL_RDMA_V1_INVALIDATE, it contains the offset, in bytes, from the beginning of the SMB2 header to the channel data as specified by the Channel field of the request. WriteChannelInfoLength (2 bytes): For the SMB 3.x dialect family and the Channel field of the request contains SMB2_CHANNEL_RDMA_V1 or SMB2_CHANNEL_RDMA_V1_INVALIDATE, it contains the length, in bytes, of the channel data as specified by the Channel field of the request. ... In Section 3.3.5.13, Receiving an SMB2 WRITE Request, clarified what the server must do if the Connection.Dialect belongs to the SMB 3.x dialect family and the Channel field contains the value SMB2_CHANNEL_RDMA_V1 or SMB2_CHANNEL_RDMA_V1_INVALIDATE. Changed from: ... The server SHOULD<309> ignore undefined bits in the Flags field. If the server implements the SMB 3.0.2 or SMB 3.1.1 dialect, Connection.Dialect is not "3.0.2" or "3.1.1", and the SMB2_WRITEFLAG_WRITE_UNBUFFERED bit is set in the Flags field, the server MUST ignore the bit. If the request Channel field contains the value SMB2_CHANNEL_RDMA_V1 or

Errata Published*	Description
	SMB2_CHANNEL_RDMA_V1_INVALIDATE, then the data MUST be first obtained via the processing specified in [MS-SMBD] section 3.1.4.6 RDMA Read from Peer Buffer, providing the Connection, a newly allocated buffer to receive the data, and the array of SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structures passed in the request at offset WriteChannelInfoOffset and of length WriteChannelInfoLength fields. ... Changed to: ... The server SHOULD<309> ignore undefined bits in the Flags field. If the server implements the SMB 3.0.2 or SMB 3.1.1 dialect, Connection.Dialect is not "3.0.2" or "3.1.1", and the SMB2_WRITEFLAG_WRITE_UNBUFFERED bit is set in the Flags field, the server MUST ignore the bit. If Connection.Dialect belongs to the SMB 3.x dialect family and the Channel field contains the value SMB2_CHANNEL_RDMA_V1 or SMB2_CHANNEL_RDMA_V1_INVALIDATE, the server MUST do the following: ● The server MUST return STATUS_INVALID_PARAMETER to the client in the following conditions: ● RemainingBytes field is greater than Connection.MaxWriteSize. ● Length field of the first SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structure is zero. ● Sum of the values of Length fields in all SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structures is less than RemainingBytes. ● The data MUST be first obtained via the processing specified in [MS-SMBD] section 3.1.4.6, providing the Connection, a newly allocated buffer to receive the data, and the array of SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structures passed in the request at offset WriteChannelInfoOffset and of length WriteChannelInfoLength fields. ...

*Date format: YYYY/MM/DD

[MS-SMBD]: SMB2 Remote Direct Memory Access (RDMA) Transport Protocol

This topic lists the Errata found in [MS-SMBD] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

Errata below are for Protocol Document Version V12.0 – 2018/09/12.

Errata Published*	Description
2019/11/11	In Section 2.2.3.1, Buffer Descriptor V1 Structure, changed the structure name from SMB_DIRECT_BUFFER_DESCRIPTOR_1 to SMB_DIRECT_BUFFER_DESCRIPTOR_V1. Changed from: The SMB_DIRECT_BUFFER_DESCRIPTOR_1 structure represents a registered RDMA buffer and is used to Advertise the source and destination of RDMA Read and RDMA Write operations, respectively. The upper layer optionally embeds one or more of these structures in its payload when requesting RDMA direct placement of peer data via the protocol. ... Changed to: The SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structure represents a registered RDMA buffer and is used to Advertise the source and destination of RDMA Read and RDMA Write operations, respectively. The upper layer optionally embeds one or more of these structures in its payload when requesting RDMA direct placement of peer data via the protocol. ...
2019/11/11	In Section 3.1.5.1, Sending Upper Layer Messages, the following was changed from: ... The new messages to be sent, if any, MUST be appended to the list of messages in the Connection.SendQueue. If there are no messages to be sent and Connection.SendImmediate is TRUE, a newly constructed Data Transfer Message MUST be added to Connection.SendQueue. • the credit processing specified in section 3.1.5.9 MUST be performed, and the CreditsGranted field of the first message in Connection.SendQueue MUST be incremented by the number of new credits returned. For each message in Connection.SendQueue: • If Connection.SendCredits is 0, stop processing messages, and break the loop. • If Connection.SendCredits is 1 and the CreditsGranted field of the message is 0, then at least one credit MUST be granted to the peer to prevent deadlock. If the processing specified in section 3.1.5.9 returns zero, stop processing Sends, and break the loop. Otherwise, increment the CreditsGranted field of the current first message in Connection.SendQueue by the number of

Errata Published*	Description
	new credits returned. ● The first message MUST be removed from Connection.SendQueue. ● The value of Connection.SendCredits MUST be decremented by one. ● The value of the CreditsRequested field of the message MUST be set to Connection.SendCreditTarget. ● If Connection.KeepaliveRequested is "PENDING", the Flags field of the message MUST be set to SMB_DIRECT_RESPONSE_REQUESTED, Connection.KeepaliveRequested MUST be set to "SENT", and the Idle Connection Timer SHOULD<3> be set to an implementation-specific value. Otherwise, the Flags field of the message MUST be set to 0x0000. ● If the message to be sent was provided with an optional remote memory token to be invalidated on the receiving peer, the token SHOULD be provided in an implementation-specific manner to the RDMA provider when sending. If sending of remote invalidation is not supported by the RDMA provider, the token MAY be ignored. ● The message MUST be sent on the connection in an implementation-specific manner, and any error MUST be returned to the caller. ● If Connection.SendQueue is empty, Connection.SendImmediate MUST be set to FALSE and success MUST be returned to the caller. Changed to: ... For each message in Connection.SendQueue: ● If Connection.SendCredits is 0, stop processing. ● If CreditsGranted field of the first message in Connection.SendQueue is zero, the credit processing specified in section 3.1.5.9 MUST be performed, and the CreditsGranted field of the message MUST be set to the number of new credits returned. ● If Connection.SendCredits is 1 and the CreditsGranted field of the message is 0, stop processing. ● The first message MUST be removed from Connection.SendQueue. ● The value of Connection.SendCredits MUST be decremented by one. ● The value of the CreditsRequested field of the message MUST be set to Connection.SendCreditTarget. ● If Connection.KeepaliveRequested is "PENDING", the Flags field of the message MUST be set to SMB_DIRECT_RESPONSE_REQUESTED, Connection.KeepaliveRequested MUST be set to "SENT", and the Idle Connection Timer SHOULD<3> be set to an implementation-specific value. Otherwise, the Flags field of the message MUST be set to 0x0000. ● If the message to be sent was provided with an optional remote memory token to be invalidated on the receiving peer, the token SHOULD be provided in an implementation-specific manner to the RDMA provider when sending. If sending of remote invalidation is not supported by the RDMA provider, the token MAY be ignored. ● The message MUST be sent on the connection in an implementation-specific manner. ● Connection.SendImmediate MUST be set to FALSE. In Section 3.1.5.8, Receiving a Data Transfer Message, the following was changed from: ... If Connection.SendQueue is empty, the credit processing specified in section 3.1.5.9 MUST be performed. If the number of new credits returned is greater than zero, the receiver MUST set Connection.SendImmediate to TRUE and MUST promptly send a Data Transfer message on the Connection, as specified in section 3.1.5.1.

Errata Published*	Description
	... Changed to: ... If Connection.SendQueue is empty, the credit processing specified in section 3.1.5.9 MUST be performed. If the number of new credits returned is greater than zero, the receiver MUST promptly send a newly constructed Data Transfer message with its CreditsGranted field set to the number of new credits on the Connection, as specified in section 3.1.5.1. ...

*Date format: YYYY/MM/DD

[MS-SPNG]: Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) Extension

This topic lists the Errata found in [MS-SPNG] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-SQOS]: Storage Quality of Service Protocol

This topic lists the Errata found in [MS-SQOS] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-SSTP]: Secure Socket Tunneling Protocol (SSTP)

This topic lists the Errata found in [MS-SSTP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-SSTR]: Smooth Streaming Protocol

This topic lists the Errata found in the [MS-SSTR] document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

March 16, 2018 - Download

[MS-SWN]: Service Witness Protocol

This topic lists the Errata found in [MS-SWN] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

Errata below are for Protocol Document Version V11.0 – 2018/09/12.

Errata Published*	Description
2019/02/19	In Section 7, Appendix B: Product Behavior Product Behavior, note 2 has been changed from: <2> Section 3.1.3: Windows Server 2012 sets this value to 0x00010001. Windows Server 2012 R2, Windows Server 2016, Windows Server operating system, and Windows Server 2019 set this value to 0xFFFFFFFF. Changed to: <2> Section 3.1.3: Windows Server 2012 sets this value to 0x00010001. Windows Server 2012 R2, Windows Server 2016, Windows Server operating system, and Windows Server 2019 set this value to 0x00020000.

*Date format: YYYY/MM/DD

[MS-TCC]: Tethering Control Channel Protocol

This topic lists the Errata found in [MS-TCC] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

March 20, 2017 - Download

August 21, 2017 - Download

December 1, 2017 - Download

September 12, 2018 - Download

October 14, 2019 - Download

Errata below are for Protocol Document Version V28.0 – 2019/11/01.

Errata Published*	Description

2020/02/17	In Section 2.2.5.3.3, Trace Activity Header, the ABNF definition of ActivityId has been added.
	
	Changed from:
	
	`Stream-Specific Rules:`
	`GUID_ActivityID = 16BYTE ; client application activity id`
	`; used for debugging purposes`
	`ActivitySequence = ULONG ; client application activity sequence`
	`; used for debugging purposes`
	
	Changed to:
	
	`Stream-Specific Rules:`
	`GUID_ActivityID = 16BYTE ; client application activity id`
	`; used for debugging purposes`
	`ActivitySequence = ULONG ; client application activity sequence`
	`; used for debugging purposes`
	
	`ActivityId = GUID_ActivityID`
	`ActivitySequence`
	
	In Section 2.2.6.5, PRELOGIN, the ABNF definition of ACTIVITYID has been added to stream-specific rules and ACTIVITYID has been added to the definition of TRACEID in the

Errata Published*	Description

PL_OPTION_TOKEN table.

Changed from:

```
Stream-Specific Rules:
...
GUID_CONNID        =   16BYTE          ; client application trace id
                                       ; used for debugging purposes
GUID ActivityID    =   16BYTE          ; client application activity id
                                       ; used for debugging purposes
ActivitySequence   =   ULONG           ; client application activity sequence
                                       ; used for debugging purposes
```

...

PL_OPTION_TOKEN	Value	Description
...		
TRACEID	0x05	PL_OPTION_DATA = GUID_CONNID ACTIVITY_GUID SEQUENCE_ID
...		

Changed to:

```
Stream-Specific Rules:
...
GUID_CONNID       =   16BYTE          ; client application trace id
                                      ; used for debugging purposes
                                      ; introduced in TDS 7.4
GUID_ActivityID   =   16BYTE          ; client application activity id
                                      ; used for debugging purposes
                                      ; introduced in TDS 7.4
ActivitySequence  =   ULONG           ; client application activity sequence
                                      ; used for debugging purposes
                                      ; introduced in TDS 7.4

ACTIVITYID        =   GUID_ActivityID   ; client application activity id token
                      ActivitySequence  ; used for debugging purposes
                                        ; introduced in TDS 7.4
...
```

PL_OPTION_TOKEN	Value	Description
...		
TRACEID	0x05	PL_OPTION_DATA = GUID_CONNID ACTIVITYID Introduced in TDS 7.4.
...		

Errata Published*	Description

2019/12/09	In Section 2.2.5.1.2, Collation Rule Definition, the collation rule definition has been changed from:
	
	. . .
	
	`fBinary2 = BIT`
	`ColFlags = fIgnoreCase fIgnoreAccent fIgnoreKana fIgnoreWidth fBinary fBinary2 FRESERVEDBIT FRESERVEDBIT`
	`Version = 4BIT`
	
	. . .
	
	Changed to:
	
	. . .
	
	`fBinary2 = BIT`
	`fUTF8 = BIT`
	`ColFlags = fIgnoreCase fIgnoreAccent fIgnoreKana fIgnoreWidth fBinary fBinary2 fUTF8 FRESERVEDBIT`
	`Version = 4BIT`
	
	. . .

*Date format: YYYY/MM/DD

[MS-TLSP]: Transport Layer Security (TLS) Profile

This topic lists the Errata found in [MS-TLSP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

September 15, 2017 - Download

[MS-TPMVSC]: Trusted Platform Module (TPM) Virtual Smart Card Management Protocol

This topic lists the Errata found in [MS-TPMVSC] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-TSCH]: Task Scheduler Service Remoting Protocol

This topic lists the Errata found in [MS-TSCH] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

[MS-TSGU]: Terminal Services Gateway Server Protocol

This topic lists the Errata found in [MS-TSGU] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

December 1, 2017 - Download

September 12, 2018 - Download

Errata below are for Protocol Document Version V39.0 – 2018/09/12.

Errata Published*	Description
2019/10/28	In Section 3.1.1, Abstract Data Model, changed HTTP_CHANNEL_REQUEST to HTTP_CHANNEL_PACKET in the Target server names and Channel id element descriptions. Changed from: Target server names: An array of alias names for a target server. A target server alias name is a string of Unicode characters. The server name applies to the machine to which the RDG server connects.<23> ... ● For HTTP transport, this is initialized when the RDG server receives an HTTP_CHANNEL_REQUEST from the RDG client. ... Channel id: An unsigned long representing the channel identifier for tracking purposes on the RDG server. The Channel id, which is then generated on the server, is stored by the RDG server and RDG client and can later be used for subsequent channel-related calls.<25> ... ● For HTTP transport, this is generated after the RDG server receives HTTP_CHANNEL_REQUEST .... Changed to: Target server names: An array of alias names for a target server. A target server alias name is a string of Unicode characters. The server name applies to the machine to which the RDG server connects.<23> ... ● For HTTP transport, this is initialized when the RDG server receives an HTTP_CHANNEL_PACKET (section 2.2.10.2) from the RDG client. ...

Errata Published*	Description
	Channel id: An unsigned long representing the channel identifier for tracking purposes on the RDG server. The Channel id, which is then generated on the server, is stored by the RDG server and RDG client and can later be used for subsequent channel-related calls.<25> ... ● For HTTP transport, this is generated after the RDG server receives HTTP_CHANNEL_PACKET. ...
2019/10/28	In Section 2.2.9.2.1.1, TSG_PACKET_HEADER, changed the field names ComponentID to ComponentId and PacketID to PacketId. Changed from: The TSG_PACKET_HEADER structure contains information about the ComponentID and PacketID fields of the TSG_PACKET structure. The value of PacketID in TSG_PACKET MUST be set to TSG_PACKET_TYPE_HEADER. ... Changed to: The TSG_PACKET_HEADER structure contains information about the ComponentId and PacketId fields of the TSG_PACKET structure. The value of PacketId in TSG_PACKET MUST be set to TSG_PACKET_TYPE_HEADER. ... In Section 3.5.1, Abstract Data Model, changed the structure name AUTHENTICATION_COOKIE_DATA to AUTHN_COOKIE_DATA in the UDPAuthCookie description. Changed from: ... UDPAuthCookie: A signed and encoded byte BLOB containing an AUTHENTICATION_COOKIE_DATA structure. ... Changed to: ... UDPAuthCookie: A signed and encoded byte BLOB containing an AUTHN_COOKIE_DATA structure. ... In Section 3.7.1, Abstract Data Model, changed the structure name AUTHENTICATION_COOKIE_DATA to AUTHN_COOKIE_DATA in the UDPAuthCookie description. Changed from: UDPAuthCookie: A signed and encoded byte BLOB containing an AUTHENTICATION_COOKIE_DATA structure. ... Changed to: UDPAuthCookie: A signed and encoded byte BLOB containing an AUTHN_COOKIE_DATA structure. ...

Errata Published*	Description
	In Section 4.3.1, Normal Scenario, changed the structure name AUTHENTICATION_COOKIE_DATA to AUTHN_COOKIE_DATA and the ADM element name AUTHENTICATION_COOKIE_DATA.szServerName to AUTHN_COOKIE_DATA.szServerName. Changed from: .. 6. The RDG server decrypts the packet received with DTLS. The RDG server decodes the message and verifies the signature on the decoded message. The RDG server maps the decoded message to the AUTHENTICATION_COOKIE_DATA structure. 7. The RDG server connects to the target server specified in the ADM element AUTHENTICATION_COOKIE_DATA.szServerName. ... Changed to: ... 6. The RDG server decrypts the packet received with DTLS. The RDG server decodes the message and verifies the signature on the decoded message. The RDG server maps the decoded message to the AUTHN_COOKIE_DATA structure. 7. The RDG server connects to the target server specified in the ADM element AUTHN_COOKIE_DATA.szServerName....

*Date format: YYYY/MM/DD

[MS-TSTS]: Terminal Services Terminal Server Runtime Interface Protocol

This topic lists the Errata found in [MS-TSTS] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

Errata below are for Protocol Document Version V26.0 – 2018/09/12.

Errata Published*	Description
2019/04/15	In Section 3.10.4.1.1, RpcShadow2 (Opnum 0), the format of the pszInvitation field has been clarified. In addition, a reference to a Windows platform-specific API has been removed and substituted with a link to MS-RAI Section 2.2.2. Changed from: pszInvitation: The output data containing the invitation string for the shadow session. The data returned is an invitation string in an XML format that can be used with the Windows Desktop Sharing API IRDPSRAPIViewer::Connect method to connect to the session running in the target session (specified by TargetSessionId). The caller must allocate a buffer to hold this data and specify the size of the buffer in cchInvitation. Changed to: pszInvitation: The output data containing the invitation string for the shadow session. The data returned is a Unicode string in the XML format specified in [MS-RAI] section 2.2.2 that can be used to connect to a session running in the target session (specified by TargetSessionId). The caller must allocate a buffer to hold this data and specify the size of the buffer in cchInvitation.

*Date format: YYYY/MM/DD

[MS-TSWP]: Terminal Services Workspace Provisioning Protocol

This topic lists the Errata found in [MS-TSWP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 12, 2018 - Download

[MS-UAMG]: Update Agent Management Protocol

This topic lists the Errata found in [MS-UAMG] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-UCODEREF]: Windows Protocols Unicode Reference

This topic lists the Errata found in [MS-UCODEREF] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-VAPR]: Virtual Application Publication and Reporting (App-V) Protocol

This topic lists the Errata found in [MS-VAPR] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-VHDX]: Virtual Hard Disk v2 (VHDX) File Format

This topic lists the Errata found in [MS-VHDX] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-W32T]: W32Time Remote Protocol

This topic lists the Errata found in [MS-W32T] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-WCCE]: Windows Client Certificate Enrollment Protocol

This topic lists the Errata found in [MS-WCCE] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

Errata below are for Protocol Document Version V43.0 – 2018/09/12.

Errata Published*	Description
2019/12/16	In Section 3.1.2.4.2.2.2.8 , Certificate.Template.msPKI-Private-Key-Flag, added missing 'CT_FLAG_HELLO_LOGON_KEY' flag and description to the processing rules table. Also added new informative reference [MSDOCS-WHfB] to the description for the missing flag Changed from: 0x000001000 CT_FLAG_ATTEST_PREFERRED * This flag instructs the client to generate a certificate request as explained in section 3.1.1.4.3.4.1.1 if the Client_HardwareKeyInfo and Client_KeyAttestationStatement ADM elements are not empty (as described in section 3.1.2.4.2.2.2.2). Changed to: 0x000001000 CT_FLAG_ATTEST_PREFERRED * This flag instructs the client to generate a certificate request as explained in section 3.1.1.4.3.4.1.1 if the Client_HardwareKeyInfo and Client_KeyAttestationStatement ADM elements are not empty (as described in section 3.1.2.4.2.2.2.2). 0x00200000 CT_FLAG_HELLO_LOGON_KEY This flag instructs the client to generate a certificate request for the Windows Hello Logon key. For more information about

Errata Published*	Description
	* Windows Hello for Business, see [MSDOCS-WHfB].

*Date format: YYYY/MM/D

[MS-WCFESAN]: WCF-Based Encrypted Server Administration and Notification Protocol

This topic lists the Errata found in [MS-WCFESAN] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

[MS-WDSMT]: Windows Deployment Services Multicast Transport Protocol

This topic lists the Errata found in [MS-WDSMT] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-WDSOSD]: Windows Deployment Services Operation System Deployment Protocol

This topic lists the Errata found in the MS-FAX document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

September 12, 2018 - Download

[MS-WFDAA]: Wi-Fi Direct (WFD) Application to Application Protocol

This topic lists the Errata found in [MS-WFDAA] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

June 1, 2017 - Download

[MS-WFDPE]: Wi-Fi Display Protocol Extension

This topic lists the Errata found in [MS-WFDPE] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

March 20, 2017 - Download

June 1, 2017 - Download

[MS-WKST]: Workstation Service Remote Protocol

This topic lists the Errata found in [MS-WKST] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

Errata below are for Protocol Document Version V28.0 – 2018/09/12.

Errata Published*	Description
2018/11/12	In Section 3.2.4.8, NetrUseGetInfo (Opnum 9), changed from: … The server MUST fill the return structures as follows: • If the Level member is 0, the server MUST return the information about the connection by filling the USE_INFO_0_CONTAINER (section 2.2.5.25) structure in the Buffer field of the InfoStruct parameter as follows. USE_INFO_0_CONTAINER contains an array of USE_INFO_0 structures. • ui0_local set to Connection.local • ui0_remote set to Connection.Remote • If the Level member is 1, the server MUST return the information about the connection by filling the USE_INFO_1_CONTAINER (section 2.2.5.26) structure in the Buffer field of the InfoStruct parameter as follows. USE_INFO_1_CONTAINER contains an array of USE_INFO_1 structures. • ui1_local set to Connection.local • ui1_remote set to Connection.remote • ui1_password set to NULL • ui1_status set to Connection.status • ui1_asg_type set to Connection.asgtype • ui1_refcount set to Connection.refcount • ui1_usecount set to Connection.useCount • If the Level member is 2, the server MUST return the information about the connection by filling the USE_INFO_2_CONTAINER (section 2.2.5.27) structure in the Buffer field of the InfoStruct parameter as follows. USE_INFO_2_CONTAINER contains an array of USE_INFO_2 structures. • ui2_local set to Connection.local • ui2_remote set to Connection.remote • ui2_password set to NULL • ui2_status set to Connection.status • ui2_asg_type set to Connection.asgtype • ui2_refcount set to Connection.refcount • ui2_usecount set to Connection.useCount • ui2_domainname set to Connection.domain • If the Level member is 3, the server MUST return the information about the connection by filling the USE_INFO_3_CONTAINER structure in the Buffer field of the InfoStruct parameter as follows. USE_INFO_3_CONTAINER contains an array of

Errata Published*	Description
	USE_INFO_3 structures. ● ui2_local set to Connection.local ● ui2_remote set to Connection.remote ● ui2_password set to NULL ● ui2_status set to Connection.status ● ui2_asg_type set to Connection.asgtype ● ui2_refcount set to Connection.refcount ● ui2_usecount set to Connection.useCount ● ui2_domainname set to Connection.domain ● ui2_flag set to 0 The server MUST invoke the event to end the client impersonation ([MS-RPCE] section 3.3.3.4.3.3). Changed to: … The server MUST fill the return structures as follows: ● If the Level member is 0, the server MUST return the information about the connection by filling the USE_INFO_0_CONTAINER (section 2.2.5.25) structure in the Buffer field of the InfoStruct parameter as follows. USE_INFO_0_CONTAINER contains an array of USE_INFO_0 structures. ● ui0_local set to Connection.local ● ui0_remote set to Connection.Remote ● If the Level member is 1, the server MUST return the information about the connection by filling the USE_INFO_1_CONTAINER (section 2.2.5.26) structure in the Buffer field of the InfoStruct parameter as follows. USE_INFO_1_CONTAINER contains an array of USE_INFO_1 structures. ● ui1_local set to Connection.local ● ui1_remote set to Connection.remote ● ui1_password set to NULL ● ui1_status set to Connection.status ● ui1_asg_type set to Connection.asgtype ● ui1_refcount set to Connection.refcount ● ui1_usecount set to Connection.usecount ● If the Level member is 2 or 3, the server MUST return the information about the connection by filling the USE_INFO_2_CONTAINER (section 2.2.5.27) structure in the Buffer field of the InfoStruct parameter as follows. USE_INFO_2_CONTAINER contains an array of USE_INFO_2 structures. ● ui2_local set to Connection.local ● ui2_remote set to Connection.remote ● ui2_password set to NULL ● ui2_status set to Connection.status ● ui2_asg_type set to Connection.asgtype ● ui2_refcount set to Connection.refcount ● ui2_usecount set to Connection.usecount ● ui2_username set to Connection.username ● ui2_domainname set to Connection.domain The server MUST invoke the event to end the client impersonation ([MS-RPCE] section 3.3.3.4.3.3).
2018/11/12	In Section 3.2.4.13, NetrJoinDomain2 (Opnum 22), changed from:

Errata Published*	Description

Value/code	Meaning
…	…
NETSETUP_MACHINE_PWD_PASSED 0x00000080	Indicates that the Password parameter SHOULD<58> specify the password for the machine joining the domain. This flag is valid only for unsecured joins, which MUST be indicated by setting the NETSETUP_JOIN_UNSECURE flag. If this flag is set, the value of Password determines the value stored for the computer password during the join process.
…	…
NETSETUP_INSTALL_INVOCATION 0x00040000	Indicates that the protocol method was invoked during installation

Changed to:

Value/code	Meaning
…	…
NETSETUP_MACHINE_PWD_PASSED 0x00000080	Indicates that the Password parameter SHOULD<58> specify the password for the machine joining the domain. This flag is valid only for unsecured joins, which MUST be indicated by setting the NETSETUP_JOIN_UNSECURE flag, or read-only joins, which MUST be indicated by setting the NETSETUP_JOIN_READONLY flag. If this flag is set, the value of Password determines the value stored for the computer password during the join process.
…	…
NETSETUP_JOIN_READONLY 0x00000800	Specifies that the join SHOULD <121> be performed in a read-only manner against an existing account object. This option is intended to enable the server to join a domain using a read-only domain controller.
NETSETUP_INSTALL_INVOCATION 0x00040000	Indicates that the protocol method was invoked during installation

<121> Section 3.2.4.13: Windows NT, Windows 2000, Windows XP, Windows Server 2003, and Windows Server 2003 R2 do not implement this option.

In Section 3.2.4.13.3, Domain Join Specific Message Processing, changed from:

Errata Published*	Description
	The following statements define the sequence of message-processing operations: 1. If the NETSETUP_MACHINE_PWD_PASSED bit is set in Options, and the NETSETUP_JOIN_UNSECURE bit is not set in Options, the server MUST return ERROR_INVALID_PARAMETER. Otherwise, message processing continues. 2. If the NETSETUP_MACHINE_PWD_PASSED bit is set in Options, and AccountName is not NULL, the server MUST return ERROR_INVALID_PARAMETER. Otherwise, message processing continues. 3. If the NETSETUP_MACHINE_PWD_PASSED bit is set in Options, and either Password is NULL or the length of the PasswordString is zero, the server MUST return ERROR_PASSWORD_RESTRICTION. Otherwise, message processing continues. 4. If the NETSETUP_MACHINE_PWD_PASSED bit is set in Options, the value of PasswordString MUST be copied to the value of ComputerPasswordString, and PasswordString MUST be set to NULL. 5. If the server processing the message is already joined to a domain, and the NETSETUP_DOMAIN_JOIN_IF_JOINED bit is not set in Options, the server MUST return NERR_SetupAlreadyJoined. Otherwise, message processing continues. … 6. If DomainNameString contains the character "\",… The specified domain controller MUST be validated by invoking the DsrGetDcNameEx2 method ([MS-NRPC] section 3.5.4.3.1) on the DomainControllerString computer, specifying the following parameters: … ● Flags = B | J | R … If the call fails, or the returned domain controller name does not match DomainControllerString, the server MUST invoke the DsrGetDcNameEx2 method ([MS-NRPC] section 3.5.4.3.1) on the DomainControllerString computer, specifying the following parameters: … ● Flags = B | J | S … 29. The following LDAP attributes… Changed to: The following statements define the sequence of message-processing operations: 1. If the NETSETUP_MACHINE_PWD_PASSED bit is set in Options, and the NETSETUP_JOIN_UNSECURE bit is not set in Options, the server MUST return ERROR_INVALID_PARAMETER. Otherwise, message processing continues. 2. If the NETSETUP_MACHINE_PWD_PASSED bit is set in Options, and AccountName is not NULL, the server MUST return ERROR_INVALID_PARAMETER. Otherwise, message processing continues. 3. If the NETSETUP_MACHINE_PWD_PASSED bit is set in Options, and either Password is NULL or the length of the PasswordString is zero, the server MUST return ERROR_PASSWORD_RESTRICTION. Otherwise, message processing continues. 4. If the NETSETUP_MACHINE_PWD_PASSED bit is set in Options, the value of PasswordString MUST be copied to the value of ComputerPasswordString, and PasswordString MUST be set to NULL. 5. If the NETSETUP_JOIN_READONLY bit is set in Options, and NETSETUP_MACHINE_PWD_PASSED bit is not set in Options, the server MUST return

Errata Published*	Description
	ERROR_INVALID_PARAMETER. Otherwise, message processing continues. 6. If the NETSETUP_JOIN_READONLY bit is set in Options, and the NETSETUP_ACCT_CREATE bit is set in Options, the server MUST return ERROR_INVALID_PARAMETER. Otherwise, message processing continues. 7. If the NETSETUP_JOIN_READONLY bit is set in Options, the server MUST perform all subsequent message processing as if NETSETUP_DEFER_SPN_SET and NETSETUP_JOIN_UNSECURE bits are set in Options. 8. If the server processing the message is already joined to a domain, and the NETSETUP_DOMAIN_JOIN_IF_JOINED bit is not set in Options, the server MUST return NERR_SetupAlreadyJoined. Otherwise, message processing continues.… 9. If DomainNameString contains the character "\",… The specified domain controller MUST be validated by invoking the DsrGetDcNameEx2 method ([MS-NRPC] section 3.5.4.3.1) on the DomainControllerString computer, specifying the following parameters: … ● Flags : if NETSETUP_JOIN_READONLY bit is set in Options, set Flags = (B | R); otherwise set Flags to (B | J | R) … If the call fails, or the returned domain controller name does not match DomainControllerString, the server MUST invoke the DsrGetDcNameEx2 method ([MS-NRPC] section 3.5.4.3.1) on the DomainControllerString computer, specifying the following parameters: … ● Flags : if NETSETUP_JOIN_READONLY bit is set in Options, set Flags = (B | S); otherwise set Flags to (B | J | S) … 32. If the NETSETUP_JOIN_READONLY bit is not set in Options, the following LDAP attributes…

*Date format: YYYY/MM/DD

[MS-WMIO]: Windows Management Instrumentation Encoding Version 1.0 Protocol

This topic lists the Errata found in [MS-WMIO] 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 V15.0 – 2018/09/12.

Errata Published*	Description
2019/06/10	In Section 3 Structure Examples, we revised the octet value of PropertyInfoRef. Changed from: A0 00 00 00 Changed to: 0A 00 00 00

*Date format: YYYY/MM/DD

[MS-WMF]: Windows Metafile Format

This topic lists the Errata found in [MS-WMF] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-WPO]: Windows Protocols Overview

This topic lists the Errata found in [MS-WPO] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

[MS-WSDS]: WS-Enumeration Directory Services Protocol Extensions

This topic lists the Errata found in [MS-WSDS] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

March 20, 2017 - Download

June 1, 2017 - Download

[MS-WSMV]: Web Services Management Protocol Extensions for Windows Vista

This topic lists the Errata found in [MS-WSMV] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

December 1, 2017 - Download

September 12, 2018 - Download

[MS-WSP]: Windows Search Protocol

This topic lists the Errata found in [MS-WSP] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

September 23, 2019 - Download

Errata below are for Protocol Document Version V33.0 – 2019/09/23.

Errata Published*	Description
2019/11/25	In Section 2.2.1.42, CTableVariant, described the various string types in the Offset field description and clarified how to use the Offset field when the vType is not a string type. Changed from: ... Offset (variable): An offset to variable length data (for example, a string). This MUST be a 32-bit value (4 bytes long) if 32-bit offsets are being used (per the rules in section 2.2.3.12), or a 64-byte value (8 bytes long) if 64-bit offsets are being used. Changed to: ... Offset (variable): Only applies to string type (VT_VARIANT, VT_LPSTR, VT_LPWSTR, VT_BSTR, or VT_VECTORs or VT_ARRAYs of those base types). This MUST be a 32-bit value (4 bytes long) if 32-bit offsets are being used (per the rules in section 2.2.3.12), or a 64-bit value (8 bytes long) if 64-bit offsets are being used. Value: Other data types should put value in place rather than count/offset. VT_DECIMAL (see section 2.2.1.1.1.1) type: Size of value field is 4 bytes. All other types: Size of value field is 8 bytes.
2019/10/16	In Section 1.2.2, Informative References, a reference to Named Pipes has been added. Changed from: ... [MSDN-PROPLIST] Microsoft Corporation, "Windows Properties", http://msdn.microsoft.com/en-us/library/windows/desktop/dd561977(v=VS.85).aspx [MSDOCS-NLST] Microsoft Corporation, "National Language Support Terminology", https://docs.microsoft.com/en-us/windows/win32/intl/nls-terminology Changed to: ...

Errata Published*	Description
	[MSDN-PROPLIST] Microsoft Corporation, "Windows Properties", http://msdn.microsoft.com/en-us/library/windows/desktop/dd561977(v=VS.85).aspx [MSDOCS-NLST] Microsoft Corporation, "National Language Support Terminology", https://docs.microsoft.com/en-us/windows/win32/intl/nls-terminology [PIPE] Microsoft Corporation, "Named Pipes", http://msdn.microsoft.com/en-us/library/aa365590.aspx In Section 2.1, Transport, named pipe capitalization case sensitivity has been addressed by changing "MSFTEWDS" to "MsFteWds" and adding the reference [PIPE]. Changed from: All messages MUST be transported using a named pipe, as specified in [MS-SMB] or [MS-SMB2]. The following pipe name is used: ● \pipe\MSFTEWDS ... Changed to: All messages MUST be transported using a named pipe, as specified in [MS-SMB] or [MS-SMB2]. The following pipe name is used. For more information, see [PIPE]. ● \pipe\MsFteWds ...

*Date format: YYYY/MM/DD

[MS-WSTEP]: WS-Trust X.509v3 Token Enrollment Extensions

This topic lists the Errata found in [MS-WSTEP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-WSUSAR]: Windows Server Update Services: Administrative API Remoting Protocol

This topic lists the Errata found in the MS-WSUSAR document 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

June 1, 2017 - Download

Errata below are for Protocol Document Version V6.0 – 2018/09/12

Errata Published*	Description
2019/11/11	In Section 3.1.4.168.3.1, ExecuteSPPurgeReportingEventInstancesRequestBody, the element name in the WSDL has been updated. Changed from: name="ExecuteSPPurgeReportingEventInstances" Changed to: name="ExecuteSPPurgeReportingEventInstancesRequestBody" In Section 3.1.4.168.3.2, ExecuteSPPurgeReportingEventInstancesResponseBody, the element name in the WSDL has been updated: Changed from: name="ExecuteSPPurgeReportingEventInstancesResponse" Changed to: name="ExecuteSPPurgeReportingEventInstancesResponseBody" In Section 6, Appendix, corresponding changes were made to the WSDL. In Section 3.1.4.97.3.2, ExecuteSPGetApprovedUpdateMetadataResponseBody, the issue is "By Design", per the following text: "This data type is opaque for client protocol implementations. The server protocol implementation MUST provide a mechanism for how to manage these cookies."
2019/11/11	Throughout the document, the names and punctuation of the input and output messages have

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>been standardized.

Changed from:

<message description, possibly with spaces> Input Message

Changed to:

<message description, without spaces>_InputMessage

and correspondingly for output messages.</td></tr>
<tr><td>2019/11/11</td><td>In sections 2.2.5.11, UpdateState, 3.1.4.166, ExecuteSPResumeAllDownloads, and 3.1.4.167, ExecuteSPResumeDownload, the references to "LicenseAgreementFailed" have all been updated for consistency to "EulaFailed"

In Section 2.2.4.9, CompleteUpdates, the language has been updated to remove the incorrect inference that a special type Additional Information exists.

Changed from:

infoUrls: This field MUST be present. It gets or sets the array of Additional Information URLs.

Changed to:

infoUrls: This field MUST be present. It gets or sets the array of additional information URLs, which provides supplementary information—for example, the URL for a support article about the update.

In Section 3.1.4, Message Processing Events and Sequencing Rules, descriptions have been updated for GetSubscriptionCategories and ExecuteSPSearchEventHistory.

Changed from:

<table><tr><td>GetSubscriptionCategories</td><td>Gets the categories for a subscription which contains an array of Category Table Rows.</td></tr></table>
Changed to:

<table><tr><td>GetSubscriptionCategories</td><td>Gets the categories for a subscription.</td></tr></table>
and

Changed from:</td></tr>
</table>

Errata Published*	Description
	<table><tr><td>ExecuteSPSearchEventHistory</td><td>Returns the EventHistory based on EventHistoryFilter.</td></tr></table> Changed to:: <table><tr><td>ExecuteSPSearchEventHistory</td><td>Returns the list of EventHistoryTableRow based on EventHistoryFilter.</td></tr></table>
2019/11/11	In Section 3.1.4.51.3.4, EventHistoryFilter, the field names of two of the parameters have been changes as follows: Changed from: eventInstanceIdFilter: This field MUST be present. It represents the InstanceId to use when filtering the event history. eventIdFilter: This field MUST be present. It specifies the eventId of the event filter. Changed to: eventInstanceIdFilter: This field MUST be present. It represents the instance identifier to use when filtering the event history.eventIdFilter: This field MUST be present. It specifies the event identifier of the event filter. In the following sections, the field names of EventState parameter has been changed. 3.1.4.51.3.5 EventHistoryTableRow 3.1.4.51.3.6 EventIdFilter 3.1.4.51.3.7 EventSourceFilter Changed from: NamespaceId: This field MUST be present. It identifies the namespace that defines the EventId, EventState, and SourceId. Changed to: NamespaceId: This field MUST be present. It identifies the namespace that defines the EventId, StateId, and SourceId. In Section 3.1.4.69, ExecuteSPGetSdpXmlForUpdate, the name of a referenced parameter was updated as follows: Changed from: If the updateId or revisionsNumber fields are not found in the database, the server MUST send a SOAP fault as specified in section 3.1.4.1. Changed to:

Errata Published*	Description
	If the updateId or revisionNumber fields are not found in the database, the server MUST send a SOAP fault as specified in section 3.1.4.1. In Section 3.1.4.70.2.1, ExecuteSPSetEmailNotificationConfiguration, the referenced message name was updated: Changed from: This element contains the body of the ApiRemotingSoap_ ExecuteSPSetEmailNotificationRecipients _InputMessage WSDL message defined in section 3.1.4.71.1.1. Changed to: This element contains the body of the ApiRemotingSoap_ExecuteSPSetEmailNotificationRecipients_InputMessage WSDL message defined in section 3.1.4.71.1.1. In Section 3.1.4.70.2.2, ExecuteSPSetEmailNotificationConfigurationResponse, the referenced message name was updated: Changed from: This element contains the body of the ApiRemotingSoap_ ExecuteSPSetEmailNotificationRecipients _OutputMessage WSDL message defined in section 3.1.4.71.1.2. Changed to: This element contains the body of the ApiRemotingSoap_ExecuteSPSetEmailNotificationRecipients_OutputMessage WSDL message defined in section 3.1.4.71.1.2. In sections 3.1.4.134, ExecuteSPGetUpdateById, 3.1.4.134.3, Complex Types, and 3.1.4.134.3.1, ExecuteSPGetUpdateByIdRequestBody, references to updateRevisionID have been changed. Changed from: updateRevisionID Changed to: update revision identifier In Section 6, Appendix A: Full WSDL, the element definitions below have all had a parameter added to their definition: StatusNotification

Errata Published*	Description
	SmtpHostName SmtpUserName SmtpUserDisplayName SmtpUserMailAddress EmailLanguageLastModifiedBy Added the parameter: nillable="true"

*Date format: YYYY/MM/DD

[MS-WSUSOD]: Windows Server Update Services Protocols Overview

This topic lists the Errata found in [MS-WSUSOD] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

June 1, 2017 - Download

[MS-WSUSSS]: Windows Update Services: Server-Server Protocol

This topic lists the Errata found in the MS-WSUSSS document 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

July 18, 2016 - Download

June 1, 2017 - Download

[MS-WUSP]: Windows Update Services: Client-Server Protocol

This topic lists the Errata found in [MS-WMF] 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

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

June 1, 2017 - Download

Errata below are for Protocol Document Version V28.0 – 2018/09/12.

Errata Published*	Description
2020/02/17	In Section 3.1.5.1, Self-Update, the text has been updated to clarify the steps taken by clients which use the self-update procedure. Additionally, information has been provided as to how to initiate a license request for the Microsoft-signed redistributable files needed by Windows 7, 8, and 8.1 clients for a third-party implementation of the self-update process and more details provided regarding the processing of the files. Changed from: At the start of the protocol, the client SHOULD<31> check if it needs to self-update. Client/server communications can fail if the server does not expose the self-update content directory, as specified in this section. As specified in section 2, the server implementation MUST expose the self-update content directory as a virtual directory. The client issues HTTP GET requests (as specified in [RFC2616] section 9.3) to obtain files from the self-update content directory; therefore, the server MUST support HTTP requests on this virtual directory. The directory MUST include the wuident.cab file, which is implementation-specific.<32> <31> Section 3.1.5.1: Only Windows 8.1 and earlier Windows versions use this self-update mechanism. <32> Section 3.1.5.1: Windows Server Update Services includes a version of wuident.cab that is compatible with clients running Windows 8.1 and earlier. A current copy of this wuident.cab file is available at https://go.microsoft.com/fwlink/?linkid=2111079. Changed to: At the start of the protocol, a client that implements the self-update protocol SHOULD<31> check if it needs to self-update. Client/server communications can fail if the server does not expose the self-update content directory, as specified in this section. As specified in section 2, the server implementation MUST expose the self-update content directory as a virtual directory. Clients that support the self-update protocol issue HTTP GET

Errata Published*	Description
	requests (as specified in [RFC2616] section 9.3) to obtain files from the self-update content directory; therefore, the server MUST support HTTP requests on this virtual directory to support these clients. The files that the client accesses in the self-update directory, and the way in which the client uses them, are implementation-specific.<32> <31> Section 3.1.5.1: Only Windows 8.1 and earlier Windows versions use this self-update mechanism. <32> Windows 8.1 and earlier Windows versions perform self-update using the contents of the self-update content directory. Since these versions of Windows require that these files be Microsoft-signed, the files cannot be modified. For information on how to receive a redistributable copy of these files, contact dochelp@microsoft.com. Windows 10 does not perform self-update and does not access the files in the self-update content directory. For reference, here is a summary of how Windows 8.1 and earlier Windows versions perform self-update. 1. The client retrieves the file wuident.cab from the self-update content directory and verifies that it is signed by Microsoft. If the wuident.cab file is not present or is not signed by Microsoft, the client stops self-update and does not further interoperate with the server. 2. The client retrieves the file wuident.txt from the wuident.cab file. 3. The client compares the version number of the current version of the Windows Update client files to the versions listed in the [OS], [OS2], and [OS3] sections of the wuident.txt file, and finds the matching line in the file. 4. If the matching line in wuident.txt contains the directive "/SKIP", then the client is already up-to-date and self-update is complete. 5. If the matching line in wuident contains the directive "/UNSUPPORTED", then the client cannot self-update and does not further interoperate with the server. 6. If the matching line in wuident contains neither "/SKIP" nor "/UNSUPPORTED", then it specifies a subdirectory. 7. The client compares the architecture of the current version of windows to the list of architectures in the [ARCH] section of the wuident.txt file and retrieves a subdirectory name from the matching line. 8. The client compares the primary language of the current version of windows to the list of languages in the [LANG] section of the wuident.txt file and retrieves a subdirectory name from the matching line. 9. The client assembles the full subdirectory path from the three subdirectory names it has retrieved, using the format specified in the StructureKeyEx, StructureKeyEx2, or StructureKeyEx3 lines in the [SelfClientUpdate] section of the wuident.txt file. 10. The client accesses the resulting self-directory within the self-update content directory. This directory will contain a set of several files containing the appropriate Windows Update client binaries for the current version, architecture, and language of Windows. 11. The client verifies that these files are Microsoft-signed, unpacks the contents into a secure directory, and executes them to update the client binaries. 12. If the installation succeeds, then self-update is complete. 13. If the installation fails, then the client cannot self-update and does not further interoperate with the server.
2019/12/09	In Section 3.1.1.1, Populating the Data Model, the text was updated to include the OSUpgrade attribute in the description of the concatenated strings of the XMLFragment component of both the Core and Extended metadata entries. Changed from:

Errata Published*	Description
	Core: There is exactly one "Core" entry created from the revisions metadata. RevisionID: References the entry in the revision table for the revisions metadata. FragmentType: Core. Locale: NULL. XmlFragment: MUST be derived from the original metadata by: • Collecting the following XmlNodes: • The XmlNode identified by XPATH /Update/UpdateIdentity • The XmlNode identified by XPATH /Update/Properties, all attributes removed except: UpdateType, ExplicitlyDeployable, AutoSelectOnWebSites, and EulaID ... Extended: There is exactly one "Extended" entry created from the revisions metadata. RevisionID: References the entry in the revision table for the revisions metadata. FragmentType: Extended. Locale: NULL. XmlFragment: MUST be derived from the original metadata by concatenating the following strings together after removing all XML namespace definitions from each string (the result of which is not well-formed XML): • The XmlNode identified by XPATH /Update/Properties, with the following attributes removed: UpdateType, ExplicitlyDeployable, AutoSelectOnWebSites, EulaID, PublicationState, PublisherID, CreationDate, IsPublic, LegacyName, DetectoidType... Changed to: Core: There is exactly one "Core" entry created from the revisions metadata. RevisionID: References the entry in the revision table for the revisions metadata. FragmentType: Core. Locale: NULL. XmlFragment: MUST be derived from the original metadata by: • Collecting the following XmlNodes: • The XmlNode identified by XPATH /Update/UpdateIdentity • The XmlNode identified by XPATH /Update/Properties, all attributes removed except: UpdateType, ExplicitlyDeployable, AutoSelectOnWebSites, OSUpgrade, and EulaID ... Extended: There is exactly one "Extended" entry created from the revisions metadata. RevisionID: References the entry in the revision table for the revisions metadata. FragmentType: Extended. Locale: NULL. XmlFragment: MUST be derived from the original metadata by concatenating the following strings together after removing all XML namespace definitions from each string (the result of which is not well-formed XML): • The XmlNode identified by XPATH /Update/Properties, with the following attributes removed: UpdateType, ExplicitlyDeployable, AutoSelectOnWebSites, EulaID, PublicationState, PublisherID, CreationDate, IsPublic, LegacyName, DetectoidType, OSUpgrade.

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>...</td></tr>
<tr><td>2019/11/25</td><td>In Section 3.1.5.1, Self-Update, the text was changed to provide information as to where the update files could be accessed.

Changed from:

At the start of the protocol, the client MUST check if it needs to self-update. Client/server communications fail if the server does not expose the self-update content directory, as specified in this section.

As specified in section 2, the server implementation MUST expose the self-update content directory as a virtual directory. The client issues HTTP GET requests (as specified in [RFC2616] section 9.3) to obtain files from the self-update content directory; therefore, the server MUST support HTTP requests on this virtual directory.

Changed to:

At the start of the protocol, the client SHOULD<31> check if it needs to self-update. Client/server communications can fail if the server does not expose the self-update content directory, as specified in this section.

As specified in section 2, the server implementation MUST expose the self-update content directory as a virtual directory. The client issues HTTP GET requests (as specified in [RFC2616] section 9.3) to obtain files from the self-update content directory; therefore, the server MUST support HTTP requests on this virtual directory. The directory MUST include the wuident.cab file, which is implementation-specific<32>.

<31>Only Windows 8.1 and earlier Windows versions use this self-update mechanism.

<32>Windows Server Update Services includes a version of wuident.cab that is compatible with clients running Windows 8.1 and earlier. A current copy of this wuident.cab file is available at https://go.microsoft.com/fwlink/?linkid=2111079.</td></tr>
<tr><td>2019/11/11</td><td>In Section 2.2.2.3.1, ReportEventBatch, an event was added to the Install Events section of the EventID table.

Added:

<table><tr><td>AGENT_INSTALLING_STARTED</td><td>181</td><td>User initiated installation started</td><td></td><td>Installation Started:Windows has started installing the following update: %1</td></tr></table></td></tr>
<tr><td>2019/06/24</td><td>In this document, the properties of the LastChangeTime element of the Deployment complex type have been updated to indicate that it is mandatory.

In Section 2.2.2.2.4, SyncUpdates, the minOccurs value for the LastChangeTime element of the Deployment complex type has been changed from "0" to "1".

Changed from:</td></tr>
</table>

Errata Published*	Description

	…
	
	 `<s:complexType name="Deployment">`
	`  <s:sequence>`
	`  <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:int" />`
	`  <s:element minOccurs="1" maxOccurs="1" name="Action"`
	`  type="s1:DeploymentAction" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="Deadline"`
	`  type="s:string" />`
	`  <s:element minOccurs="1" maxOccurs="1" name="IsAssigned"`
	`  type="s:boolean" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="LastChangeTime"`
	`  type="s:string" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="DownloadPriority"`
	`  type="s:string" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="HardwareIds"`
	`  type="s1:ArrayOfString" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="AutoSelect"`
	`  type="s:string" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="AutoDownload"`
	`  type="s:string" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="SupersedenceBehavior"`
	`  type="s:string" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="FlagBitmask"`
	`  type="s:string" />`
	`  </s:sequence>`
	`  </s:complexType>`
	
	…
	
	LastChangeTime: Specifies when the deployment was created, in the syntax specified for s:date (as specified in [XMLSCHEMA2] section 3.2.9).
	
	…
	
	Changed to:
	
	…
	
	 `<s:complexType name="Deployment">`
	`  <s:sequence>`
	`  <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:int" />`
	`  <s:element minOccurs="1" maxOccurs="1" name="Action"`
	`  type="s1:DeploymentAction" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="Deadline"`
	`  type="s:string" />`
	`  <s:element minOccurs="1" maxOccurs="1" name="IsAssigned"`
	`  type="s:boolean" />`
	`  <s:element minOccurs="1" maxOccurs="1" name="LastChangeTime"`
	`  type="s:string" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="DownloadPriority"`
	`  type="s:string" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="HardwareIds"`
	`  type="s1:ArrayOfString" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="AutoSelect"`
	`  type="s:string" />`
	`  <s:element minOccurs="0" maxOccurs="1" name="AutoDownload"`
	`  type="s:string" />`

Errata Published*	Description
	<s:element minOccurs="0" maxOccurs="1" name="SupersedenceBehavior" type="s:string" /> <s:element minOccurs="0" maxOccurs="1" name="FlagBitmask" type="s:string" /> </s:sequence> </s:complexType> … LastChangeTime: Specifies when the deployment was created, in the syntax specified for s:date (as specified in [XMLSCHEMA2] section 3.2.9). This element MUST be present. … In Section 6.2, Client Web Service WSDL, changed from: … <s:element minOccurs="0" maxOccurs="1" name="LastChangeTime" type="s:string" /> … Changed to: … <s:element minOccurs="1" maxOccurs="1" name="LastChangeTime" type="s:string" /> …
2019/04/29	In Section 3.1.5.7, SyncUpdates, and Section 3.1.5.12, SyncPrinterCatalog, a note has been added to clarify that only one revision can be sent for a given update. In Section 3.1.5.7, SyncUpdates, changed from: … Xml: The revision's associated "core" metadata (FragmentType = "Core"). SyncUpdatesResponse.OutOfScopeRevisionIDs: Populated with the IDs of revision that are in the CachedRevisions list that are not in the NeededRevisions list. … Changed to: ... Xml: The revision's associated "core" metadata (FragmentType = "Core"). Note: The server implementation MUST send no more than one Revision for a given Update. It is recommended that the implementation SHOULD, in the event of multiple matches, select only the latest Revision (the one with the highest revision number). SyncUpdatesResponse.OutOfScopeRevisionIDs: Populated with the IDs of revision that are in the CachedRevisions list that are not in the NeededRevisions list.

Errata Published*	Description
	... In Section 3.1.5.12, SyncPrinterCatalog, changed from: … Xml: The "core" metadata (FragmentType = "Core") associated with the revision. SyncPrinterCatalogResponse.OutOfScopeRevisionIDs: Populated with the revision's IDs in the CachedRevisions list that are not in the NeededRevisions list. ... Changed to: … Xml: The "core" metadata (FragmentType = "Core") associated with the revision. Note: The server implementation MUST send no more than one Revision for a given Update. It is recommended that the implementation SHOULD, in the event of multiple matches, select only the latest Revision (the one with the highest revision number). SyncPrinterCatalogResponse.OutOfScopeRevisionIDs: Populated with the revision's IDs in the CachedRevisions list that are not in the NeededRevisions list. ...

*Date format: YYYY/MM/DD

[MS-XCA]: Xpress Compression Algorithm

This topic lists the Errata found in [MS-XCA] 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 V5.0 – 2018/09/12.

Errata Published*	Description

2020/02/17	In Section 2.3.4, Processing, we updated the pseudocode for the encoding method for match lengths greater than 65535.
	
	Changed from:
	
	`If MatchLength >= 7`
	`  MatchLength -= 7`
	`  If LastLengthHalfByte == 0`
	`  LastLengthHalfByte = OutputPosition`
	`  Write the byte value min(MatchLength, 15) to OutputPosition`
	`  OutputPosition += 1`
	`  Else`
	`  OutputBuffer[LastLengthHalfByte] |= min(15, MatchLength) << 4`
	`  LastLengthHalfByte = 0`
	`  If MatchLength >= 15`
	`  MatchLength -= 15`
	`  Write the byte value min(MatchLength, 255) to OutputPosition`
	`  OutputPosition += 1`
	`  If MatchLength >= 255`
	`  MatchLength += 15 + 7`
	`  Write the 2-byte value MatchLength to OutputPosition`
	`  OutputPosition += 2`
	
	Changed to:
	
	`If MatchLength < 7`
	`  // This is the simple case. The length fits in 3 bits.`

Errata Published*	Description

```
        MatchOffset += MatchLength
        Write MatchOffset the 2-byte value to OutputPosition
        OutputPosition += 2
    Else
        // The length does not fit 3 bits. Record a special value to
        // indicate a longer length.
        MatchOffset |= 7
        Write MatchOffset the 2-byte value to OutputPosition
        OutputPosition += 2
        MatchLength -= 7
        // Try to encode the length in the next 4 bits. If we previously
        // encoded a 4-bit length, we'll use the high 4 bits from that byte.
        If LastLengthHalfByte == 0
            LastLengthHalfByte = OutputPosition
            If MatchLength < 15
                Write single byte value of MatchLength to OutputPosition
                OutputPosition += 1
            Else
                Write single byte value of 15 to OutputPosition
                OutputPosition++
                goto EncodeExtraLen
        Else
            If MatchLength < 15
                OutputBuffer[LastLengthHalfByte] |= MatchLength << 4
                LastLengthHalfByte = 0
            Else
                OutputBuffer[LastLengthHalfByte] |= 15 << 4
                LastLengthHalfByte = 0
    EncodeExtraLen:
                // We've already used 3 bits + 4 bits to encode the length
                // Next use the next byte.
                MatchLength -= 15
                If MatchLength < 255
```

Errata Published*	Description

	`Write single byte value of MatchLength to OutputPosition`
	`OutputPosition += 1`
	`Else`
	`// Use two more bytes for the length`
	`Write single byte value of 255 to OutputPosition`
	`OutputPosition += 1`
	`MatchLength += 7 + 15`
	`If MatchLength < (1 << 16)`
	`Write two-byte value MatchLength to OutputPosition`
	`OutputPosition += 2`
	`Else`
	`Write two-byte value of 0 to OutputPosition`
	`OutputPosition += 2`
	`Write four-byte value of MatchLength to OutputPosition`
	`OutputPosition += 4`

2020/02/17	In Section 2.3.4 Processing, we added clarifying information about the maximum MatchLength. Changed from: The fastest variant of the Xpress Compression Algorithm avoids the cost of the Huffman[IEEE-MRC] pass by encoding the LZ77 [UASDC] literals and matches in a simple way. The encoding process is similar to the method described in section 2.1.4.1, with the key difference that the largest match offset it can encode is 8192 instead of the 65535 limit of the Huffman format. The literal or match flags are encoded in 32-bit chunks. Literals are encoded with a simple byte value. Matches are encoded with a 16-bit value, where the high 13 bits represent the offset and the low 3 bits represent the length. Long lengths are encoded with an additional 4 bits, then 8 bits, and then 16 bits. The following pseudocode provides an outline of the encoding method. Changed to: The fastest variant of the Xpress Compression Algorithm avoids the cost of the Huffman[IEEE-MRC] pass by encoding the LZ77 [UASDC] literals and matches in a simple way. The encoding process is similar to the method described in section 2.1.4.1, with the key difference that the largest match offset it can encode is 8192 instead of the 65535 limit of the Huffman format. The literal or match flags are encoded in 32-bit chunks. Literals are encoded with a simple byte value. Matches are encoded with a 16-bit value, where the high 13 bits represent the offset and the low 3 bits represent the length. Long lengths are encoded with an additional 4 bits, then 8 bits, and then 16 bits. The MatchLength is represented by a ULONG, a 32-bit unsigned integer (see [MS-DTYP] section 2.2.51); therefore, the maximum value is 4,294,967,295. The following pseudocode provides an outline of the encoding method.
2020/02/17	In Section 2.2.4 Processing, we corrected the pseudocode by replacing DecodedValue with

Errata Published*	Description
	HuffmanSymbol and added a clarifying comment to the pseudocode to explain why the HuffmanSymbol needs to be right-shifted by 4 bits. Changed from: … Loop until a decompression terminating condition Build the decoding table CurrentPosition = 256 // start at the end of the Huffman table NextBits = Read16Bits(InputBuffer + CurrentPosition) CurrentPosition += 2 NextBits <<= 16 NextBits |= Read16Bits(InputBuffer + CurrentPosition) CurrentPosition += 2 ExtraBits = 16 BlockEnd = OutputPosition + 65536 Loop until a block terminating condition If OutputPosition >= BlockEnd then terminate block processing Loop until a literal processing terminating condition Next15Bits = NextBits >> (32 – 15) HuffmanSymbol = DecodingTable[Next15Bits] HuffmanSymbolBitLength = the bit length of HuffmanSymbol, from the table in the input buffer If HuffmanSymbol <= 0 NextBits <<= HuffmanSymbolBitLength ExtraBits -= HuffmanSymbolBitLength Do HuffmanSymbol = - HuffmanSymbol HuffmanSymbol += (NextBits >> 31) NextBits *= 2 ExtraBits = ExtraBits - 1 HuffmanSymbol = DecodingTable[HuffmanSymbol] **While DecodedValue <= 0** Else **DecodedBitCount = DecodedValue & 15**

Errata Published*	Description
	NextBits <<= DecodedBitCount ExtraBits -= DedcodedBitCount HuffmanSymbol >>= 4 HuffmanSymbol -= 256 If ExtraBits < 0 NextBits |= Read16Bits(InputBuffer + CurrentPosition) << (-ExtraBits) ExtraBits += 16 CurrentPosition += 2 If HuffmanSymbol >= 0 If HuffmanSymbol == 0 If the entire input buffer has been read and the expected decompressed size has been written to the output buffer Decompression is complete. Return with success. Terminate literal processing Else Output the byte value of HuffmanSymbol to the output stream End of literal processing Loop MatchLength = HuffmanSymbol mod 16 MatchOffsetBitLength = HuffmanSymbol / 16 If MatchLength == 15 MatchLength = ReadByte(InputBuffer + CurrentPosition) CurrentPosition += 1 If MatchLength == 255 MatchLength = Read16Bits(InputBuffer + CurrentPosition) CurrentPosition += 2 If MatchLength < 15 The compressed data is invalid. Return error. MatchLength = MatchLength - 15 MatchLength = MatchLength + 15 MatchLength = MatchLength + 3 MatchOffset = NextBits >> (32 – MatchOffsetBitLength) MatchOffset += (1 << MatchOffsetBitLength)

Errata Published*	Description

	``
	NextBits <<= MatchOffsetBitLength
	ExtraBits -= MatchOffsetBitLength
	If ExtraBits < 0
	NextBits |= Read16Bits(InputBuffer + CurrentPosition) << (-
	ExtraBits)
	ExtraBits += 16
	CurrentPosition += 2
	For i = 0 to MatchLength - 1
	Output OutputBuffer[CurrentOutputPosition – MatchOffset + i]
	End of block loop
	End of decoding loop
	``
	
	Changed to:
	
	``
	…
	Loop until a decompression terminating condition
	Build the decoding table
	CurrentPosition = 256 // start at the end of the Huffman table
	NextBits = Read16Bits(InputBuffer + CurrentPosition)
	CurrentPosition += 2
	NextBits <<= 16
	NextBits |= Read16Bits(InputBuffer + CurrentPosition)
	CurrentPosition += 2
	ExtraBits = 16
	BlockEnd = OutputPosition + 65536
	Loop until a block terminating condition
	If OutputPosition >= BlockEnd then terminate block processing
	Loop until a literal processing terminating condition
	Next15Bits = NextBits >> (32 – 15)
	HuffmanSymbol = DecodingTable[Next15Bits]
	HuffmanSymbolBitLength = the bit length of HuffmanSymbol, from the
	table in
	the input buffer
	If HuffmanSymbol <= 0
	``
	
	…
	Loop until a decompression terminating condition
	Build the decoding table
	CurrentPosition = 256 // start at the end of the Huffman table
	NextBits = Read16Bits(InputBuffer + CurrentPosition)
	CurrentPosition += 2
	NextBits <<= 16
	NextBits |= Read16Bits(InputBuffer + CurrentPosition)
	CurrentPosition += 2
	ExtraBits = 16
	BlockEnd = OutputPosition + 65536
	Loop until a block terminating condition
	If OutputPosition >= BlockEnd then terminate block processing
	Loop until a literal processing terminating condition
	Next15Bits = NextBits >> (32 – 15)
	HuffmanSymbol = DecodingTable[Next15Bits]
	HuffmanSymbolBitLength = the bit length of HuffmanSymbol, from the
	table in
	the input buffer
	If HuffmanSymbol <= 0

Errata Published*	Description
	NextBits <<= HuffmanSymbolBitLength ExtraBits -= HuffmanSymbolBitLength Do HuffmanSymbol = - HuffmanSymbol HuffmanSymbol += (NextBits >> 31) NextBits *= 2 ExtraBits = ExtraBits - 1 HuffmanSymbol = DecodingTable[HuffmanSymbol] **While HuffmanSymbol <= 0** Else **DecodedBitCount = HuffmanSymbol & 15** NextBits <<= DecodedBitCount ExtraBits -= DedcodedBitCount HuffmanSymbol >>= 4 // Shift by 4 bits to get the symbol value // (the lower 4 bits are the bit length of the symbol) HuffmanSymbol -= 256 If ExtraBits < 0 NextBits |= Read16Bits(InputBuffer + CurrentPosition) << (-ExtraBits) ExtraBits += 16 CurrentPosition += 2 If HuffmanSymbol >= 0 If HuffmanSymbol == 0 If the entire input buffer has been read and the expected decompressed size has been written to the output buffer Decompression is complete. Return with success. Terminate literal processing Else Output the byte value of HuffmanSymbol to the output stream End of literal processing Loop MatchLength = HuffmanSymbol mod 16 MatchOffsetBitLength = HuffmanSymbol / 16 If MatchLength == 15

Errata Published*	Description

	`MatchLength = ReadByte(InputBuffer + CurrentPosition)`
	`CurrentPosition += 1`
	`If MatchLength == 255`
	`MatchLength = Read16Bits(InputBuffer + CurrentPosition)`
	`CurrentPosition += 2`
	`If MatchLength < 15`
	`The compressed data is invalid. Return error.`
	`MatchLength = MatchLength - 15`
	`MatchLength = MatchLength + 15`
	`MatchLength = MatchLength + 3`
	`MatchOffset = NextBits >> (32 – MatchOffsetBitLength)`
	`MatchOffset += (1 << MatchOffsetBitLength)`
	`NextBits <<= MatchOffsetBitLength`
	`ExtraBits -= MatchOffsetBitLength`
	`If ExtraBits < 0`
	`NextBits |= Read16Bits(InputBuffer + CurrentPosition) << (-ExtraBits)`
	`ExtraBits += 16`
	`CurrentPosition += 2`
	`For i = 0 to MatchLength - 1`
	`Output OutputBuffer[CurrentOutputPosition – MatchOffset + i]`
	`End of block loop`
	`End of decoding loop`

2019/12/09	In Section 2.1, LZ77+Huffman Compression Algorithm Details, described how data is processed for the Huffman variant. Changed from: The overall compression algorithm for the Huffman [IEEE-MRC] variant can be divided into three stages, which are performed in this order: ... Changed to: The overall compression algorithm for the Huffman [IEEE-MRC] variant can handle an arbitrary amount of data. Data is processed in 64k blocks, and the encoded results are stored in-order. After the final block, the end-of-file (EOF) symbol is encoded. Each 64k block is run through three stages, which are performed in this order:

Errata Published*	Description

	...
	
	In Section 2.2.4, Processing, described the decompression process and clarified how the compression stream handles the bytes for long match lengths in the pseudocode.
	
	Changed from:
	The decompression algorithm uses the 256-byte Huffman table to reconstruct the canonical Huffman [IEEE-MRC] representations of each symbol. Next, the Huffman stream of LZ77 ([UASDC]) literals and matches is decoded to reproduce the original data.
	
	The following method can be used to construct a decoding table. The decoding table will have 2^15 entries because 15 is the maximum bit length permitted by the Xpress Compression Algorithm for a Huffman code. If a symbol has a bit length of X, it has 2^(15 – X) entries in the table that point to its value. The order of symbols in the table is sorted by bit length (from low to high), and then by symbol value (from low to high). These requirements represent the agreement of canonicalness with the compression end of the algorithm. The following pseudocode shows the table construction method:
	...
	The compression stream is designed to be read in (mostly) 16-bit chunks, with a 32-bit register maintaining at least the next 16 bits of input. This strategy allows the code to seamlessly handle the bytes for long match lengths, which would otherwise be awkward. The following pseudocode demonstrates this method.
	
	Build the decoding table
	CurrentPosition = 256 // start at the end of the Huffman table
	NextBits = Read16Bits(InputBuffer + CurrentPosition)
	CurrentPosition += 2
	NextBits <<= 16
	NextBits |= Read16Bits(InputBuffer + CurrentPosition)
	CurrentPosition += 2
	ExtraBits = 16
	Loop until a terminating condition
	  Next15Bits = NextBits >> (32 – 15)
	  HuffmanSymbol = DecodingTable[Next15Bits]
	  HuffmanSymbolBitLength = the bit length of HuffmanSymbol, from the table in
	  the input buffer
	  NextBits <<= HuffmanSymbolBitLength
	  ExtraBits -= HuffmanSymbolBitLength
	  If ExtraBits < 0
	  NextBits |= Read16Bits(InputBuffer + CurrentPosition) << (-ExtraBits)
	  ExtraBits += 16
	  CurrentPosition += 2
	  If HuffmanSymbol < 256
	  Output the byte value HuffmanSymbol to the output stream.
	  Else If HuffmanSymbol == 256 and
	  the entire input buffer has been read and
	  the expected decompressed size has been written to the output buffer
	  Decompression is complete. Return with success.
	  Else

Errata Published*	Description

	HuffmanSymbol = HuffmanSymbol - 256
	MatchLength = HuffmanSymbol mod 16
	MatchOffsetBitLength = HuffmanSymbol / 16
	If MatchLength == 15
	MatchLength = ReadByte(InputBuffer + CurrentPosition)
	CurrentPosition += 1
	If MatchLength == 255
	MatchLength = Read16Bits(InputBuffer + CurrentPosition)
	CurrentPosition += 2
	If MatchLength < 15
	The compressed data is invalid. Return error.
	MatchLength = MatchLength - 15
	MatchLength = MatchLength + 15
	MatchLength = MatchLength + 3
	MatchOffset = NextBits >> (32 – MatchOffsetBitLength)
	MatchOffset += (1 << MatchOffsetBitLength)
	NextBits <<= MatchOffsetBitLength
	ExtraBits -= MatchOffsetBitLength
	If ExtraBits < 0
	Read the next 2 bytes the same as the preceding (ExtraBits < 0) case
	For i = 0 to MatchLength - 1
	Output OutputBuffer[CurrentOutputPosition – MatchOffset + i]
	...
	
	Changed to:
	The decompression processes a series of blocks to form the decompressed output. Each block is processed in-order, and its decoded content written to the output stream is in-order. When processing a block, we check for terminating conditions for both block and overall decoding.
	
	The decompression algorithm uses the 256-byte Huffman table to reconstruct the canonical Huffman [IEEE-MRC] representations of each symbol. Next, the Huffman stream of LZ77 ([UASDC]) literals and matches is decoded to reproduce the original data.
	
	The following method can be used to construct a decoding table. The decoding table will have 2^15 entries because 15 is the maximum bit length permitted by the Xpress Compression Algorithm for a Huffman code. If a symbol has a bit length of X, it has 2^(15 – X) entries in the table that point to its value. The order of symbols in the table is sorted by bit length (from low to high), and then by symbol value (from low to high). These requirements represent the agreement of canonicalness with the compression end of the algorithm. The following pseudocode shows the table construction method:
	...
	The compression stream is designed to be read in (mostly) 16-bit chunks, with a 32-bit register maintaining at least the next 16 bits of input. This strategy allows the code to seamlessly handle the bytes for long match lengths, which would otherwise be awkward. The following pseudocode demonstrates this method.
	
	Loop until a decompression terminating condition
	Build the decoding table
	CurrentPosition = 256 // start at the end of the Huffman table
	NextBits = Read16Bits(InputBuffer + CurrentPosition)

Errata Published*	Description

	CurrentPosition += 2
	NextBits <<= 16
	NextBits |= Read16Bits(InputBuffer + CurrentPosition)
	CurrentPosition += 2
	ExtraBits = 16
	BlockEnd = OutputPosition + 65536
	
	Loop until a block terminating condition
	  If OutputPosition >= BlockEnd then terminate block processing
	  Loop until a literal processing terminating condition
	  Next15Bits = NextBits >> (32 – 15)
	  HuffmanSymbol = DecodingTable[Next15Bits]
	  HuffmanSymbolBitLength = the bit length of HuffmanSymbol, from the table in
	  the input buffer
	  If HuffmanSymbol <= 0
	  NextBits <<= HuffmanSymbolBitLength
	  ExtraBits -= HuffmanSymbolBitLength
	
	  Do
	  HuffmanSymbol = - HuffmanSymbol
	  HuffmanSymbol += (NextBits >> 31)
	  NextBits *= 2
	  ExtraBits = ExtraBits - 1
	  HuffmanSymbol = DecodingTable[HuffmanSymbol]
	  While DecodedValue <= 0
	  Else
	  DecodedBitCount = DecodedValue & 15
	  NextBits <<= DecodedBitCount
	  ExtraBits -= DedcodedBitCount
	  HuffmanSymbol >>= 4
	  HuffmanSymbol -= 256
	  If ExtraBits < 0
	  NextBits |= Read16Bits(InputBuffer + CurrentPosition) << (-ExtraBits)
	  ExtraBits += 16
	  CurrentPosition += 2
	  If HuffmanSymbol >= 0
	  If HuffmanSymbol == 0
	  If the entire input buffer has been read and
	  the expected decompressed size has been written to the output buffer
	  Decompression is complete. Return with success.
	  Terminate literal processing
	  Else
	  Output the byte value of HuffmanSymbol to the output stream
	  End of literal processing Loop
	
	  MatchLength = HuffmanSymbol mod 16
	  MatchOffsetBitLength = HuffmanSymbol / 16

Errata Published*	Description
	If MatchLength == 15 MatchLength = ReadByte(InputBuffer + CurrentPosition) CurrentPosition += 1 If MatchLength == 255 MatchLength = Read16Bits(InputBuffer + CurrentPosition) CurrentPosition += 2 If MatchLength < 15 The compressed data is invalid. Return error. MatchLength = MatchLength - 15 MatchLength = MatchLength + 15 MatchLength = MatchLength + 3 MatchOffset = NextBits >> (32 – MatchOffsetBitLength) MatchOffset += (1 << MatchOffsetBitLength) NextBits <<= MatchOffsetBitLength ExtraBits -= MatchOffsetBitLength If ExtraBits < 0 NextBits |= Read16Bits(InputBuffer + CurrentPosition) << (-ExtraBits) ExtraBits += 16 CurrentPosition += 2 For i = 0 to MatchLength - 1 Output OutputBuffer[CurrentOutputPosition – MatchOffset + i] End of block loop End of decoding loop ...

2019/09/02	In Section 2.4.4, Processing, pseudocode supporting longer matches has been updated
	
	Changed from:
	...
	The match length can be greater than the match offset, and this necessitates the 1-byte-at-a-time copying strategy shown in the following pseudocode.
	
	`BufferedFlags = 0`
	`BufferedFlagCount = 0`
	`InputPosition = 0`
	`OutputPosition = 0`
	`LastLengthHalfByte = 0`
	`Loop until break instruction or error`
	`If BufferedFlagCount == 0`
	`BufferedFlags = read 4 bytes at InputPosition`
	`InputPosition += 4`
	`BufferedFlagCount = 32`
	`BufferedFlagCount = BufferedFlagCount – 1`
	`If (BufferedFlags & (1 << BufferedFlagCount)) == 0`
	`Copy 1 byte from InputPosition to OutputPosition.  Advance both.`
	`Else`
	`If InputPosition == InputBufferSize`
	`Decompression is complete.  Return with success.`
	`MatchBytes = read 2 bytes from InputPosition`
	`InputPosition += 2`
	`MatchLength = MatchBytes mod 8`
	`MatchOffset = (MatchBytes / 8) + 1`

Errata Published*	Description

	<pre>If MatchLength == 7
	    If LastLengthHalfByte == 0
	        MatchLength = read 1 byte from InputPosition
	        MatchLength = MatchLength mod 16
	        LastLengthHalfByte = InputPosition
	        InputPosition += 1
	    Else
	        MatchLength = read 1 byte from LastLengthHalfByte position
	        MatchLength = MatchLength / 16
	        LastLengthHalfByte = 0
	    If MatchLength == 15
	        MatchLength = read 1 byte from InputPosition
	        InputPosition += 1
	        If MatchLength == 255
	            MatchLength = read 2 bytes from InputPosition
	            InputPosition += 2
	            If MatchLength < 15 + 7
	               Return error.
	            MatchLength -= (15 + 7)
	        MatchLength += 15
	    MatchLength += 7
	MatchLength += 3
	For i = 0 to MatchLength - 1
	    Copy 1 byte from OutputBuffer[OutputPosition - MatchOffset]
	    OutputPosition += 1</pre>
	Changed to:
	...
	The match length can be greater than the match offset, and this necessitates the 1-byte-at-a-time copying strategy shown in the following pseudocode.
	
	<pre>BufferedFlags = 0
	BufferedFlagCount = 0
	InputPosition = 0
	OutputPosition = 0
	LastLengthHalfByte = 0
	Loop until break instruction or error
	    If BufferedFlagCount == 0
	        BufferedFlags = read 4 bytes at InputPosition
	        InputPosition += 4
	        BufferedFlagCount = 32
	    BufferedFlagCount = BufferedFlagCount - 1
	    If (BufferedFlags & (1 << BufferedFlagCount)) == 0
	        Copy 1 byte from InputPosition to OutputPosition. Advance both.
	    Else
	        If InputPosition == InputBufferSize
	            Decompression is complete. Return with success.
	        MatchBytes = read 2 bytes from InputPosition
	        InputPosition += 2
	        MatchLength = MatchBytes mod 8
	        MatchOffset = (MatchBytes / 8) + 1
	        If MatchLength == 7
	            If LastLengthHalfByte == 0
	                MatchLength = read 1 byte from InputPosition
	                MatchLength = MatchLength mod 16
	                LastLengthHalfByte = InputPosition
	                InputPosition += 1
	            Else
	                MatchLength = read 1 byte from LastLengthHalfByte position
	                MatchLength = MatchLength / 16
	                LastLengthHalfByte = 0</pre>

Errata Published*	Description

	<pre>If MatchLength == 15
	MatchLength = read 1 byte from InputPosition
	InputPosition += 1
	If MatchLength == 255
	MatchLength = read 2 bytes from InputPosition
	InputPosition += 2
	If MatchLength == 0
	MatchLength = read 4 bytes from InputPosition
	InputPosition += 4 bytes
	If MatchLength < 15 + 7
	Return error.
	MatchLength -= (15 + 7)
	MatchLength += 15
	MatchLength += 7
	MatchLength += 3
	For i = 0 to MatchLength - 1
	Copy 1 byte from OutputBuffer[OutputPosition - MatchOffset]
	OutputPosition += 1</pre>

2019/07/08	In Section 2.1.4.2, Huffman Code Construction Phase, clarified that the sorting algorithm used in the Huffman Code construction phase is stable. Changed from: ... The following flowchart illustrates the length-limited canonical Huffman code construction method. ... Changed to: ... The following flowchart illustrates the length-limited canonical Huffman code construction method. Note that the sorting algorithm used in the Huffman Code construction phase is stable. ...
2019/07/08	In Section 2.1.4.3 Final Encoding Phase, clarified that some implementations of the decompression algorithm expect a terminating Huffman symbol and that it is recommended the encoding algorithm append this symbol. Changed from: Some implementations of the decompression algorithm expect an extra symbol to mark the end of the data. For example, certain implementations fail during decompression if the Huffman symbol 256 is not found after the actual data. For this reason, the encoding algorithm appends this symbol and increments the count of symbol 256 before the Huffman codes are constructed. Changed to: Implementations of the decompression algorithm may expect an extra symbol to mark the end of the data. For example, certain implementations fail during decompression if the Huffman symbol 256 is not found after the actual data. For this reason, the encoding algorithm SHOULD append this symbol and increment the count of symbol 256 before the Huffman codes are constructed.

*Date format: YYYY/MM/DD

[MS-XCEP]: X.509 Certificate Enrollment Policy Protocol

This topic lists the Errata found in [MS-XCEP] 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

No errata are available for the latest version of this Windows Protocols document. To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download