Errata in Open Specification Protocol Documentation

The Errata pages may be updated more frequently than the protocols documents. To receive notifications of changes to Errata pages, you can subscribe to these RSS or Atom feeds. **Errata are subject to the same terms as the Open Specifications documentation referenced.**	RSS Atom

The pages below contain the Errata for the indicated protocols documents since those documents were last published. For more information, please contact dochelp@microsoft.com.

What are Errata?

Errata are content issues in previously 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.

What kinds of issues are not included in Errata?

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

Windows Protocols Errata

Office Protocols Errata

SharePoint Protocols Errata

Exchange Server Protocols Errata

SQL Server Protocols Errata

Contents

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.

Windows Protocols Errata for Protocols documents titled [MS-A…] through [MS-L…]

Windows Protocols Errata for Protocols documents titled [MS-M…] through [MS-R…]

Windows Protocols Errata for Protocols documents titled [MS-S…] through [MS-Z…]

Windows Protocols Errata A-L

This topic lists the Errata found in the Windows Protocols Technical Specifications, Overview Documents, and Reference documents titled [MS-A…] **through** [MS-L…] **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.** **To find out more about the types of issues that are included in Errata, see Windows Protocols Errata.** **Errata are subject to the same terms as the Open Specifications documentation referenced.**	**RSS** **Atom**

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

[MS-ADSC]: Active Directory Schema Classes

[MS-ADTS]: Active Directory Technical Specification

[MS-AIPS]: Authenticated Internet Protocol

[MS-APDS]: Authentication Protocol Domain Support

[MS-AZOD]: Authorization Protocols Overview

[MS-BKRP]: BackupKey Remote Protocol

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

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

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

[MS-CSRA]: Certificate Services Remote Administration Protocol

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

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

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

[MS-DTYP]: Windows Data Types

[MS-DVRE]: Device Registration Enrollment Protocol

[MS-ECS]: Enterprise Client Synchronization Protocol

[MS-EMF]: Enhanced Metafile Format

[MS-EMFPLUS]: Enhanced Metafile Format Plus Extensions

[MS-FSA]: File System Algorithms

[MS-FSCC]: File System Control Codes

[MS-FSRVP]: File Server Remote VSS Protocol

[MS-FSVCA]: File Set Version Comparison Algorithms

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

[MS-GPOL]: Group Policy: Core Protocol

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

[MS-KILE]: Kerberos Protocol Extensions

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

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/05/11	V3.0 – 2014/05/15	In Section 6, Appendix A: Full JSON Schema, updated the "Serialized Request with Certificate" object to add a missing “Content” property as follows. Changed from: { "title" : "Serialized Request with Certificate", "type" : "object", "properties" : { "Request" :

Errata Published YYYY/MM/DD	Protocol Document Version	Description

		<pre>{
		"type" : "object",
		"properties" :
		{
		"AcceptTypes" : {"type" : "string"},
		"ContentEncoding" : {"type" : "string"},
		...</pre>
		Changed to:
		<pre>{
		"title" : "Serialized Request with Certificate",
		"type" : "object",
		"properties" :
		{
		"Request" :
		{
		"type" : "object",
		"properties" :
		{
		"AcceptTypes" : {"type" : "string"},
		"Content" : [<byte>, *],
		"ContentEncoding" : {"type" : "string"},
		...</pre>

2014/10/13	V3.0 – 2014/05/15	In Section 2.2.2.11, Serialized Request with Certificate, updated the definition of the Serialized Request with Certificate object to accurately reflect protocol behavior. Section 2.2.2.11 now reads as follows:
		2.2.2.11 Serialized Request with Certificate
		This is a JSON object containing a serialized request plus a serialized client certificate and its usage. The format of the object is as follows:
		<pre>{
		"Request" :
		{
		"AcceptTypes" : ["<accept-type>", *],
		"Content" : [<byte>, *],
		"ContentEncoding" : "<content-encoding>",
		"ContentLength" : "<content-length>",
		"ContentType" : "<content-type>",
		"Cookies" :
		[{
		"Name" : "<cookie-name>",
		"Value" : "<cookie-value>",
		"Path" : "<cookie-path>",
		"Domain" : "<cookie-domain>",
		"Expires" : "<cookie-expires>",
		"Version" : "<cookie-version>",
		}, *],
		"Headers" :
		[{ "Name" : "<header-name>", "Value" :
		"<header-value>" }, *],</pre>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		"HttpMethod" : "<http-method>", "RequestUri" : "<request-uri>", "QueryString" : [{ "Name" : "<query-param>", "Value" : "<query-value>" }, *], "UserAgent" : "<user-agent>", "UserHostAddress" : "<user-host-address>", "UserHostName" : "<user-host-name>", "UserLanguages" : ["<user-language>", *] }, "SerializedClientCertificate" : "<serialized-client-certificate>", "CertificateUsage" : "<certificate-usage>", } accept-type: A string that represents a MIME accept type supported by the client. This corresponds to a value of the Accept header of the request. byte: An 8 bit integer in decimal form. content-encoding: Character set of the entity-body of the request. content-length: Length in bytes of content sent in the request. content-type: MIME content type of the request. cookie-name: Name of the cookie. cookie-value: Value of the cookie. cookie-path: Virtual path transmitted with the cookie. cookie-domain: Domain associated with the cookie. cookie-expires: Expiration date and time of the cookie. cookie-version: Version of the cookie. header-name: Name of header. header-value: Value of header. http-method: HTTP data transfer method of the request, for example GET, POST, HEAD. request-uri: URI of the request. query-param: Name of the query parameter. query-value: Value of the query parameter. user-agent: User agent presented in the request. user-host-address: IP address and port number to which the request was directed. user-host-name: DNS name and port number (if provided) specified in the request. user-language: Natural language preferred for the response. serialized-client-certificate: Client certificate obtained from TLS handshake base64 string encoded. certificate-usage: Certificate Type (section 2.2.2.16) for certificate. For details of these changes, see [MS-ADFSPIP-Diff]: Active Directory Federation Services Proxy and Web Application Proxy Integration Protocol. This PDF shows the differences between the May 2014 release and the current Preview document.

Errata Published YYYY/MM/DD	Protocol Document Version	Description

2014/10/13	V3.0 – 2014/05/15	In Section 2.2.2.4, Configuration, added information for "DiscoveredUpnSuffixes" and "CustomUpnSuffixes" to "ServiceConfiguration" and information for "SupportsNTLM" to "EndpointConfiguration".
		
		{
		"ServiceConfiguration" :
		{
		"ServiceHostName" : "<service-host-name>",
		"HttpPort" : "<http-port-number>",
		"HttpsPort" : "<https-port-number >",
		"HttpsPortForUserTlsAuth" : "<user-TLS-port-number>",
		"DeviceCertificateIssuers" : ["<device-certificate-issuer>", *],
		"ProxyTrustCertificateLifetime" : "<trust-renewal-interval>",
		"DiscoveredUpnSuffixes" : ["<upn-suffix>", *],
		"CustomUpnSuffixes" : ["<upn-suffix>", *]
		},
		"EndpointConfiguration" :
		[
		{
		"Path" : "<endpoint-uri>",
		"PortType" : "<port-type>",
		"AuthenticationSchemes" : "<credential-collection-scheme>",
		"ClientCertificateQueryMode" : "<tls-query-behavior>",
		"CertificateValidation" : "<certificate-validation>",
		"SupportsNtlm" : "<support-ntlm>",
		"ServicePath" : "<service-endpoint-uri>",
		"ServicePortType" : "<service-port-type>"
		}, *
		]
		}
		
		upn-suffix: Possible User Principal Name (UPN) suffixes for principals that can be pre-authorized.
		support-ntlm: Boolean value that indicates whether the client supports NTLM authentication for SPNEGO-based HTTP authentication [RFC4559].
		In Section 3.11.5, Message Processing Events and Sequencing Rules, updated the processing rules to reflect the preceding changes. For details of these changes, see [MS-ADFSPIP-Diff]: Active Directory Federation Services Proxy and Web Application Proxy Integration Protocol. This PDF shows the differences between the May 2014 release and the current Preview document.

2014/10/13	V3.0 – 2014/05/15	In Section 3.13.5.2.1, Initiate Redirect-based Preauthentication, and Section 3.13.5.2.2, Response to Active Requests, corrected the description of the parameter apprealm. Changed from:

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Parameter</th><th>Value</th></tr>
<tr><td>apprealm</td><td>URL of the endpoint of the application being accessed.</td></tr>
</table>
Changed to:
<table>
<tr><th>Parameter</th><th>Value</th></tr>
<tr><td>apprealm</td><td>objectIdentifier of the application being accessed (section 2.2.2.6).</td></tr>
</table>
</td></tr>
<tr><td>2014/09/16</td><td>V3.0 – 2014/05/15</td><td>In the sections listed below, revised ADM element [Proxy Service State] to read [Client State]:
▪ Section 2.2.2.17, Proxy Token
▪ Section 3.3.5.1.1.3, Processing Details
▪ Section 3.3.5.2.1.3, Processing Details
▪ Section 3.3.5.3.2.3, Processing Details
▪ Section 3.3.5.3.3.3, Processing Details
▪ Section 3.5.5.1.1.3, Processing Details
▪ Section 3.13.5.2.1, Initiate Redirect based Preauthentication
In the sections listed below, revised ADM element [Service State Data] to read [Server State]:
▪ Section 3.2.5.1.1.3, Processing Details
▪ Section 3.2.5.2.1, POST
▪ Section 3.2.5.2.1.3, Processing Details
▪ Section 3.2.5.3.1, GET
▪ Section 3.2.5.3.1.3, Processing Details
▪ Section 3.2.5.3.2.3, Processing Details
▪ Section 3.2.5.3.3.3, Processing Details
▪ Section 3.4.5.1.1, GET
▪ Section 3.4.5.1.1.3, Processing Details
▪ Section 3.4.5.2.1, GET
▪ Section 3.4.5.2.1.3, Processing Details
▪ Section 3.4.5.3.1, GET</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ Section 3.4.5.3.1.3, Processing Details ▪ Section 3.6.5, Message Processing Events and Sequencing Rules ▪ Section 3.8.5, Message Processing Events and Sequencing Rules ▪ Section 3.8.5.1.1.3, Processing Details ▪ Section 3.8.5.1.2.3, Processing Details In Section 3.1.1, Server State, corrected the spelling of "Certificates" in the below snippet: Changed from: "ProxyTrustedCertficates" : ["<certificate-identifier>", *], Changed to: "ProxyTrustedCertificates" : ["<certificate-identifier>", *], In Section 3.1.1.3, Relying Party Trust State, removed the element below: pre-auth-required: Boolean denoting that access from outside the network needs preauthentication. In Section 3.6.5.1.1.3, Processing Details, added the following: Upon successful authentication the server MUST return [Server State].ProxyStore section 3.1.1.

Return to top of page

[MS-ADSC]: Active Directory Schema Classes

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/11/24	V18.0 – 2014/05/15	In Section 2.146, Class msDS-ResourcePropertyList, clarified that the defaultHidingValue attribute is set to TRUE in new domains running Windows Server 2012. Changed from: Version-Specific Behavior: Implemented on Windows Server 2012 operating system and Windows Server 2012 R2 operating system. Changed to: Version-Specific Behavior: Implemented on Windows Server 2012 operating system and Windows Server 2012 R2 operating system. The defaultHidingValue attribute is set to TRUE in new domains running Windows Server 2012.

Return to top of page

[MS-ADTS]: Active Directory Technical Specification

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/03/16	V40.0 – 2014/05/15	In Section 3.1.1.4.5.19, tokenGroups, tokenGroupsNoGCAcceptable, added processing rules for OperationType. Text added is shown in **bold**. Changed to: ... Let U be the object from which the tokenGroups or tokenGroupsNoGCAcceptable attribute is being read. ▪ If U!objectSid does not exist, U!tokenGroups and U!tokenGroupsNoGCAcceptable are not present. ▪ **For AD DS in mixed mode, let OperationType=RevMembGetGroupsForUser; otherwise, for AD LDS and AD DS not in mixed mode, let OperationType=RevMembGetAccountGroups.** ▪ Otherwise, U!tokenGroups and U!tokenGroupsNoGCAcceptable are the result of the algorithm in [MS-DRSR] section 4.1.8.3 (IDL_DRSGetMemberships) using DRS_MSG_REVMEMB_REQ_V1.OperationType=OperationType , DRS_MSG_REVMEMB_REQ_V1.ppDsNames=U, and DRS_MSG_REVMEMB_REQ_V1.pLimitingDomain = the domain for which the server is a DC. ...
2015/01/19	V40.0 – 2014/05/15	In Section 3.1.1.5.2.1, Security Considerations, added a cross-reference to [MS-SAMR] for the security details on new object creation. Changed from: In the case of Windows Server 2008 R2 operating system, Windows Server 2012 operating system, and Windows Server 2012 R2 operating system, in the absence of RIGHT_DS_CREATE_CHILD, computer object creation requires that the RpcImpersonationAccessToken.Privileges[] field MUST have the SE_MACHINE_ACCOUNT_NAME privilege (defined in [MS-LSAD] section 3.1.1.2.1). Changed to: In the case of Windows Server 2008 R2 operating system, Windows Server 2012 operating system, and Windows Server 2012 R2 operating system, in the absence of RIGHT_DS_CREATE_CHILD, computer object creation requires that the security constraints and state changes specified in step 13 of [MS-SAMR] section 3.1.5.4.4 be followed.
2014/12/22	V40.0 – 2014/05/15	In Section 3.1.1.5.3.7.2, Undelete Constraints, added the following clarification text: If the object class of the target object is part of the base schema, the objectCategory attribute of the target object cannot be specified as part of the undelete operation.
2014/10/27	V40.0 – 2014/05/15	In Section 3.1.1.11.2.6, FillClaimsSetMetadata, updated the CompressionFormat value in the logical processing for the FillClaimsSetMetadata procedure as follows: Changed from:

Errata Published YYYY/MM/DD	Protocol Document Version	Description

		`CompressionFormat := COMPRESSION_FORMAT_LZNT1;`
		Changed to:
		`CompressionFormat := COMPRESSION_FORMAT_XPRESS_HUFF;`

2014/09/16	V40.0 – 2014/05/15	In Section 6.1.6.7.3, msDs-supportedEncryptionTypes, replaced the bit flag table and description with this reference to the complete and authoritative set of definitions in [MS-KILE]: Contains bitmapped values as specified in [MS-KILE] section 2.2.6 that define the encryption types supported by this trust relationship.
2014/09/16	V40.0 – 2014/05/15	In Section 3.1.1.3.4.1.3, LDAP_SERVER_DIRSYNC_OID, clarified the text for the case where more changes are available for retrieval: Changed from: The structure of the controlValue in the response control is the same as the structure of the controlValue in the request control, but the fields are interpreted differently. MoreResults is nonzero if there are more changes to retrieve, unused is not used, and CookieServer is the value to be used for Cookie in the next LDAP_SERVER_DIRSYNC_OID control sent in a search request to the server. Changed to: The structure of the controlValue in the response control is the same as the structure of the controlValue in the request control, but the fields are interpreted differently. MoreResults is nonzero if there are more changes to retrieve, unused is not used, and CookieServer is the value to be used for Cookie in the next LDAP_SERVER_DIRSYNC_OID control sent in a search request to the server to retrieve more changes.
2014/09/16	V40.0 – 2014/05/15	In Section 5.1.3.2.1, Control Access Rights, the DS-Bypas-Quota's 'Identifying GUID used for ACE' has been revised: Changed from: DS-Bypass-Quota 4125c71f-7fac-4ff0-bcb7-f09a41325286 Changed to: DS-Bypass-Quota 88a9933e-e5c8-4f2a-9dd7-2527416b8092
2014/09/16	V40.0 – 2014/05/15	In Section 7.6.2.7, Autoreconnecting to a Directory Server, removed two actions performed by the task: Removed action 2: Set TaskInputConnectionInfo.isAutoReconnecting to TRUE. Removed action 7: Set TaskInputConnectionInfo.isAutoReconnecting to FALSE.

Return to top of page

[MS-APDS]: Authentication Protocol Domain Support

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/04/27	V28.0 – 2014/05/15	In Section 3.1.5.2, NTLM Network Logon, updated information about the location of the Domain Controller and added new processing rules for NTLM v2 authentication. Added text is shown in **bold**. Changed from: The DC of the user account domain MUST be located ([MS-NRPC] section 3.5.4.3) and sent the request. This request MUST contain the NTLM challenge-response pair that was exchanged between the NTLM server and the client ([MS-NLMP] sections 2.2.1.2 and 2.2.1.3). Changed to: The DC of the server's domain MUST be located ([MS-NRPC] section 3.5.4.3) and sent the request. This request MUST contain the NTLM challenge-response pair that was exchanged between the NTLM server and the client ([MS-NLMP] sections 2.2.1.2 and 2.2.1.3). … **For NTLMv2 authentication, the DC MUST verify that the request originated from the NTLM server that generated the challenge:** ▪ **The DC extracts the MsvAvNbComputerName and MsvAvNbDomainName AV pairs ([MS-NLMP] section 2.2.2.1) in the NTLMv2_CLIENT_CHALLENGE ([MS-NLMP] section 2.2.2.7) of the AUTHENTICATE_MESSAGE ([MS-NLMP] section 2.2.1.3).** ▪ **If MsvAvNbDomainName does not match the NetBIOS name of the DC's domain, then return STATUS_LOGON_FAILURE.** ▪ **If MsvAvNbComputerName does not match the NetBIOS name of the server that established the secure channel ([MS-NRPC] section 3.5.4.4.2), then return STATUS_LOGON_FAILURE.**

Return to top of page

[MS-AIPS]: Authenticated Internet Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/03/02	V24.0 – 2014/05/15	In Section 3.10.4.1, New Connection Initiated, updated the information about connection set-up. Added text is shown in **bold**. Changed to: If the IsAuthenticatedFirewallConnection flag is set to TRUE in the connection state table entry corresponding to the connection, the first packet of every new connection (that is, the first packet sent by the connection initiator after creating the new entry in the connection state table) MUST be sent twice: initially with IPSec encapsulation

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		and then again without IPSec encapsulation. **These messages are known as the *ESP SYN* and *cleartext SYN* messages, respectively.<21>** **<21>It is possible for the cleartext SYN message to be received before the ESP SYN message. If this scenario occurs, a common practice for the server is to drop both messages, after which the client must attempt to reconnect. This reconnection attempt will delay a connection by approximately three seconds. For inbound TCP connections where NAT-T is not enabled, Windows can be configured to decrypt the ESP SYN message and send it up the stack as if it was the cleartext SYN message. By taking this action, the client is not required to reconnect. Windows Server 2012 R2 with [MSKB-3023555] and all subsequent versions of Windows according to the applicability list at the beginning of this section support this behavior.** The preceding changes are supported in Windows Server 2012 R2 with [MSKB-3023555].

Return to top of page

[MS-AZOD]: Authorization Protocols Overview

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/09/16	V1.1 – 2014/05/15	In Section 1.1.1.3, Security Descriptor, added information on the order of ACEs in the case of inheritance. Changed from: When access is requested to an Active Directory object, the Local Security Authority (LSA) compares the access token of the account that is requesting access to the object to the DACL. The security protocols check the object's DACL, searching for ACEs that apply to the user and group SIDs that are referenced in the user's access token. The security protocols then step through the DACL until they find any ACEs that allow or deny access to the user or to one of the user's groups. The protocols do this by first examining ACEs that have been explicitly assigned to the object and then examining the ACEs that have been inherited by the object. The following diagram shows the evaluation process for an access token and a DACL when a request is evaluated. Changed to: When access is requested to an Active Directory object, the Local Security Authority (LSA) compares the access token of the account that is requesting access to the object to the DACL. The security protocols check the object's DACL, searching for ACEs that apply to the user and group SIDs that are referenced in the user's access token. The security protocols then step through the DACL until they find any ACEs that allow or deny access to the user or to one of the user's groups. The protocols do this by first examining ACEs that have been explicitly assigned to the object and then examining the

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		ACEs that have been inherited by the object. Inherited ACEs are placed in the order in which they are inherited. ACEs inherited from the child object's parent come first, then ACEs inherited from the grandparent, and so on up the tree of objects. The following diagram shows the evaluation process for an access token and a DACL when a request is evaluated.

Return to top of page

[MS-BKRP]: BackupKey Remote Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/05/25	V19.0 – 2014/05/15	In Section 3.1.4.1.1, BACKUPKEY_BACKUP_GUID, and Section 3.1.4.1.2.1, Processing a Valid ServerWrap Wrapped Secret, corrected the byte size of SrvKey derived from ServerWrap from 64 bytes to 256 bytes. In Section 3.1.4.1.1, BACKUPKEY_BACKUP_GUID, changed from: … 3. At this stage, we have the value of the current ServerWrap key. Let SrvKey denote the leading 64 bytes of this key. .. Changed to: … 3. At this stage, we have the value of the current ServerWrap key. Let SrvKey denote the leading 256 bytes of this key. .. In Section 3.1.4.1.2.1, Processing a Valid ServerWrap Wrapped Secret, changed from: … Otherwise, let SrvKey denote the leading 64 bytes of the ServerWrap key. … Changed to: … Otherwise, let SrvKey denote the leading 256 bytes of the ServerWrap key. …

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/10/13	V3.0 – 2014/05/15	In Section 3.1.3, Initialization, changed the security support provider that must be used from Netlogon to SPNEGO. Changed from: The CAPR server implementation registers an endpoint with RPC over TCP/IP. The server MUST register the Netlogon security support provider authentication_type constant [0x44] as the security provider used by the RPC interface, as specified in [MS-RPCE] section 3.3.3.3.1.3. Changed to: The CAPR server implementation registers an endpoint with RPC over TCP/IP. The server MUST register the SPNEGO security support provider authentication_type constant [0x09] as the security provider used by the RPC interface, as specified in [MS-RPCE] section 3.3.3.3.1.3.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/10/13	V24.0 – 2014/05/15	In Section 2.2.4.46.1, Request, corrected the statement that the SMB_COM_TRANSACTION2 request format is identical to that of the SMB_COM_TRANSACTION request, changed the datatype of the Name field from SMB_STRING to UCHAR, and revised the description of the Name field. Changed from: The SMB_COM_TRANSACTION2 request format is identical to that of the SMB_COM_TRANSACTION request. The differences are in the subcommands supported, and in the purposes and usages of some of the fields. Changed to: The SMB_COM_TRANSACTION2 request format is similar to that of the SMB_COM_TRANSACTION request except for the Name field. The differences are in the subcommands supported, and in the purposes and usages of some of the fields. Changed from: Name (variable): This field is present but not used in SMB_COM_TRANSACTION2 requests. If Unicode support has been negotiated, then this field MUST be aligned to a 16-bit boundary and MUST consist of two terminating null characters. If Unicode support has not been negotiated this field will contain only one terminating null character. The Name field MUST be the first field in this section. Changed to: Name (1 byte): This field is not used in SMB_COM_TRANSACTION2 requests. This field MUST be set to zero, and the server MUST ignore it on receipt.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/09/16	V24.0 – 2014/05/15	In Section 2.2.4.30, SMB_COM_SET_INFORMATION2 (0x22), updated the processing rules for a date or time value of 0 by removing the line "A date or time value of 0 indicates to the server that the values MUST NOT be changed." Changed from: This command MAY be sent by a client to set attribute information about an open file. The client MUST provide a valid FID in the SMB Header (section 2.2.3.1). The FID SHOULD have been acquired through a previously successful use of one of the SMB commands for opening a file. The client MUST have at least write permission on the file. The target file is updated from the values specified in the request. A date or time value of 0 indicates to the server that the values MUST NOT be changed. This command allows the client to set more attribute information for the file than the SMB_COM_SET_INFORMATION (section 2.2.4.10) command. Changed to: This command MAY be sent by a client to set attribute information about an open file. The client MUST provide a valid FID in the SMB Header (section 2.2.3.1). The FID SHOULD have been acquired through a previously successful use of one of the SMB commands for opening a file. The client MUST have at least write permission on the file. The target file is updated from the values specified in the request. This command allows the client to set more attribute information for the file than the SMB_COM_SET_INFORMATION (section 2.2.4.10) command. In Section 3.3.5.29, Receiving an SMB_COM_SET_INFORMATION2 Request, updated the processing rules for nonzero attributes. Changed from: For each nonzero attribute in the request, the server MUST attempt to set the attribute on the file indicated by the FID. If an error is detected, the Status field of the response MUST be set to the error; otherwise, Status MUST be set to success. The response messages MUST be sent to the client as described in section 3.3.4.1.<284> Changed to: The server MUST attempt to set the attribute on the file indicated by the FID. If an error is detected, the Status field of the response MUST be set to the error; otherwise, Status MUST be set to success. The response messages MUST be sent to the client as described in section 3.3.4.1.<284>
2014/09/16	V24.0 – 2014/05/15	In Section 2.2.4.64.2, Response, and Section 2.2.6.7.2, Response, added the error code ERRbadfile(0x0002) to the tables of error codes.
2014/09/16	V24.0 – 2014/05/15	In Section 2.2.8.1.2, SMB_INFO_QUERY_EA_SIZE; Section 2.2.8.1.3, SMB_INFO_QUERY_EAS_FROM_LIST; Section 2.2.8.1.4, SMB_FIND_FILE_DIRECTORY_INFO; Section 2.2.8.1.5, SMB_FIND_FILE_FULL_DIRECTORY_INFO; Section 2.2.8.1.6, SMB_FIND_FILE_NAMES_INFO; and Section 2.2.8.1.7, SMB_FIND_FILE_BOTH_DIRECTORY_INFO, the data type of FileName was changed from SMB_STRING to an array of UCHARs.
2014/09/16	V24.0 –	In Section 2.2.8.1.7, SMB_FIND_FILE_BOTH_DIRECTORY_INFO, changed the descriptions for the ShortNameLength and ShortName

Errata Published YYYY/MM/DD	Protocol Document Version	Description
	2014/05/15	fields. Changed from: ShortNameLength: (1 byte): This field MUST contain the length of the ShortName, in bytes. ShortName: (24 bytes): This field MUST contain the 8.3 name of the file in Unicode format. Changed to: ShortNameLength: (1 byte): This field MUST contain the length of the ShortName, in bytes, or zero if no 8.3 name is present. ShortName: (24 bytes): This field MUST contain the 8.3 name, if any, of the file in Unicode format.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/06/08	V30.0 – 2014/05/15	In Section 6.2, Appendix A.2: clusapi3.idl536, the code has been changed as follows to enable the IDL to compile correctly: ▪ Removed embedded non-ascii characters ▪ Added a missing "*" before the rpc_status token in the line "[out] error_status_t rpc_status"

Return to top of page

[MS-CSRA]: Certificate Services Remote Administration Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/03/16	V33.0 – 2014/05/15	Updated an existing product behavior note and created a new product behavior note to more specifically express the values of Config_Product_Version and OnNextRestart_Config_Product_Version In product behavior note <11> for Section 3.1.1.10, Configuration Data, corrected the Default Value for Windows Server 2012 R2. Changed from: - 0x00060001: Windows Server 2012 R2 Changed to: - 0x00050001: Windows Server 2012 R2 without [MSKB-3013769] - 0x00060001: Windows Server 2012 R2 with [MSKB-3013769] In Section 3.1.4.2.14, (ICertAdminD2::GetConfigEntry (Opnum 44)), added a new product behavior note <71> as follows:

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Input parameters</th><th>Processing rule for pVariant</th></tr>
<tr><td>pwszNodePath is EMPTY and pwszEntry is "Version"</td><td>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 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<71></td></tr>
</table>
<71> Section 3.1.4.2.14: Windows Server 2012 R2 without [MSKB-3013769] sends 0x00050001, whereas Windows Server 2012 R2 with [MSKB-3013769] sends 0x00060001.
</td></tr>
</table>

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/09/16	V20.0 – 2014/05/15	In Appendix A: Full IDL, the code for the CprepCheckNetFtBindings3 method has been changed to denote that the parameter to the method is void.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/11/24	V28.0 – 2014/05/15	In Section 2.2.5.2.4.1, DNS_RPC_ZONE_INFO_W2K, revised the documentation to clarify that the fNotifyLevel flag is ignored for primary non-AD integrated zones. Changed from: fNotifyLevel: A value that specifies the settings for sending zone

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		notifications for a zone from the master DNS server. This value MUST be set to one of the allowed values as specified in DNS_ZONE_ NOTIFY_LEVEL (section 2.2.5.1.3). Changed to: fNotifyLevel: This parameter is ignored, and for dwZoneType parameter setting DNS_ZONE_TYPE_PRIMARY (specified in section 2.2.5.1.1) and for DNS servers that are not directory service-integrated, the zone notification setting is set to ZONE_NOTIFY_ALL_SECONDARIES. Otherwise, it is set to ZONE_NOTIFY_LIST_ONLY, as specified in section 2.2.5.1.3.
2014/09/16	V28.0 – 2014/05/15	In Section 2.2.6.2.4, DNS_RPC_TRUST_POINT, revised the TRUSTPOINT_STATE values to be 4 bytes, consistent with Section 2.2.1.1.3, TRUSTPOINT_STATE. For example, revised the value of TRUSTPOINT_STATE_INITIALIZED from (0x0000) to (0x00000000). In Section 2.2.6.2.6, DNS_RPC_TRUST_ANCHOR, revised the TRUSTANCHOR_STATE values to be consistent with Section 2.2.1.1.4, TRUSTANCHOR_STATE. For example, revised the value of TRUSTANCHOR_STATE_DSPENDING from (0x0001) to (0x00000001).
2014/09/16	V28.0 – 2014/05/15	In Section 2.2.4.2.2.2, DNS_RPC_SERVER_INFO_DOTNET, revised the descriptions of the dwReserved0 and pExtensions fields as follows: Changed from: dwReserved0: This field is reserved for future use and it MUST be ignored by receiver. pExtensions: Reserved for future use and MUST be ignored by the receiver. Changed to: dwReserved0: This field is reserved for future use. Senders MUST set this to zero and it MUST be ignored by receiver. pExtensions: Reserved for future use. Senders MUST set this to NULL and it MUST be ignored by the receiver.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/06/22	V37.0 – 2014/05/15	In Section 4.1.19.2, Server Behavior of the IDL_DRSReplicaAdd Method, and Section 4.1.23.2, Server Behavior of the IDL_DRSReplicaSync Method, added clarifying details to the pseudocode for server processing of the IDL_DRSReplicaAdd method and the IDL_DRSReplicaSync method. In addition, updated the pseudocode to add a missing carriage return in Section 4.1.21.2.3, ReplicateOffChanges(). In Section 4.1.19.2, Server Behavior of the IDL_DRSReplicaAdd

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Method: Changed from: `...` `updRefs: DRS_MSG_UPDREFS /* See IDL_DRSUpdateRefs structures. */` `hDrsSrc: DRS_HANDLE` `ValidateDRSInput(hDrs, 5)` `...` Changed to: `...` `updRefs: DRS_MSG_UPDREFS /* See IDL_DRSUpdateRefs structures. */` `hDrsSrc: DRS_HANDLE` `msgRequest: DRS_MSG_GETCHGREQ` `msgOut: DRS_MSG_GETCHGREPLY` `outVersion: DWORD` `cMaxObjects: ULONG` `cMaxBytes: ULONG` `versionRequestMsg: DWORD` `err: ULONG` `ValidateDRSInput(hDrs, 5)` `...` Changed from: `...` `Perform a replication cycle as a client of IDL_DRSGetNCChanges. Call ReplicateNCRequestMsg (section 4.1.10.4.1) to form the first request and send it. If not DRS_MAIL_REP in msgIn.ulOptions, then wait for the response, process it (section 4.1.10.6), send the next request, etc. until the replication cycle is complete. If there are any failures from this replication attempt, err should be assigned an appropriate error value.` `...`  Changed to:  `...` `/* Perform a replication cycle as a client of IDL_DRSGetNCChanges. */` `versionRequestMsg := The version number of the input message negotiated between the client and server (section 4.1.10.4.1).`

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		cMaxObjects := Implementation-specific value. cMaxBytes := Implementation-specific value. /* Form the first request */ ReplicateNCRequestMsg(hDrsSrc, versionRequestMsg, nc, rf, options, cMaxObjects, cMaxBytes, ADDR(msgRequest)) err := IDL_DRSGetNCChanges(hDrsSrc, versionRequestMsg, msgRequest, ADDR(outVersion), ADDR(msgOut)) if err = 0 and not DRS_MAIL_REP in msgIn.ulOptions then Wait for the response, process it (section 4.1.10.6), send the next request, etc. until the replication cycle is complete. If there are any failures from this replication attempt, err should be assigned an appropriate error value. endif return err In Section 4.1.23.2, Server Behavior of the IDL_DRSReplicaSync Method: Changed from: ... options: DRS_OPTIONS nc: DSName rf: sequence of RepsFrom msgIn: DRS_MSG_REPSYNC_V1 ... Changed to: options: DRS_OPTIONS nc: DSName rf: sequence of RepsFrom msgIn: DRS_MSG_REPSYNC_V1 err: ULONG

Errata Published YYYY/MM/DD	Protocol Document Version	Description

		...
		
		Changed from:
		
		...
		foreach r in rf
		
		if DRS_UPDATE_NOTIFICATION in options
		and not DRS_TWOWAY_SYNC in options
		and DRS_NEVER_NOTIFY in r.V2.ulReplicaFlags then
		return ERROR_DS_DRA_NO_REPLICA
		endif
		
		/* Replicate nc from the DC specified by r.uuidDsa. */
		Perform a replication cycle as a client of IDL_DRSGetNCChanges.
		Call ReplicateNCRequestMsg (section 4.1.10.4.1) to form the
		first request and send it. If not DRS_MAIL_REP in r.options,
		then wait for the response, process it, send the next request
		(section 4.1.10.6), etc., until the replication cycle is
		complete.
		endfor
		
		return 0
		
		Changed to:
		
		...
		foreach r in rf
		msgRequest: DRS_MSG_GETCHGREQ
		cMaxObjects: ULONG
		cMaxBytes: ULONG
		versionRequestMsg: DWORD
		outVersion: DWORD
		msgOut: DRS_MSG_GETCHGREPLY
		
		versionRequestMsg := The version number of the input message negotiated between the
		client and server (section 4.1.10.4.1).
		cMaxObjects := Implementation-specific value.
		cMaxBytes := Implementation-specific value.
		
		if DRS_UPDATE_NOTIFICATION in options
		and not DRS_TWOWAY_SYNC in options
		and DRS_NEVER_NOTIFY in r.V2.ulReplicaFlags then
		return ERROR_DS_DRA_NO_REPLICA
		endif
		
		/* Replicate nc from the DC specified by r.uuidDsa. */
		
		ReplicateNCRequestMsg(
		hDrs,

Errata Published YYYY/MM/DD	Protocol Document Version	Description

		versionRequestMsg,
		nc,
		r,
		options,
		cMaxObjects,
		cMaxBytes,
		ADDR(msgRequest))
		err := IDL_DRSGetNCChanges(
		hDrsSrc,
		versionRequestMsg,
		msgRequest,
		ADDR(outVersion),
		ADDR(msgOut))
		
		if err = 0
		and not DRS_MAIL_REP in msgIn.ulOptions
		
		then
		
		Wait for the response, process it (section 4.1.10.6), send the next request, etc.
		until the replication cycle is complete.
		
		If there are any failures from this replication attempt, assign an
		appropriate error value to err, and then break out of the for loop.
		
		endif
		endfor
		
		In Section 4.1.21.2.3, ReplicateOffChanges() changed from:
		
		...
		msgSyncReq.ulOptions := DRS_WRIT_REP ret := IDL_DRSReplicaSync(hDRS, 1,ADR(msgSyncReq))
		
		Changed to:
		
		...
		msgSyncReq.ulOptions := DRS_WRIT_REP
		ret := IDL_DRSReplicaSync(hDRS, 1,ADR(msgSyncReq))
		...

2015/03/16	V37.0 – 2014/05/15	In Section 4.1.1.2.3, CreateNtdsDsa, corrected the pseudocode to change an attribute name from serverPrincipalName to servicePrincipalName. Changed from: /* Find the computer object and update its SPN. */ if sl ≠ null then dcObj := select one v from subtree DefaultNC() where v = sl spn := ConstructReplSpn(domainCR!dnsHostName,

Errata Published YYYY/MM/DD	Protocol Document Version	Description

		dcObj.guid)
		  dcObj!serverPrincipalName := dcObj!serverPrincipalName + {spn}
		endif
		
		Changed to:
		
		/* Find the computer object and update its SPN. */
		if sl ≠ null then
		  dcObj := select one v from subtree DefaultNC() where v = sl
		  spn := ConstructReplSpn(domainCR!dnsHostName, dcObj.guid)
		  dcObj!servicePrincipalName := dcObj!servicePrincipalName + {spn}
		endif

2015/03/16	V37.0 – 2014/05/15	In Section 4.1.8.3, Server Behavior of the IDL_DRSGetMemberships Method, corrected the pseudocode that tests for a Read-Only Domain Controller (RODC).
		Changed from:
		
		if((u!userAccountControl & ADS_UF_WORKSTATION_TRUST_ACCOUNT = ADS_UF_WORKSTATION_TRUST_ACCOUNT) or (u!userAccountControl & ADS_UF_PARTIAL_SECRETS_ACCOUNT = ADS_UF_PARTIAL_SECRETS_ACCOUNT))
		
		wSet := wSet + GetDSNameOfEnterpriseRODCsGroup()
		endif if((u!userAccountControl & ADS_UF_WORKSTATION_TRUST_ACCOUNT = ADS_UF_WORKSTATION_TRUST_ACCOUNT) or (u!userAccountControl & ADS_UF_PARTIAL_SECRETS_ACCOUNT = ADS_UF_PARTIAL_SECRETS_ACCOUNT))
		
		wSet := wSet + GetDSNameOfEnterpriseRODCsGroup()
		endif
		
		Changed to:
		
		if((u!userAccountControl & ADS_UF_WORKSTATION_TRUST_ACCOUNT = ADS_UF_WORKSTATION_TRUST_ACCOUNT) and (u!userAccountControl & ADS_UF_PARTIAL_SECRETS_ACCOUNT = ADS_UF_PARTIAL_SECRETS_ACCOUNT))
		
		wSet := wSet + GetDSNameOfEnterpriseRODCsGroup()
		endif

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/03/16	V37.0 – 2014/05/15	In Section 4.1.6.2.1, ExecuteKCCTasks, added more information, including a reference to a list of the tasks. Changed from: `procedure ExecuteKCCTasks(): ULONG` This procedure executes the tasks necessary for maintaining the replication topology between DCs. Changed to: `procedure ExecuteKCCTasks(): ULONG` This procedure executes the tasks necessary for maintaining the replication topology between DCs. These tasks include activities such as maintenance of kCCFailedLinks and kCCFailedConnections, maintenance of intrasite and intersite connections, and updates of RODC objects (as appropriate). See [MS-ADTS] section 6.2.2 for a full list of these tasks.
2014/11/24	V37.0 – 2014/05/15	In Section 2.2.2, Service Principal Names for Domain Controllers, updated the document to include information about the "RPC/<DSA GUID based DNS hostname>" Service Principal Name (SPN). Added the following: In either DC-to-DC or client-to-DC operations, to allow use of the DRS Remote Protocol when the RPC endpoint mapper has been configured to disallow anonymous clients (see [MS-RPCE] section 3.1.1.1.3), the DC stores an SPN with the following format<51>: ▪ "RPC/<DSA GUID> .msdcs.<DNS forest name>" In the preceding SPN description, <DSA GUID> is the DSA GUID of the DC and <DNS forest name> is the FQDN of the forest of the DC. <51>Section 5.208: Windows 2000 Server, Windows Server 2003, Windows Server 2003 R2 Datacenter x64 Edition, Windows Server 2008, and Windows Server 2008 R2 AD DS DCs do not store the "RPC" SPN.

2014/10/13	V37.0 – 2014/05/15	In Section 4.1.4.2.11, LookupAttr, updated the pseudocode to include a call to IsGC().
		Changed from:
		`if DS_NAME_FLAG_GCVERIFY in flags then`
		Changed to:
		`if DS_NAME_FLAG_GCVERIFY in flags or IsGC() then`

Return to top of page

[MS-DTYP]: Windows Data Types

Errata Published YYYY/MM/DD	Protocol Document Version	Description

2014/10/27	V38.0 – 2014/05/15	Section 2.2.56, UNC, was updated to provide a more formal specification of the construct using ABNF. The section now reads as follows:
		2.2.57 UNC
		A Universal Naming Convention (UNC) string is used to specify the location of resources such as shared files or devices.
		For RPC implementations, this type is declared as follows:
		`typedef STRING UNC;`
		There are three UNC schemes based on namespace selectors: filespace selector, Win32API selector, and device selector. Only the filespace selector is parsed for on-wire traffic, the other two pass opaque BLOBs to the consuming entity. The filespace selector is a null-terminated Unicode character string in the following ABNF syntax:
		`UNC = "\\" host-name "\" share-name [ "\" object-name ]`
		`host-name = host ; as specified in [RFC3986] minus IPvFuture`
		`share-name = 1*80pchar`
		`pchar = %x20-21 / %x23-29 / %x2D-2E / %x30-39 / %x40-5A / %x5E-7B / %x7D-FF`
		`object-name = *path-name [ "\" file-name ]`
		`path-name = 1*255pchar`
		`file-name = 1*255fchar [ ":" stream-name [ ":" stream-type ] ]`
		`fchar = %x20-21 / %x23-29 / %x2B-2E / %x30-39 / %x3B / %x3D / %x40-5B / %x5D-7B / %x7D-FF`
		`stream-name = *schar`
		`schar = %x01-2E / %x30-39 / %x3B-5B /%x5D-FF`
		`stream-type = 1*schar`
		host-name: The host name of a server or the domain name of a domain hosting resource, using the syntax of host from [RFC3986], with the exception that IPvFuture is not supported. The host-name string MUST be a NetBIOS name as specified in [MS-NBTE] section 2.2.1, a fully qualified domain name (FQDN) as specified in [RFC1035] and [RFC1123], or a textual IPv4 as specified in [RFC1123] section 2.1 or IPv6 address as specified in [RFC4291] section 2.2.
		share-name: The name of a share or a resource to be accessed. The format of this name depends on the actual file server protocol that is used to access the share. Examples of file server protocols include SMB (as specified in [MS-SMB]), NFS (as specified in [RFC3530]), and NCP (as specified in [NOVELL]).
		object-name: The name of an object; this name depends on the actual resource accessed.
		The notation "[\object-name]*" indicates that zero or more object

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>names may exist in the path, and each object-name is separated from the immediately preceding object-name with a backslash path separator. In a UNC path used to access files and directories in an SMB share, for example, object-name may be the name of a file or a directory. The host-name, share-name, and object-name are referred to as "pathname components" or "path components". A valid UNC path consists of two or more path components. The host-name is referred to as the "first pathname component", the share-name as the "second pathname component", and so on. The last component of the path is also referred to as the "leaf component". The protocol that is used to access the resource, and the type of resource that is being accessed, define the size and valid characters for a path component. The only limitations that a Distributed File System (DFS) places on path components are that they MUST be at least one character in length and MUST NOT contain a backslash or null.

path-name: One or more pathname components separated by the "\" backslash character. All pathname components other than the last pathname component denote directories or reparse points.

file-name: The "leaf component" of the path, optionally followed by a ":" colon character and a stream-name , optionally followed by a ":" colon character and a stream type. The stream-name, if specified, MAY be zero-length only if stream-type is also specified; otherwise, it MUST be at least one character. The stream-type, if specified, MUST be at least one character.</td></tr>
<tr><td>2014/09/16</td><td>V38.0 – 2014/05/15</td><td>In Section 2.4.5, ACL, updated the text for the description of the ACL_REVISION value of the AclRevision field.

Changed from:
<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>ACL_REVISION 0x02</td><td>When set to 0x02, only AceTypes 0x00, 0x01, 0x02, 0x03, and 0x11 can be present in the ACL. An AceType of 0x11 is used for SACLs but not for DACLs. For more information about ACE types, see section 2.4.4.1.</td></tr></table>
Changed to:
<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>ACL_REVISION 0x02</td><td>When set to 0x02, only AceTypes 0x00, 0x01, 0x02, 0x03, 0x11, 0x12, and 0x13 can be present in the ACL. An AceType of 0x11 is used for SACLs but not for DACLs. For more information about ACE types, see section 2.4.4.1.</td></tr></table></td></tr>
<tr><td>2014/09/16</td><td>V38.0 – 2014/05/15</td><td>In Section 2.5.1.1, Syntax, added the following product behavior note for the “LA” SDDL alias (well-known SID name, ADMINISTRATOR):

<66> Section 2.5.1.1: For the domain built-in ADMINISTRATOR (S-1-5-21-<domain>-500), Windows passes the actual SID, not the “LA” token. Reporting tools may convert this back to a token when</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		examining the SDDL.

Return to top of page

[MS-DVRE]: Device Registration Enrollment Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/09/16	V3.0 – 2014/05/15	In Section 6, Appendix A: Full WSDL, the missing XML header was added to the WSDL: Changed from: <wsdl:definitions xmlns:q2="http://schemas.datacontract.org/2004/07/Microsoft.DeviceRegistration" xmlns:xsd="http://www.w3.org/2001/XMLSchema" Changed to: <?xml version="1.0" encoding="utf-8"?> <wsdl:definitions xmlns:q2="http://schemas.datacontract.org/2004/07/Microsoft.DeviceRegistration" xmlns:xsd="http://www.w3.org/2001/XMLSchema"

Return to top of page

[MS-ECS]: Enterprise Client Synchronization Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/01/19	V3.0 – 2014/05/15	In Section 3.6.1.4, Per DownloadFile, updated the processing rules for ConcurrencyInfo. Changed from: ConcurrencyInfo: An opaque stream of bytes that identifies the file version, as specified in section 2.2.2.1. Changed to: ConcurrencyInfo: An opaque stream of bytes that identifies the file version. ConcurrencyInfo MUST be in the format specified in [MS-FSVCA] section 2.9. In Section 3.6.5.2, Download Scenario, updated the processing rules for DownloadEntryVector within the Data Transfer section. Changed from: The client MUST send a PUT request on the Download Data resource, as specified in section 3.5.5.2.1. The DownloadEntryVector in the request body MUST be set to a VECTOR_DOWNLOAD_ENTRY constructed in an implementation-specific manner using

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>each DownloadFile object in DownloadFileList.
Changed to:
The client MUST send a PUT request on the Download Data resource, as specified in section 3.5.5.2.1. The DownloadEntryVector in the request body MUST be set to a VECTOR_DOWNLOAD_ENTRY structure constructed as a collection of DOWNLOAD_ENTRY structures using each DownloadFile object in DownloadFileList:
▪ DOWNLOAD_ENTRY.SyncItemId is set to DownloadFile.SyncId
▪ DOWNLOAD_ENTRY.FileVersion is set to DownloadFile.ConcurrencyInfo
In Section 3.6.5.2, Download Scenario, updated the processing rules for BatchMetadata entry in the Download Batch response within the Obtain Change Batch details section.
Step 3 has been changed from:
Otherwise, the client inserts the DownloadFile objects from the BatchMetadata entry in the Download Batch response body into the DownloadFileList. The client MUST set ContinueToken to the x-ecs-continue token that is received in the response header.
Changed to:
Otherwise, for each BatchMetadata entry in the Download Batch response, the Client MUST construct DownloadFile objects as follows:
▪ DownloadFile.SyncId is set to SYNC_CHANGE_BATCH.File.FileId.
▪ DownloadFile.ConcurrencyInfo is set to SYNC_CHANGE_BATCH.File.SyncVersion.
▪ The client inserts the DownloadFile objects into the DownloadFileList. The client MUST set ContinueToken to the x-ecs-continue token that is received in the response header.</td></tr>
<tr><td>2014/12/22</td><td>V3.0 – 2014/05/15</td><td>In Section 2, Messages, and Section 3, Protocol Details, all references to header x-ecs-session-location-url have been removed as follows:
<table>
<tr><th>Subsection</th><th>Description of changes</th></tr>
<tr><td>2.2.1, HTTP Headers</td><td>Removed the entry for x-ecs-session-location-url from the table.</td></tr>
<tr><td>2.2.1.7, x-ecs-session-location-url</td><td>Deleted entire subsection.</td></tr>
<tr><td>3.4.5.2.1, GET</td><td>Changed from:
The operation can be invoked through the following URI suffix on the x-ecs-session-location-url returned in the response header of the PUT method on the Creation Session resource:
<code>/SyncBatchParameters</code></td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td>Changed to:
The operation can be invoked through the following URI suffix using Session.SessionId:

<code>/SyncBatchParameters</code></td></tr>
<tr><td>3.4.5.5.1, DELETE</td><td>Changed from:
The client MUST use the x-ecs-session-location-url returned by the server in the Creation Session response (see section 3.4.5.1.1) as the URI for this operation.
Changed to:
The client MUST use Session.SessionId as the URI for this operation.</td></tr>
<tr><td>3.4.5.6.1, GET</td><td>Changed from:
The operation can be invoked through the following URI on the x-ecs-session-location-url that is returned in the response header of the PUT method on the Creation Session resource:

<code>/DownloadBatch</code>

Changed to:
The operation can be invoked through the following URI suffix using Session.SessionId:

<code>/DownloadBatch</code></td></tr>
</table>
</td></tr>
<tr><td>2014/11/10</td><td>V3.0 – 2014/05/15</td><td>In Section 4.2, Upload Scenario, changed ReplicaId to ClientID within the HTTP PUT request for the Create Session resource.
Step 1 now reads:

<pre>- Http: Request, PUT /sync/1.0/session
  Command: PUT
+ URI: /sync/1.0/session
  ProtocolVersion: HTTP/1.1
  Cache-Control: no-cache
  Connection: Keep-Alive
  Pragma: no-cache
  UserAgent: MS_WorkFoldersClient
  x-ecs-devicename: HOMEPC,Windows,6,3,MS_WorkFoldersClient
  x-ecs-partnershipId:
U2FsZXNTaGFyZXxBZG1pbmlzdHJhdG9yXERvY3VtZW50c3xTLTEtNS0yMS0yNTA3MDM5Mj
UxLTIzNjkyNjc3NTEtMzkxMTk0MDczMy01MDA=
  ContentLength: 17
  Host: syncsvr.contoso.com
  Type Upload(1)</pre></td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		ClientID 60f4c9fd-9c9d-421c-97a0-bdcb740424c3
2014/10/27	V3.0 – 2014/05/15	In Section 3.4.5.2.2, PUT, corrected the first paragraph to reflect that the PUT method of Sync Batch Parameters is for the download scenario, not for the upload scenario. Changed from: The PUT method on the Sync Batch Parameters resource is issued by the client to update the client's sync knowledge to the server in an upload scenario. Changed to: The PUT method on the Sync Batch Parameters resource is issued by the client to update the client's sync knowledge to the server in a download scenario.
2014/09/16	V3.0 – 2014/05/15	In Section 3, Protocols Details, modified existing subsections and added new subsections to provide new processing details for preparing a batch request. View this Word document with Track Changes turned on to see the information added to Section 3: [MS-ECS]_Section_3_Diff.
2014/08/21	V3.0 – 2014/05/15	In Section 1.2.2, Informative References, added a new reference: [MSKB-2891638] Microsoft Corporation, "Work Folders is available on Windows 7 client", April 2014, http://support.microsoft.com/kb/2891638. In Appendix A, Product Behavior, added a new supported product version: Windows 7 operating system with Service Pack 1 (SP1) and [MSKB-2891638]. In product behavior note <1>, added the following to the table of Windows operating system versions that support the ECS client: Windows 7 operating system with Service Pack 1 (SP1) and [MSKB-2891638]
2014/08/21	V3.0 – 2014/05/15	In the sections listed below, changed the ADM element ClientID to Client_ID: ▪ Section 3.4.5.1.1.3, Processing Details ▪ Section 3.6.1.1, Global ▪ Section 3.6.3, Initialization ▪ Section 3.6.5.1, Upload Scenario ▪ Section 3.6.5.2, Download Scenario

Return to top of page

[MS-EMF]: Enhanced Metafile Format

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/09/16	V10.0 – 2014/05/15	In Section 2.3.2, Clipping Record Types, the EMR_SETMETARGN description was updated. Changed from:

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Intersets the current metaregion with the current clipping region to form a new metaregion for the playback device context. The current clipping region SHOULD be reset to null. This EMF record specifies no parameters. Changed to: If the current metaregion is null, sets the metaregion to the current clip region. Otherwise, the current metaregion is intersected against the current clipping region. The resulting region will be set as the new metaregion. After the operation, the current clipping region will always be reset to null. During playback, any rendering that does not intersect the current metaregion will be clipped. This EMF record specifies no parameters.

Return to top of page

[MS-EMFPLUS]: Enhanced Metafile Format Plus Extensions

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/11/10	V13.0 – 2014/05/15	In Section 2.2.1.6, EmfPlusPath Object, corrected the description of the values in the PathPointFlags field, and in Section 2.3.4.4, EmfPlusDrawClosedCurve Record, corrected the point data type definition for 32-bit coordinates. View this Word document with Track Changes turned on to see the information added to Section 2: [MS-EMFPLUS]_Sections_2_2_1_and_2_2_3_4_4_Diff.

Return to top of page

[MS-FSA]: File System Algorithms

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/04/27	V18.0 – 2014/05/15	In Section 2.1.5.7, Server Requests a Byte-Range Lock, updated information about the interaction between leases and byte range lock. Text added is shown in **bold**. Changed to: ▪ [Processing] ▪ **If (FileOffset < Open.Stream.AllocationSize)<51> and Open.Stream.Oplock is not empty, the object store MUST check for an oplock break according to the algorithm in section 2.1.4.12, with input values as follows:**

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ **Open equal to this operation's Open** ▪ **Oplock equal to Open.Stream.Oplock** ▪ **Operation equal to "LOCK_CONTROL"** ▪ **OpParams empty** **<51> Section 2.1.5.7: In Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2, NTFS checks for an oplock break even when (FileOffset >= Open.Stream.AllocationSize).**
2015/01/19	V18.0 – 2014/05/15	In various places in Section 2, updated information for detecting if open files exist under a directory. Changed the title of Section 2.1.4.2, Algorithm for Detecting If Open Files Exist Within a Directory, and updated the algorithm input and pseudocode related to the RootDirectory and Link.File.OpenList objects as follows: **2.1.4.2 Algorithm for Detecting If Open Files Exist Under a Directory** The inputs for this algorithm are: ▪ RootDirectory: The DirectoryFile indicating the top-level directory under which to search for open files. ▪ Open: The Open for the request that is calling this algorithm. ▪ Operation: A code describing the operation being processed, per section 2.1.4.12. ▪ OpParams: Parameters associated with Operation, passed in from the calling request, per section 2.1.4.12. The output is a Boolean. If the return value is TRUE, then no open files exist under the directory; if FALSE, then at least one open exists even after attempting to break oplocks. Pseudocode for the algorithm is as follows: ▪ For each Link in RootDirectory.DirectoryList: ▪ // Check for oplock breaks in this directory. ▪ If Link.File.OpenList contains an Open with Open.Link equal to Link: ▪ If Stream.Oplock is not empty and Stream.Oplock.State contains either BATCH_OPLOCK or HANDLE_CACHING, the object store MUST check for an oplock break according to the algorithm in section 2.1.4.12, with input values as follows: ▪ Open equal to this algorithm's Open.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ Open equal to this algorithm's Open. ▪ Operation equal to this algorithm's Operation. ▪ Operation equal to this algorithm's Operation. ▪ EndIf ▪ EndFor ▪ EndIf ▪ // See if all oplock holders have gotten out of the way. ▪ If Link.File.OpenList contains an Open with Open.Link equal to Link: ▪ Return FALSE // An open still exists, deny the operation. ▪ EndIf ▪ // Recurse into any subdirectories. ▪ If Link.File.FileType is DirectoryFile, determine whether Link.File contains open files per section 2.1.4.2, with input values as follows: ▪ RootDirectory equal to Link.File. ▪ Open equal to this algorithm's Open. ▪ Operation equal to this algorithm's Operation. ▪ OpParams equal to this algorithm's OpParams. ▪ EndIf ▪ If Link.File contains open files as determined above: ▪ Return FALSE. // An open exists deeper in the directory hierarchy. ▪ EndIf ▪ EndFor ▪ Return TRUE // No opens remaining. . In Section 2.1.5.14.11, FileRenameInformation, added a reference to section 2.1.4.2 in a processing rule for when Open.File contains open files. Changed from: If Open.File contains open files, the operation MUST be failed with STATUS_ACCESS_DENIED. Changed to:

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		If Open.File contains open files per section 2.1.4.2, the operation MUST be failed with STATUS_ACCESS_DENIED. In Section 2.1.5.14.13, FileShortNameInformation, added If Open.File.FileType is DirectoryFile condition to a processing rule and added reference to section 2.1.4.2 to another processing rule. Changed from: Determine whether Open.File contains open files as per section 2.1.4.2, with input values as follows: Changed to: If Open.File.FileType is DirectoryFile, determine whether Open.File contains open files as per section 2.1.4.2, with input values as follows: Changed from: If Open.File contains open files, the operation MUST be failed with STATUS_ACCESS_DENIED. Changed to: If Open.File contains open files as per section 2.1.4.2, the operation MUST be failed with STATUS_ACCESS_DENIED.
2014/12/22	V18.0 – 2014/05/15	In Section 2.1.1.1, Per Volume, updated the ReFS cluster size for the different supported versions of Windows. Changed from: ClusterSize: A 32-bit unsigned integer specifying the size of a cluster (2) for this volume in bytes. ClusterSize MUST be a power of two, and MUST be greater than or equal to LogicalBytesPerSector and a power-of-two multiple of LogicalBytesPerSector.<1> <1> Section 2.1.1.1: NTFS uses a default cluster size of 4 KB, a maximum cluster size of 64 KB, and a minimum cluster size of 512 bytes. ReFS uses a default cluster size of 64 KB, a maximum cluster size of 128k, and a minimum cluster size of 4 KB. ReFS is supported only on Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2. Changed to: ClusterSize: A 32-bit unsigned integer specifying the size of a cluster (2) for this volume in bytes. ClusterSize MUST be a power of two, and MUST be greater than or equal to LogicalBytesPerSector and a power-of-two multiple of LogicalBytesPerSector.<1> <1> Section 2.1.1.1: NTFS uses a default cluster size of 4 KB, a maximum cluster size of 64 KB, and a minimum cluster size of 512 bytes. ReFS in Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 uses a fixed cluster size of 64 KB.
2014/09/16	V18.0 – 2014/05/15	In Section 2.1.5.2, Server Requests a Read, updated parameters provided by the server and updated pseudocode. Added: Unbuffered: A Boolean value. TRUE indicates that the read should be unbuffered (read directly from disk after writing and removing any cached data for this range), otherwise the value of Open.Mode.FILE_NO_INTERMEDIATE_BUFFERING should determine if the read is unbuffered.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/09/16	V18.0 – 2014/05/15	In Section 2.1.5.9.4, FSCTL_FILE_LEVEL_TRIM, updated the processing rules for when ((TrimOffset % Open.File.Volume.SystemPageSize) != 0). Added: The object store MUST check for byte range lock conflicts using the algorithm described in section 2.1.4.10 with ByteOffset set to TrimOffset, Length set to TrimLength, IsExclusive set to TRUE, LockIntent set to FALSE, and Open set to Open. If a conflict is detected, the operation MUST be failed with STATUS_FILE_LOCK_CONFLICT. In Section 2.1.5.9.17, FSCTL_OFFLOAD_READ, added object store check for byte range lock conflicts to the read operation pseudocode. Changed from: If InputBuffer.FileOffset / Open.File.Volume.BytesPerCluster is less than 0, the operation MUST be failed with STATUS_INVALID_PARAMETER. Changed to: If InputBuffer.FileOffset / Open.File.Volume.BytesPerCluster is less than 0, the operation MUST be failed with STATUS_INVALID_PARAMETER. The object store MUST check for byte range lock conflicts using the algorithm described in section 2.1.4.10 with ByteOffset set to InputBuffer.FileOffset, Length set to InputBuffer.CopyLength, IsExclusive set to FALSE, LockIntent set to FALSE, and Open set to Open. If a conflict is detected, the operation MUST be failed with STATUS_FILE_LOCK_CONFLICT. In Section 2.1.5.9.18, FSCTL_OFFLOAD_WRITE, added object store check for byte range lock conflicts to the write operation pseudocode. Changed from: If (InputBuffer.FileOffset + InputBuffer.CopyLength) is greater than Open.File.Volume.MaxFileSize, the operation MUST be failed with STATUS_INVALID_PARAMETER. Changed to: If (InputBuffer.FileOffset + InputBuffer.CopyLength) is greater than Open.File.Volume.MaxFileSize, the operation MUST be failed with STATUS_INVALID_PARAMETER. The object store MUST check for byte range lock conflicts using the algorithm described in section 2.1.4.10 with ByteOffset set to InputBuffer.FileOffset, Length set to InputBuffer.CopyLength, IsExclusive set to TRUE, LockIntent set to FALSE, and Open set to Open. If a conflict is detected, the operation MUST be failed with STATUS_FILE_LOCK_CONFLICT.

Return to top of page

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/12/22	V37.0 – 2014/05/15	In Section 2.3.81, FSCTL_OFFLOAD_WRITE Reply, clarified that the LengthWritten field must not be greater than the CopyLength field specified in the FSCTL_OFFLOAD_WRITE_INPUT data element. Changed from: LengthWritten (8 bytes): A 64-bit unsigned integer that contains the amount, in bytes, of data that was written. The value of this field MUST be greater than or equal to zero and MUST be aligned to a logical sector boundary on the volume. This value can be smaller than the CopyLength field specified in the FSCTL_OFFLOAD_WRITE_INPUT data element. A smaller value indicates that less data was logically written with the specified Token than was requested. Changed to: LengthWritten (8 bytes): A 64-bit unsigned integer that contains the amount, in bytes, of data that was written. The value of this field MUST be greater than or equal to zero and MUST be aligned to a logical sector boundary on the volume. This value can be smaller than the CopyLength field specified in the FSCTL_OFFLOAD_WRITE_INPUT data element. A smaller value indicates that less data was logically written with the specified Token than was requested. This field MUST NOT be greater than the CopyLength field specified in the FSCTL_OFFLOAD_WRITE_INPUT data element, meaning it is incorrect to copy more than what was requested.<71> <71> While it is valid to issue a single Offload Write operation for the full contents of a file, the Win32 CopyFileEx API does not perform this. Instead, CopyFileEx issues Offload Write operations in 256-MB chunks so that components like Internet Explorer can show proper progress of file copy operations.
2014/10/27	V37.0 – 2014/05/15	In Section 2.5.10, FileFsDeviceInformation, the description of FILE_DEVICE_ALLOW_APPCONTAINER_TRAVERSAL was corrected as follows: Changed from: The IO Manager performs a full security check for traverse access if the client is an appcontainer.<157> Changed to: The IO Manager performs a full security check for traverse access on every file open when the client is an appcontainer. Setting of this flag bypasses this enforced traverse access check if the client token already has traverse privileges.<157>
2014/10/27	V37.0 – 2014/05/15	In Section 2.5.9, FileFsVolumeInformation, a new processing rule was added to handle volume labels that exceed 32 characters. Text added is shown in **bold**. VolumeLabel (variable): A variable-length Unicode field containing the name of the volume. The content of this field can be a null-terminated string or can be a string padded with the space character to be VolumeLabelLength bytes long. This operation returns a status code, as specified in [MS-ERREF] section 2.3. The status code returned directly by the function that processes this file information class MUST be STATUS_SUCCESS or

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>one of the following.
If the volume label is greater than 32 characters, return the first 32 characters of the label and STATUS_SUCCESS.</td></tr>
<tr><td>2014/10/13</td><td>V37.0 – 2014/05/15</td><td>In Section 2.5.10, FileFsDeviceInformation, the constant FILE_PORTABLE_DEVICE has been added to the Characteristics field as follows:
<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>FILE_PORTABLE_DEVICE
0x0004000</td><td>Indicates that the given device resides on a portable bus like USB or Firewire and that the entire device (not just the media) can be removed from the system.</td></tr></table></td></tr>
</table>

Return to top of page

[MS-FSRVP]: File Server Remote VSS Protocol

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td>2014/09/16</td><td>V7.0 – 2014/05/15</td><td>In Section 2.2.4, Error Codes, added FSRVP_E_SHADOWCOPYSET_ID_MISMATCH to the table:
<table><tr><th>Return value/code</th><th>Description</th></tr><tr><td>FSRVP_E_SHADOWCOPYSET_ID_MISMATCH
(0x80042501)</td><td>The provided ShadowCopySetId does not exist.</td></tr></table>In sections 3.1.4.4, 3.1.4.5, 3.1.4.6, 3.1.4.7, 3.1.4.8, 3.1.4.12, and 3.1.4.13, added FSRVP_E_SHADOWCOPYSET_ID_MISMATCH to the table of error code return values and updated ShadowCopySet processing rules.
Added:
<table><tr><th>Return value/code</th><th>Description</th></tr><tr><td>0x80042501
FSRVP_E_SHADOWCOPYSET_ID_MISMATCH</td><td>The provided ShadowCopySetId does not exist.</td></tr></table>Changed from:
The server MUST look up the ShadowCopySet from GlobalShadowCopysetTable using the index ShadowCopySetId. If no shadow copy set is found, the server MUST fail the call with E_INVALIDARG.
Changed to:
The server MUST look up the ShadowCopySet from GlobalShadowCopysetTable using the index ShadowCopySetId. If no</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		shadow copy set is found, the server MUST fail the call with FSRVP_E_SHADOWCOPYSET_ID_MISMATCH.

Return to top of page

[MS-FSVCA]: File Set Version Comparison Algorithms

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/10/13	V1.0 – 2014/05/15	The following two new sections have been added to describe the algorithm used to update the FSVCA ADM metadata when an item changes: **3.1.4.4 Update the FSVCA metadata when an item changes** The inputs for this algorithm are the following: SyncGid: The 24-byte SYNC_GID that uniquely identifies an item within the dataset. IsDelete: A flag that indicates if an item is deleted from the dataset. The Participant store MUST update the metadata as follows: ▪ Increment the tick count of all the ClockVectorElements associated with the local replica in ClockVectorList. ▪ Search for a FileItemEntry in FileItemList where SyncGid is equal to FileItemEntry.SyncGid. If an entry is found: ▪ Set FileItemEntry.ChangeTickCount to the TickCount associated with the local replica in ClockVectorList. ▪ If IsDelete is TRUE, set FileItemEntry.IsDeleted to TRUE. ▪ Otherwise, create a FileItemEntry and initialize it as follows: ▪ Set FileItemEntry.SyncGid to SyncGid. ▪ Set FileItemEntry.CreateReplicaKey to the index of the local replica in the ReplicaTable. ▪ Set FileItemEntry.CreateTickCount to the TickCount associated with the local replica in ClockVectorList. ▪ Set FileItemEntry.ChangeReplicaKey to the index of the local replica in the ReplicaTable. ▪ Set FileItemEntry.ChangeTickCount to the TickCount associated with the local replica in ClockVectorList. ▪ Set FileItemEntry.IsDeleted to FALSE. ▪ Set WinnerExists to FALSE. ▪ Set WinningFileItem to 0. **3.1.4.5 Update the FSVCA metadata for the successfully**

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		applied items The inputs for this algorithm are the following: SyncMetadata: SYNC_CHANGE_INFORMATION structure provided by other replica. The Participant store MUST update the metadata as follows: ▪ For each CHANGE_SET_ENTRY in SyncMetadata.ChangeSetList, the Participant store MUST update the metadata as follows: ▪ Search for a FileItemEntry in FileItemList where CHANGE_SET_ENTRY.SyncGid is equal to FileItemEntry.SyncGid. If an entry is not found, construct a FileItemEntry, initialize it as follows, and insert it in FileItemList. ▪ Set FileItemEntry.SyncGid to CHANGE_SET_ENTRY.SyncGid. ▪ Update FileItemEntry as follows: ▪ Get the local index of SyncMetadata.MadeWithKnowledge.ReplicaKeys.REPLICA_GID referred by CHANGE_SET_ENTRY.CreateVersion.ReplicaKey in the ReplicaTable. Set FileItemEntry.CreateReplicaKey to the preceding index. ▪ Set FileItemEntry.CreateTickCount to CHANGE_SET_ENTRY.CreateVersion.ChangeTickCount. ▪ Get the local index of SyncMetadata.MadeWithKnowledge.ReplicaKeys.REPLICA_GID referred by CHANGE_SET_ENTRY.ChangeVersion.ReplicaKey in the ReplicaTable. Set FileItemEntry.ChangeReplicaKey to the preceding index. ▪ Set FileItemEntry.ChangeTickCount to CHANGE_SET_ENTRY.ChangeVersion.ChangeTickCount. ▪ If CHANGE_SET_ENTRY.SyncChange is equal to 0x00000001, set FileItemEntry.IsDeleted to TRUE. ▪ Set FileItemEntry.WinnerExists to TRUE if CHANGE_SET_ENTRY.WinnerExists is 0x01, FALSE otherwise. ▪ If FileItemEntry.WinnerExists is TRUE, set FileItemEntry.WinningFileItem to CHANGE_SET_ENTRY.WinnerSyncGid. ▪ Update RangeList and ClockVectorList using SyncMetadata.MadeWithKnowledge.

Return to top of page

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td>2014/10/13</td><td>V19.0 – 2014/05/15</td><td>In Section 2.2.11.2, ConsentPromptBehaviorAdmin, revised data values to specify the default (0x00000005), add missing values, and specify that values 1 and 2 occur on the secure desktop. Revisions are in italics and additions are shown in bold text.
Data: This MUST be a value in the following table.
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0x00000000</td><td>This option SHOULD be used to allow the Consent Admin to perform an operation that requires elevation without consent or credentials.</td></tr>
<tr><td>0x00000001</td><td>This option SHOULD be used to prompt the Consent Admin to enter his or her user name and password (or another valid admin) when an operation requires elevation of privilege. This operation occurs on the secure desktop.</td></tr>
<tr><td>0x00000002</td><td>This option SHOULD be used to prompt the administrator in Admin Approval Mode to select either "Permit" or "Deny" an operation that requires elevation of privilege. If the Consent Admin selects Permit, the operation will continue with the highest available privilege. "Prompt for consent" removes the inconvenience of requiring that users enter their name and password to perform a privileged task. This operation occurs on the secure desktop.</td></tr>
<tr><td>0x00000003</td><td>This option SHOULD be used to prompt the Consent Admin to enter his or her user name and password (or that of another valid admin) when an operation requires elevation of privilege.</td></tr>
<tr><td>0x00000004</td><td>This option SHOULD be used to prompt the administrator in Admin Approval Mode to select either "Permit" or "Deny" an operation that requires elevation of privilege. If the Consent Admin selects Permit, the operation will continue with the highest available privilege. "Prompt for consent" removes the inconvenience of requiring that users enter their name and password to perform a privileged task.</td></tr>
<tr><td>0x00000005</td><td>This option is the default. This option SHOULD be used to prompt the administrator in Admin Approval Mode to select either "Permit" or "Deny" an operation that requires elevation of privilege for any non-Windows binaries. If the Consent Admin selects Permit, the</td></tr>
</table></td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		operation will continue with the highest available privilege. This operation will happen on the secure desktop. Windows binaries will be allowed to perform an operation that requires elevation without consent or credentials.
2014/09/16	V19.0 – 2014/05/15	In Section 2.2.1.2, Account Lockout Policies, revised the description of LockoutDurationTEXT. Changed from: Number of minutes that a locked-out account MUST remain locked out before automatically becoming unlocked. The value MUST be in the range -2^32 to 2^32. If the account lockout duration value is set to a negative value, the account MUST be locked out until an administrator explicitly unlocks it. If an account lockout threshold is defined, the account lockout duration MUST be greater than or equal to the reset time, ResetLockoutCount. This setting only has meaning when an account lockout threshold is specified. Changed to: The number of minutes that a locked-out account MUST remain locked out before automatically becoming unlocked. The value MUST be either -1 or in the range 1 to 99,999. If the account lockout duration value is set to negative 1, the account MUST be locked out until an administrator explicitly unlocks it. If an account lockout threshold is defined, the account lockout duration MUST be greater than or equal to the reset time, ResetLockoutCount. This setting only has meaning when an account lockout threshold is specified.

Return to top of page

[MS-GPOL]: Group Policy: Core Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/10/27	V31.0 – 2014/05/15	In Section 2.2.4, GPO Search, revised the description of gPCMachineExtensionNames. Changed from:

Attribute	Format
gPCMachineExtensionNames	A directory string with the format:[<CSE GUID1><TOOL GUID1>][<CSE GUID 2><TOOL GUID2>]where <CSE GUIDn> is a CSE GUID and <TOOL GUIDn> is a tool extension GUID, and the "[" and "]" characters are to be interpreted literally. The CSE GUID and tool extension GUID are each a 38-character "curly braced GUID

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td>string" as defined in [MS-GLOS].</td></tr>
</table>
Changed to:
<table>
<tr><th>Attribute</th><th>Format</th></tr>
<tr><td>gPCMachineExtensionNames</td><td>A directory string with the format:[<CSE GUID1><TOOL GUID1>][<CSE GUID2><TOOL GUID2>] sorted in case-insensitive ascending order by <CSE GUID> where <CSE GUIDn> is a CSE GUID and <TOOL GUIDn> is a tool extension GUID, and the "[" and "]" characters are to be interpreted literally. The CSE GUID and tool extension GUID are each a 38-character "curly braced GUID string" as defined in [MS-GLOS]. Group Policy processing terminates at the first <CSE GUIDn> out of sequence.</td></tr>
</table>
</td></tr>
</table>

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/11/10	V2.0 – 2014/05/15	In Section 2.2.2, Host Header, added information for the Server Name Indication extension. Added the following sentence at the end of the first paragraph: When using HTTPS and the server name indication extension specified in [RFC6066], the hostname specified in the Host header MUST match the hostname specified in the server name indication extension.

Return to top of page

[MS-KILE]: Kerberos Protocol Extensions

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/05/11	V29.0 – 2014/05/15	Updated product behavior notes with the correct product versions and features that support Resource SID compression. Changed from: <9> Section 2.2.6: The Resource-SID-compression-disabled bit is

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2. <66> Section 3.3.5.7.3: Resource SID compression is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, or Windows Server 2008 R2. Changed to: <9> Section 2.2.6: The Resource-SID-compression-disabled bit is not supported in Windows 2000, Windows Server 2003, Windows Server 2008, or Windows Server 2008 R2 KDCs. <66> Section 3.3.5.7.3: Resource SID compression is not supported in Windows 2000, Windows Server 2003, Windows Server 2008, or Windows Server 2008 R2 KDCs.
2014/12/22	V29.0 – 2014/05/15	In Section 3.3.5.6, AS Exchange, references to the UseDESOnly flag have been removed. Changed from: The KDC SHOULD check whether the krbtgt account has the UseDESOnly flag: ▪ If the UseDESOnly flag is set: the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the AS-REP message, include PA-DATA with the padata-type set to PA-SUPPORTED-ENCTYPES (165), and the padata-value set to 0x3 (section 2.2.6). ▪ Otherwise: ▪ If domainControllerFunctionality returns a value < 3 ([MS-ADTS] section 3.1.1.3.2.25): the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the AS-REP message, include PA-DATA with the padata-type set to PA-SUPPORTED-ENCTYPES (165), and the padata-value set to 0x7 (section 2.2.6). ▪ If domainControllerFunctionality returns a value >= 3: the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the AS-REP message, include PA-DATA with the padata-type set to PA-SUPPORTED-ENCTYPES (165), and the padata-value set to 0x1F (section 2.2.6). Changed to: The KDC SHOULD check the value that is returned by domainControllerFunctionality ([MS-ADTS] section 3.1.1.3.2.25). ▪ If domainControllerFunctionality returns a value < 3: the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the AS-REP message, include PA-DATA with the padata-type set to PA-SUPPORTED-ENCTYPES (165), and the padata-value set to 0x7 (section 2.2.6). ▪ If domainControllerFunctionality returns a value >= 3: the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11],

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Appendix A) of the AS-REP message, include PA-DATA with the padata-type set to PA-SUPPORTED-ENCTYPES (165), and the padata-value set to 0x1F (section 2.2.6).
2014/09/16	V29.0 – 2014/05/15	In Section 3.3.5.4, Determining Authentication Policy Silo Membership, updated text to use BelongsToSilo rather than SiloName: Changed from: If domainControllerFunctionality returns a value < 6 ([MS-ADTS] section 3.1.1.3.2.25), the KDC SHOULD set SiloName to NULL.<45> If domainControllerFunctionality returns a value >= 6, the KDC SHOULD check whether the account is a member of an Authentication Policy Silo: ▪ If the AssignedSilo (section 3.3.1.1) is NULL, the KDC SHOULD set SiloName to NULL. ▪ If the AssignedSilo is not NULL and AssignedSilo.msDS-AuthNPolicySiloMembers does not contain the account, the KDC SHOULD set SiloName to NULL. ▪ If the AssignedSilo is not NULL and AssignedSilo.msDS-AuthNPolicySiloMembers contains the account, the KDC SHOULD set SiloName to AssignedSilo.RDN. Changed to: If domainControllerFunctionality returns a value < 6 ([MS-ADTS] section 3.1.1.3.2.25), the KDC SHOULD set BelongsToSilo to FALSE.<45> Note The BelongsToSilo variable is a Boolean variable that is used for illustrative purposes in the processing rules of this section and section 3.3.5.5. The value of BelongsToSilo is not persisted across client requests. If domainControllerFunctionality returns a value >= 6, the KDC SHOULD check whether the account is a member of an Authentication Policy Silo: ▪ If the AssignedSilo (section 3.3.1.1) is NULL, the KDC SHOULD set BelongsToSilo to FALSE. ▪ If the AssignedSilo is not NULL and AssignedSilo.msDS-AuthNPolicySiloMembers does not contain the account, the KDC SHOULD set BelongsToSilo to FALSE. ▪ If the AssignedSilo is not NULL and AssignedSilo.msDS-AuthNPolicySiloMembers contains the account, the KDC SHOULD set BelongsToSilo to TRUE. In Section 3.3.5.5, Determining Authentication Policy Settings, updated the text to clarify that BelongsToSilo == TRUE indicates that the account belongs to a silo, as follows: Changed from: If the account belongs to a Silo (section 3.3.5.4), when the account

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		is of type: If BelongsToSilo == TRUE (section 3.3.5.4) for the account, the account belongs to a Silo. In this case, when the account is of type: Changed from: If the account does not belong to a Silo and AssignedPolicy (section 3.3.1.1) is NULL, the KDC SHOULD set PolicyName to NULL and Enforced to FALSE. If the account does not belong to a Silo and the AssignedPolicy is not NULL, the KDC SHOULD set PolicyName to AssignedPolicy.RDN, Enforced to AssignedPolicy.msDS-AuthNPolicyEnforced, and when the account is of type: Changed to: If the account does not belong to a Silo (BelongsToSilo == TRUE (section 3.3.5.4)) and AssignedPolicy (section 3.3.1.1) is NULL, the KDC SHOULD set PolicyName to NULL and Enforced to FALSE. If the account does not belong to a Silo (BelongsToSilo == TRUE (section 3.3.5.4)) and the AssignedPolicy is not NULL, the KDC SHOULD set PolicyName to AssignedPolicy.RDN, Enforced to AssignedPolicy.msDS-AuthNPolicyEnforced, and when the account is of type: In Section 3.3.5.6, AS Exchange, updated the text to remove reference to additional rights GUID, as follows: Changed from: If AllowedToAuthenticateFrom is not NULL, the PAC of the armor TGT MUST be used to perform an access check for the ACTRL_DS_CONTROL_ACCESS right with additional rights GUID against the AllowedToAuthenticateFrom. If the access check fails, the KDC MUST return KDC_ERR_POLICY. Changed to: If AllowedToAuthenticateFrom is not NULL, the PAC of the armor TGT MUST be used to perform an access check for the ACTRL_DS_CONTROL_ACCESS right against the AllowedToAuthenticateFrom. If the access check fails, the KDC MUST return KDC_ERR_POLICY. In Section 3.3.5.7, TGS Exchange, updated the text to clarify behavior if domainControllerFunctionality returns a value >= 6, as follows: Changed from: If domainControllerFunctionality returns a value >= 6 ([MS-ADTS] section 3.1.1.3.2.25), the KDC MUST determine whether an Authentication Policy is applied to the server or service (section 3.3.5.5); if Enforced is TRUE then: <62> ▪ If AllowedToAuthenticateTo is not NULL, the PAC of the user and the PAC of the armor TGT MUST be used to perform an access check for the ACTRL_DS_CONTROL_ACCESS right with additional rights GUID against the AllowedToAuthenticateTo. If the access check fails, the KDC MUST return KDC_ERR_POLICY. Changed to: If domainControllerFunctionality returns a value >= 6 ([MS-ADTS]

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		section 3.1.1.3.2.25) and the account is not also the application service account, the KDC MUST determine whether an Authentication Policy is applied to the server or service (section 3.3.5.5); if Enforced is TRUE then:<62> ▪ If AllowedToAuthenticateTo is not NULL, the PAC of the user and the PAC of the armor TGT MUST be used to perform an access check for the ACTRL_DS_CONTROL_ACCESS right against the AllowedToAuthenticateTo. If the access check fails, the KDC MUST return KDC_ERR_POLICY.
2014/09/16	V29.0 – 2014/05/15	In Section 3.3.5.6, AS Exchange, the description of the return error code was changed as follows: Changed from: If pre-authentication used DES or RC4, the KDC MUST return KDC_ERR_POLICY. Changed to: If pre-authentication used DES or RC4, the KDC MUST return KDC_ERR_ETYPE_NOTSUPP.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/06/08	V37.0 – 2014/05/15	In Section 2.2.7.9, LSAPR_TRUSTED_DOMAIN_INFORMATION_EX, added the TRUST_ATTRIBUTE_CROSS_ORGANIZATION_NO_TGT_DELEGATION flag (0x00000200) to the possible values of the TrustAttributes field of the LSAPR_TRUSTED_DOMAIN_INFORMATION_EX structure. Changed from: TrustAttributes: This field contains bitmapped values that define the attributes of the trust.<24> <24> Section 2.2.7.9: The following is a timeline of when each flag value was introduced. Unless otherwise specified, all flag values continue to be available in subsequent versions of Windows according to the applicability list at the beginning of this section.

Possible value	Value	Product
…	…	…
TATE (TRUST_ATTRIBUTE_TREAT_AS_EXTERNAL)	0x00000040	Windows Server 2003.
Obsolete	0x00400000	Introduced in Windows 2000 RTM. Became obsolete in

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>Windows 2000 operating system Service Pack 4 (SP4).</td></tr>
<tr><td>…</td><td>…</td><td>…</td></tr>
</table>
Changed to:
TrustAttributes: This field contains bitmapped values that define the attributes of the trust.<24>
<24> Section 2.2.7.9: The following is a timeline of when each flag value was introduced. Unless otherwise specified, all flag values continue to be available in subsequent versions of Windows according to the applicability list at the beginning of this section.
<table>
<tr><th>Possible value</th><th>Value</th><th>Product</th></tr>
<tr><td>…</td><td>…</td><td>…</td></tr>
<tr><td>TATE (TRUST_ATTRIBUTE_TREAT_AS_EXTERNAL)</td><td>0x00000040</td><td>Windows Server 2003.</td></tr>
<tr><td>TANC (TRUST_ATTRIBUTE_CROSS_ORGANIZATION_NO_TGT_DELEGATION)</td><td>0x00000200</td><td>Windows Server 2012 operating system</td></tr>
<tr><td>Obsolete</td><td>0x00400000</td><td>Introduced in Windows 2000 RTM. Became obsolete in Windows 2000 operating system Service Pack 4 (SP4).</td></tr>
<tr><td>…</td><td>…</td><td>…</td></tr>
</table>
</td></tr>
</table>

Return to top of page

Windows Protocols Errata M-R

This topic lists the Errata found in the Windows Protocols Technical Specifications, Overview Documents, and Reference documents titled [MS-M…] **through** [MS-R…] **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.** **To find out more about the types of issues that are included in Errata, see Windows Protocols Errata.** **Errata are subject to the same terms as the Open Specifications documentation referenced.**	**RSS** **Atom**

[MS-MDM]: Mobile Device Management Protocol

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

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

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

[MS-NRPC]: Netlogon Remote Protocol

[MS-PSRDP]: PowerShell Remoting Debugging Protocol

[MS-PSRP]: PowerShell Remoting Protocol

[MS-RA]: Remote Assistance Protocol

[MS-RAI]: Remote Assistance Initiation Protocol

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

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

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

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

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

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

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

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

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

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

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

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

[MS-RPRN]: Print System Remote Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description

2014/12/08	V3.0 – 2014/05/15	In Sections 3 and 6, added two classes to support user-defined security measures and access to Windows Update policies, and the corresponding class values for the SyncML Request Get and Replace commands versions.
		In Section 6.2, MDMSettingsProv MOF File, added the following to the MOF file:
		
		`[Description("This class provides user-specific security health metrics on the device")]`
		`class MDM_SecurityStatusUser`
		`{`
		`  [Key,Description("The key to identify the instance of MDM_SecurityStatusUser class")]`
		`  Uint32 Key;`
		`  [Read,Description("This property returns the presence of connected account")]`
		`  boolean HasConnectedAccount;`
		`  [Description("This property gets/sets the connected account policy to require"),`
		`  Values {"Don't require", "Require"}]`
		`  uint32 ConnectedAccountPolicy;`
		`  [Read,Description("This property returns the status of the account password"),`
		`  Values {"Not compliant", "No policy", "Compliant"}]`
		`  Uint32 PasswordStatus;`
		`  [Read,Description("This property returns the device encryption status as reported by EasClientSecurityPolicy.CheckCompliance result")]`
		`  uint32 EncryptionStatus;`
		`  [Description("This property gets/sets the device encryption policy to require"),`
		`  Values {"Don't require", "Require"}]`
		`  uint32 DeviceEncryptionPolicy;`
		`};`
		
		`[Description("This class provides access to Windows Update policies")]`
		`class MDM_Updates`
		`{`
		`  [Key,Description("The key to identify the instance of MDM_Updates class")]`
		`  Uint32 Key;`
		`  [Description("This property gets/sets the policy for regular application of Windows updates"), Values {"None", "Important", "Recommended"}]`
		`  Uint32 AutoUpdatePolicy;`
		`};`
		
		In Section 3.1.5.1.6 Get, and Section 3.1.5.1.7, Replace, added the following two class values:

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Class reference</th><th>LocURI</th><th>Description</th></tr>
<tr><td>[MDM_SecurityStatusUser]</td><td>./cimv2/MDM_SecurityStatusUser</td><td>A class defined by the MDM protocol (see section 6) that provides security health metrics on the device for the current user.</td></tr>
<tr><td>[MDM_Updates]</td><td>./cimv2/MDM_Updates</td><td>A class defined by the MDM protocol (see section 6) for managing device update settings.</td></tr>
</table>
The preceding changes are supported in Windows Server 2012 R2 with [MSKB-3013816].
</td></tr>
</table>

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/09/16	V9.0 – 2014/05/15	Modified the product behavior notes for the Query String Response Transfer Protocol and SAML1.1 Assertion Extension to correct the supported product versions: Changed from: The Query String Response Transfer Protocol is applicable where the Microsoft Web Browser Federated Sign-On Protocol is applicable. Changed to: The Query String Response Transfer Protocol is applicable where the Microsoft Web Browser Federated Sign-On Protocol is applicable.<3>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		<3> Section 1.6: The Query String Response Transfer Protocol is supported only in Windows Server 2003 R2, Windows Server 2008, Windows Server 2008 R2, and Windows Server 2012. Changed from: The SAML 1.1 Assertion Extension is applicable when the protected HTTP web application requires SIDs to perform authorization.<3> <3> Section 1.6: All Windows behavior documented in this specification applies to Windows Server 2003 R2, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2. The Windows behavior does not apply to Windows Vista, Windows 7, Windows 8, or Windows 8.1 because the software that implements the protocol extensions described in this document is included only as part of Windows Server. Changed to: The SAML 1.1 Assertion Extension is applicable when the protected HTTP web application requires SIDs to perform authorization.<4> <4> Section 1.6: SAML1.1 Assertion Extension is supported only in Windows Server 2003 R2, Windows Server 2008, and Windows Server 2008 R2.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/04/13	V9.0 – 2014/05/15	The Windows client products have been removed from the list of supported products. In Section 6, Appendix A: Product Behavior: Changed from: ▪ Windows 2000 operating system ▪ Windows XP operating system ▪ Windows Server 2003 R2 operating system ▪ Windows Vista operating system ▪ Windows Server 2008 operating system ▪ Windows 7 operating system ▪ Windows Server 2008 R2 operating system ▪ Windows 8 operating system ▪ Windows Server 2012 operating system ▪ Windows 8.1 operating system

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ Windows Server 2012 R2 operating system Changed to: ▪ Windows Server 2003 R2 operating system ▪ Windows Server 2008 operating system ▪ Windows Server 2008 R2 operating system ▪ Windows Server 2012 operating system ▪ Windows Server 2012 R2 operating system In Section 1.6, Applicability Statement, removed the following product behavior note: <1> Section 1.6: All Windows behavior documented in this specification applies only to Windows Server 2003 R2, Windows Server 2008, Windows Server 2008 R2 operating system, Windows Server 2012, and Windows Server 2012 R2. In Section 3.4.5, Processing Events and Sequencing Rules: Changed from: <86> Section 3.4.5: Internet Explorer in Windows XP, Windows Server 2003, Windows Server 2003 R2, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 always passes protocol messages through unaltered. The RMS 2.0 client in Windows Vista operating system with Service Pack 1 (SP1), Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 adds a whr parameter to the wsignin 1.0 Request Message (section 2.2.3) if the wsignin 1.0 Request Message does not already contain a whr parameter. Changed to: <86> Section 3.4.5: The RMS 2.0 client in Windows Vista operating system with Service Pack 1 (SP1), Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 adds a whr parameter to the wsignin 1.0 Request Message (section 2.2.3) if the wsignin 1.0 Request Message does not already contain a whr parameter.
2014/09/16	V9.0 – 2014/05/15	In Section 2.2.3, wsignin1.0 Request Message, updated the descriptions of wreply and wtrealm. Removed the following line from the description of wtrealm: If present, the wreply parameter MUST NOT be present. For processing semantics on wtrealm and wreply, see section 3.1.5.4.2. Revised the description of wreply to read as follows: wreply (optional): This parameter MAY be included in request messages to the same security realm as the relying party. If present, this value MUST be a URL to which responses MUST be directed. The requestor IP/STS MUST validate that this URL belongs to the relying party before directing responses to this URL.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		In Section 3.1.5.4.2, Message Validation, removed the following text: If a resource IP/STS receives a request that has both the wreply and wtrealm parameters set, the resource IP/STS MUST return an HTTP 1.1 status code 500 server error. If a requestor IP/STS receives a request that has both the wreply and wtrealm parameters set, the requestor IP/STS MUST ignore the wreply.
2014/09/16	V9.0 – 2014/05/15	In Section 2.2.4.2.1.3, Subject Element, revised the following text: Changed from: Subject element MUST specify the NameIdentifier element, as specified in [SAMLCore] section 2.4.2.2. Changed to: Subject element MAY<26> specify the NameIdentifier element, as specified in [SAMLCore] section 2.4.2.2. <26>Section 2.2.4.2.1.3: On Active Directory Federation Services that shipped with Windows Server 2003 R2, Windows Server 2008, and Windows Server 2008 R2, Windows always specifies the NameIdentifier claim, and the value for the NameIdentifier element is the value of the EmailAddress, user principal name (UPN), or CommonName claim, as specified in the Abstract Data Model in section 3.1.1.4. The NameIdentifier element specifies the Format attribute, as specified in [SAMLCore] section 2.4.2.2. The corresponding value of the Format attribute is one of the following, as specified in the Abstract Data Model (see section 3.1.1.4). Added the following text to Appendix A: Product Behavior (Note <54>): <54> Section 3.1.5.4.7: The following Windows behaviors apply for response message processing: On Active Directory Federation Services that shipped with Windows Server 2003 R2, Windows Server 2008, and Windows Server 2008 R2, Windows does not specify the SubjectConfirmation element when emitting wsignin1.0 response messages.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/12/22	V26.0 – 2014/05/15	Section 3.2.5.1.2, Server Receives an AUTHENTICATE_MESSAGE from the Client, was updated to clarify the case sensitivity of NTLM user names. Changed from: Both the client and the server now have the session, signing, and sealing keys. When the client runs an integrity check on the next message from the server, it detects that the server has determined (either directly or indirectly) the user password.

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>Changed to:
Both the client and the server now have the session, signing, and sealing keys. When the client runs an integrity check on the next message from the server, it detects that the server has determined (either directly or indirectly) the user password.
Note User names MUST be case-insensitive. For additional information about the case sensitivity of user names, see [MS-AUTHSOD] section 1.1.1.2.</td></tr>
<tr><td>2014/11/24</td><td>V26.0 – 2014/05/15</td><td>Section 3.2.5, Message Processing Events and Sequencing Rules, was updated to add conditions to the test for NegFlg during AUTHENTICATE_MESSAGE processing.
In Section 3.2.5.1.2, Server Receives an AUTHENTICATE_MESSAGE from the Client:
Changed a portion of the pseudocode from:
<pre>If (NTLMSSP_NEGOTIATE_KEY_EXCH flag is set in NegFlg)</pre>Changed to:
<pre>If (NTLMSSP_NEGOTIATE_KEY_EXCH flag is set in NegFlg
  AND (NTLMSSP_NEGOTIATE_SIGN OR NTLMSSP_NEGOTIATE_SEAL
are set in NegFlg))</pre>In Section 3.2.5.2.2, Server Response Checking:
Changed a portion of the pseudocode from:
<pre>If (NTLMSSP_NEGOTIATE_KEY_EXCH flag is set in NegFlg)</pre>Changed to:
<pre>If (NTLMSSP_NEGOTIATE_KEY_EXCH flag is set in NegFlg
  AND (NTLMSSP_NEGOTIATE_SIGN OR NTLMSSP_NEGOTIATE_SEAL
are set in NegFlg))</pre></td></tr>
<tr><td>2014/11/10</td><td>V26.0 – 2014/05/15</td><td>In Section 2.2.2.1, AV_PAIR, clarified that the structure associated with MsvAvTimestamp is sent in the CHALLENGE_MESSAGE. Text added is shown in bold.
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>MsvAvTimestamp
0x0007</td><td>A FILETIME structure ([MS-DTYP] section 2.3.3) in little-endian byte order that contains the server local time. This structure is always sent in the CHALLENGE_MESSAGE.<14></td></tr>
</table></td></tr>
<tr><td>2014/11/10</td><td>V26.0 – 2014/05/15</td><td>In Section 2.2.1.1, NEGOTIATE_MESSAGE, Section 2.2.1.2, CHALLENGE_MESSAGE, and Section 2.2.1.3, AUTHENTICATE_MESSAGE, the description for NEGOTIATE_MESSAGE has been revised to clarify that the Version field of the various NTLM</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		messages is required, though not always populated. Changed from: Version (8 bytes): A VERSION structure (section 2.2.2.10) that is present only when the NTLMSSP_NEGOTIATE_VERSION flag is set in the NegotiateFlags field. This structure is used for debugging purposes only. In normal protocol messages, it is ignored and does not affect the NTLM message processing. Changed to: Version (8 bytes): A VERSION structure (section 2.2.2.10) that is populated only when the NTLMSSP_NEGOTIATE_VERSION flag is set in the NegotiateFlags field. This structure is used for debugging purposes only. In normal protocol messages, it is ignored and does not affect the NTLM message processing. In Section 2.2.2.10 VERSION, a related clarification was made as follows: Changed from: The VERSION structure contains Windows version information that SHOULD be ignored. This structure is used for debugging purposes only and its value does not affect NTLM message processing. It is present in the NEGOTIATE_MESSAGE, CHALLENGE_MESSAGE, and AUTHENTICATE_MESSAGE messages only if NTLMSSP_NEGOTIATE_VERSION is negotiated.<28> Changed to: The VERSION structure contains Windows version information that SHOULD be ignored. This structure is used for debugging purposes only and its value does not affect NTLM message processing. It is populated in the NEGOTIATE_MESSAGE, CHALLENGE_MESSAGE, and AUTHENTICATE_MESSAGE messages only if NTLMSSP_NEGOTIATE_VERSION is negotiated.<28>
2014/11/10	V26.0 – 2014/05/15	In Section 3.2.1.1, Variables Internal to the Protocol, revised the description of DnsForestName to include details about machines that are not domain joined. Text added is shown in **bold**. DnsForestName: A string that indicates the FQDN (2) of the server's forest. **The DnsForestName is NULL on machines that are not domain joined.**
2014/11/10	V26.0 – 2014/05/15	In Section 2.2.2.7, (NTLM v2: NTLMv2_CLIENT_CHALLENGE), added a product behavior note that is relevant to the NTLMv2_CLIENT_CHALLENGE structure. Text added is shown in **bold**. The NTLMv2_CLIENT_CHALLENGE structure defines the client challenge in the AUTHENTICATE_MESSAGE. This structure is used only when NTLM v2 authentication is configured and is transported in the NTLMv2_RESPONSE (section 2.2.2.8) structure. **<28>** **<28> In some situations, Windows adds bytes to the end of the variable-length section. These bytes are considered to be part of the NTLMv2_CLIENT_CHALLENGE structure, but have no defined contents.**
2014/10/13	V26.0 – 2014/05/15	In Section 3.1.5.1.2, Client Receives a CHALLENGE_MESSAGE from the Server, clarified the information about sending the LmChallengeResponse during NTLM v2 authentication as follows:

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Changed from: If NTLM v2 authentication is used and the CHALLENGE_MESSAGE contains a TargetInfo field, the client SHOULD NOT send the LmChallengeResponse and SHOULD set the LmChallengeResponseLen and LmChallengeResponseMaxLen fields in the AUTHENTICATE_MESSAGE to zero. <43> <43> Section 3.1.5.1.2: This functionality is not supported in Windows NT, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008. Changed to: If NTLM v2 authentication is used and the CHALLENGE_MESSAGE TargetInfo field (section 2.2.1.2) has an MsvAvTimestamp present, the client SHOULD NOT send the LmChallengeResponse and SHOULD send Z(24) instead.<43> <43> Section 3.1.5.1.2: This functionality is not supported in Windows NT and Windows 2000.
2014/09/16	V26.0 – 2014/05/15	In Section 2.2.2.2, Single_Host_Data, removed the DataPresent field and expanded the CustomData field and updated its description as follows: Changed from: CustomData (4 bytes): An optional 4 byte platform-specific blob. Changed to: CustomData (8 bytes): An 8 byte platform-specific blob containing info only relevant when the client and the server are on the same host.

Return to top of page

[MS-NRPC]: Netlogon Remote Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/11/10	V31.0 – 2014/05/15	In Section 3.5.4.4.9, NetrLogonGetDomainInfo (Opnum 29), added information that describes how the SupportedEncTypes field of the NETLOGON_DOMAIN_INFO structure is used. Text added is shown in **bold**. WkstaBuffer.WorkstationInfo.OsName and WkstaBuffer.WorkstationInfo.OsVersion SHOULD be processed as specified in section 2.2.1.3.6. If WkstaBuffer.WorkstationInfo.OsName and WkstaBuffer.WorkstationInfo.OsVersion are not specified, then a generic string SHOULD be used to update the operatingSystem attribute ("Windows unknown version"). If WkstaBuffer.WorkstationInfo.OsVersion is specified but WkstaBuffer.WorkstationInfo.OsName is not, then a different generic string SHOULD be used to update the operatingSystem attribute, depending on the value of WkstaBuffer.WorkstationInfo.OsVersion.wProductType. If the wProductType is VER_NT_WORKSTATION, then the string that SHOULD be used is "Windows Workstation", otherwise the string SHOULD be "Windows Server".<258>

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>If WkstaBuffer.WorkstationInfo.KerberosSupportedEncryptionTypes is set, NETLOGON_DOMAIN_INFO.SupportedEncTypes is set to the msDS-SupportedEncryptionTypes attribute ([MS-ADA2] section 2.444) of the client account.</td></tr>
<tr><td>2014/10/27</td><td>V31.0 – 2014/05/15</td><td>In Section 2.2.1.2.1, DOMAIN_CONTROLLER_INFOW, the bit flags for the Flags field have been updated as follows:
Changed from:
Flags bit layout: bit 16=Q, 17=0
<table><tr><th>Value</th><th>Description</th></tr><tr><td>…</td><td>…</td></tr><tr><td>Q</td><td>The DC has a functional level of DS_BEHAVIOR_WIN2012 or later.</td></tr></table>Changed to:
Flags bit layout: bit 16=R, 17=Q
<table><tr><th>Value</th><th>Description</th></tr><tr><td>…</td><td>…</td></tr><tr><td>Q</td><td>The DC has a functional level of DS_BEHAVIOR_WIN2012 or later.</td></tr><tr><td>R</td><td>The DC has a functional level of DS_BEHAVIOR_WIN2012R2 or later.</td></tr></table>In Section 3.5.4.3.1, DsrGetDcNameEx2 (Opnum 34) for Return Values:
Changed from:
Flags B, Q, and U MUST NOT be combined with each other.
Changed to:
Flags B, Q, U, V, and W MUST NOT be combined with each other.</td></tr>
<tr><td>2014/09/16</td><td>V31.0 – 2014/05/15</td><td>In Section 2.2.1.3.7, NL_TRUST_PASSWORD, corrected the random data size in the diagram illustrating the domain trust password buffer format, as follows:
Changed from:
Random Data of size (256 – Length - 12) bytes
Changed to:
Random Data of size (512 – Length - 12) bytes</td></tr>
</table>

Return to top of page

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/10/27	V1.0 – 2014/05/15	Section 2.2.8, REMOTE_DEBUGGER_BREAKPOINT_UPDATED_EVENT Message, and Section 2.2.9, REMOTE_DEBUGGER_STOP_EVENT Message, have been updated to reflect that the PSRP USER_EVENT message does not have a property called "event name". In Section 2.2.8, REMOTE_DEBUGGER_BREAKPOINT_UPDATED_EVENT Message: Changed from: The PSRP USER_EVENT event name MUST be PSInternalRemoteDebuggerBreakpointUpdatedEvent. Changed to: The PSRP USER_EVENT source identifier string MUST be PSInternalRemoteDebuggerBreakpointUpdatedEvent. In Section 2.2.9, REMOTE_DEBUGGER_STOP_EVENT Message: Changed from: The PSRP USER_EVENT name MUST be PSInternalRemoteDebuggerStopEvent. Changed to: The PSRP USER_EVENT source identifier string MUST be PSInternalRemoteDebuggerStopEvent.

Return to top of page

[MS-PSRP]: PowerShell Remoting Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/04/27	V14.0 – 2014/05/15	In Section 1.2.1, Normative References, added a reference to [MS-UCODEREF] Windows Protocols Unicode Reference and in Section 2.2.3.20, Wilcard, made the following change: Changed from: All character comparisons are locale-invariant, ordinal-based, and case-insensitive. Changed to: All character comparisons are locale-invariant, ordinal-based, and case-insensitive, as defined in [MS-UCODEREF] section 3.1.5.5.
2014/09/16	V14.0 – 2014/05/15	In Section 2.2.1, PowerShell Remoting Protocol Message, changed the value of CONNECT_RUNSPACEPOOL to 0x00010008 and changed the value of RUNSPACE_INIT_DATA to 0x0002100B.
2014/09/16	V14.0 – 2014/05/15	In Section 2.2.2.28, CONNECT_RUNSPACEPOOL Message, updated the content as follows: Default Runspace pool containing a single Runspace Empty string field is the same as MinRunspaces and MaxRunspaces properties set to the value of 1. The following example specifies a Runspace pool with up to five Runspaces.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		<Obj RefId="1"> <MS> <I32 N="MinRunspaces">1</I32> <I32 N="MaxRunspaces">5</I32> </MS> </Obj> The following example specifies a default Runspace pool with a single Runspace. <Obj RefId="1"> <MS> <S></S> </MS> </Obj>
2014/09/16	V14.0 – 2014/05/15	In Section 2.2.3.11, PowerShell Pipeline, added the following content: String, conveying command history information to the higher layer. The PSRP layer MUST NOT interpret this data. ▪ Property name: History. ▪ Property type: String (see section 2.2.5.1.1). Boolean, indicating to the higher layer if error output should be redirected. The PSRP layer MUST NOT interpret this data. ▪ Property name: RedirectShellErrorOutputPipe. ▪ Property type: Boolean (see section 2.2.5.1.3).
2014/09/16	V14.0 – 2014/05/15	In Section 2.2.3.28, BufferCell, corrected the capitalization of the following Property names from uppercase to lowercase: character, foregroundColor, backgroundColor, bufferCellType.
2014/09/16	V14.0 – 2014/05/15	In Section 3.1.5.3.15, Rules for the wxf:ConnectResponse Message, added the following to specify how a SESSION_CAPABILITY message is sent as a part of a ConnectResponse message: The SESSION_CAPABILITY message is included in the ConnectResponse message as a base-64 encoded string inside a <connectResponseXml> tag. The base-64 encoded string is a serialized complex object (section 2.2.5.2). Example response:

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		<rsp:ConnectResponse> <rsp:InputStreams>stdinpr</rsp:InputStreams> <rsp:OutputStreams>stdout</rsp:OutputStreams> <connectResponseXml xmlns="http://schemas.microsoft.com/powershell"> Base64-Encoded data </connectResponseXml> </rsp:ConnectResponse> Decoded SESSION_CAPABILITY <connectResponseXml xmlns="http://schemas.microsoft.com/powershell"> <Obj RefId="0"> <MS> <Version N="protocolversion">2.2</Version> <Version N="PSVersion">2.0</Version> <Version N="SerializationVersion">1.1.0.1</Version> </MS> </Obj> </connectResponseXml>
2014/09/16	V14.0 – 2014/05/15	In Section 2.2.4.15, DisconnectType, added the BufferMode element as shown below: <xs:complexType name="DisconnectType"> <xs:sequence> <xs:element name="IdleTimeOut" type="xs: duration" minOccurs="0" /> <xs:element name="BufferMode" type=" OutputBufferingModeEnumeration" minOccurs="0"

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		/>
</xs:sequence>
</xs:complexType>

IdleTimeOut: This overrides the IdleTimeOut value specified in the Shell data type when the Shell was created.

BufferMode: This overrides the BufferMode value specified in the Shell data type when the Shell was created. |
| 2014/09/16 | V14.0 – 2014/05/15 | In Section 2.2.2.25, PROGRESS_RECORD Message, revised the example to the following:

```
<Obj RefId="0">
  <MS>
    <S N="Activity">Activity Name</S>
    <I32 N="ActivityId">4</I32>
    <S N="StatusDescription">Good</S>
    <S N="CurrentOperation">Down loading</S>
    <I32 N="ParentActivityId">-1</I32>
    <I32 N="PercentComplete">20</I32>
    <Obj N="Type" RefId="1">
      <TN RefId="0">

<T>System.Management.Automation.ProgressRecordType</T>
        <T>System.Enum</T>
        <T>System.ValueType</T>
        <T>System.Object</T>
      </TN>
      <ToString>Processing</ToString>
      <I32>0</I32>
    </Obj>
    <I32 N="SecondsRemaining">30</I32>
  </MS>
</Obj>
```

In Section 2.5.1.25, Progress Record, revised the description, element descriptions, and example to the following:

Represents the status of an ongoing operation at a point in time. The Progress Record is serialized as a complex object as described in section 2.2.5.2.

Activity: An <S N="Activity"> XML element with a string describing the activity for which progress is being reported.

ActivityId: An <I32 N="ActivityId"> XML element with an integer identifying the activity for which progress is being reported.

CurrentOperation: An <S N="CurrentOperation"> XML element with a string describing the current operation of the many required to accomplish the activity (such as copying sample.txt).

ParentActivityId: An <I32 N="ParentActivityId"> XML element with an integer identifying the parent activity for which this record is a subordinate; a negative value indicates that the activity for which progress is being reported has no parent

PercentComplete: An <I32 N="PercentComplete"> XML element with an integer with an estimate of the percentage of total work that is |

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		completed for the activity RecordType: An <Obj N="Type" RefId="1"> XML element defining the type of the record. SecondsRemaining: An <I32 N="SecondsRemaining"> XML element with an integer estimating the time needed to complete the activity for which progress is being reported StatusDescription: An <S N="StatusDescription"> XML element with a string containing the current status of the operation; for example, 35 of 50 items copied, 95% completed, or 100 files purged. Example: <Obj RefId="0"> <MS> <S N="Activity">Activity Name</S> <I32 N="ActivityId">4</I32> <S N="StatusDescription">Good</S> <S N="CurrentOperation">Down loading</S> <I32 N="ParentActivityId">-1</I32> <I32 N="PercentComplete">20</I32> <Obj N="Type" RefId="1"> <TN RefId="0"> <T>System.Management.Automation.ProgressRecordType</T> <T>System.Enum</T> <T>System.ValueType</T> <T>System.Object</T> </TN> <ToString>Processing</ToString> <I32>0</I32> </Obj> <I32 N="SecondsRemaining">30</I32> </MS> </Obj>

Return to top of page

[MS-RA]: Remote Assistance Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/02/02	V11.0 – 2014/05/15	In Section 2.1, Transport, added information on the remdesk static virtual channel. Changed from: When the Remote Assistance connection is started, it MUST create three virtual channels: ▪ The session initialization virtual channel MUST be named "RC_CTL", and is used to initialize the Remote Assistance session. ▪ …

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Changed to: When the Remote Assistance connection is started, it MUST create three virtual channels: ▪ The session initialization virtual channel is used to initialize the Remote Assistance session. If setup as a dynamic virtual channel, it MUST be named "RC_CTL". If using a static virtual channel, it MUST be named "remdesk". ▪ … In Section 2.2.1.2, REMOTEDESKTOP_CTL_PACKETHEADER, added the following entry for remdesk to the table of ChannelName values: "remdesk" Specifies the session initialization static virtual channel if not using the dynamic channel RC_CTL.
2014/11/10	V11.0 – 2014/05/15	In Section 2.2.7.2, Client Info PDU, updated the content to reflect that if the Remote Assistance invitation file is protected by a password, then the AlternateShell field MUST be initialized with the password string, and if the invitation is not password protected, this field MUST be initialized with "*". Changed from: When used in context of the Remote Assistance protocol, the following variables in the infoPacket field of Client Info PDU, as specified in [MS-RDPBCGR] section 2.2.1.11.1 need to be replaced. The format and maximum length of the following fields is specified in [MS-RDPBCGR] section 2.2.1.11.1.1. ▪ WorkingDir(Variable): Variable length ID string from the Auth String Node (the length in bytes is given by the cbWorkingfDir field). Auth String Node is present in the Remote Assistance Connection String as specified in the [MS-RAI] section 2.2. ▪ AlternateShell(Variable): This field MUST be filled with "*" (the length in bytes is given by the cbAlternateShell field). Changed to: When used in context of the Remote Assistance protocol, the following variables in the infoPacket field of Client Info PDU, as specified in [MS-RDPBCGR] section 2.2.1.11.1, need to be replaced. The format and maximum length of the following fields is specified in [MS-RDPBCGR] section 2.2.1.11.1.1: ▪ WorkingDir (Variable): Variable length ID string from the Auth String Node (the length in bytes is given by the cbWorkingfDir field). Auth String Node is present in the Remote Assistance Connection String as specified in the [MS-RAI] section 2.2. ▪ AlternateShell (Variable): If the Remote Assistance invitation file is protected by a password, then the AlternateShell field MUST be initialized with the password string. If the invitation is not password protected, then this field MUST be initialized with "*". The length in bytes is given by the cbAlternateShell field.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/09/16	V11.0 – 2014/05/15	In Section 3.5.5, Message Processing Events and Sequencing Rules, added the following two paragraphs: If the expert obtains the Remote Assistance Connection String 2 during the connection sequence, and encryption is selected for the RDP session; that is. a nonzero encryptionMethod in TS_UD_SC_SEC1 (see [MS-RDPBCGR] section 2.2.1.4.3), the client validates the public key of the server certificate contained in the Server Security Data (TS_UD_SC_SEC1). On receiving the TS_UD_SC_SEC1 from the server, the client calculates the SHA1 hash of the public key, and compares its base64-encoded string against the value of the KH parameter of the Auth String Node <A> as specified in [MS-RAI] section 2.2.2. The validation is successful if they are an exact match. Otherwise, if the validation fails, the server certificate is considered invalid and the client disconnects the session.

Return to top of page

[MS-RAI]: Remote Assistance Initiation Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/09/16	V5.0 – 2014/05/15	In Section 2.2.2, Remote Assistance Connection String 2, added the following two paragraphs: The novice (server) generates the KH attribute of the Auth String Node <A> in the Remote Assistance Connection String 2. The expert (client) validates the value of KH during the RDP connection sequence. The KH value is a base64-encoded string of the SHA1 hash of the PublicKeyBlob field of the server certificate received in TS_UD_SC_SEC1. The length, in bytes, of the PublicKeyBlob is given by the wPublicKeyBlobLen field as specified in [MS-RDPBCGR] sections 2.2.1.4.3 and 2.2.1.4.3.1.1.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/06/08	V38.0 – 2014/05/15	In Section 2.2.4.4, RFX_AVC420_BITMAP_STREAM, clarified the byte stream format to which AVC420 implementation conforms. Changed from: The RFX_AVC420_BITMAP_STREAM structure encapsulates regions of a graphics frame compressed using MPEG-4 AVC/H.264 compression techniques [ITU-H.264-201201] in YUV420p mode as specified in [ITU-H.264-201201] Annex B. The data compressed using these techniques is transported in the bitmapData

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		field of the RD PGFX_WIRE_TO_SURFACE_PDU_1 (section 2.2.2.2) message or encapsulated in the RFX_AVC444_BITMAP_STREAM structure (section 2.2.4.5). avc420EncodedBitstream (variable): An array of bytes that represents a single frame encoded using the MPEG-4 AVC/H.264 codec in YUV420p mode as specified in [ITU-H.264-201201] Annex B. Color conversion is described in section 3.3.8.3.1. Changed to: The RFX_AVC420_BITMAP_STREAM structure encapsulates regions of a graphics frame compressed using the MPEG-4 AVC/H.264 codec in YUV420p mode (as specified in [ITU-H.264-201201]) and conforming to the byte stream format specified in [ITU-H.264-201201] Annex B. The data compressed using these techniques is transported in the bitmapData field of the RD PGFX_WIRE_TO_SURFACE_PDU_1 (section 2.2.2.2) message or encapsulated in the RFX_AVC444_BITMAP_STREAM structure (section 2.2.4.5). avc420EncodedBitstream (variable): An array of bytes that represents a single frame encoded using the MPEG-4 AVC/H.264 codec in YUV420p mode (as specified in [ITU-H.264-201201]) and conforming to the byte stream format specified in [ITU-H.264-201201] Annex B. Color conversion is described in section 3.3.8.3.1.
2015/05/25	V38.0 – 2014/05/15	In Section 3.2.5.3.2, Processing X.224 Connection Confirm PDU, updated that once the External Security Protocol handshake has successfully run to completion and all authentication requirements have been fulfilled, the client SHOULD continue with the connection sequence. Changed from: Once the External Security Protocol handshake has run to completion, the client MUST continue with the connection sequence by sending the MCS Connect Initial PDU (section 2.2.1.3) to the server over the newly established secure channel (section 3.2.5.3.3). Changed to: Once the External Security Protocol handshake has successfully run to completion and all authentication requirements have been fulfilled, the client SHOULD continue with the connection sequence by sending the MCS Connect Initial PDU (section 2.2.1.3) to the server over the newly established secure channel (section 3.2.5.3.3).
2015/05/25	V38.0 – 2014/05/15	In various sections, defined the height and width of the bitmap image encoded in the TS_BITMAP_DATA_EX structure. In Section 2.2.9.2.1, Set Surface Bits Command (TS_SURFCMD_SET_SURF_BITS), changed from: The Set Surface Bits Command is used to transport encoded bitmap data destined for a rectangular region of the current target surface from an RDP server to an RDP client. … destRight (2 bytes): A 16-bit, unsigned integer. Exclusive right bound of the destination rectangle that will contain the decoded bitmap data. destBottom (2 bytes): A 16-bit, unsigned integer. Exclusive bottom bound of the destination rectangle that will contain the decoded bitmap data. bitmapData (variable): An Extended Bitmap Data (section 2.2.9.2.1.1) structure that contains an encoded bitmap image.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Changed to: The Set Surface Bits Command is used to transport encoded bitmap data destined for a rectangular region of the primary drawing surface from an RDP server to an RDP client. … destRight (2 bytes): A 16-bit, unsigned integer. Exclusive right bound of the destination rectangle that will contain the decoded bitmap data. This field SHOULD be ignored, as the width of the encoded bitmap image is specified in the Extended Bitmap Data (section 2.2.9.2.1.1) present in the variable-length bitmapData field. destBottom (2 bytes): A 16-bit, unsigned integer. Exclusive bottom bound of the destination rectangle that will contain the decoded bitmap data. This field SHOULD be ignored, as the height of the encoded bitmap image is specified in the Extended Bitmap Data present in the variable-length bitmapData field. bitmapData (variable): An Extended Bitmap Data structure that contains an encoded bitmap image. In Section 2.2.9.2.2, Stream Surface Bits Command (TS_SURFCMD_STREAM_SURF_BITS), changed from: The Stream Surface Bits Command is used to transport encoded bitmap data destined for a rectangular region of the current target surface from an RDP server to an RDP client. Changed to: The Stream Surface Bits Command is used to transport encoded bitmap data destined for a rectangular region of the primary drawing surface from an RDP server to an RDP client.
2015/04/13	V38.0 – 2014/05/15	In Section 5.5, Automatic Reconnection, corrected that the client random array size is 32 zero bytes, not 16 zero bytes. Changed from: When Enhanced RDP Security is in effect the client random value is not generated (see section 5.3.2). In this case, for the purpose of generating the security verifier, the client random is assumed to be an array of 16 zero bytes, that is, { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }. Changed to: When Enhanced RDP Security is in effect the client random value is not generated (see section 5.3.2). In this case, for the purpose of generating the security verifier, the client random is assumed to be an array of 32 zero bytes.
2015/03/30	V38.0 – 2014/05/15	In various sections, added information on missing multitransport error flag 0x200. In Section 1.3.10, Connection Health Monitoring, changed from: The Heartbeat PDU (section 2.2.16.1) allows a client to monitor the state of the connection to the server in real time. If the client and server both support connection health monitoring, then the server will send Heartbeat PDUs to the client at a regular cadence. If a predetermined number of heartbeats are not received by the client, then the server may be down or the network link may be in a disconnected state. If this is the case, the client can respond by displaying a warning or initiating a reconnection attempt.

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
Changed to:
The Heartbeat PDU (section 2.2.16.1) allows a client to monitor the state of the connection to the server in real time. If the client and server both support connection health monitoring, then the server will send Heartbeat PDUs to the client at a regular cadence when no other data is sent. If no data has been received over a predetermined number of heartbeat intervals by the client, then the server may be down or the network link may be in a disconnected state. If this is the case, the client can respond by displaying a warning or initiating a reconnection attempt.

In Section 2.2.1.3.8, Client Multitransport Channel Data (TS_UD_CS_MULTITRANSPORT) and Section2.2.1.4.6, Server Multitransport Channel Data (TS_UD_SC_MULTITRANSPORT), added the SOFTSYNC_TCP_TO_UDP value definition.
Changed from:
flags (4 bytes): A 32-bit, unsigned integer that specifies protocols supported by the client-side multitransport layer.
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>TRANSPORTTYPE_UDPFECR
0x01</td><td>RDP-UDP Forward Error Correction (FEC) reliable transport ([MS-RDPEUDP] sections 1 to 3).</td></tr>
<tr><td>TRANSPORTTYPE_UDPFECL
0x04</td><td>RDP-UDP FEC lossy transport ([MS-RDPEUDP] sections 1 to 3).</td></tr>
<tr><td>TRANSPORTTYPE_UDP_PREFERRED
0x100</td><td>Indicates that tunneling of static virtual channel traffic over UDP is supported.</td></tr>
</table>
Changed to:
flags (4 bytes): A 32-bit, unsigned integer that specifies protocols supported by the client-side multitransport layer.
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>TRANSPORTTYPE_UDPFECR
0x01</td><td>RDP-UDP Forward Error Correction (FEC) reliable transport ([MS-RDPEUDP] sections 1 to 3).</td></tr>
<tr><td>TRANSPORTTYPE_UDPFECL
0x04</td><td>RDP-UDP FEC lossy transport ([MS-RDPEUDP] sections 1 to 3).</td></tr>
<tr><td>TRANSPORTTYPE_UDP_PREFERRED
0x100</td><td>Indicates that tunneling of static virtual channel traffic over UDP is supported.</td></tr>
<tr><td>SOFTSYNC_TCP_TO_UDP
0x200</td><td>Indicates that switching dynamic virtual channels from the TCP to UDP transport is supported.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
The sections following were renamed:
<table>
<tr><th>Changed from</th><th>Changed to</th></tr>
<tr><td>2.2.15.2 Client Initiate Multitransport Error PDU</td><td>2.2.15.2 Client Initiate Multitransport Response PDU</td></tr>
<tr><td>3.2.5.15.2 Sending the Initiate Multitransport Error PDU</td><td>3.2.5.15.2 Sending the Initiate Multitransport Response PDU</td></tr>
<tr><td>3.3.5.15.2 Processing the Initiate Multitransport Error PDU</td><td>3.3.5.15.2 Processing the Initiate Multitransport Response PDU</td></tr>
</table>

In Section 2.2.16.1, Server Heartbeat PDU:
Changed from:
The Heartbeat PDU is sent by the server to the client and allows the client to monitor the state of the connection to the server in real time.
This PDU MUST only be sent over the MCS message channel. The ID of the message channel is specified in the Server Message Channel Data (section 2.2.1.4.5).
Changed to:
The Heartbeat PDU is sent by the server to the client and allows the client to monitor the state of the connection to the server in real time.
This PDU MUST only be sent over the MCS message channel. The ID of the message channel is specified in the Server Message Channel Data (section 2.2.1.4.5). It SHOULD only be sent when no other PDUs have been sent to the client in a given heartbeat interval.

In Section 3.2.5.15.1, Processing the Initiate Multitransport Request PDU:
Changed from:
The structure and fields of the Initiate Multitransport Request PDU are described in section 2.2.14.1. Upon successfully decoding this PDU the client MUST attempt to establish a sideband channel (as described in [MS-RDPEMT] sections 1.3 and 3) using the transport protocol requested in the requestedProtocol field (for reliable or lossy UDP). If the client is unable to initiate the creation of a sideband channel, then the Initiate Multitransport ErrorResponse PDU SHOULD be sent to the server (section 3.2.5.15.2).
Changed to:
The structure and fields of the Initiate Multitransport Request PDU are described in section 2.2.14.1. Upon successfully decoding this PDU the client MUST attempt to establish a sideband channel (as described in [MS-RDPEMT] sections 1.3 and 3) using the transport protocol requested in the requestedProtocol field (for reliable or lossy UDP). If the client is unable to initiate the creation of a sideband channel, then the Initiate Multitransport ErrorResponse PDU SHOULD be sent to the server (section 3.2.5.15.2).
If Soft-Sync (switching dynamic virtual channels from the TCP to UDP) is supported by the client and server, as indicated by the SOFTSYNC_TCP_TO_UDP (0x200) flag in the Client Multitransport Channel Data (section 2.2.1.3.8) and Server Multitransport Channel Data PDU (section 2.2.1.4.6), the Initiate
</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Multitransport Response PDU MUST be sent to the server regardless of whether the sideband channel creation succeeded or failed.
2015/02/02	V38.0 – 2014/05/15	In Section 5.5, Automatic Reconnection, deleted the following incorrect statement regarding the impact of one-way HMAC transformation: The one-way HMAC transformation prevents an unauthenticated server from obtaining the original auto-reconnect random and replaying it for the purpose of connecting to the user's existing session.
2015/01/19	V38.0 – 2014/05/15	In Section 2.2.7.1.6, Input Capability Set (TS_INPUT_CAPABILITYSET), added that the keyboardLayout, keyboardType, keyboardSubType, and keyboardFunctionKey fields SHOULD be set to zero, that the imeFileName field SHOULD be filled with zeros, and added two related product behavior notes. Changed from: The TS_INPUT_CAPABILITYSET structure is used to advertise support for input formats and devices. This capability is sent by both client and server. Changed to: The TS_INPUT_CAPABILITYSET structure is used to advertise support for input formats and devices. This capability is sent by both client and server. The keyboardLayout, keyboardType, keyboardSubType, and keyboardFunctionKey fields of the server-to-client TS_INPUT_CAPABILITYSET structure SHOULD<24> be set to zero, and the imeFileName fields of the server-to-client TS_INPUT_CAPABILITYSET structure SHOULD<25> be filled with zeros. <24> Section 2.2.7.1.6: Microsoft RDP 4.0, 5.0, 5.1, and 5.2 servers do not explicitly set the keyboardLayout, keyboardType, keyboardSubType, and keyboardFunctionKey fields to zero. <25> Section 2.2.7.1.6: Microsoft RDP 4.0, 5.0, 5.1, and 5.2 servers do not explicitly fill the imeFileName field with zeros.
2014/12/22	V38.0 – 2014/05/15	In Section 1.3.1.1, Connection Sequence, updated the RDP connection sequence diagram and the processing steps to reflect that the optional Monitor Layout PDU is sent by the server after the Demand Active PDU. View this Word document with Track Changes turned on to see the information added: MS-RDPBCGR_Section_1_3_1_1_Diff.
2014/12/08	V38.0 – 2014/05/15	In Section 1.3.1.1, Connection Sequence, updated the section reference number in phase number six to correctly reference Section 2.2.14, Network Characteristics Detection. Changed from: 6.Optional Connect-Time Auto-Detection: During the Optional Connect-Time Auto-Detection phase the goal is to determine characteristics of the network, such as the round-trip latency time and the bandwidth of the link between the server and client. This is accomplished by exchanging a collection of PDUs (specified in section 2.2.1.4) over a predetermined period of time with enough data to ensure that the results are statistically relevant. Changed to: 6.Optional Connect-Time Auto-Detection: During the Optional Connect-Time Auto-Detection phase, the goal is to determine characteristics of the network, such as the round-trip latency time and the bandwidth of the link between the server and client. This is accomplished by exchanging a collection of PDUs (specified in section 2.2.14) over a predetermined period of time with enough

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>data to ensure that the results are statistically relevant.</td></tr>
<tr><td>2014/10/13</td><td>V38.0 – 2014/05/15</td><td>In Section 2.2.1.1.1, RDP Negotiation Request (RDP_NEG_REQ), and Section 2.2.1.2.1, RDP Negotiation Response (RDP_NEG_RSP), added a product behavior note to each regarding the support of the credential-less logon over CredSSP functionality:
2.2.1.1.1RDP Negotiation Request (RDP_NEG_REQ)
flags (1 byte): An 8-bit, unsigned integer that contains protocol flags.
<table>
<tr><th>flag</th><th>meaning</th></tr>
<tr><td>RESTRICTED_ADMIN_MODE_REQUIRED
0x01</td><td>Indicates that the client requires credential-less logon over CredSSP (also known as "restricted admin mode"). If the server supports this mode then it is acceptable for the client to send empty credentials in the TSPasswordCreds structure defined in [MS-CSSP] section 2.2.1.2.1.<2></td></tr>
</table>
<2> Section 2.2.1.2.2: Microsoft RDP 4.0, 5.0, 5.1, 5.2, 6.0, 6.1, and 7.0 clients do not support credential-less logon over CredSSP. This functionality is not supported in Windows NT, Windows 2000, Windows XP, and Windows Vista.
2.2.1.2.1RDP Negotiation Response (RDP_NEG_RSP)
flags (1 byte): An 8-bit, unsigned integer that contains protocol flags.
<table>
<tr><th>flag</th><th>meaning</th></tr>
<tr><td>RESTRICTED_ADMIN_MODE_SUPPORTED
0x08</td><td>Indicates that the server supports credential-less logon over CredSSP (also known as "restricted admin mode") and it is acceptable for the client to send empty credentials in the TSPasswordCreds structure defined in [MS-CSSP] section 2.2.1.2.1.<3></td></tr>
</table>
<3> Section 2.2.1.2.1: Microsoft RDP 4.0, 5.0, 5.1, 5.2, 6.0, 6.1, and 7.0 servers do not support credential-less logon over CredSSP. This functionality is not supported in Windows Server 2003 and Windows Server 2008.</td></tr>
<tr><td>2014/09/16</td><td>V38.0 – 2014/05/15</td><td>In Section 2.2.11.2, Client Refresh Rect PDU, updated the text below and added a new behavior note:
The Refresh Rect PDU allows the client to request that the server redraw one or more rectangles of the session screen area. The client can use it to repaint sections of the client window that were obscured by local applications. <27> Server support for this PDU is indicated in the General Capability Set (section 2.2.7.1.1).
<27> Section 2.2.11.2: Microsoft RDP 8.0 servers do not correctly process the Refresh Rect PDU in either single monitor or multiple monitor scenarios.</td></tr>
</table>

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>Microsoft RDP 8.1 servers do not correctly process the Refresh Rect PDU in multiple monitor scenarios. The workaround in both cases is to force a refresh of the entire virtual desktop by sending two Suppress Output PDUs (section 2.2.11.3): one Suppress Output PDU to suppress display updates, followed by another Suppress Output PDU to resume display updates.</td></tr>
<tr><td>2014/09/16</td><td>V38.0 – 2014/05/15</td><td>In Section 2.2.5.2, Server Status Info PDU, updated the statusCode field values as follows:
Changed from:
ValueMeaning
TS_STATUS_VM_BOOTING
0x00000503The destination virtual machine is being booted.
Changed to:
ValueMeaning
TS_STATUS_VM_STARTING
0x00000503The destination virtual machine is being started.
TS_STATUS_VM_STARTING_MONITORING
0x00000504Monitoring of the destination virtual machine is being initiated.
TS_STATUS_VM_RETRYING_MONITORING
0x00000505Monitoring of the destination virtual machine is being reinitiated.</td></tr>
<tr><td>2014/09/16</td><td>V38.0 – 2014/05/15</td><td>In Section 2.2.8.1.1.3.1.1.1, Keyboard Event (TS_KEYBOARD_EVENT), added KBDFLAGS_EXTENDED1 to the table for keyboardFlags:
<table><tr><th>Flag</th><th>Meaning</th></tr><tr><td>KBDFLAGS_EXTENDED1
0x0200</td><td>Used to send keyboard events triggered by the PAUSE key.
A PAUSE key press and release MUST be sent as the following sequence of keyboard events:
▪ CTRL (0x1D) DOWN
▪ NUMLOCK (0x45) DOWN
▪ CTRL (0x1D) UP
▪ NUMLOCK (0x45) UP
The CTRL DOWN and CTRL UP events MUST both include the KBDFLAGS_EXTENDED1 flag.</td></tr></table>The Refresh Rect PDU allows the client to request that the server redraw one or more rectangles of the session screen area. The client can use it to repaint sections of the client window that were obscured by local applications. <27> Server support for this PDU is indicated in the General Capability Set (section 2.2.7.1.1).
<27> Section 2.2.11.2: Microsoft RDP 8.0 servers do not correctly process the Refresh Rect PDU in either single monitor or multiple monitor scenarios. Microsoft RDP 8.1 servers do not correctly process the Refresh Rect PDU in multiple monitor scenarios. The workaround in both cases is to force a refresh of</td></tr>
</table>

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>the entire virtual desktop by sending two Suppress Output PDUs (section 2.2.11.3): one Suppress Output PDU to suppress display updates, followed by another Suppress Output PDU to resume display updates.
In Section 2.2.8.1.2.2.1, Fast-Path Keyboard Event (TS_FP_KEYBOARD_EVENT), added FASTPATH_INPUT_KBDFLAGS_EXTENDED1 to the table for eventFlags:
<table>
<tr><th>5-Bit Codes</th><th>Meaning</th></tr>
<tr><td>FASTPATH_INPUT_KBDFLAGS_EXTENDED1
0x04</td><td>Used to send keyboard events triggered by the PAUSE key.
A PAUSE key press and release MUST be sent as the following sequence of keyboard events:
▪ CTRL (0x1D) DOWN
▪ NUMLOCK (0x45) DOWN
▪ CTRL (0x1D) UP
▪ NUMLOCK (0x45) UP
The CTRL DOWN and CTRL UP events MUST both include the FASTPATH_INPUT_KBDFLAGS_EXTENDED1 flag.</td></tr>
</table></td></tr>
</table>

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/10/13	V14.0 – 2014/05/15	In Section 2.2.2.2, Client Audio Formats and Version PDU, added a product behavior note that clarifies the conditions when the AAC encoder format is used. The Client Audio Formats and Version PDU is a PDU that is used to send version information, capabilities, and a list of supported audio formats from the client to the server<25>. <25> Section 2.2.2.2.2: AAC is only used if the client includes an AUDIO_FORMAT entry with wFormatTag equal to Microsoft AAC, and nAvgBytesPerSec equal to 12000 in sndFormats when sending the Client Audio Formats and Version PDU.
2014/09/16	V14.0 – 2014/05/15	In Section 1.3.2.2, Data Transfer Sequences, updated the description of the data transfer sequence. Changed from: The data transfer sequence over virtual channels has a very simple protocol. If the client version or server version is less than 8, the

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		server sends two consecutive packets of audio data: a WaveInfo PDU (section 2.2.3.3) and a Wave PDU (section 2.2.3.4). Upon consuming the audio data, the client sends back a Wave Confirm PDU (section 2.2.3.8) to the server to notify the server that it has consumed the audio data. Changed to: The data transfer sequence over virtual channels has a very simple protocol. If the client version or server version is less than 8, the server sends two consecutive packets of audio data: a WaveInfo PDU (section 2.2.3.3) and a Wave PDU (section 2.2.3.4). Upon consuming the audio data, the client sends back a Wave Confirm PDU (section 2.2.3.8) to the server to notify the server that it has consumed the audio data. Consuming the audio data means it was processed, cancelled, or dropped by the client. See section 3.2.5.2.1.6 for details of how the wTimeStamp field of the Wave Confirm PDU is set. In Section 3.2.5.2.1.6, Sending a Wave Confirm PDU, changed "packet" to "complete wave PDU" in the last paragraph. Changed from: The wTimeStamp field MUST be set to the same field of the originating WaveInfo PDU, Wave Encrypt PDU, or UDP Wave Last PDU, plus the time, in milliseconds, between receiving the packet from the network and sending this PDU. This enables the server to calculate the amount of time it takes for the client to receive the audio data PDU and send the confirmation. Changed to: The wTimeStamp field MUST be set to the same field of the originating WaveInfo PDU, Wave Encrypt PDU, or UDP Wave Last PDU, plus the time, in milliseconds, between receiving the complete wave PDU from the network and sending this PDU. This enables the server to calculate the amount of time it takes for the client to receive the audio data PDU and send the confirmation.
2014/09/16	V14.0 – 2014/05/15	In Section 3.2.5.2.1.6, Sending a Wave Confirm PDU, updated information concerning when the client MUST send a Wave Confirm PDU. Changed from: The client MUST send a Wave Confirm PDU in response to any audio sample sent by the server. The client MUST send the PDU over the same channel used to receive the audio sample. That is, if the client received a WaveInfo PDU and Wave PDU, then the client MUST send the Wave Confirm PDU over virtual channels. If the client received a Wave Encrypt PDU, or several UDP Wave PDUs and a UDP Wave Last PDU, then the client MUST send the Wave Confirm PDU over UDP. Changed to: Unless an unreliable UDP transport, as specified in [MS-RDPEUDP], is used, the client MUST send a Wave Confirm PDU in response to any audio sample sent by the server. The client MUST send the PDU over the same channel used to receive the audio sample. That is, if the client received a WaveInfo PDU and Wave PDU, then the client MUST send the Wave Confirm PDU over virtual channels. If the

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		client received a Wave Encrypt PDU, or several UDP Wave PDUs and a UDP Wave Last PDU, then the client MUST send the Wave Confirm PDU over UDP.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/12/22	V9.0 – 2014/05/15	In Section 2.2.1, Clipboard PDU Header (CLIPRDR_HEADER), added a product behavior note regarding the dataLen field. Changed from: dataLen (4 bytes): An unsigned, 32-bit integer that specifies the size, in bytes, of the data which follows the Clipboard PDU Header. Changed to: dataLen (4 bytes): An unsigned, 32-bit integer that specifies the size, in bytes, of the data which follows the Clipboard PDU Header.<1> <1> Section 2.2.1: The operating systems Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 append four bytes to the end of clipboard PDUs. These four bytes are not included in the PDU size specified by the dataLen field and can be ignored.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/05/25	V21.0 – 2014/05/15	In the following sections, updated the Padding field description to change the text from "This field… MUST be ignored on receipt." to "This field… MUST be ignored.": ▪ 2.2.1.4.2 -Device Close Request (DR_CLOSE_REQ) ▪ 2.2.1.4.3 -Device Read Request (DR_READ_REQ) ▪ 2.2.1.4.4 -Device Write Request (DR_WRITE_REQ) ▪ 2.2.1.4.5 -Device Control Request (DR_CONTROL_REQ) ▪ 2.2.1.5.4 -Device Write Response (DR_WRITE_RSP) ▪ 2.2.2.7- Server Core Capability Request (DR_CORE_CAPABILITY_REQ)

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ 2.2.3.3.6 -Server Drive Query Volume Information Request (DR_DRIVE_QUERY_VOLUME_INFORMATION_REQ) ▪ 2.2.3.3.7 -Server Drive Set Volume Information Request (DR_DRIVE_SET_VOLUME_INFORMATION_REQ) ▪ 2.2.3.3.8 -Server Drive Query Information Request (DR_DRIVE_QUERY_INFORMATION_REQ) ▪ 2.2.3.3.9 -Server Drive Set Information Request (DR_DRIVE_SET_INFORMATION_REQ) ▪ 2.2.3.3.10 -Server Drive Query Directory Request (DR_DRIVE_QUERY_DIRECTORY_REQ) ▪ 2.2.3.3.11- Server Drive NotifyChange Directory Request (DR_DRIVE_NOTIFY_CHANGE_DIRECTORY_REQ) ▪ 2.2.3.3.12 -Server Drive Lock Control Request (DR_DRIVE_LOCK_REQ) ▪ 2.2.3.4.1 -Client Drive Create Response (DR_DRIVE_CREATE_RSP) ▪ 2.2.3.4.6 -Client Drive Query Volume Information Response (DR_DRIVE_QUERY_VOLUME_INFORMATION_RSP) ▪ 2.2.3.4.9 -Client Drive Set Information Response (DR_DRIVE_SET_INFORMATION_RSP) ▪ 2.2.3.4.10 -Client Drive Query Directory Response (DR_DRIVE_QUERY_DIRECTORY_RSP) ▪ 2.2.3.4.11 -Client Drive NotifyChange Directory Response (DR_DRIVE_NOTIFY_CHANGE_DIRECTORY_RSP) ▪ 2.2.3.4.12 -Client Drive Lock Control Response (DR_DRIVE_LOCK_RSP) For example, in Section 2.2.1.4.2, Device Close Request (DR_CLOSE_REQ), changed from: Padding (32 bytes): An array of 32 bytes. Reserved. This field can be set to any value, and MUST be ignored on receipt. Changed to: Padding (32 bytes): An array of 32 bytes. Reserved. This field can be set to any value, and MUST be ignored. In Section 2.2.1.5.2, Device Close Response (DR_CLOSE_RSP), changed the Padding field description as indicated above and also revised the Padding field value from 5 bytes to 4 bytes. Changed from: Padding (5 bytes): An array of 5 bytes. Reserved. This field can be set to any value, and MUST be ignored on receipt. Changed to: Padding (4 bytes): An array of 4 bytes. Reserved. This field can be set

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>to any value, and MUST be ignored.

In Section 2.2.1.5.5, Device Control Response (DR_CONTROL_RSP), updated size information for the Output field.
Changed from:
OutputBuffer (variable): A variable-length array of bytes whose size is specified by the OutputBufferLength field. The minimum size is 1 byte; that is, if OutputBufferLength is 0, this field MUST have 1 byte of extra padding.
Changed to:
OutputBuffer (variable): A variable-length array of bytes whose size is specified by the OutputBufferLength field.

In Section 2.2.2.4, Client Name Request (DR_CORE_CLIENT_NAME_REQ), clarified that only the least significant bit of the UnicodeFlag field is valid.
Changed from:
UnicodeFlag (4 bytes): A 32-bit unsigned integer that indicates the format of the ComputerName field. This field MUST be set to one of the following values.
<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0x00000001</td><td>ComputerName is in Unicode characters.</td></tr><tr><td>0x00000000</td><td>ComputerName is in ASCII characters.</td></tr></table>Changed to:
UnicodeFlag (4 bytes): A 32-bit unsigned integer that indicates the format of the ComputerName field. Only the least significant bit of this field is valid (the most significant 31 bits MUST be ignored). The least significant bit MUST be set to one of the following values.
<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0x1</td><td>ComputerName is in Unicode characters.</td></tr><tr><td>0x0</td><td>ComputerName is in ASCII characters.</td></tr></table>
In Section 2.2.2.7.1, General Capability Set (GENERAL_CAPS_SET), clarified that the osVersion field SHOULD be ignored.
Changed from:
osVersion (4 bytes): A 32-bit unsigned integer. This field is unused, and MUST be set to 0.
Changed to:
osVersion (4 bytes): A 32-bit unsigned integer. This field is unused and SHOULD be ignored.</td></tr>
<tr><td>2015/05/25</td><td>V21.0 – 2014/05/15</td><td>In various sections, updated the definition of the fields FileFsVolumeInformation and FileBothDirectoryInformation to add that</td></tr>
</table>

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
the Reserved field MUST be ignored.
In Section 2.2.3.3.6, Server Drive Query Volume Information Request (DR_DRIVE_QUERY_VOLUME_INFORMATION_REQ), changed from:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>FileFsVolumeInformation
0x00000001</td><td>Used to query information for a volume on which a file system is mounted.</td></tr>
</table>
Changed to:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>FileFsVolumeInformation
0x00000001</td><td>Used to query information for a volume on which a file system is mounted. The Reserved field of the FileFsVolumeInformation structure ([MS-FSCC] section 2.5.9) MUST be ignored.</td></tr>
</table>
In Section 2.2.3.3.10, Server Drive Query Directory Request (DR_DRIVE_QUERY_DIRECTORY_REQ), changed from:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>FileBothDirectoryInformation
0x00000003</td><td>Basic information plus extended attribute size and short name about a file or directory.</td></tr>
</table>
Changed to:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>FileBothDirectoryInformation
0x00000003</td><td>Basic information plus extended attribute size and short name about a file or directory. The Reserved field of the FileBothDirectoryInformation structure ([MS-FSCC] section 2.4.8) MUST be ignored.</td></tr>
</table>
</td></tr>
<tr><td>2015/05/25</td><td>V21.0 – 2014/05/15</td><td>
In Section 2.2.1.2.1,Capability Message (CAPABILITY_SET), updated the definitions for the Header and capabilityData fields.
Changed from:
Header (8 bytes): A CAPABILITY_HEADER header. The CapabilityType field of the CAPABILITY_HEADER determines the CapabilityMessage type (section 2.2.2.7).
capabilityData (variable): Capability set data (section 2.2.1.2) which conforms to the structure of the type given by the CapabilityType field.
Changed to:
Header (8 bytes): A CAPABILITY_HEADER structure. The CapabilityType field of the CAPABILITY_HEADER specifies the format of the data in the
</td></tr>
</table>

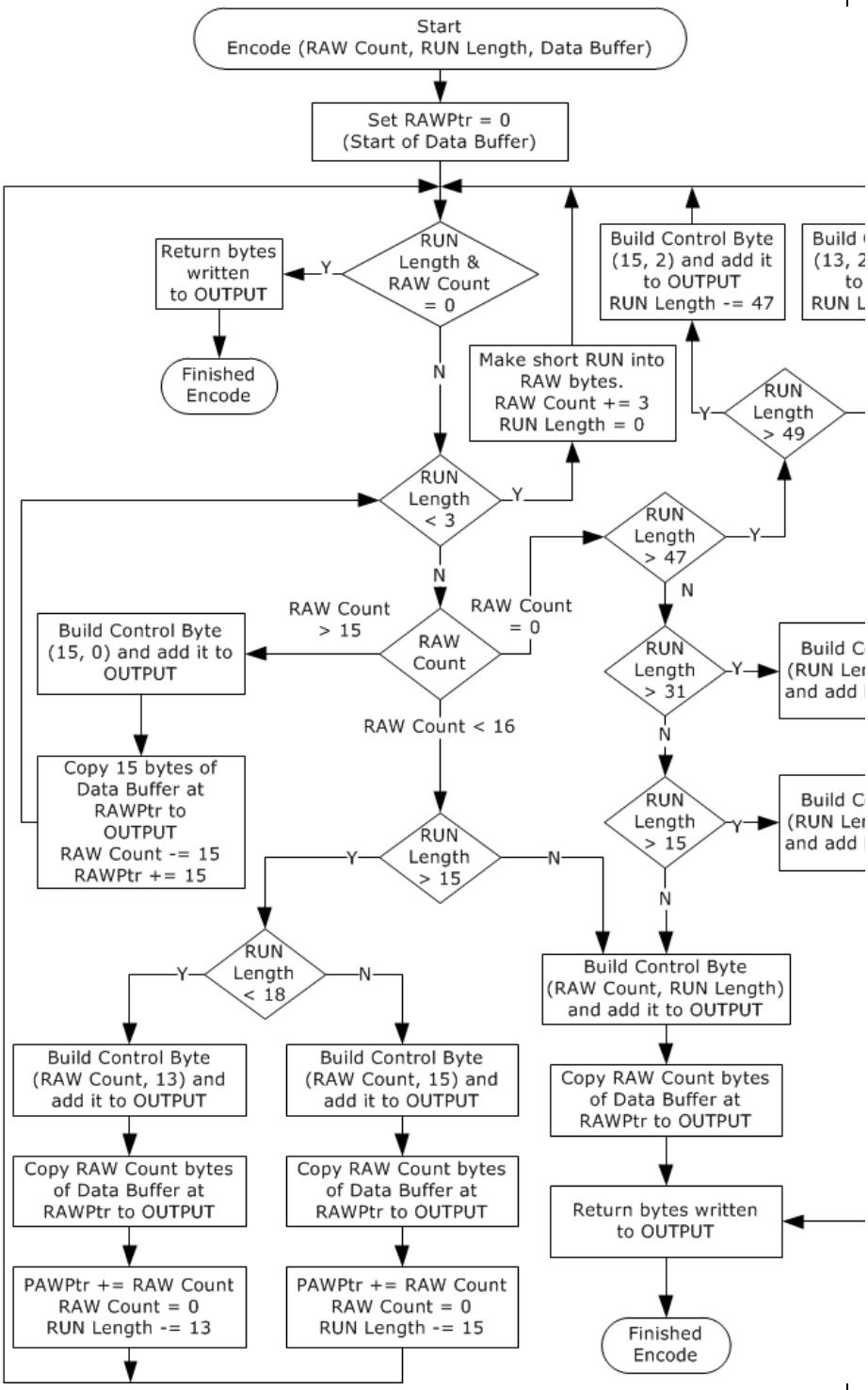
<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>capabilityData field.
capabilityData (variable): Capability set data that conforms to the structure of the type specified by the CapabilityType field of the CAPABILITY_HEADER.</td></tr>
<tr><td>2015/02/16</td><td>V21.0 – 2014/05/15</td><td>In Section 2.2.3.3.8,Server Drive Query Information Request (DR_DRIVE_QUERY_INFORMATION_REQ), added that the Reserved field of the FileBasicInformation and FileStandardInformation structures MUST be ignored.
Changed from:
FsInformationClass (4 bytes): A 32-bit unsigned integer. The possible values for this field are defined in [MS-FSCC] section 2.4. This field MUST contain one of the following values.
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>FileBasicInformation
0x00000004</td><td>This information class is used to query a file for the times of creation, last access, last write, and change, in addition to file attribute information.</td></tr>
<tr><td>FileStandardInformation
0x00000005</td><td>This information class is used to query for file information such as allocation size, end-of-file position, and number of links.</td></tr>
<tr><td>FileAttributeTagInformation
0x00000023</td><td>This information class is used to query for file attribute and reparse tag information.</td></tr>
</table>
Changed to:
FsInformationClass (4 bytes): A 32-bit unsigned integer. The possible values for this field are defined in [MS-FSCC] section 2.4. This field MUST contain one of the following values.
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>FileBasicInformation
0x00000004</td><td>This information class is used to query a file for the times of creation, last access, last write, and change, in addition to file attribute information. The Reserved field of the FileBasicInformation structure ([MS-FSCC] section 2.4.7) MUST be ignored.</td></tr>
<tr><td>FileStandardInformation
0x00000005</td><td>This information class is used to query for file information such as allocation size, end-of-file position, and number of links. The Reserved field of the FileStandardInformation structure ([MS-FSCC] section 2.4.38) MUST be ignored.</td></tr>
</table></td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		FileAttributeTagInformation 0x00000023 — This information class is used to query for file attribute and reparse tag information.

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/11/24	V27.0 – 2014/05/15	In Section 4.1.15, FastGlyph, the code example was updated to reflect the correct use of the encode flags in OpTop, as follows. FASTGLYPH_ORDER::OpLeft not present 0d 00 -> FASTGLYPH_ORDER::OpTop = 0x0D = 13 fe 7f -> FASTGLYPH_ORDER::OpRight = 0x7ffe = 32766 00 80 -> FASTGLYPH_ORDER::OpBottom = -32768 OpBottom = -32768, so OpTop (0x0D) contains encoding flags in the low 4 bits. 0x0D = 0x08 | 0x04 | 0x01 = OPRECT_LEFT_ABSENT | OPRECT_TOP_ABSENT | OPRECT_BOTTOM_ABSENT Hence, the left, top, and bottom coordinates of the opaque rectangle all match the background text rectangle and are absent. The actual value of the right coordinate is present. OpLeft = BkLeft = 139 OpTop = BkTop = 177 OpBottom = BkBottom = 190 OpRight = 32766 00 80 -> FASTGLYPH_ORDER::X = -32768 bb 00 -> FASTGLYPH_ORDER::Y = 187 13 -> VARIABLE1_FIELD::cbData = 0x13 = 19 bytes 00 01 4a 06 0a 80 80 80 b8 c4 84 84 84 84 84 00 00 68 00 -> VARIABLE1_FIELD::rgbData 00 -> TS_CACHE_GLYPH_DATA_REV2::cacheIndex = 0 01 -> TS_CACHE_GLYPH_DATA_REV2::x = 1 4a -> TS_CACHE_GLYPH_DATA_REV2::y = -10 06 -> TS_CACHE_GLYPH_DATA_REV2::cx = 6 0a -> TS_CACHE_GLYPH_DATA_REV2::cy = 10 80 80 80 b8 c4 84 84 84 84 84 00 00 -> TS_CACHE_GLYPH_DATA_REV2::aj 0x80 -> X.....

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		0x80 -> X..... 0x80 -> X..... 0xb8 -> X XXX. 0xc4 -> XX...X 0x84 -> X....X 0x84 -> X....X 0x84 -> X....X 0x84 -> X....X 0x84 -> X....X 00 00 -> padding 68 00 -> Unicode = 0x68 = 104 = h
2014/09/16	V27.0 – 2014/05/15	In Section 3.1.9.2.1, Encoding Run-Length Sequences, updated figure 15, the flow chart illustrating how a given set of RAW values and a RUN (which can be zero) is encoded using RDP 6.0 RLE. The updated figure is shown below.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		 **Encoding data using RDP 6.0 Run-Length Encoding (RLE)**

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/06/22	V7.0 – 2014/05/15	In Section 2.2.4.1.1.2.1.1.2, SHORT_VBAR_CACHE_HIT, and Section 2.2.4.1.1.2.1.1.3, SHORT_VBAR_CACHE_MISS, updated that the V-Bar Storage Cursor ADM element MUST also wrap to zero when incremented from 32767. Changed from: The V-Bar Storage Cursor (section 3.3.1.11) ADM element MUST be incremented by 1. Changed to: The V-Bar Storage Cursor (section 3.3.1.11) ADM element MUST be incremented by 1, and MUST wrap to zero when incremented from 32767.
2015/06/08	V7.0 – 2014/05/15	In Section 2.2.7.1.9, Offscreen Bitmap Cache Capability Set (TS_OFFSCREEN_CAPABILITYSET), updated the definitions for the offscreenCacheSize and offscreenCacheEntries fields. Changed from: offscreenCacheSize (2 bytes): A 16-bit, unsigned integer. The maximum size in kilobytes of the offscreen bitmap cache (largest allowed value is 10,240 KB). offscreenCacheEntries (2 bytes): A 16-bit, unsigned integer. The maximum number of cache entries (largest allowed value is 500 entries). Changed to: offscreenCacheSize (2 bytes): A 16-bit, unsigned integer. The maximum size, in kilobytes, of the client-side offscreen bitmap cache. offscreenCacheEntries (2 bytes): A 16-bit, unsigned integer. The maximum number of cache entries allowed in the client-side offscreen bitmap cache.
2015/05/25	V7.0 – 2014/05/15	In Section 2.2.4.4, RFX_AVC420_BITMAP_STREAM, clarified the compression algorithm used for avc420EncodedBitstream by adding a reference to [ITU-H.264-201201] Annex B. Changed from: The RFX_AVC420_BITMAP_STREAM structure encapsulates regions of a graphics frame compressed using MPEG-4 AVC/H.264 compression techniques [ITU-H.264-201201] in YUV420p mode. Changed to: The RFX_AVC420_BITMAP_STREAM structure encapsulates regions of a graphics frame compressed using MPEG-4 AVC/H.264 compression techniques [ITU-H.264-201201] in YUV420p mode as specified in [ITU-H.264-201201] Annex B.
2015/01/19	V7.0 – 2014/05/15	In Section 3.1.8.1.5, Simplified Run-Length (SRL), updated that an extra zero byte is always emitted after the last SRL byte array. Changed from: The Simplified Run-Length (SRL) Encoder uses the same zero run-

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		length engine as the RLGR entropy encoder ([MS-RDPRFX] section 3.1.8.1.7). However, it differs when encoding nonzero elements, because these elements are unary-encoded (there is no Golomb-Rice coding). Changed to: The Simplified Run-Length (SRL) Encoder uses the same zero run-length engine as the RLGR entropy encoder ([MS-RDPRFX] section 3.1.8.1.7). However, it differs when encoding nonzero elements, because these elements are unary-encoded (there is no Golomb-Rice coding). An extra zero byte is always emitted after the last SRL byte.
2014/09/16	V7.0 – 2014/05/15	Added information to the sections below on converting to RGB: In Section 2.2.4.4, RFX_H264_BITMAP_STREAM, revised the description for h264EncodedBitstream to read: h264EncodedBitstream (variable): An array of bytes that represents a single frame encoded using the H.264 codec. Color conversion is described in section 3.3.8.2.3.1. Added a new Section 3.3.8.2.3.1, Color conversion for the H.264 Codec, with the following text: The forward transformation from ARGB to AYUV is based on full-range BT.709 ([ITU-BT.709-5] section 4) and is described by the following two formulas: A = A The resultant Y, U, and V components MUST be clamped to the range 0...255 inclusive. The reverse transformation from AYUV to ARGB is described by the following two formulas: A = A The resultant R, G, and B components MUST be clamped to the range 0...255 inclusive.
2014/09/16	V7.0 – 2014/05/15	In Section 2.2.2.1, RDPGFX_WIRE_TO_SURFACE_PDU_1, added the following to the description for RDPGFX_CODECID_UNCOMPRESSED. Pixels in the uncompressed data are ordered from left to right and then top to bottom.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/05/11	V10.0 – 2014/05/15	In Section 2.2.2.11,GetStatusChangeA_Call, and Section 6, Appendix A: Full IDL, updated the GetStatusChangeA_Call structure definition. Changed from: 2.2.2.11 - GetStatusChangeA_Call

Errata Published YYYY/MM/DD	Protocol Document Version	Description

		The GetStatusChangeA_Call structure provides the state change in the reader as specified in section 3.1.4.23.
		
		`typedef struct _GetStatusChangeA_Call {`
		`  REDIR_SCARDCONTEXT Context;`
		`  unsigned long dwTimeOutlong;`
		`  [range(0,11)] unsigned long cReaders;`
		`  [size_is(cReaders)] ReaderStateA* rgReaderStates;`
		`} GetStatusChangeA_Call;`
		
		Context: A valid context, as specified in section 2.2.1.1.
		dwTimeOutlong: The maximum amount of time, in milliseconds, to wait for an action. If this member is set to 0xFFFFFFFF (INFINITE), the caller MUST wait until an action occurs.
		Changed to:
		2.2.2.11 - GetStatusChangeA_Call
		The GetStatusChangeA_Call structure provides the state change in the reader as specified in section 3.1.4.23.
		
		`typedef struct _GetStatusChangeA_Call {`
		`  REDIR_SCARDCONTEXT Context;`
		`  unsigned long dwTimeOut;`
		`  [range(0,11)] unsigned long cReaders;`
		`  [size_is(cReaders)] ReaderStateA* rgReaderStates;`
		`} GetStatusChangeA_Call;`
		
		Context: A valid context, as specified in section 2.2.1.1.
		dwTimeOut: The maximum amount of time, in milliseconds, to wait for an action. If this member is set to 0xFFFFFFFF (INFINITE), the caller MUST wait until an action occurs.
		
		Changed from:
		6 - Appendix A: Full IDL
		…
		
		`typedef struct _GetStatusChangeA_Call`
		`{`
		`  REDIR_SCARDCONTEXT Context;`
		`  [range(0, 65536)] unsigned long cBytes;`
		`  [size_is(cBytes)] const byte *mszCards;`
		`  [range(0, 10)] unsigned long cReaders;`
		`  [size_is(cReaders)] ReaderStateA *rgReaderStates;`
		`} GetStatusChangeA_Call;`
		
		Changed to:
		6 - Appendix A: Full IDL
		…
		
		`typedef struct _GetStatusChangeA_Call`

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>

```
   {
                            REDIR_SCARDCONTEXT      Context;
                            unsigned long           dwTimeOut;
       [range(0, 11)]       unsigned long           cReaders;
       [size_is(cReaders)] ReaderStateA           *rgReaderStates;
   } GetStatusChangeA_Call;
```

</td></tr>
<tr><td>2015/05/11</td><td>V10.0 – 2014/05/15</td><td>
In Section 3.1.4, Message Processing Events and Sequencing Rules, Section 3.1.4.47, SCARD_IOCTL_GETREADERICON (IOCTL 0x00090104), and Section 3.1.4.48, SCARD_IOCTL_GETDEVICETYPEID (IOCTL 0x00090108), updated the method return values.
Changed from:
3.1.4 - Message Processing Events and Sequencing Rules
…
<table>
<tr><th>Function number</th><th>Value for IoControl Code</th><th>IRP_MJ_DEVICE_CONTROL request</th><th>Input packet, Output packet</th></tr>
<tr><td>68</td><td>0x00090108</td><td>SCARD_IOCTL_GETDEVICETYPEID</td><td>GetDeviceTypeId_Call (section 2.2.2.32),GetReaderIcon_Return (section 2.2.3.14)</td></tr>
</table>
3.1.4.47 - SCARD_IOCTL_GETREADERICON (IOCTL 0x00090104)
Get Reader Icon retrieves the icon from the INF file for a specific smart card reader name (for more information, see GetReaderIcon_Call.szReaderName, section 2.2.2.31). On success, GetReaderIcon_Return.pbData contains the icon; for more information, see section 2.2.3.14.
Return Values: This method sets Long_Return.ReturnCode (for more information, see section 2.2.3.3) to SCARD_S_SUCCESS on success; otherwise, it sets one of the smart card-specific errors or another error code. No specialized error codes are associated with this method.

3.1.4.48 - SCARD_IOCTL_GETDEVICETYPEID (IOCTL 0x00090108)
Get Device Type ID retrieves the device type from the INF file for a specific smart card reader name (GetGetDeviceTypeId_Call.szReaderName; for more information, see section 2.2.2.32). On success, GetGetDeviceTypeId_Return.dwDeviceId contains the device type ID; for more information, see section 2.2.3.15.
Return Values: This method sets Long_Return.ReturnCode (for more information, see section 2.2.3.3) to SCARD_S_SUCCESS on success; otherwise, it sets one of the smart card-specific errors or another error code. No specialized error codes are associated with this method.

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

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
3.1.4 - Message Processing Events and Sequencing Rules
<table>
<tr><th>Function number</th><th>Value for IoControlCode</th><th>IRP_MJ_DEVICE_CONTROL request</th><th>Input packet, Output packet</th></tr>
<tr><td>68</td><td>0x00090108</td><td>SCARD_IOCTL_GETDEVICETYPEID</td><td>GetDeviceTypeId_Call (section 2.2.2.32), GetDeviceTypeId_Return (section 2.2.3.15)</td></tr>
</table>
3.1.4.47 - SCARD_IOCTL_GETREADERICON (IOCTL 0x00090104)
Get Reader Icon retrieves the icon from the INF file for a specific smart card reader name (for more information, see GetReaderIcon_Call.szReaderName, section 2.2.2.31). On success, GetReaderIcon_Return.pbData contains the icon; for more information, see section 2.2.3.14.
Return Values: This method sets GetReaderIcon_Return.ReturnCode (for more information, see section 2.2.3.14) to SCARD_S_SUCCESS on success; otherwise, it sets one of the smart card-specific errors or another error code. No specialized error codes are associated with this method.

3.1.4.48 - SCARD_IOCTL_GETDEVICETYPEID (IOCTL 0x00090108)
Get Device Type ID retrieves the device type from the INF file for a specific smart card reader name (GetDeviceTypeId_Call.szReaderName; for more information, see section 2.2.2.32). On success, GetDeviceTypeId_Return.dwDeviceId contains the device type ID; for more information, see section 2.2.3.15.
Return Values: This method sets GetDeviceTypeId_Return.ReturnCode (for more information, see section 2.2.3.15) to SCARD_S_SUCCESS on success; otherwise, it sets one of the smart card-specific errors or another error code. No specialized error codes are associated with this method.
</td></tr>
</table>

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/04/27	V7.0 – 2014/05/15	In Section 3.1.1.8, Congestion Control, added that the sender SHOULD set the RDPUDP_FLAG_CWR flag when a retransmit occurs and that no action is needed if the receiver did not set the RDPUDP_FLAG_CN flag. Changed from: On the other side, the sender will then ignore the set RDPUDP_FLAG_CN flags on subsequent acknowledgments from any

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		receiver that has an snSourceAck ADM in the acknowledgment that is less than the previously remembered sequence number. The sender reacts to losses that take place every round-trip time (RTT) only. There could be multiple losses in an RTT, and the sender MUST NOT react to those events. This behavior is similar to the NewReno variants behavior, as described in [RFC3782]. Changed to: On the other side, the sender will then ignore the set RDPUDP_FLAG_CN flags on subsequent acknowledgments from any receiver that has an snSourceAck ADM in the acknowledgment that is less than the previously remembered sequence number. Additionally, the sender SHOULD set the RDPUDP_FLAG_CWR flag whenever a retransmit occurs due to the Retransmit Timer (section 3.1.6.1) firing to indicate that a datagram loss was detected, even if the RDPUDP_FLAG_CN flag was not set by the receiver. If the receiver is not setting the RDPUDP_FLAG_CN flag, no action is needed on receipt of the RDPUDP_FLAG_CWR flag. The sender reacts to losses that take place every round-trip time (RTT) only. There could be multiple losses in an RTT, and the sender MUST NOT react to those events. This behavior is similar to the NewReno variants behavior, as described in [RFC3782].
2014/09/16	V7.0 – 2014/05/15	In Section 2.2.2.2, RDPUDP_FEC_PAYLOAD_HEADER Structure, changed A – uSourceRange to uRange. Changed from: A - uSourceRange (1 byte): An unsigned 8-bit value that, when added to snSourceStart, yields the range of packets that are contained in the FEC payload. Changed to: uRange (1 byte): An unsigned 8-bit value that, when added to snSourceStart, yields the range of packets that are contained in the FEC payload In Sections 3.1.5.1.5, ACK and FEC Packets Data, 4.2.2, FEC Packet, and 4.2.2.1, Payload of an FEC Packet, changed the uSourceRange variable name to uRange.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/01/19	V6.0 – 2014/05/15	In Section 2.1, Transport, added that the client SHOULD also send the Client Multitransport Channel Data to the server, to ensure that the transport is utilized effectively. Changed from: To ensure that the transport is utilized effectively, continuous network characteristics detection SHOULD be enabled as specified in [MS-RDPBCGR] sections 1.3.9 and 2.2.14.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Changed to: To ensure that the transport is utilized effectively, continuous network characteristics detection SHOULD be enabled (as specified in [MS-RDPBCGR] sections 1.3.9 and 2.2.14) and the client SHOULD send the Client Multitransport Channel Data ([MS-RDPBCGR] section 2.2.1.3.8) to the server.
2014/09/16	V6.0 – 2014/05/15	In Section 2.1, Transport, content was added to explain the relationship between network detect and VOR throttling. Text added is shown in **bold**. All PDUs except TSMM_VIDEO_DATA flow on the control channel, whereas TSMM_VIDEO_DATA flows on the data channel. **To ensure that the transport is utilized effectively, continuous network characteristics detection SHOULD be enabled as specified in [MS-RDPBCGR] sections 1.3.9 and 2.2.14.** … Client-to-server graphics messages are not encapsulated within any external structure when sent on the "Microsoft::Windows::RDS::Graphics" dynamic virtual channel. **To ensure that the transport is utilized effectively, continuous network characteristics detection SHOULD be enabled as specified in [MS-RDPBCGR] sections 1.3.9 and 2.2.14.**

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/02/02	V16.0 – 2014/05/15	In Section 3.1.8.2.5, Color Conversion (YCbCr to RGB), updated the YCbCr to RGB conversion matrix, and in numerous places in Section 4, Protocol Examples, revised existing examples and added new ones. View this Word document with Track Changes turned on to see the information added: MS-RDPRFX_Section_3_4_Diff
2014/09/16	V16.0 – 2014/05/15	In Section 2.2.2.3.3, TS_RFX_REGION, documented the processing rule when numRects = 0. Changed from: numRects (2 bytes): A 16-bit, unsigned integer. Specifies the number of TS_RFX_RECT (section 2.2.2.1.6) structures present in the rects field. Changed to: numRects (2 bytes): A 16-bit, unsigned integer. Specifies the number of TS_RFX_RECT (section 2.2.2.1.6) structures present in the rects field. If this value is zero, the decoder MUST generate a rectangle with coordinates (0, 0, width, height) that reflects the width and height of the channel's frame (section 2.2.2.1.3).

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/06/08	V32.0 – 2014/05/15	Added a new Section 3.7.4.3, GetServerInfo Operation to describe the request used to query the server for general configuration information. View this Word document to see the information added: MS-RMPR_Section_3_7_4_3. In Section 3.1.4.2, Server Endpoint URLs, added information about GetServerInfo. Changed from: ▪ … ▪ [baseURL]/licensing/ServiceLocator.asmx: FindServiceLocationsForUser Changed to: ▪ … ▪ [baseURL]/licensing/ServiceLocator.asmx: FindServiceLocationsForUser ▪ [baseURL]/licensing/server.asmx: GetServerInfo Added a new Section 4.6, GetServerInfoResponse Example: 4.6 GetServerInfoResponse Example The following is an example of the response data in a GetServerInfoResponse element. <Results xmlns=""> <ServerInfoRequest Type="VersionInfo" AdditionalInfo=""> <VersionInfo Version="6.0.0.0" /> </ServerInfoRequest> <ServerInfoRequest Type="ServerFeatureInfo" AdditionalInfo=""> <ServerFeatureInfo> <Feature Name="GroupExpansionWebService" Value="true" /> <Feature Name="ActiveDirectoryServicesRemoting" Value="false" /> <Feature Name="FederatedServicesEnabled" Value="0" /> </ServerFeatureInfo> </ServerInfoRequest> <ServerInfoRequest Type="ServerLicensorCertificate" AdditionalInfo=""> <ServerLicensorCertificateChain> <XrML xmlns="" version="1.2"> ... </XrML> <XrML xmlns="" version="1.2"> ... </XrML> <XrML xmlns="" version="1.2">

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		... </XrML> <XrML xmlns="" version="1.2"> ... </XrML> </ServerLicensorCertificateChain> </ServerInfoRequest> <ServerInfoRequest Type="ServiceLocations" AdditionalInfo=""> <ServiceLocations> <ServiceLocation Type="LicensingService" Url="" /> <ServiceLocation Type="PublishingService" Url="" /> <ServiceLocation Type="CertificationService" Url="http://localhost/_wmcs/certification/" /> <ServiceLocation Type="PrecertificationService" Url="" /> <ServiceLocation Type="ServerService" Url="" /> <ServiceLocation Type="GroupExpansionService" Url="HTTP://SMCCRAW64/_wmcs/GroupExpansion/GroupExpansion.asmx" /> </ServiceLocations> </ServerInfoRequest> </Results>

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/02/16	V14.0 – 2014/05/15	In Section 3.2.2.5.6, OUT_R1/A2 and OUT_R2/A2 RTS PDUs, removed an incorrect section reference in the first step of the RTS PDU processing rule sequence. Changed from: Create a successor OUT channel instance and send an OUT channel request to the outbound proxy as specified in section 3.2.2.4.1.2. Changed to: Create a successor OUT channel instance and send an OUT channel request to the outbound proxy.

[MS-RPRN]: Print System Remote Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/08/21	V26.0 –	In Sections 3.1.4.10.3, RpcRemoteFindFirstPrinterChangeNotification, and 3.1.4.10.4, RpcRemoteFindFirstPrinterChangeNotificationEx, the

Errata Published YYYY/MM/DD	Protocol Document Version	Description
	2014/05/15	allowed values for fdwOptions have been updated. This parameter was previously not used and was publicly documented as Reserved. These are the new flags allowed for fdwOptions: ▪ PRINTER_NOTIFY_CATEGORY_ALL. When specified, FindFirstPrinterChangeNotification will return notifications for both 2-D and 3-D printers. ▪ PRINTER_NOTIFY_CATEGORY_3D. When specified, FindFirstPrinterChangeNotification will return notifications only for 3-D printers. ▪ If no flag is specified, FindFirstPrinterChangeNotification will only return notifications for 2-D printers. This is the only behavior currently. The preceding changes are supported in Windows Server 2012 R2 with [MSKB-2975719].

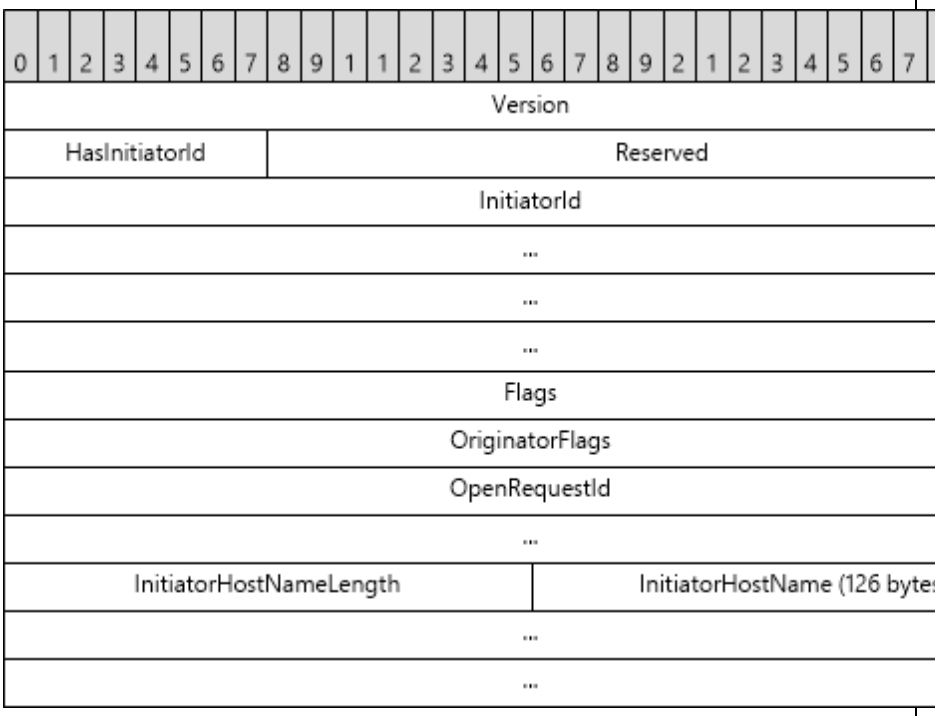
Return to top of page

[MS-RSVD]: Remote Shared Virtual Disk Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/06/08	V4.0 – 2014/05/15	In Section 3.2.5.3, Receiving a Read Request and 3.2.5.4, Receiving a Write Request, updated product behavior notes 11 and 13 to reflect the return of actual sense data in Windows Server 2012 R2 with KB3025091. In Section 3.2.5.3, Receiving a Read Request, changed from: ▪ The server MUST update SenseError in Open.SenseErrorDataList at index Open.SenseErrorSequence as follows: ▪ … ▪ SenseError.SenseData MUST be set to the data provided by the virtual SCSI disk.<11> <11>Section 3.2.5.3: Windows Server 2012 R2 sets the returned sense data to an arbitrary value. Changed to: ▪ The server MUST update SenseError in Open.SenseErrorDataList at index Open.SenseErrorSequence as follows: ▪ … ▪ SenseError.SenseData MUST be set to the data provided by the virtual SCSI disk.<11> <11> Section 3.2.5.3: Windows Server 2012 R2 without [MSKB-3025091] sets the returned sense data to an arbitrary value.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		In Section 3.2.5.4, Receiving a Write Request, changed from: ▪ The server MUST update SenseError in Open.SenseErrorDataList at index Open.SenseErrorSequence as follows: ▪ … ▪ SenseError.SenseData MUST be set to the data provided by the virtual SCSI disk.<13> <13>Section 3.2.5.4: Windows Server 2012 R2 sets the returned sense data to an arbitrary value. Changed to: ▪ The server MUST update SenseError in Open.SenseErrorDataList at index Open.SenseErrorSequence as follows: ▪ … ▪ SenseError.SenseData MUST be set to the data provided by the virtual SCSI disk.<13> <13> Section 3.2.5.4: Windows Server 2012 R2 without [MSKB-3025091] sets the returned sense data to an arbitrary value.
2015/04/27	V4.0 – 2014/05/15	In Section 3.1.4.8, Application Requests Shared Virtual Disk Information, removed the following initialization steps: The SVHDX_TUNNEL_DISK_INFO_REQUEST Request MUST be initialized as follows: ▪ The Reserved field MUST be set to zero. ▪ The BlockSize field MUST be set to zero. ▪ The LinkageId field MUST be set to zero. ▪ The IsMounted field MUST be set to zero. ▪ The Is4kAligned field MUST be set to zero. ▪ The FileSize field MUST be set to zero. ▪ The VirtualDiskId field MUST be set to zero.
2015/03/30	V4.0 – 2014/05/15	Added a new Section 2.2.4.31, SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE Structure: **2.2.4.31 SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE Structure** The SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE packet is sent by the server in response to the open shared virtual disk request.

Errata Published YYYY/MM/DD	Protocol Document Version	Description



.

Version (4 bytes): The version of the create context. It MUST be set to the highest supported version of the protocol, as specified in section 1.7.

HasInitiatorId (1 bytes): A Boolean value, where zero represents FALSE and nonzero represents TRUE.

Reserved (3 bytes): This field MUST be set to zero when sent and MUST be ignored on receipt.

InitiatorId (16 bytes): A GUID that optionally identifies the initiator of the open request.

Flags (4 bytes): Reserved. The client SHOULD set this field to 0x00000000, and the server MUST ignore it on receipt.

OriginatorFlags (4 bytes): This field is used to indicate which component has originated or issued the operation. This field MUST be set to one of the following values.

Name	Meaning
SVHDX_ORIGINATOR_PVHDPARSER 0x00000001	Shared virtual disk file to be opened as a virtual SCSI disk device
SVHDX_ORIGINATOR_VHDMP 0x00000004	Shared virtual disk file to be opened in underlying object store

OpenRequestId (8 bytes): A 64-bit value assigned by the client for an outgoing

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		request. The server MUST ignore it on receipt. **InitiatorHostNameLength (2 bytes):** The length, in bytes, of the **InitiatorHostName**. This value MUST be less than or equal to RSVD_MAXIMUM_NAME_LENGTH. **InitiatorHostName (126 bytes):** A 126-byte buffer containing a null-terminated Unicode UTF-16 string that specifies the computer name which initiated the request. In Section 3.2.5.1, Receiving an Open Request: Changed from: If the underlying object store or virtual SCSI disk returns success, the server MUST initialize the Open as follows, and return STATUS_SUCCESS to the client: … ▪ Open.IsVHDSET MUST be set to TRUE if the application-provided file name has the .vhds extension. FALSE otherwise. Changed to: If the underlying object store or virtual SCSI disk returns success, the server MUST initialize the Open as follows: … ▪ Open.IsVHDSET MUST be set to TRUE if the application-provided file name has the .vhds extension. FALSE otherwise. The server MUST construct a SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE structure using the received SVHDX_OPEN_DEVICE_CONTEXT. The server SHOULD<9> return the constructed SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE and STATUS_SUCCESS to the client. <9> Section 3.2.5.1: Windows Server 2012 R2 without [MSKB-3025091] doesn't return SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE. The preceding changes are supported in Windows Server 2012 R2 with [MSKB-3025091].
2014/12/08	V4.0 – 2014/05/15	In Section 3.2.5.1, Receiving an Open Request, clarified the RSVD parser behavior when a VHDX does not exist. Changed from: If the OriginatorFlags in the request is not set to SVHDX_ORIGINATOR_VHDMP, the server SHOULD<7> pass the request to the virtual SCSI disk. <7> If the OriginatorFlags in the request is 0x00000008, Windows-based RSVD servers will not consider it as a virtual SCSI disk device and will incorrectly pass the request to the underlying object store. Changed to: If the OriginatorFlags field in the request is not set to SVHDX_ORIGINATOR_VHDMP, the server SHOULD<7> pass the request to the virtual SCSI disk. <7> When the shared virtual disk file does not previously exist, Windows Server 2012 R2 attempts to open with a create disposition, which allows an empty file to

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		be created. This in turn will cause the RSVD request to return STATUS_FILE_CORRUPT_ERROR.
2014/11/10	V4.0 – 2014/05/15	In Sections 2 and 3, clarified ProtocolId, ProtocolVersion, and OperationCode in the SVHDX_TUNNEL_OPERATION_HEADER structure. In Section 2.2.2, Operation Codes, corrected all the control code values used in shared virtual disk operations: Changed the value of RSVD_TUNNEL_GET_INITIAL_INFO_OPERATION from 0x001 to 0x02001001. Changed the value of RSVD_TUNNEL_SCSI_OPERATION from 0x002 to 0x02001002. Changed the value of RSVD_TUNNEL_CHECK_CONNECTION_STATUS_OPERATION from 0x003 to 0x02001003. Changed the value of RSVD_TUNNEL_SRB_STATUS_OPERATION from 0x004 to 0x02001004. Changed the value of RSVD_TUNNEL_GET_DISK_INFO_OPERATION from 0x005 to 0x02001005. Changed the value of RSVD_TUNNEL_VALIDATE_DISK_OPERATION from 0x006 to 0x02001006. In Section 2.2.4.11, SVHDX_TUNNEL_OPERATION_HEADER Structure, the ProtocolId and ProtocolVersion fields were removed and the OperationCode field length revised from 12 bits to 4 bytes. In Section 3.2.5.5, Receiving a Tunnel Operation Request, the processing rules related to the bitwise AND of the OperationCode value and 0xFF000000 have been updated. Changed from: If an Open is found and if ProtocolId is equal to 0, the server MUST fail the request with STATUS_INVALID_DEVICE_REQUEST. If any of the following conditions are TRUE, server MUST construct the SVHDX_TUNNEL_OPERATION_HEADER with the ProtocolId, ProtocolVersion, OperationCode, and RequestId fields set to the value received in the request, and with the Status field set as follows. ▪ If the ProtocolId value is not equal to 0x01 or 0x02, the Status field MUST be set to STATUS_NOT_IMPLEMENTED. ▪ If the ProtocolId value is equal to 0x01, the Status field MUST be set to STATUS_INVALID_DEVICE_REQUEST. ▪ If the ProtocolVersion value 00Fis not equal to 1, the Status field MUST be set to STATUS_SVHDX_VERSION_MISMATCH. Changed to: If an Open is found and if the bitwise AND of the OperationCode value and 0xFF000000 is equal to 0, the server MUST fail the request with STATUS_INVALID_DEVICE_REQUEST. If any of the following conditions is TRUE, the server MUST construct the SVHDX_TUNNEL_OPERATION_HEADER with the OperationCode and RequestId fields set to the value received in the request, and with the Status field set as follows.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ If the bitwise AND of the OperationCode value and 0xFF000000 is not equal to 0x01000000 or 0x02000000, the Status field MUST be set to STATUS_NOT_IMPLEMENTED. ▪ If the bitwise AND of the OperationCode value and 0xFF000000 is equal to 0x01000000, the Status field MUST be set to STATUS_INVALID_DEVICE_REQUEST. ▪ If the bitwise AND of the OperationCode value and 0x00FFF000 is not equal to 0x00001000 or 0x00002000, the Status field MUST be set to STATUS_SVHDX_VERSION_MISMATCH. In Section 3.2.5.5.5, Receiving a SCSI Command Request, ProtocolId and ProtocolVersion were removed from the list of fields that must be set to the values received in the request. Changed from: The SVHDX_TUNNEL_OPERATION_HEADER MUST be initialized. ProtocolId, ProtocolVersion, OperationCode and RequestId fields MUST be set to the values received in the request. Changed to: The SVHDX_TUNNEL_OPERATION_HEADER MUST be initialized. OperationCode and RequestId fields MUST be set to the values received in the request.
2014/11/10	V4.0 – 2014/05/15	In Section 3.2.5.5, Receiving a Tunnel Operation Request, updated the content to clarify when STATUS_INVALID_DEVICE_REQUEST is returned. Changed from: When the server receives a tunnel operation request, the server MUST locate the Open in the OpenTable where Open.LocalOpen matches the Open provided by the SMB2 server, as specified in [MS-SMB2] section 3.3.5.13. If no Open is found, the server MUST fail the request and return STATUS_INVALID_PARAMETER. If an Open is found and if the bitwise AND of the OperationCode value and 0xFF000000 is equal to 0, the server MUST fail the request with STATUS_INVALID_DEVICE_REQUEST. If any of the following conditions are TRUE, the server MUST construct the SVHDX_TUNNEL_OPERATION_HEADER with the OperationCode and RequestId fields set to the value received in the request, and with the Status field set as follows. ▪ If the bitwise AND of the OperationCode value and 0xFF000000 is not equal to 0x01000000 or 0x02000000, the Status field MUST be set to STATUS_NOT_IMPLEMENTED. ▪ If the bitwise AND of the OperationCode value and 0xFF000000 is equal to 0x01000000, the Status field MUST be set to STATUS_INVALID_DEVICE_REQUEST. ▪ If the bitwise AND of the OperationCode value and 0xFF000000 is not equal to 0x00001000 or 0x00002000, the Status field MUST be set to STATUS_SVHDX_VERSION_MISMATCH. ▪ If the OperationCode value does not exist in the tunnel operation list as

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		specified in section 2.2.2, the Status field MUST be set to STATUS_INVALID_PARAMETER. The server MUST return SVHDX_TUNNEL_OPERATION_HEADER to the client. If the size of the tunnel operation request including header is less than 16, the server MUST fail the request with STATUS_BUFFER_TOO_SMALL. Processing for a specific OperationCode is as specified in subsequent sections. Changed to: When the server receives a tunnel operation request, the server MUST locate the Open in the OpenTable where Open.LocalOpen matches the Open provided by the SMB2 server, as specified in [MS-SMB2] section 3.3.5.13. If no Open is found, the server MUST fail the request and return STATUS_INVALID_PARAMETER. If the size of the tunnel operation request including the header is less than 16, the server MUST fail the request with STATUS_BUFFER_TOO_SMALL. If an Open is found and if the bitwise AND of the OperationCode value and 0xFF000000 is not equal to 0x02000000, the server MUST fail the request with STATUS_INVALID_DEVICE_REQUEST. If any of the following conditions are TRUE, the server MUST construct the SVHDX_TUNNEL_OPERATION_HEADER with the OperationCode and RequestId fields set to the value received in the request, and with the Status field set as follows. ▪ If ServerServiceVersion is equal to RSVD protocol version 1(0x00000001) and the bitwise AND of the OperationCode value and 0x00FFF000 is not equal to 0x00001000, the Status field MUST be set to STATUS_SVHDX_VERSION_MISMATCH. ▪ If ServerServiceVersion is equal to RSVD protocol version 2(0x00000002) and the bitwise AND of the OperationCode value and 0x00FFF000 is not equal to 0x00001000 or 0x00002000, the Status field MUST be set to STATUS_SVHDX_VERSION_MISMATCH. ▪ If the OperationCode value does not exist in the tunnel operation list as specified in section 2.2.2, the Status field MUST be set to STATUS_INVALID_PARAMETER. The server MUST return SVHDX_TUNNEL_OPERATION_HEADER to the client. Processing for a specific OperationCode is as specified in subsequent sections.
2014/10/27	V4.0 – 2014/05/15	In two sections, added clarifications on Read or Write beyond disk end of file (EOF) and sense data. In Section 3.2.5.3, Receiving a Read Request, added a new product behavior note <10> to the following line: SenseError.SenseData MUST be set to the data provided by the virtual SCSI disk.<10> <10> Section 3.2.5.3: Windows Server 2012 R2 sets the returned sense data to an arbitrary value. In Section 3.2.5.4, Receiving a Write Request, added a new product behavior note <12> to the following line: SenseError.SenseData MUST be set to the data provided by the virtual SCSI

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		disk.<12> <12> Section 3.2.5.4: Windows Server 2012 R2 sets the returned sense data to an arbitrary value.
2014/10/27	V4.0 – 2014/05/15	In 3 sections, updated the processing rules for Open.InitiatorId to clarify the handling of STATUS_SVHDX_ERROR_xxx. Text deleted is shown in *italics*; text added is shown in **bold**. Section 3.2.5.3, Receiving a Read Request, now reads as follows: When the server receives a read request, the server MUST locate the Open in the OpenTable, where Open.LocalOpen matches the Open provided by the SMB2 server, as specified in [MS-SMB2] section 3.2.5.12. If no Open is found, the server MUST fail the request with the STATUS_INVALID_PARAMETER code. **If the Open is found and Open.InitiatorId is zero, the server MUST process as follows:** ▪ **If Open.SenseErrorSequence is 0xFF, the server MUST reset Open.SenseErrorSequence to 0x00. Otherwise, the server MUST increment the Open.SenseErrorSequence by 0x01.** ▪ **The server MUST update SenseError in Open.SenseErrorDataList at index Open.SenseErrorSequence as follows:** ▪ **SenseError.StatusKey MUST be set to Open.SenseErrorSequence.** ▪ **Update other fields of SenseError with an implementation-specific<9> value.** ▪ **The server MUST return the error (STATUS_SVHDX_ERROR_STORED | SenseError.StatusKey) to the client.** If the Open is found and the FILE_NO_INTERMEDIATE_BUFFERING bit is not set in Open.CreateOptions, the server MUST fail the request with the STATUS_NOT_SUPPORTED code. If the Open is found and Open.IsVirtualSCSIDisk is FALSE, the server MUST issue a read to the underlying object store; otherwise the server MUST pass the request to the virtual SCSI disk in an implementation-specific manner. If the underlying object store or virtual SCSI disk indicates the read is successful, the server MUST return the read data buffer and success to the client. If the underlying object store or virtual SCSI disk indicates that the read failed, and the received error code is specified in the section 2.2.3, the server MUST return the error code. If the underlying object store is the virtual SCSI disk and the error code is not specified in section 2.2.3, the server MUST store the received sense data **and return an error** as follows: … ▪ The server MUST return the error (STATUS_SVHDX_ERROR_STORED | SenseError.StatusKey) to the client. Section 3.2.5.4, Receiving a Write Request, now reads as follows: When the server receives a write request, the server MUST locate the Open in the

Errata Published YYYY/MM/DD	Protocol Document Version	Description

		OpenTable where Open.LocalOpen matches the Open provided by the SMB2 server, as specified in [MS-SMB2] section 3.3.5.13.
		If no Open is found, the server MUST fail the request with the STATUS_INVALID_PARAMETER error code.
		If the Open is found and Open.InitiatorId is zero, the server MUST **process as follows**:
		▪ **If Open.SenseErrorSequence is 0xFF, the server MUST reset Open.SenseErrorSequence to 0x00. Otherwise, the server MUST increment the Open.SenseErrorSequence by 0x01.**
		▪ **The server MUST update SenseError in Open.SenseErrorDataList at index Open.SenseErrorSequence as follows:**
		  ▪ **SenseError.StatusKey MUST be set to Open.SenseErrorSequence.**
		  ▪ **Update other fields of SenseError with an implementation-specific<10> value.**
		▪ **The server MUST return the error (STATUS_SVHDX_ERROR_STORED | SenseError.StatusKey) to the client.**
		If the Open is found and the FILE_NO_INTERMEDIATE_BUFFERING bit is not set in Open.CreateOptions, the server MUST fail the request with the STATUS_NOT_SUPPORTED code.
		…
		If the underlying object store is the virtual SCSI disk and the error code is not specified in section 2.2.3, the server MUST store the received sense data **and return an error** as follows:
		…
		▪ The server MUST return the error (STATUS_SVHDX_ERROR_STORED | SenseError.StatusKey) to the client.
		In Section 3.2.5.5, Receiving a Tunnel Operation Request, the fifth bullet was removed as shown in the following:
		If any of the following conditions are TRUE, server MUST construct the SVHDX_TUNNEL_OPERATION_HEADER with the ProtocolId, ProtocolVersion, OperationCode, and RequestId fields set to the value received in the request, and with the Status field set as follows.
		▪ If the ProtocolId value is not equal to 0x01 or 0x02, the Status field MUST be set to STATUS_NOT_IMPLEMENTED.
		▪ If the ProtocolId value is equal to 0x01, the Status field MUST be set to STATUS_INVALID_DEVICE_REQUEST.
		▪ If the ProtocolVersion value is not equal to 1, the Status field MUST be set to STATUS_SVHDX_VERSION_MISMATCH.
		▪ If the OperationCode value does not exist in the tunnel operation list as specified in section 2.2.2, the Status field MUST be set to STATUS_INVALID_PARAMETER.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ *If the OperationCode is equal to RSVD_TUNNEL_SCSI_OPERATION and Open.InitiatorId is zero, the Status field MUST be set to STATUS_INVALID_HANDLE.* Section 3.2.5.5.5, Receiving a SCSI Command Request, now reads as follows: When the server receives a request with an OperationCode equal to RSVD_TUNNEL_SCSI_OPERATION, the request handling proceeds as follows: If MaxOutputResponse is less than 52 (size of SVHDX_TUNNEL_OPERATION_HEADER + size of SVHDX_TUNNEL_SCSI_RESPONSE), the server MUST fail the request with STATUS_INVALID_PARAMETER. If any of the following conditions is TRUE, the server MUST generate an error response as specified below: ▪ Size of the request is less than 36 ▪ … ▪ **Open.InitiatorId is zero** The SVHDX_TUNNEL_OPERATION_HEADER MUST be initialized. ProtocolId, ProtocolVersion, OperationCode and RequestId fields MUST be set to the values received in the request. **If Open.InitiatorId is zero, the Status field MUST be set to STATUS_INVALID_HANDLE. Otherwise, the** Status field MUST be set to STATUS_INVALID_PARAMETER. The SVHDX_TUNNEL_SCSI_RESPONSE MUST be set to the data received in SVHDX_TUNNEL_SCSI_REQUEST structure of the request. The server MUST send SVHDX_TUNNEL_OPERATION_HEADER and SVHDX_TUNNEL_SCSI_RESPONSE to the client and SHOULD NOT<11> add additional data to the response.
2014/10/13	V4.0 – 2014/05/15	In Section 3.2.5.1, Receiving an Open Request, updated the processing rules that apply when Open.LocalOpen is not NULL. Changed from: When the server receives a request to open a shared virtual disk file, by providing the file name and SVHDX_OPEN_DEVICE_CONTEXT, the server MUST verify the following: If the file name does not contain":SharedVirtualDisk" at the end, the server MUST fail the request with error STATUS_INVALID_PARAMETER. If HasInitiatorId is neither FALSE (0x00) nor TRUE (0x01), the server MUST fail the request with STATUS_INVALID_PARAMETER. … Changed to: If Open.LocalOpen is not NULL, the server MUST look up an Open in OpenTable where Open.LocalOpen matches the Open provided by the SMB2 server, as specified in [MS-SMB2] section 3.3.5.9.12. If an Open is found, the server MUST stop processing the request. If no Open is found, the server MUST fail the request with STATUS_INVALID_HANDLE. If Open.LocalOpen is NULL, this indicates that the server received a request to open a shared virtual disk file. The server MUST process the request as described in the steps that follow, providing the file name and SVHDX_OPEN_DEVICE_CONTEXT.

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>When the server receives a request to open a shared virtual disk file, by providing the file name and SVHDX_OPEN_DEVICE_CONTEXT, the server MUST verify the following:
If HasInitiatorId is neither FALSE (0x00) nor TRUE (0x01), the server MUST fail the request with STATUS_INVALID_PARAMETER.
…</td></tr>
<tr><td>2014/09/16</td><td>V4.0 – 2014/05/15</td><td>In Section 2.2.4.8, SVHDX_TUNNEL_SCSI_RESPONSE Structure, changed SenseDataEx description to sense data instead of command description buffer.
Changed from:
SenseDataEx (variable): A buffer of maximum size 20 bytes that contains the command descriptor buffer.
Changed to:
SenseDataEx (variable): A buffer of maximum size 20 bytes that contains the sense data.</td></tr>
<tr><td>2014/09/16</td><td>V4.0 – 2014/05/15</td><td>Renamed Section 3.2.5.5.3, Receiving a Sense Code, to 3.2.5.5.3, Receiving a Status Request for a Prior Operation.</td></tr>
<tr><td>2014/09/16</td><td>V4.0 – 2014/05/15</td><td>In Section 2.2.4.7, SVHDX_TUNNEL_SCSI_REQUEST Structure, and Section 2.2.4.8, SVHDX_TUNNEL_SCSI_RESPONSE Structure, updated the DataIn and SrbFlags values.
Changed from:
DataIn (1 byte): A Boolean, indicating the SCSI command descriptor block transfer type. The value TRUE (0x01) indicates that the operation is to store the data onto the disk. The value FALSE (0x00) indicates that the operation is to retrieve the data from the disk.
Changed to:
DataIn (1 byte): This field indicates the SCSI command descriptor block transfer type and MUST be set to one of the following values:
<table>
<tr><th>Value</th><th>Meaning</th><th>0x00</th><th>Indicates that the client is requesting data from the server</th><th>0x01</th><th>Indicates that the client is sending data to the server</th><th>0x02</th><th>Indicates that the client is neither sending nor requesting an additional data buffer</th></tr>
</table>
Changed from:
SrbFlags (4 bytes): An optional, application-provided flag to indicate the options of the SCSI request.
Changed to:
SrbFlags (4 bytes): An optional, application-provided flag to indicate the options of the SCSI request. This field MUST contain zero or more of the following values:</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		In Section 3.1.4.9, Application Requests Execution of SCSI Command, updated the values supplied by the application and updated the SVHDX_TUNNEL_SCSI_REQUEST initialization rules. Changed from: The DataIn value is set to the application-provided Boolean value. The SrbFlags field MUST be set to the application-provided special flag values. Changed to: The SrbFlags field MUST be set to the application-provided SrbFlags. If SrbFlags includes SRB_FLAGS_DATA_OUT, set DataIn to 0x00. If SrbFlags includes SRB_FLAGS_DATA_IN, DataIn SHOULD<5> be set to 0x01. If SrbFlags includes neither SRB_FLAGS_DATA_OUT nor SRB_FLAGS_DATA_IN, set DataIn to 0x02. In Section 3.2.5.5.5, Receiving a SCSI Command Request, updated the error response and initialization rules. Changed from: **3.2.5.5.5 Receiving a SCSI Command Request** When the server receives a request with an **OperationCode** equal to RSVD_TUNNEL_SCSI_OPERATION, the request handling proceeds as follows: If **MaxOutputResponse** is less than 52 (size of SVHDX_TUNNEL_OPERATION_HEADER + size of SVHDX_TUNNEL_SCSI_RESPONSE), the server MUST fail the request with STATUS_INVALID_PARAMETER. If any of the following conditions is TRUE, the server MUST generate an error response as specified below: ▪ Size of the request is less than 36 ▪ **Length** field of the request is not equal to 36 bytes ▪ **SenseInfoExLength** is greater than (size of CDBBuffer + 4) ▪ **CDBLength** is greater than the size of **CDBBuffer** The SVHDX_TUNNEL_OPERATION_HEADER MUST be initialized. **ProtocolId**, **ProtocolVersion**, **OperationCode** and **RequestId** fields MUST be set to the values received in the request. **Status** field MUST be set to STATUS_INVALID_PARAMETER. The SVHDX_TUNNEL_SCSI_RESPONSE MUST be set to the data received in SVHDX_TUNNEL_SCSI_REQUEST structure of the request. The server MUST send SVHDX_TUNNEL_OPERATION_HEADER and SVHDX_TUNNEL_SCSI_RESPONSE to the client and SHOULD NOT add additional data to the response. Otherwise, the server MUST issue a CDB command to the virtual SCSI disk in an implementation-specific manner. The server MUST construct a SVHDX_TUNNEL_SCSI_RESPONSE structure as specified in section 2.2.4.8 with the following values: The SVHDX_TUNNEL_OPERATION_HEADER MUST be initialized as follows: ▪ The **OperationCode** field MUST be set to the **OperationCode** value of the request. ▪ The **Status** field MUST be set to the value received from the virtual **SCSI** disk.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ The **RequestId** field MUST be set to the value received in the request. The SVHDX_TUNNEL_SCSI_RESPONSE MUST be initialized as follows: ▪ The **SenseInfoAutoGenerated** field MUST be set to the value received from the virtual SCSI disk. ▪ The **SrbStatus** field MUST be set to one of the values specified in section SRB Status Code reflecting the status indicated by the virtual SCSI disk. ▪ The **SCSIStatus** field MUST be set to value received from the virtual SCSI disk. ▪ The **DataIn** field MUST be set to the value received in the request. ▪ The **Length** field is set to the size, in bytes, of the SVHDX_TUNNEL_SCSI_RESPONSE structure that is constructed following the syntax specified in section 2.2.4.8. ▪ The **SenseInfoExLength** field is set to the length of the sense information received from the virtual SCSI disk, if any. ▪ The **SenseDataEx** field is set to the sense information received from the virtual SCSI disk, if any. The response MUST be sent to the client. Changed to: **3.2.5.5.5 Receiving a SCSI Command Request** When the server receives a request with an **OperationCode** equal to RSVD_TUNNEL_SCSI_OPERATION, the request handling proceeds as follows: If **MaxOutputResponse** is less than 52 (size of SVHDX_TUNNEL_OPERATION_HEADER + size of SVHDX_TUNNEL_SCSI_RESPONSE), the server MUST fail the request with STATUS_INVALID_PARAMETER. If any of the following conditions is TRUE, the server MUST generate an error response as specified below: ▪ Size of the request is less than 36 ▪ **Length** field of the request is not equal to 36 bytes ▪ **SenseInfoExLength** is greater than (size of CDBBuffer + 4) ▪ **CDBLength** is greater than the size of **CDBBuffer** ▪ **DataIn** is 0x01 and **DataTransferLength** is less than the size of the **DataBuffer** field The SVHDX_TUNNEL_OPERATION_HEADER MUST be initialized. **ProtocolId**, **ProtocolVersion**, **OperationCode** and **RequestId** fields MUST be set to the values received in the request. **Status** field MUST be set to STATUS_INVALID_PARAMETER. The SVHDX_TUNNEL_SCSI_RESPONSE MUST be set to the data received in SVHDX_TUNNEL_SCSI_REQUEST structure of the request. The server MUST send SVHDX_TUNNEL_OPERATION_HEADER and SVHDX_TUNNEL_SCSI_RESPONSE to the client and SHOULD NOT add additional

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		data to the response. Otherwise, the server MUST issue a CDB command to the virtual SCSI disk in an implementation-specific manner. If **DataIn** is 0 and the data returned by the virtual SCSI disk is greater than **DataTransferLength** in the request, the server MUST fail the request with STATUS_INVALID_PARAMETER. The server MUST construct a SVHDX_TUNNEL_SCSI_RESPONSE structure as specified in section 2.2.4.8 with the following values: The SVHDX_TUNNEL_OPERATION_HEADER MUST be initialized as follows: ▪ The **OperationCode** field MUST be set to the **OperationCode** value of the request. ▪ The **Status** field MUST be set to the value received from the virtual SCSI disk. ▪ The **RequestId** field MUST be set to the value received in the request. The SVHDX_TUNNEL_SCSI_RESPONSE MUST be initialized as follows: ▪ The **SenseInfoAutoGenerated** field MUST be set to the value received from the virtual SCSI disk. ▪ The **SrbStatus** field MUST be set to one of the values specified in section 2.2.5 reflecting the status indicated by the virtual SCSI disk. ▪ The **SCSIStatus** field MUST be set to value received from the virtual SCSI disk. ▪ The **DataIn** field MUST be set to the value received in the request. ▪ **SrbFlags** MUST be set to the value received in the request. ▪ The **CDBLength** field MUST be set to the value received in the request. ▪ The **Length** field is set to the size, in bytes, of the SVHDX_TUNNEL_SCSI_RESPONSE structure that is constructed following the syntax specified in section 2.2.4.8. ▪ The **SenseInfoExLength** field is set to the length of the sense information received from the virtual SCSI disk, if any. ▪ The **SenseDataEx** field is set to the sense information received from the virtual SCSI disk, if any. ▪ The **DataTransferLength** is set to the length of the additional data returned by the virtual SCSI disk, if any. ▪ The **DataBuffer** is set to the additional data returned by the virtual SCSI disk, if any. The response MUST be sent to the client.
2014/09/16	V4.0 – 2014/05/15	In Section 3.2.1.1, Global, added the IsSVHDXSupported entry so the section now reads as follows: The server MUST implement the following:IsSVHDXSupported: A Boolean value

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		that, if set, indicates support for operations on shared virtual disks. OpenTable: A table of opened shared virtual disk files, as specified in section 3.2.1.2, indexed by InitiatorId. ServerServiceVersion: The highest protocol version supported by the server. In Section 3.2.3, Initialization, changed the initialization rules so that the section now reads as follows. The server MUST enumerate the shares by calling NetrShareEnum as specified in [MS-SRVS] section 3.1.4.8. In the enumerated list, if any of the shares have shi*_type set to STYPE_CLUSTER_SOFS, as specified in [MS-SRVS] section 2.2.2.4, the server MUST set IsSVHDXSupported to TRUE. When the server is started: 1. If IsSVHDXSupported is TRUE, the server MUST notify the SMB server that it is ready to process client requests, as specified in [MS-SMB2] section 3.3.4.25. 2. The server MUST initialize ServerServiceVersion to the highest protocol version supported by the server. In Section 3.2.5, Message Processing Events and Sequencing Rules, updated the processing rules for when IsSVHDXSupported is FALSE so that the section now reads as follows: For this section, if IsSVHDXSupported is FALSE, the server MUST fail the request with STATUS_INVALID_DEVICE_REQUEST. In Section 3.2.5.6, Receiving a Query Shared Virtual Disk Support Request, updated the processing rules for when IsSVHDXSupported is TRUE. Changed from: When the server receives a request to open a shared virtual disk file, by providing the file name and SVHDX_OPEN_DEVICE_CONTEXT, the server MUST verify the following: If the file name does not contain":SharedVirtualDisk" at the end, the server MUST fail the request with error STATUS_INVALID_PARAMETER. If HasInitiatorId is neither FALSE (0x00) nor TRUE (0x01), the server MUST fail the request with STATUS_INVALID_PARAMETER. Changed to: If Open.LocalOpen is not NULL, the server MUST look up an Open in OpenTable where Open.LocalOpen matches the Open provided by the SMB2 server, as specified in [MS-SMB2] section 3.3.5.9.12. If an Open is found, the server MUST stop processing the request. If no Open is found, the server MUST fail the request with STATUS_INVALID_HANDLE. If Open.LocalOpen is NULL, this indicates that the server received a request to open a shared virtual disk file. The server MUST process the request as described in the steps that follow, providing the file name and SVHDX_OPEN_DEVICE_CONTEXT. When the server receives a request to open a shared virtual disk file, by providing the file name and SVHDX_OPEN_DEVICE_CONTEXT, the server MUST verify the following: If HasInitiatorId is neither FALSE (0x00) nor TRUE (0x01), the server MUST fail the request with STATUS_INVALID_PARAMETER.

Return to top of page

Windows Protocols Errata S-Z

This topic lists the Errata found in the Windows Protocols Technical Specifications, Overview Documents, and Reference documents titled [MS-S…] **through** [MS-Z…] **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.** **To find out more about the types of issues that are included in Errata, see Windows Protocols Errata.** **Errata are subject to the same terms as the Open Specifications documentation referenced.**	**RSS** **Atom**

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

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

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

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

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

[MS-TDS]: Tabular Data Stream Protocol

[MS-TSGU]: Terminal Services Gateway Server Protocol

[MS-UCODEREF]: Windows Protocols Unicode Reference

[MS-WCCE]: Windows Client Certificate Enrollment Protocol

[MS-WCFESAN]: WCF-Based Encrypted Server Administration and Notification Protocol

[MS-WDSMT]: Windows Deployment Services Multicast Transport Protocol

[MS-WFDAA]: Wi-Fi Direct (WFD) Application to Application Protocol

[MS-WPO]: Windows Protocols Overview

[MS-WMF]: Windows Metafile Format

[MS-WSMV]: Web Services Management Protocol Extensions for Windows Vista

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

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/06/08	V27.0 – 2014/05/15	In Section 1.3.1, Object-Based Perspective, added the SamrRidToSid method to the Domain Object list. In Section 3.1.5.13.5, SamrRidToSid (Opnum 65), changed from: Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
The ObjectHandle.HandleType MUST be "Domain", "User", "Group", or "Alias".
The output parameter Sid MUST be set to the objectSid attribute value of the object referenced by the Rid parameter.
Changed to:
Upon receiving this message, the server MUST process the data from the message subject to the following constraints:
The ObjectHandle.HandleType MUST be "Domain", "User", "Group", or "Alias".
The output parameter Sid MUST be set to a SID whose domain SID prefix is equal to the domain SID prefix of the objectSid attribute of the object identified by ObjectHandle, and whose RID suffix is equal to the Rid parameter.
</td></tr>
<tr><td>2015/05/25</td><td>V27.0 – 2014/05/15</td><td>In Section 3.1.5.5.1, SamrQueryInformationDomain2 (Opnum 46), and Section 3.1.5.6.1, SamrSetInformationDomain (Opnum 9), changed from:
4. If DomainInformationClass does not meet the criteria of constraint 2, the constraints associated with the DomainInformationClass input value in the following subsections MUST be satisfied; if there is no subsection for the DomainInformationClass value, an error MUST be returned to the client.
Changed to:
4. If DomainInformationClass does not meet the criteria of constraint 3, the constraints associated with the DomainInformationClass input value in the following subsections MUST be satisfied; if there is no subsection for the DomainInformationClass value, an error MUST be returned to the client.</td></tr>
<tr><td>2015/01/19</td><td>V34.0 – 2014/05/15</td><td>In Section 3.1.1.8.10, userAccountControl, added information about userAccountControl per-bit ACLs.
Changed from:
…
5. If any of the following bits are set, the client MUST have the associated control access right (defined in [MS-ADTS] section 5.1.3.2.1) on the ntSecurityDescriptor for the account domain object, per an access check. (Information about the access check mechanism is specified in [MS-ADTS] section 5.1.3.3.) If this constraint fails, the server MUST abort processing and return STATUS_ACCESS_DENIED.
<table>
<tr><th>userAccountControlBit</th><th>Required control access right</th></tr>
<tr><td>UF_PASSWD_NOTREQD</td><td>Update-Password-Not-Required-Bit</td></tr>
<tr><td>UF_DONT_EXPIRE_PASSWD</td><td>Unexpire-Password</td></tr>
</table>
</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description

UF_ENCRYPTED_TEXT_PASSWORD_ALLOWED	Enable-Per-User-Reversibly-Encrypted-Password
UF_SERVER_TRUST_ACCOUNT	DS-Install-Replica

…

8. If none of the following bits are set, the server MUST set the UF_NORMAL_ACCOUNT bit.

userAccountControlBit
UF_NORMAL_ACCOUNT
UF_INTERDOMAIN_TRUST_ACCOUNT
UF_WORKSTATION_TRUST_ACCOUNT
UF_SERVER_TRUST_ACCOUNT
UF_TEMP_DUPLICATE_ACCOUNT

.

Changed to:

…

5. If any of the following bits are set, the client MUST have the associated control access right (defined in [MS-ADTS] section 5.1.3.2.1) on the ntSecurityDescriptor for the account domain object, per an access check. (Information about the access check mechanism is specified in [MS-ADTS] section 5.1.3.3.) If this constraint fails, the server MUST abort processing and return STATUS_ACCESS_DENIED.

userAccountControlBit	Required control access right
UF_PASSWD_NOTREQD	Update-Password-Not-Required-Bit
UF_DONT_EXPIRE_PASSWD	Unexpire-Password
UF_ENCRYPTED_TEXT_PASSWORD_ALLOWED	Enable-Per-User-Reversibly-Encrypted-Password
UF_SERVER_TRUST_ACCOUNT	DS-Install-Replica
UF_PARTIAL_SECRETS_ACCOUNT	DS-Install-Replica

…

8. If the UF_INTERDOMAIN_TRUST_ACCOUNT bit is set, and the write request did not originate over the MS-LSAD protocol (see MS-ADTS

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>section 6.1.6.9.7), the server MUST abort processing and return an error status.
9. If both UF_USER_PARTIAL_SECRETS_ACCOUNT and UF_TRUSTED_FOR_DELEGATION are set the server MUST abort processing and return an error status.
10. If UF_USER_PARTIAL_SECRETS_ACCOUNT is set and UF_WORKSTATION_TRUST_ACCOUNT is not set the server MUST abort processing and return an error status.
11. If more than one of the following bits are set, the server MUST abort processing and return an error status.
<table>
<tr><th>userAccountControlBit</th></tr>
<tr><td>UF_NORMAL_ACCOUNT</td></tr>
<tr><td>UF_INTERDOMAIN_TRUST_ACCOUNT</td></tr>
<tr><td>UF_WORKSTATION_TRUST_ACCOUNT</td></tr>
<tr><td>UF_SERVER_TRUST_ACCOUNT</td></tr>
<tr><td>UF_TEMP_DUPLICATE_ACCOUNT</td></tr>
</table>
12. If the UF_TEMP_DUPLICATE_ACCOUNT is set, the server MUST abort processing and return an error status.
13. If none of the following bits are set, the server MUST set the UF_NORMAL_ACCOUNT bit.
<table>
<tr><th>userAccountControlBit</th></tr>
<tr><td>UF_NORMAL_ACCOUNT</td></tr>
<tr><td>UF_INTERDOMAIN_TRUST_ACCOUNT</td></tr>
<tr><td>UF_WORKSTATION_TRUST_ACCOUNT</td></tr>
<tr><td>UF_SERVER_TRUST_ACCOUNT</td></tr>
<tr><td>UF_TEMP_DUPLICATE_ACCOUNT</td></tr>
</table></td></tr>
<tr><td>2014/09/16</td><td>V27.0 – 2014/05/15</td><td>In Section 2.2.4.15, DisconnectType, changed IdleTimeout to IdleTimeOut.
In Section 2.2.4.37, Shell, changed MaxIdleTimeout to MaxIdleTimeOut and changed IdleTimeout to IdleTimeOut.</td></tr>
</table>

Return to top of page

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/03/02	V43.0 – 2014/05/15	In various places in Section 3, Protocol Details, revised the information about the SessionKey length used for message signing in SMB v1. In Section 3.2.1.3, Per SMB Session, corrected the descriptions of Client.Session.SessionKey and Server.Session.SessionKey. Changed from: Client.Session.SessionKey: The 16-byte cryptographic session key associated with this session, as obtained from the authentication subsystem after successful authentication. Server.Session.SessionKey: The 16-byte session key associated with this session, as obtained from the authentication packages after successful authentication. Changed to: Client.Session.SessionKey: The cryptographic session key associated with this session, as obtained from the authentication subsystem after successful authentication. Server.Session.SessionKey: The session key associated with this session, as obtained from the authentication packages after successful authentication. In Section 3.2.4.15, Application Requests the Session Key for a Connection, changed from: If a session is found, Client.Session.AuthenticationState is Valid, and Client.Session.SessionKeyState is Available, then the 16-byte Client.Session.SessionKey MUST be returned to the calling application. Changed to: If a session is found, Client.Session.AuthenticationState is Valid, and Client.Session.SessionKeyState is Available, then the first 16-bytes of Client.Session.SessionKey MUST be returned to the calling application. In Section 3.3.4.2, Server Application Queries a User Session Key, changed from: The server MUST locate an SMB connection that uses either the application-supplied ServerName to look in the Server.ConnectionTable[ClientName] or the application-supplied Open.Connection. If a valid Connection is found, then the server MUST scan for an SMB session in the Server.Connection.SessionTable that matches the security context of the user. If no entry is found, then the application request MUST be failed with STATUS_INVALID_PARAMETER. If a Session is found but Server.Session.SessionKeyState is Unavailable, the request MUST be failed with STATUS_ACCESS_DENIED and ServerStatistics.sts0_permerrors MUST be increased by 1. If Server.Session.SessionKeyState is Available, then the 16-byte Server.Session.SessionKey MUST be returned to the calling application. Changed to: The server MUST locate an SMB connection that uses either the application-supplied ServerName to look in the Server.ConnectionTable[ClientName] or the application-supplied Open.Connection. If a valid Connection is found, then the server MUST scan for an SMB session in the Server.Connection.SessionTable that

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>matches the security context of the user. If no entry is found, then the application request MUST be failed with STATUS_INVALID_PARAMETER. If a Session is found but Server.Session.SessionKeyState is Unavailable, the request MUST be failed with STATUS_ACCESS_DENIED and ServerStatistics.sts0_permerrors MUST be increased by 1. If Server.Session.SessionKeyState is Available, then the first16-bytes of Server.Session.SessionKey MUST be returned to the calling application.

In Section 3.3.5.3, Receiving an SMB_COM_SESSION_SETUP_ANDX Request, changed from:
If authentication is successful, the server MUST query the session key from the authentication package (as specified in [MS-NLMP] for implicit NTLM and in [RFC4178] for extended security). If the session key is equal to or longer than 16 bytes, only the least significant 16 bytes MUST be stored in Server.Session.SessionKey. Otherwise, the session key MUST be stored in Server.Session.SessionKey and MUST be padded with zeros up to 16 bytes. The server MUST set Server.Session.SessionKeyState to Unavailable.
Changed to:
If authentication is successful, the server MUST query the session key from the authentication package (as specified in [MS-NLMP] for implicit NTLM and in [RFC4178] for extended security). If the session key is equal to or longer than 16 bytes, the session key MUST be stored in Server.Session.SessionKey. Otherwise, the session key MUST be stored in Server.Session.SessionKey and MUST be padded with zeros up to 16 bytes. The server MUST set Server.Session.SessionKeyState to Unavailable.</td></tr>
<tr><td>2014/11/10</td><td>V43.0 – 2014/05/15</td><td>Updated the content to clarify the SECURITY_INFORMATION flags that should be filtered before passing them to the [MS-FSA] layer and to match the information in [MS-SMB2].
In Section 2.2.7.3, NT_TRANSACT_SET_SECURITY_DESC (0x0003) Extensions, changed from:
An SMB_COM_NT_TRANSACT command (section 2.2.4.8) with an NT_TRANSACT_SET_SECURITY_DESC allows a client to set the security descriptors for a file or device on the server. The NT_TRANSACT_SET_SECURITY_DESC subcommand is specified in [MS-CIFS] section 2.2.7.3. This extension adds an additional parameter value to the SecurityInformation field. The LABEL_SECURITY_INFORMATION constant is added to allow the server to set the integrity label in the security descriptor of the file or named pipe.
SecurityInformation (4 bytes): A ULONG. Fields of the security descriptor to be set. These values can be logically OR-ed together to set several descriptors in one request. The server MUST set only the descriptors requested by SecurityInformation.
<table>
<tr><th>Name and bitmask</th><th>Meaning</th></tr>
<tr><td>OWNER_SECURITY_INFORMATION
0x00000001</td><td>Owner of the object or resource.</td></tr>
</table>
</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description

GROUP_SECURITY_INFORMATION 0x00000002	Group associated with the object or resource.
DACL_SECURITY_INFORMATION 0x00000004	DACL associated with the object or resource.
SACL_SECURITY_INFORMATION 0x00000008	SACL associated with the object or resource.
LABEL_SECURITY_INFORMATION 0x00000010	Integrity label in the security descriptor of the file or named pipe.
ATTRIBUTE_SECURITY_INFORMATION 0x00000020	Resource attribute in the security descriptor of the file or named pipe.
SCOPE_SECURITY_INFORMATION 0x00000040	Central access policy of resource in the security descriptor of the file or named pipe.

Changed to:

An SMB_COM_NT_TRANSACT command (section 2.2.4.8) with an NT_TRANSACT_SET_SECURITY_DESC allows a client to set the security descriptors for a file or device on the server. The NT_TRANSACT_SET_SECURITY_DESC subcommand is specified in [MS-CIFS] section 2.2.7.3. This extension adds LABEL_SECURITY_INFORMATION, ATTRIBUTE_SECURITY_INFORMATION, SCOPE_SECURITY_INFORMATION, and BACKUP_SECURITY_INFORMATION parameter values to the SecurityInformation field.

SecurityInformation (4 bytes): A ULONG. Fields of the security descriptor to be set. These values can be logically OR-ed together to set several descriptors in one request. Bits and security descriptors not mentioned in the following table MUST be ignored and MUST NOT be processed.

Name and bitmask	Meaning
OWNER_SECURITY_INFORMATION 0x00000001	Owner of the object or resource.
GROUP_SECURITY_INFORMATION 0x00000002	Group associated with the object or resource.
DACL_SECURITY_INFORMATION 0x00000004	DACL associated with the object or resource.
SACL_SECURITY_INFORMATION 0x00000008	SACL associated with the object or resource.

Errata Published YYYY/MM/DD	Protocol Document Version	Description

LABEL_SECURITY_INFORMATION 0x00000010	Integrity label in the security descriptor of the file or named pipe.
ATTRIBUTE_SECURITY_INFORMATION 0x00000020	Resource attribute in the security descriptor of the file or named pipe.
SCOPE_SECURITY_INFORMATION 0x00000040	Central access policy of resource in the security descriptor of the file or named pipe.
BACKUP_SECURITY_INFORMATION 0x00010000	Security descriptor information used for backup operation.

.

In Section 2.2.7.4, NT_TRANSACT_QUERY_SECURITY_DESC (0x0006) Extensions, changed from:

An SMB_COM_NT_TRANSACT command (section 2.2.4.8) with an NT_TRANSACT_QUERY_SECURITY_DESC allows a client to retrieve the security descriptors for a file or device on the server. The NT_TRANSACT_QUERY_SECURITY_DESC subcommand is specified in [MS-CIFS] section 2.2.7.6. This extension adds an additional parameter value to the SecurityInfoFields field. The LABEL_SECURITY_INFORMATION constant is added to allow the server to query the integrity label from the security descriptor of the file or named pipe.

SecurityInfoFields (4 bytes): A ULONG. This field represents the requested fields of the security descriptor to be retrieved. These values can be logically OR-ed together to request several descriptors in one request. The descriptor response format contains storage for all of the descriptors. The client MUST ignore the values returned for descriptors corresponding to bits that were not included in this field as part of the request.

Name and bitmask	Meaning
OWNER_SECURITY_INFORMATION 0x00000001	Owner of the object or resource.
GROUP_SECURITY_INFORMATION 0x00000002	Group associated with the object or resource.
DACL_SECURITY_INFORMATION 0x00000004	DACL associated with the object or resource.
SACL_SECURITY_INFORMATION 0x00000008	SACL associated with the object or resource.
LABEL_SECURITY_INFORMATION	Integrity label in the security descriptor of the file or

Errata Published YYYY/MM/DD	Protocol Document Version	Description

0x00000010	named pipe.

.

Changed to:

An SMB_COM_NT_TRANSACT command (section 2.2.4.8) with an NT_TRANSACT_QUERY_SECURITY_DESC allows a client to retrieve the security descriptors for a file or device on the server. The NT_TRANSACT_QUERY_SECURITY_DESC subcommand is specified in [MS-CIFS] section 2.2.7.6. This extension adds LABEL_SECURITY_INFORMATION, ATTRIBUTE_SECURITY_INFORMATION, SCOPE_SECURITY_INFORMATION, and BACKUP_SECURITY_INFORMATION parameter values to the SecurityInfoFields field.

SecurityInfoFields (4 bytes): A ULONG. This field represents the requested fields of the security descriptor to be retrieved. These values can be logically OR-ed together to request several descriptors in one request. The descriptor response format contains storage for all the descriptors. The values returned for security descriptors corresponding to bits not mentioned in the following table MUST be ignored.

Name and bitmask	Meaning
OWNER_SECURITY_INFORMATION 0x00000001	Owner of the object or resource.
GROUP_SECURITY_INFORMATION 0x00000002	Group associated with the object or resource.
DACL_SECURITY_INFORMATION 0x00000004	DACL associated with the object or resource.
SACL_SECURITY_INFORMATION 0x00000008	SACL associated with the object or resource.
LABEL_SECURITY_INFORMATION 0x00000010	Integrity label in the security descriptor of the file or named pipe.
ATTRIBUTE_SECURITY_INFORMATION 0x00000020	Resource attribute in the security descriptor of the file or named pipe.
SCOPE_SECURITY_INFORMATION 0x00000040	Central access policy of resource in the security descriptor of the file or named pipe.
BACKUP_SECURITY_INFORMATION 0x00010000	Security descriptor information used for backup operation.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/06/22	V46.0 – 2014/05/15	In Section 3.3.5.15.7, Handling a Content Information Retrieval Request, clarified how Offset and Length are handled in SRV_HASH_RETRIEVE_FILE_BASED. Changed from: ▪ If the HashRetrievalType is SRV_HASH_RETRIEVE_HASH_BASED, the server MUST copy a SRV_READ_HASH Response following the syntax specified in section 2.2.32.4.2 into the Buffer field at the OutputOffset computed above. ▪ If the HashRetrievalType is SRV_HASH_RETRIEVE_FILE_BASED, the server MUST copy a SRV_READ_HASH Response following the syntax specified in section 2.2.32.4.3 into the Buffer field at the OutputOffset computed above. If the Offset field in the SRV_READ_HASH request is zero, the server MUST also copy the HASH_HEADER from the Content Information File, as specified in section 2.2.32.4.1, at the beginning of the Buffer[] field of the response. Changed to: ▪ If the HashRetrievalType is SRV_HASH_RETRIEVE_HASH_BASED, the server MUST copy a SRV_READ_HASH Response following the syntax specified in section 2.2.32.4.2 into the Buffer field at the OutputOffset computed above. The server MUST set the Offset to the Offset field in the SRV_READ_HASH request and BufferLength to the length of the returned content. ▪ If the HashRetrievalType is SRV_HASH_RETRIEVE_FILE_BASED, the server MUST copy a SRV_READ_HASH Response following the syntax specified in section 2.2.32.4.3 into the Buffer field at the OutputOffset computed above. The server SHOULD<331> set the FileDataOffset and FileDataLength fields to the offset and length of the region of the object that is covered by the returned content. If the Offset field in the SRV_READ_HASH request is zero, the server MUST also copy the HASH_HEADER from the Content Information File, as specified in section 2.2.32.4.1, at the beginning of the Buffer[] field of the response. <331> Section 3.3.5.15.7: Windows-based servers set the FileDataOffset field to the starting offset from the segment covering the Offset requested in the SRV_READ_HASH request.
2015/04/13	V46.0 – 2014/05/15	In sections 1-3, added content that describes the reconnect behavior on port 139. In Section 1.5, Prerequisites/Preconditions: Changed from: ▪ Either TCP or NetBIOS over TCP to support reliable, in-order message delivery. The SMB 3.x dialect family optionally supports the additional use of RDMA to support reliable, in-order message delivery with direct placement. Changed to: ▪ The SMB2 protocol required a transport to support reliable, in-order message delivery. Three such transports are used, depending on dialect, as specified in section 2.1. The SMB 3.x dialect family optionally supports the additional use

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		of RDMA to support reliable, in-order message delivery with direct placement. In Section 2.1, Transport, added the following text shown in **bold**: … The SMB 2 Protocol supports Direct TCP, NetBIOS over TCP [RFC1001] [RFC1002], and SMB2 Remote Direct Memory Access (RDMA) Transport [MS-SMBD] as transports. These transports are supported by the various SMB2 dialects as follows: … SMB2Message (variable): The body of the SMB2 packet. The length of an SMB2Message varies based on the SMB2 command represented by the message. ▪ **SMB2 dialects 2.0.2, 2.1, 3.0, and 3.0.2 support operation over NetBIOS over TCP [RFC1001] and [RFC1002].** ▪ **SMB2 dialects 3.0, 3.0.2, and 3.1.1 support operation over SMB2 Remote Direct Memory Access Protocol [MS-SMBD].** … In Section 3.2.1.1, Global, the ADM element ServerList has been changed from: ServerList: A list of server entries, as specified in section 3.2.1.9. Changed to: ServerList: A list of server entries, as specified in section 3.2.1.9, indexed by Server.ServerName. In Section 3.2.1.9, Per Server, added the ADM element ServerName as shown in **bold**: The client MUST implement the following: ▪ … ▪ AddressList: A list of IPv4 and IPv6 addresses hosted on the server. ▪ **ServerName: A fully qualified domain name, a NetBIOS name, or an IP address of the server machine.** In Section 3.2.4.2.1, Connecting to the Target Server: Changed from: The client MUST attempt to connect to the target server over the registered transports specified in section 2.1 and [MS-SMB] section 2.1. The ServerName and the optional TransportIdentifier provided by the caller are used to establish the connection. The client SHOULD resolve the ServerName as described in [MS-WPO] section 7.1.4, and SHOULD attempt connections to one or more of the returned addresses. The client can attempt to initiate each such SMB2 connection on all configured transports that it supports, most commonly Direct TCP and the other transports described in section 2.1. The client can choose to prioritize the addresses and/or transport order and try each one sequentially, or try to connect on them all and select one using any implementation-specific heuristic<102>.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		The client can accept the TransportIdentifier parameter from the calling application, which specifies what transport to use, and then attempt to use the transport specified. If the connection attempt is successful, a connection object MUST be created, as specified in section 3.2.1.2, with the following default parameters: … If the client implements the SMB 3.x dialect family, the client MUST look up a server entry in ServerList where Server.AddressList contains the server address to which the connection is established. If an entry is found, the client MUST set Connection.Server to the server entry found. Otherwise the client MUST initialize a server object and MUST set Connection.Server to NULL. <102> Section 3.2.4.2.1: Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 attempt to connect to the first eligible address returned from name resolution, preferring all transports supporting Direct TCP, and after 1000 milliseconds will attempt to connect to all other eligible addresses and transports in parallel. Windows Vista SP1 and Windows Server 2008 attempt to connect to the first eligible address, preferring a single Direct TCP transport, and after 500 milliseconds will attempt to connect to all other eligible addresses and all other NetBIOS over TCP transports. In each case, the first successful connection is used and all others are closed. Changed to: The client MUST attempt to connect to the target server over the registered transports specified in section 2.1 and [MS-SMB] section 2.1. The ServerName and the optional TransportIdentifier provided by the caller are used to establish the connection. The client SHOULD resolve the ServerName as described in [MS-WPO] section 7.1.4, and SHOULD attempt connections to one or more of the returned addresses. The client can attempt to initiate each such SMB2 connection on all configured transports that it supports, most commonly Direct TCP and the other transports described in section 2.1. The client can choose to prioritize the addresses and/or transport order and try each one sequentially, or try to connect on them all and select one using any implementation-specific heuristic<102>. The client can accept the TransportIdentifier parameter from the calling application, which specifies what transport to use, and then attempt to use the transport specified. If the connection attempt is successful, a connection object MUST be created, as specified in section 3.2.1.2, with the following default parameters: … If the client implements the SMB 3.x dialect family, the client MUST look up a server entry in ServerList where Server.ServerName matches the ServerName to which the connection is established. If an entry is found, the client MUST set Connection.Server to the server entry found. Otherwise the client MUST initialize a server object and MUST set Server.ServerName to ServerName and Connection.Server to NULL. <102> Section 3.2.4.2.1: Windows Vista SP1 and Windows Server 2008 clients enumerate all transports, send a Direct TCP connection request, and then, after 500 milliseconds, send connection requests to all other eligible addresses and all other NetBIOS over TCP transports. Windows 7 and Windows Server 2008 R2 clients enumerate all transports, send a Direct TCP connection request, and then, after 1,000 milliseconds, send connection requests to all other eligible addresses and all other NetBIOS over

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		TCP transports. Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 clients look up a server entry in ServerList where Server.ServerName matches the ServerName to which the connection is established. If no entry is found, the clients enumerate all transports, send a Direct TCP connection request, and then, after 1,000 milliseconds, send connection requests to all other eligible addresses over Direct TCP and NetBIOS over TCP transports. If an entry is found, the clients send a Direct TCP connection request, and then, after 1,000 milliseconds, enumerate all transports and send connection requests to all Direct TCP addresses. In each case, the first successful connection is used and all others are closed. In Section 3.2.5.2, Receiving an SMB2 NEGOTIATE Response: Changed from: If the client implements SMB 3.1.1, the DialectRevision in the SMB2 NEGOTIATE Response is 0x02FF, and the Connection is NetBIOS over TCP, the client MUST close the connection and return an implementation-specific error to the calling application. Changed to: If the client implements SMB 3.1.1, the DialectRevision in the SMB2 NEGOTIATE Response is 0x02FF, and the Connection is NetBIOS over TCP, the client MUST close the connection. The client MUST establish a new connection to the server, as specified in section 3.2.4.2.1, by providing the ServerName and TransportIdentifier indicating Direct TCP transport.
2015/04/13	V46.0 – 2014/05/15	In various places in Sections 2 and 3, added and revised content to clarify the lease behavior regarding ClientId (client guid, lease key) in case of STATUS_SHARING_VIOLATION for Windows Server 2012 R2. In the following sections: ▪ 2.2.13.2.8, SMB2_CREATE_REQUEST_LEASE ▪ 2.2.13.2.10, SMB2_CREATE_REQUEST_LEASE_V2 ▪ 2.2.14.2.10, SMB2_CREATE_RESPONSE_LEASE ▪ 2.2.14.2.11, SMB2_CREATE_RESPONSE_LEASE_V2 ▪ 2.2.23.2, Lease Break Notification ▪ 2.2.24.2, Lease Break Acknowledgment ▪ 2.2.25.2, Lease Break Response Changed from: LeaseKey (16 bytes): A unique key that identifies the owner of the lease. Changed to: LeaseKey (16 bytes): The client-generated key that identifies the owner of the lease. In Section 2.2.13.2.10, SMB2_CREATE_REQUEST_LEASE_V2, and Section

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		2.2.14.2.11, SMB2_CREATE_RESPONSE_LEASE_V2: Changed from: ParentLeaseKey (16 bytes): A unique key that identifies the owner of the lease for the parent directory. Changed to: ParentLeaseKey (16 bytes): A key that identifies the owner of the lease for the parent directory. In Section 3.3.1.4, Algorithm for Leasing in an Object Store: Changed from: If the server implements the SMB 2.1 or SMB 3.x dialect family and supports leasing, the underlying object store MUST implement an algorithm that permits multiple opens to the same object to share lease state (for valid lease states, see section 3.3.1.12). The algorithm MUST meet the following conditions: ▪ The algorithm MUST permit a create request from the server to the underlying object store to be accompanied by a global identifier that indicates the unique context for this lease, which will be referred to as the ClientId. The ClientId consists of a ClientGuid combined with a LeaseKey. ▪ The algorithm MUST allow multiple opens to an object that specifies the same ClientId. These opens MUST NOT alter the lease state on an object. ▪ The algorithm MUST permit three different caching capabilities within a lease: READ, WRITE, and HANDLE, with the following semantics: ▪ READ caching permits the SMB2 client to cache data read from the object. Before processing one of the following operations from a client with a different ClientId, the object store MUST request that the server revoke READ caching. The object store is not required to wait for acknowledgment: … ▪ WRITE caching permits the SMB2 client to cache writes and byte-range locks on an object. Before processing one of the following operations, the underlying object store MUST request that the server revoke WRITE caching, and the object store MUST wait for acknowledgment from the server before proceeding with the operation: ▪ The file is opened by a local application or via another protocol, or opened via SMB2 without providing the same ClientId, and requested access includes any flags other than FILE_READ_ATTRIBUTES, FILE_WRITE_ATTRIBUTES, and SYNCHRONIZE. ▪ HANDLE caching permits one or more SMB2 clients to delay closing handles it holds open, or to defer sending opens. Before processing one of the following operations, the underlying object store MUST request that the server revoke HANDLE caching, and the object store MUST wait for acknowledgment before proceeding with the operation: HANDLE caching on a file:

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ A file is opened with an access or share mode incompatible with opens from different ClientIds or local applications as described in [MS-FSA] section 2.1.5.1.2. ▪ The algorithm MUST allow a client to flow one or more creates with the same ClientId to the underlying object store during a lease break without blocking the create until the acknowledgment of the lease break is received. ▪ The algorithm SHOULD allow additional lease state flags on subsequent opens from the same ClientId to permit upgrading the lease state. The algorithm MUST NOT allow the client to release lease state flags on subsequent opens from the same ClientId to downgrade the lease state. ▪ If the requested lease state is not a superset of the existing lease state flags for this ClientId, then the requested lease state SHOULD be interpreted as the union of the existing lease state and the requested lease state. ▪ When the underlying object store requests that the server issue a lease break, it MUST also provide a new lease state for the server to pass to the client as part of the lease break packet, based on the local operations that caused the lease break to occur. Changed to: If the server implements the SMB 2.1 or SMB 3.x dialect family and supports leasing, the underlying object store MUST implement an algorithm that permits multiple opens to the same object to share lease state (for valid lease states, see section 3.3.1.12). The algorithm MUST meet the following conditions: ▪ The algorithm MUST permit a create request from the server to the underlying object store to be accompanied by an implementation-specific<171> identifier that indicates the unique server-local context for this lease, which will be referred to as the ClientLeaseId. <171> Section 3.3.1.4: On Windows 7 and Windows Server 2008 R2, a 128-bit ClientLeaseId is generated by an arithmetic combination of LeaseKey and ClientGuid, which is passed to the object store at open/create time. On Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2, the LeaseKey in the request is used as the ClientLeaseId. ▪ The algorithm MUST allow multiple opens to an object that shares the same ClientLeaseId. These opens MUST NOT alter the lease state on an object. ▪ The algorithm MUST permit three different caching capabilities within a lease: READ, WRITE, and HANDLE, with the following semantics: ▪ READ caching permits the SMB2 client to cache data read from the object. Before processing one of the following operations from a client that shares the same ClientLeaseId, the object store MUST request that the server revoke READ caching. The object store is not required to wait for acknowledgment: … ▪ WRITE caching permits the SMB2 client to cache writes and byte-range

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		locks on an object. Before processing one of the following operations, the underlying object store MUST request that the server revoke WRITE caching, and the object store MUST wait for acknowledgment from the server before proceeding with the operation: ▪ The file is opened by a client with a different ClientLeaseId, and requested access includes any flags other than FILE_READ_ATTRIBUTES, FILE_WRITE_ATTRIBUTES, and SYNCHRONIZE. ▪ HANDLE caching permits one or more SMB2 clients to delay closing handles it holds open, or to defer sending opens. Before processing one of the following operations, the underlying object store MUST request that the server revoke HANDLE caching, and the object store MUST wait for acknowledgment before proceeding with the operation: HANDLE caching on a file: ▪ A file is opened with an access or share mode incompatible with opens from clients with different ClientLeaseIds. ▪ The algorithm MUST allow a client to flow one or more creates with the same ClientLeaseId to the underlying object store during a lease break without blocking the create until the acknowledgment of the lease break is received. ▪ The algorithm SHOULD allow additional lease state flags on subsequent opens with the same ClientLeaseId to permit upgrading the lease state. The algorithm MUST NOT allow the client to release lease state flags on subsequent opens with the same ClientLeaseId to downgrade the lease state. ▪ If the requested lease state is not a superset of the existing lease state flags for this ClientLeaseId, then the requested lease state SHOULD be interpreted as the union of the existing lease state and the requested lease state. ▪ When the underlying object store requests that the server issue a lease break, it MUST also provide a new lease state for the server to pass to the client as part of the lease break packet, based on the operations that caused the lease break to occur. In Section 3.3.1.12, Per Lease: Changed from: If the server implements the SMB 2.1 or SMB 3.x dialect family and supports leasing, it implements the following: ▪ Lease.LeaseKey: A global identifier for this lease. ▪ Lease.Filename: The name of the file backing this lease. ▪ … If the server implements the SMB 3.x dialect family and supports leasing, it implements the following: ▪ Lease.Epoch: A sequence number incremented by the server on every lease

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		state change. ▪ Lease.Version: A number indicating the lease version. Changed to: If the server implements the SMB 2.1 or SMB 3.x dialect family and supports leasing, it implements the following: ▪ Lease.LeaseKey: The 128-bit client-generated identifier for this lease. ▪ Lease.ClientLeaseId: The implementation-defined server identifier for this lease. ▪ Lease.Filename: The name of the file backing this lease. ▪ … If the server implements the SMB 3.x dialect family and supports leasing, it implements the following: ▪ Lease.Epoch: A sequence number incremented by the server on every lease state change. ▪ Lease.ParentLeaseKey: The 128-bit client-generated identifier of the lease for the parent directory of this lease. ▪ Lease.Version: A number indicating the lease version. In Section 3.3.4.7, Object Store Indicates a Lease Break, all instances of LeaseKey were changed to ClientLeaseId. In Section 3.3.5.9.7, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT Create Context: Changed from: In the "Response Construction" phase: ▪ … ▪ ParentLeaseKey MUST be set to the ParentLeaseKey in the request. Changed to: In the "Response Construction" phase: ▪ … ▪ If Lease.ParentLeaseKey is not empty, ParentLeaseKey MUST be set to Lease.ParentLeaseKey, and the SMB2_LEASE_FLAG_PARENT_LEASE_KEY_SET bit MUST be set in the Flags field of the response. In Section 3.3.5.9.8, Handling the SMB2_CREATE_REQUEST_LEASE Create Context: Changed from:

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
If no lease is found, one MUST be allocated with the following values set:

Lease.LeaseKey is set to the LeaseKey in the SMB2_CREATE_REQUEST_LEASE create context.
Lease.Filename is set to the file being opened.
…

If the caching state requested in LeaseState of the SMB2_CREATE_REQUEST_LEASE is not a superset of Lease.LeaseState or if Lease.Breaking is TRUE, the server MUST NOT promote Lease.LeaseState. If the lease state requested is a superset of Lease.LeaseState and Lease.Breaking is FALSE, the server MUST request promotion of the lease state from the underlying object store to the new caching state.<282>
<282> Section 3.3.5.9.8: On Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2, the Lease.LeaseKey generated in section 3.3.5.9.8 is associated with the LeaseTable.ClientGuid to generate a unique OplockKey which is passed to the object store when processing continues at open/create time. On Windows 8 and Windows Server 2012, the LeaseKey in the request is passed to the object store. A new or existing lease is thereby associated with the resulting open.
To acquire or promote the lease as dictated by the SMB2_CREATE_REQUEST_LEASE Create Context, a subsequent object store call is invoked as described in [MS-FSA] section 2.1.5.17. The Open parameter passed is the Open.Local result from the above operation, and the Type parameter is LEVEL_GRANULAR to indicate a Lease request. The RequestedOplockLevel parameter is constructed to include zero or more bits as follows.
<table>
<tr><th>Object Store RequestedOplockLevel bit to be set</th><th>SMB2 Lease.LeaseState bit requested</th></tr>
<tr><td>READ_CACHING</td><td>SMB2_LEASE_READ_CACHING</td></tr>
<tr><td>WRITE_CACHING</td><td>SMB2_LEASE_WRITE_CACHING</td></tr>
<tr><td>HANDLE_CACHING</td><td>SMB2_LEASE_HANDLE_CACHING</td></tr>
</table>
The Status code returned indicates whether the requested lease was granted.
Changed to:
If no lease is found, one MUST be allocated with the following values set:

Lease.LeaseKey is set to the LeaseKey in the SMB2_CREATE_REQUEST_LEASE create context.
Lease.ClientLeaseId is set to a value as specified in section 3.3.1.4.
Lease.Filename is set to the file being opened.
…

If the caching state requested in LeaseState of the SMB2_CREATE_REQUEST_LEASE is not a superset of Lease.LeaseState or if
</td></tr>
</table>

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
Lease.Breaking is TRUE, the server MUST NOT promote Lease.LeaseState. If the lease state requested is a superset of Lease.LeaseState and Lease.Breaking is FALSE, the server MUST request promotion of the lease state from the underlying object store to the new caching state.<282>

<282> Section 3.3.5.9.8: On Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2, the Lease.ClientLeaseId is passed to the object store when processing continues at open/create time. A new or existing lease is thereby associated with the resulting open.

To acquire or promote the lease as dictated by the SMB2_CREATE_REQUEST_LEASE Create Context, a subsequent object store call is invoked as described in [MS-FSA] section 2.1.5.17. The Open parameter passed is the Open.Local result from the above operation, and the Type parameter is LEVEL_GRANULAR to indicate a Lease request. The RequestedOplockLevel parameter is constructed to include zero or more bits as follows.
<table>
<tr><th>Object Store RequestedOplockLevel bit to be set</th><th>SMB2 Lease.LeaseState bit requested</th></tr>
<tr><td>READ_CACHING</td><td>SMB2_LEASE_READ_CACHING</td></tr>
<tr><td>WRITE_CACHING</td><td>SMB2_LEASE_WRITE_CACHING</td></tr>
<tr><td>HANDLE_CACHING</td><td>SMB2_LEASE_HANDLE_CACHING</td></tr>
</table>
The Status code returned indicates whether the requested lease was granted.

In Section 3.3.5.9.11, Handling the SMB2_CREATE_REQUEST_LEASE_V2 Create Context:

Changed from:

If no lease is found, one MUST be allocated with the following values set:

▪ Lease.LeaseKey is set to the LeaseKey in the SMB2_CREATE_REQUEST_LEASE_V2 create context.

▪ Lease.Filename is set to the file being opened.

…
If the caching state requested in LeaseState of the SMB2_CREATE_REQUEST_LEASE_V2 is not a superset of Lease.LeaseState or if Lease.Breaking is TRUE, the server MUST NOT promote Lease.LeaseState. If the lease state requested is a superset of Lease.LeaseState and Lease.Breaking is FALSE, the server MUST request promotion of the lease state from the underlying object store to the new caching state.<286>

<286> Section 3.3.5.9.11: The LeaseKey and ParentLeaseKey fields in the SMB2_CREATE_REQUEST_LEASE_V2 Create Context request are passed to the object store in the form of TargetOplockKey and ParentOplockKey. A new or existing lease is thereby associated with the resulting open.

…
In the "Response Construction" phase, the server MUST construct an
</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		SMB2_CREATE_RESPONSE_LEASE_V2 response create context, following the syntax specified in section 2.2.14.2.11, and include it in the buffer described by the response CreateContextLength and CreateContextOffset. This structure MUST have the following values set: ▪ … ▪ If SMB2_LEASE_FLAG_PARENT_LEASE_KEY_SET bit is set in the Flags field of the request, ParentLeaseKey MUST be set to the ParentLeaseKey in the request and SMB2_LEASE_FLAG_PARENT_LEASE_KEY_SET bit MUST be set in the Flags field of the response. ▪ … Changed to: If no lease is found, one MUST be allocated with the following values set: ▪ Lease.LeaseKey is set to the LeaseKey in the SMB2_CREATE_REQUEST_LEASE_V2 create context. ▪ If the SMB2_LEASE_FLAG_PARENT_LEASE_KEY_SET bit is set in the Flags field of the request, Lease.ParentLeaseKey MUST be set to the ParentLeaseKey of the request. ▪ Lease.ClientLeaseId is set to a value as specified in section 3.3.1.4. ▪ Lease.Filename is set to the file being opened. … If the caching state requested in LeaseState of the SMB2_CREATE_REQUEST_LEASE_V2 is not a superset of Lease.LeaseState or if Lease.Breaking is TRUE, the server MUST NOT promote Lease.LeaseState. If the lease state requested is a superset of Lease.LeaseState and Lease.Breaking is FALSE, the server MUST request promotion of the lease state from the underlying object store to the new caching state.<286> <286> Section 3.3.5.9.11: On Windows 8, Windows Server 2012, Windows 8.1 operating system, and Windows Server 2012 R2 operating system, the Lease.ClientLeaseId and Lease.ParentLeaseKey are passed to the object store in the form of TargetOplockKey and ParentOplockKey. A new or existing lease is thereby associated with the resulting open. … In the "Response Construction" phase, the server MUST construct an SMB2_CREATE_RESPONSE_LEASE_V2 response create context, following the syntax specified in section 2.2.14.2.11, and include it in the buffer described by the response CreateContextLength and CreateContextOffset. This structure MUST have the following values set: ▪ … ▪ If Lease.ParentLeaseKey is not empty, ParentLeaseKey MUST be set to the Lease.ParentLeaseKey in the request, and the SMB2_LEASE_FLAG_PARENT_LEASE_KEY_SET bit MUST be set in the Flags field of the response.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ … In Section 3.3.5.9.12, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context: Changed from: … ▪ ParentLeaseKey MUST be set to the ParentLeaseKey in the request. Changed to: … ▪ If Lease.ParentLeaseKey is not empty, ParentLeaseKey MUST be set to Lease.ParentLeaseKey, and the SMB2_LEASE_FLAG_PARENT_LEASE_KEY_SET bit MUST be set in the Flags field of the response.
2015/04/13	V46.0 – 2014/05/15	In Section 3.3.5.9.7, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT Create Context: Changed step 12 from: Open.FileId.Volatile MUST be set to a generated value that uniquely identifies this Open in Session.OpenTable. Changed to: Open.FileId MUST be set to a generated value that uniquely identifies this Open in Session.OpenTable. In Section 3.3.5.9.12, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context: Changed from: Open.FileId.Volatile MUST be set to a generated value that uniquely identifies this Open in Session.OpenTable. Changed to: Open.FileId MUST be set to a generated value that uniquely identifies this Open in Session.OpenTable.
2015/04/13	V46.0 – 2014/05/15	In Section 3.2.5.19.2, Receiving a Lease Break Notification, documented a condition under which Windows clients do not send a Lease Break Acknowledgement. Changed from: If a lease acknowledgment is required by the server as indicated by the SMB2_NOTIFY_BREAK_LEASE_FLAG_ACK_REQUIRED bit in the Flags field of the Lease Break Notification, the client MUST send a Lease Break Acknowledgment request described as follows. Changed to: If a lease acknowledgment is required by the server as indicated by the SMB2_NOTIFY_BREAK_LEASE_FLAG_ACK_REQUIRED bit in the Flags field of the Lease Break Notification, the client SHOULD<162> send a Lease Break Acknowledgment request described as follows. <162> Section 3.2.5.19.2: Windows clients do not send a Lease Break

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>Acknowledgement when they have an outstanding SMB2 CREATE Request on the same File.</td></tr>
<tr><td>2015/04/13</td><td>V46.0 – 2014/05/15</td><td>In Section 3.2.4.1.8, Encrypting the Message, documented an additional encryption criterion.
Changed from:
If the client implements the SMB 3.x dialect family, the client MUST encrypt the message before sending, if any of the following conditions are satisfied:
▪ If Session.EncryptData is TRUE and the request being sent is not SMB2 NEGOTIATE.
If Session.EncryptData is TRUE and the request being sent is not SMB2 SESSION_SETUP with the SMB2_SESSION_FLAG_BINDING bit set in the Flags field.
▪ If Session.EncryptData is FALSE, the request being sent is not SMB2 NEGOTIATE or SMB2 SESSION_SETUP or SMB2 TREE_CONNECT, and TreeConnect.EncryptData is TRUE.
The client MUST encrypt the message as specified in section 3.1.4.3, before sending it to the server.
Changed to:
If the client does not implement the SMB 3.x dialect family, or the request being sent is SMB2 NEGOTIATE, or the request being sent is SMB2 SESSION_SETUP with the SMB2_SESSION_FLAG_BINDING bit set in the Flags field, the client MUST NOT encrypt the message.
Otherwise, the client MUST encrypt the message as specified in section 3.1.4.3 before sending, if either of the following conditions is satisfied:
▪ If Session.EncryptData is TRUE.
▪ If TreeConnect.EncryptData is TRUE.

In Section 3.3.4.1.4, Encrypting the Message, documented an additional encryption criterion.
Changed from:
If Session.EncryptData is TRUE and the response being sent is not SMB2_NEGOTIATE.
Changed to:
If Session.EncryptData is TRUE and the response being sent is not SMB2_NEGOTIATE or SMB2 SESSION_SETUP.</td></tr>
<tr><td>2015/03/30</td><td>V46.0 – 2014/05/15</td><td>In Section 2.2.14.2, SMB2_CREATE_CONTEXT Response Values, added the following value:
<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE
0x9CCBCF9E04C1E643980E158DA1F</td><td>An SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE context as specified in</td></tr></table></td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		<table><tr><td>6EC83</td><td>[MS-RSVD] section 2.2.4.31 is returned. This create context value is not valid for the SMB 2.002, SMB 2.1, and SMB 3.0 dialects.</td></tr></table> In Section 2.2.14.2.14, SVHDX_OPEN_DEVICE_CONTEXT, changed the title and updated the processing rule as follows: Changed from: **2.2.14.2.14 SVHDX_OPEN_DEVICE_CONTEXT** If the server succeeds in opening the shared virtual disk file, it sends SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE context as specified in [MS-RSVD] section 2.2.4.31. Changed to: **2.2.14.2.14 SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE** If the processing in [MS-RSVD] section 3.2.5.1 is successful, an SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE context as specified in [MS-RSVD] section 2.2.4.31 is returned. In Section 3.3.5.9.14, Handling the SVHDX_OPEN_DEVICE_CONTEXT Create Context, added the following details about the new "Response Construction" phase to the end of the section: ▪ In the "Response Construction" phase: ▪ If the RSVD server has returned an SVHDX_OPEN_DEVICE_CONTEXT_RESPONSE create context, as specified in [MS-RSVD] section 2.2.4.31, the server MUST include it in the buffer described by the response CreateContextLength and CreateContextOffset fields. The preceding changes are supported in Windows Server 2012 R2 with [MSKB-3025091]. Note: The 2015/03/30 Errata entry for [MS-RSVD] describes the related changes made to [MS-RSVD] section 2.2.4.31.
2015/03/30	V46.0 – 2014/05/15	In Section 3.3.5.18, Receiving an SMB2 QUERY_DIRECTORY Request, updated the following processing rule: Changed from: If the open is not an open to a directory, the request MUST be failed with STATUS_INVALID_PARAMETER. Changed to: If the open is not an open to a directory, the server MUST process the request as follows: ▪ If SMB2_REOPEN is set in the Flags field of the SMB2 QUERY_DIRECTORY request, the request MUST be failed with an implementation-specific error

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		code.<338> ▪ Otherwise, the request MUST be failed with STATUS_INVALID_PARAMETER. <338> Section 3.3.5.18: 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 fail the request with STATUS_NOT_SUPPORTED.
2015/03/02	V46.0 – 2014/05/15	In Section 3.3.5.6, Receiving an SMB2 LOGOFF Request, corrected the processing rule for freeing up the channel. Changed from: If Connection.Dialect belongs to the SMB 3.x dialect family, each channel in Session.ChannelList MUST be removed and freed. Changed to: If Connection.Dialect belongs to the SMB 3.x dialect family, the server MUST remove the session from each Channel.Connection.SessionTable in Session.ChannelList. All channels in Session.ChannelList MUST be removed and freed.
2015/02/16	V46.0 – 2014/05/15	In Section 3.3.5.2.1, Decrypting the Message, corrected the processing rules to add that for all errors detected during decryption, the server closes the transport connection. Changed from: If the ProtocolId in the header of the received message is 0x424d53FD, the server MUST perform the following: ▪ If the size of the message received from the client is not greater than the size of SMB2 TRANSFORM_HEADER as specified in section 2.2.41, the server MUST fail the request with STATUS_BUFFER_OVERFLOW. ▪ If OriginalMessageSize value received in the TRANSFORM_HEADER is greater than the implementation-specific limit or if it is less than the size of SMB2 Header, the server MUST fail the request with STATUS_BUFFER_OVERFLOW. ▪ If the Flags/EncryptionAlgorithm in the SMB2 TRANSFORM_HEADER is not 0x0001, the server MUST fail the request with STATUS_INVALID_PARAMETER. ▪ The server MUST look up the session in the Connection.SessionTable using the SessionId in the SMB2 TRANSFORM_HEADER of the request. If the session is not found, the server MUST fail the request with STATUS_USER_SESSION_DELETED. Changed to: If the ProtocolId in the header of the received message is 0x424d53FD, the server MUST perform the following: ▪ If the size of the message received from the client is not greater than the size of the SMB2 TRANSFORM_HEADER as specified in section 2.2.41, the server MUST disconnect the connection as specified in section 3.3.7.1. ▪ If OriginalMessageSize value received in the TRANSFORM_HEADER is greater than the implementation-specific limit or if it is less than the size of the SMB2 Header, the server MUST disconnect the connection as specified in section

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		3.3.7.1. ▪ If the Flags/EncryptionAlgorithm in the SMB2 TRANSFORM_HEADER is not 0x0001, the server MUST disconnect the connection as specified in section 3.3.7.1. ▪ The server MUST look up the session in the Connection.SessionTable using the SessionId in the SMB2 TRANSFORM_HEADER of the request. If the session is not found, the server MUST disconnect the connection as specified in section 3.3.7.1.
2015/02/02	V46.0 – 2014/05/15	In Section 3.2.7.1, Handling a Network Disconnect, updated the processing rules for when Connection.Dialect belongs to the SMB 3.x dialect family and the Session has more than one channel in Session.ChannelList. Text added is shown in **bold**. Changed to: When the underlying transport indicates a disconnect, for each Session in Connection.SessionTable, the client MUST perform the following: ▪ If Connection.Dialect belongs to the SMB 3.x dialect family, and if the Session has more than one channel in Session.ChannelList, the client MUST perform the following actions: ▪ The channel entry MUST be removed from the Session.ChannelList, where Channel.Connection matches the disconnected connection. ▪ For each outstanding create request in Connection.OutstandingRequests containing SMB2_CREATE_DURABLE_HANDLE_REQUEST_V2 context, the client MUST replay the create request on an alternate channel by setting the SMB2_FLAGS_REPLAY_OPERATION bit in the SMB2 header. ▪ Session.ChannelSequence MUST be incremented by 1. ▪ **If Session.Connection matches the disconnected connection, Session.Connection MUST be set to the first entry in Session.ChannelList.** ▪ … In Section 3.3.7.1, Handling Loss of a Connection, updated the processing rules for when connection.Dialect belongs to the SMB 3.x dialect family and the Session has more than one channel in Session.ChannelList. Text added is shown in **bold**. Changed to: When the underlying transport indicates loss of a connection or after the server initiates a transport disconnect, for each session in Connection.SessionTable, the server MUST perform the following: If Connection.Dialect belongs to the SMB 3.x dialect family and if the Session has more than one channel in Session.ChannelList, the server MUST perform the following action: ▪ All requests in Session.Channel.Connection.RequestList MUST be canceled. The server SHOULD<367> pass the CancelRequestId to the object store to

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		request cancellation of the pending operation. ▪ The channel entry MUST be removed from the Session.ChannelList where Channel.Connection matches the disconnected connection. ▪ **If Session.Connection matches the disconnected connection, Session.Connection MUST be set to the first entry in Session.ChannelList.**
2015/01/19	V46.0 – 2014/05/15	Throughout Section 3, added content describing how a Windows client uses ServerGUID for reestablishing a connection when multi-channel is enabled. Text added is shown in **bold**. In Section 3.2.1.1, Global, added an entry in the SMB 3.x dialect family section for ServerList. Changed to: The client MUST implement the following: … RequireSecureNegotiate: A Boolean that, if set, indicates that the client requires validation of an SMB2 NEGOTIATE request. **ServerList: A list of server entries, as specified in section 3.2.1.9.** In Section 3.2.1.2, Per SMB2 Transport Connection, added Connection.Server to the list of items that the client must implement for the SMB 3.x dialect family. Changed to: If the client implements the SMB 3.x dialect family, it MUST also implement the following: … Connection.ServerSecurityMode: The security mode received from the server in the SMB2 NEGOTIATE response on this connection, in a form that MUST follow the syntax as specified in section 2.2.4. **Connection.Server: A reference to the server entry to which the connection is established.** Added a new section 3.2.1.9, Per Server, containing a list of items that the client must implement: **3.2.1.9 Per Server** **The client MUST implement the following:** ▪ **ServerGUID: A globally unique identifier (GUID) that is generated by the remote server to uniquely identify the remote server.** ▪ **DialectRevision: Preferred dialect between client and server.** ▪ **Capabilities: The capabilities received from the server in the SMB2 NEGOTIATE response, in a form that MUST follow the syntax as specified in section 2.2.4.** ▪ **SecurityMode: The security mode received from the server in the SMB2 NEGOTIATE response, in a form that MUST follow the syntax as specified in section 2.2.4.**

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ **AddressList: A list of IPv4 and IPv6 addresses hosted on the server.** In Section 3.2.3, Initialization, added ServerList to the list of items that the client implements for the SMB 3.x dialect family. Changed to: … RequireSecureNegotiate: MUST be set based on the local configuration policy.<82> **ServerList: MUST be set to empty.** In Section 3.2.4.2.1, Connecting to the Target Server, updated the processing rules for when the client implements the SMB 3.x dialect family. Changed to: … If the connection attempt fails, the client returns the error code to the calling application. **If the client implements the SMB 3.x dialect family, the client MUST look up a server entry in ServerList where Server.AddressList contains the server address to which the connection is established. If an entry is found, the client MUST set Connection.Server to the server entry found. Otherwise the client MUST initialize a server object and MUST set Connection.Server to NULL.** In Section 3.2.5.2, Receiving an SMB2 NEGOTIATE Response, updated the processing rules for when the client implements the SMB 3.x dialect family and Connection.Server is not NULL. Changed to: … **If the client implements the SMB 3.x dialect family and Connection.Server is not NULL, the client MUST disconnect the connection if any of the following conditions is satisfied:** ▪ **Connection.Server.ServerGUID does not match ServerGUID in the response.** ▪ **Connection.Server.DialectRevision does not match DialectRevision in the response.** ▪ **Connection.Server.SecurityMode does not match SecurityMode in the response.** ▪ **Connection.Server.Capabilities does not match Capabilities in the response.** **If the client implements the SMB 3.x dialect family and Connection.Server is NULL, the client MUST set the following values:** ▪ **Connection.Server.ServerGUID to ServerGUID in the response.** ▪ **Connection.Server.DialectRevision to DialectRevision in the response.** ▪ **Connection.Server.SecurityMode to SecurityMode in the response.**

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ **Connection.Server.Capabilities to Capabilities in the response.** In Section 3.2.5.14.11, Handling a Network Interfaces Response, updated the processing rules for client extraction of IPv4Address and IPv6Address addresses. Changed to: **The client MUST extract IPv4Address and IPv6Address addresses from each NETWORK_INTERFACE_INFO structure and MUST insert the addresses into Connection.Server.AddressList.** The client MUST return the list of network interfaces received from the server to the calling application.
2014/12/22	V46.0 – 2014/05/15	In Section 3.3.5.15.11, Handling a Query Network Interface Request, updated the processing rules to cover when IfIndex equals zero. Changed from: The server MUST enumerate the local network interfaces in an implementation-specific manner. For each IP address in each network interface, the server MUST construct a NETWORK_INTERFACE_INFO structure as specified in section 2.2.32.5, with the following values: ▪ IfIndex, Capability, and LinkSpeed MUST be set in an implementation-specific manner. ▪ … Changed to: The server MUST enumerate the local network interfaces in an implementation-specific manner. For each IP address in each network interface, the server MUST construct a NETWORK_INTERFACE_INFO structure as specified in section 2.2.32.5, with the following values: ▪ The server MUST NOT include the IP address for a network interface with IfIndex equal to zero. ▪ IfIndex, Capability, and LinkSpeed MUST be set in an implementation-specific manner. ▪ … In Section 3.2.5.5, Receiving an SMB2 TREE_CONNECT Response, updated the product behavior note as follows: Changed from: From the list of network interfaces returned by the server, as specified in section 3.2.5.14.11, the client MUST use IfIndex to identify distinct interfaces on the server. The client MUST select a network interface for establishing a new channel using implementation-specific criteria.<155> <155>Windows-based SMB2 clients will choose the interfaces using the following criteria: ▪ For each interface returned in NETWORK_INTERFACE_INFO Response, if the interface has both link-local and non-link-local IP addresses, skip the link-local IP address.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ … Changed to: From the list of network interfaces returned by the server, as specified in section 3.2.5.14.11, the client MUST use IfIndex to identify distinct interfaces on the server. The client MUST select a network interface for establishing a new channel using implementation-specific criteria.<155> <155>Windows-based SMB2 clients will choose the interfaces using the following criteria: ▪ Skip the interfaces in NETWORK_INTERFACE_INFO Response where IfIndex is 0. ▪ For each interface returned in NETWORK_INTERFACE_INFO Response, if the interface has both link-local and non-link-local IP addresses, skip the link-local IP address. ▪ …
2014/12/22	V46.0 – 2014/05/15	In Section 3.3.5.5.3, Handling GSS-API Authentication, updated the processing rules for when the dialect verification succeeds and Session.SecurityContext is NULL. Changed step 4 from: If Session.SecurityContext is NULL, it MUST be set to a value representing the user which successfully authenticated this connection. The security context MUST be obtained from the GSS authentication subsystem. If it is not NULL, no changes are necessary. The server MUST invoke the GSS_Inquire_context call as specified in [RFC2743] section 2.2.6, passing the Session.SecurityContext as the input parameter, and set Session.UserName to the returned "src_name". Changed to: If Session.SecurityContext is NULL, it MUST be set to a value representing the user which successfully authenticated this connection. The security context MUST be obtained from the GSS authentication subsystem. If Session.SecurityContext is not NULL or the request is for binding the session, no changes are necessary. The server MUST invoke the GSS_Inquire_context call as specified in [RFC2743] section 2.2.6, passing the Session.SecurityContext as the input parameter, and set Session.UserName to the returned "src_name".
2014/12/22	V46.0 – 2014/05/15	In Section 3.3.5.5.3, Handling GSS-API Authentication, updated the processing rules for the server extraction of the GSS token from the request. Changed from: The server MUST extract the GSS token from the request. The token is SecurityBufferLength bytes in length, and located SecurityBufferOffset bytes from the beginning of the SMB2 header. The server SHOULD use the configured authentication protocol to obtain the next GSS output token for the authentication exchange. Changed to: The server MUST extract the GSS token from the request. The token is SecurityBufferLength bytes in length and located SecurityBufferOffset bytes from the beginning of the SMB2 header. The server MUST invoke a GSS_Accept_sec_context call, as specified in [RFC2743], by passing the GSS

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		token to obtain the next GSS output token for the authentication exchange.
2014/12/08	V46.0 – 2014/05/15	In Section 3.2.5.2, Receiving an SMB2 NEGOTIATE Response, removed the conditional action based on the SMB2_NEGOTIATE_SIGNING_ENABLED bit. Changed from: If the SecurityMode field in the SMB2 header of the response has the SMB2_NEGOTIATE_SIGNING_ENABLED bit set, the client MUST store the received MaxTransactSize in Connection.MaxTransactSize, the received MaxReadSize in Connection.MaxReadSize, the received MaxWriteSize in Connection.MaxWriteSize, and the received ServerGuid in Connection.ServerGuid. The client MUST store the received security buffer described by SecurityBufferOffset and SecurityBufferLength into Connection.GSSNegotiateToken. Changed to: The client MUST store the received MaxTransactSize in Connection.MaxTransactSize, the received MaxReadSize in Connection.MaxReadSize, the received MaxWriteSize in Connection.MaxWriteSize, and the received ServerGuid in Connection.ServerGuid. The client MUST store the received security buffer described by SecurityBufferOffset and SecurityBufferLength into Connection.GSSNegotiateToken.
2014/11/10	V46.0 – 2014/05/15	Section 3 has been updated to handle how ClientGUID is used in SMB2 NEGOTIATE Request. In Section 3.2.4.2.2.2, SMB2-Only Negotiate, changed from: If the client implements the SMB 3.1 dialect, ClientGuid MUST be set to the Guid provided by the application. Otherwise, if the client implements the SMB 2.1, SMB 3.0 or SMB 3.02 dialect, ClientGuid MUST be set to the global ClientGuid value. Changed to: If the client implements the SMB 2.1 or SMB 3.x dialect, ClientGuid SHOULD be set to the Guid provided by the application <105>. <105> Section 3.2.4.2.2.2: Windows 7 without [MSKB-3002286] sets ClientGuid to the global ClientGuid value. In Section 3.2.5.5, Receiving an SMB2 TREE_CONNECT Response, two related changes were made as follows. Changed from: Send an SMB2 NEGOTIATE request on the new connection, as specified in section 3.2.4.2.2.2. If the client implements the SMB 3.1 dialect, it also provides a newly generated Guid to be used as ClientGuid. Changed to: Send an SMB2 NEGOTIATE request on the new connection, as specified in section 3.2.4.2.2.2. The client also provides a newly generated Guid to be used as ClientGuid. Changed from: Guid is set to the global ClientGuid value. Changed to: Guid is set to Connection.ClientGuid. The preceding changes are supported in Windows Server 2012 R2 with [MSKB-

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		3002286].
2014/10/13	V46.0 – 2014/05/15	In Section 3.3.5.9.12, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, updated the processing rules that apply to when Open.IsSharedVHDX and Open.IsPersistent are TRUE. Text added is shown in **bold**. Changed to: ▪ The server MUST insert the Open into the Session.OpenTable with the Open.FileId as the new key. ▪ **If Open.IsSharedVHDX and Open.IsPersistent are TRUE, the request MUST be processed as specified in [MS-RSVD] section 3.2.5.1 by providing Open.LocalOpen.** The "Successful Open Initialization" and "Oplock Acquisition" phases MUST be skipped, and processing MUST continue as specified in "Response Construction".

2014/10/13	V46.0 – 2014/05/15	In Section 3.3.5.9, Receiving an SMB2 CREATE Request, updated the Create Context Validation for when the server implements the SMB 3.x dialect family. Text added is shown in **bold**.
		The following subsections detail server behavior when various create contexts are provided in the request and describe how that affects server operation.
		If the server implements the SMB 3.x dialect family and all of the following conditions are TRUE, the server MUST look up an Open in GlobalOpenTable where Open.CreateGuid matches the CreateGuid in the SMB2_CREATE_DURABLE_HANDLE_REQUEST_V2 create context and Open.ClientGuid matches the ClientGuid of the connection that received this request:
		▪ **The SMB2_FLAGS_REPLAY_OPERATION bit is set in the SMB2 header.**
		▪ **The request includes an SMB2_CREATE_DURABLE_HANDLE_REQUEST_V2 create context.**
		▪ **The Treeconnect.Share.Type is STYPE_DISKTREE.**
		If an Open is found, the server MUST perform the following:
		▪ **The server MUST fail the create request with STATUS_ACCESS_DENIED in the following cases:**
		  ▪ **Open.IsDurable is FALSE.**
		  ▪ **Open.DurableOwner is not the user represented by Open.Session.SecurityContext.**
		  ▪ **If Open.Lease is not NULL and Open.Lease.LeaseKey is not equal to the LeaseKey specified in the SMB2_CREATE_REQUEST_LEASE or SMB2_CREATE_REQUEST_LEASE_V2 Create Context.**
		▪ **If Open.Session.SessionId is not equal to the current Session.SessionId, the server MUST fail the request with STATUS_DUPLICATE_OBJECTID.**

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ **If Open.IsPersistent is TRUE and the SMB2_DHANDLE_FLAG_PERSISTENT bit is not set in the Flags field of the SMB2_CREATE_DURABLE_HANDLE_REQUEST_V2 Create Context, the server SHOULD<248> fail the request with STATUS_INVALID_PARAMETER.** ▪ **Construct the create response from Open, as specified in the "Response Construction" phase; the remaining create processing MUST be skipped.** Open Execution: If the FILE_DELETE … In Section 3.3.5.9.10, Handling the SMB2_CREATE_DURABLE_HANDLE_REQUEST_V2 Create Context, updated the processing rules for when an Open is found and the SMB2_FLAGS_REPLAY_OPERATION bit is set in the SMB2 header. Text deleted is shown in *italics*; text added is shown in **bold**. *If an Open is found and the SMB2_FLAGS_REPLAY_OPERATION bit is not set in the SMB2 header, the server MUST fail the request with STATUS_DUPLICATE_OBJECTID.* **If an Open is found and the SMB2_FLAGS_REPLAY_OPERATION bit is set in the SMB2 header, the server MUST construct an SMB2_CREATE_DURABLE_HANDLE_RESPONSE_V2 response create context. The Timeout MUST be set to Open.DurableOpenTimeout. If Open.IsPersistent is TRUE, the server MUST set the SMB2_DHANDLE_FLAG_PERSISTENT bit in the Flags field. The Buffer specified by the response MUST include the CreateContextsLength and CreateContextsOffset fields.** *The server MUST fail the create request with STATUS_ACCESS_DENIED in the following cases:* *…* *The server MUST construct the create response from Open, as specified in the "Response Construction" phase, with the following additional steps, and send the response to client.*
2014/10/13	V46.0 – 2014/05/15	In Section 3.3.5.9.12, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, added a new product behavior note <277> to the following paragraph. Text deleted is shown in *italics*; text added is shown in **bold**. If Open.IsPersistent is TRUE and the SMB2_DHANDLE_FLAG_PERSISTENT bit is not set in the Flags field of the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, the server *MUST* **SHOULD<277>** fail the request with STATUS_OBJECT_NAME_NOT_FOUND. **<277> Section 3.3.5.9.12: If Open.OplockLevel is equal to SMB2_OPLOCK_LEVEL_BATCH or Open.Lease.LeaseState includes SMB2_LEASE_HANDLE_CACHING, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 continue to process the request.**
2014/10/13	V46.0 –	Sections 3.3.5.9.7, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT

Errata Published YYYY/MM/DD	Protocol Document Version	Description
	2014/05/15	Create Context, and 3.3.5.9.12, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, have been corrected to reflect that the server does not regenerate Open.FileId (the volatile portion of the SMB2_FILEID) when it reconnects a durable handle. In Section 3.3.5.9.7, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT Create Context, bullet point 12 has been corrected. Changed from: The server MUST regenerate Open.FileId (the volatile portion of the SMB2_FILEID). Changed to: Open.FileId.Volatile MUST be set to a generated value that uniquely identifies this Open in Session.OpenTable. In Section 3.3.5.9.12, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, the 23rd bullet point has been corrected. Changed from: The server MUST regenerate Open.FileId (the volatile portion of the SMB2_FILEID). Changed to: Open.FileId.Volatile MUST be set to a generated value that uniquely identifies this Open in Session.OpenTable.
2014/10/13	V46.0 – 2014/05/15	In Section 3.3.5.9.10, Handling the SMB2_CREATE_DURABLE_HANDLE_REQUEST_V2 Create Context, added information about what happens when the persistent bit is set in the create context but the share connected is not a CA share. Changed from: If the SMB2_DHANDLE_FLAG_PERSISTENT bit is not set in the Flags field of this create context, if RequestedOplockLevel in the create request is not set to SMB2_OPLOCK_LEVEL_BATCH, and if the create request does not include a SMB2_CREATE_REQUEST_LEASE or SMB2_CREATE_REQUEST_LEASE_V2 create context with a LeaseState field that includes SMB2_LEASE_HANDLE_CACHING, the server MUST ignore this create context and skip this section. Changed to: If RequestedOplockLevel in the create request is not set to SMB2_OPLOCK_LEVEL_BATCH, and if the create request does not include a SMB2_CREATE_REQUEST_LEASE or SMB2_CREATE_REQUEST_LEASE_V2 create context with a LeaseState field that includes SMB2_LEASE_HANDLE_CACHING, and if any of the following conditions is TRUE, the server MUST ignore this create context and skip this section: ▪ The SMB2_DHANDLE_FLAG_PERSISTENT bit is set in the Flags field of this create context and TreeConnect.Share.IsCA is FALSE. ▪ The SMB2_DHANDLE_FLAG_PERSISTENT bit is not set in the Flags field of this create context.
2014/10/13	V46.0 – 2014/05/	In Section 3.3.7.1, Handling Loss of a Connection, added a new condition that indicates that the Open is to be preserved for reconnect. Text added is shown in

Errata Published YYYY/MM/DD	Protocol Document Version	Description
	15	**bold**. Changed to: When the underlying transport indicates loss of a connection or after the server initiates a transport disconnect, for each session in Connection.SessionTable, the server MUST perform the following: If Connection.Dialect belongs to the SMB 3.x dialect family and if the Session has more than one channel in Session.ChannelList, the server MUST perform the following action: ▪ All requests in Session.Channel.Connection.RequestList MUST be canceled. The server SHOULD<365> pass the CancelRequestId to the object store to request cancellation of the pending operation. ▪ The channel entry MUST be removed from the Session.ChannelList where Channel.Connection matches the disconnected connection. Otherwise, the server MUST perform the following actions: ▪ The server MUST iterate over the Session.OpenTable and determine whether each Open is to be preserved for reconnect. If any of the following conditions is satisfied, it indicates that the Open is to be preserved for reconnect. ▪ The server supports leasing and Open.IsResilient is TRUE. ▪ Open.OplockLevel is equal to SMB2_OPLOCK_LEVEL_BATCH and Open.OplockState is equal to Held, and Open.IsDurable is TRUE. ▪ Open.OplockLevel is equal to SMB2_OPLOCK_LEVEL_LEASE, Lease.LeaseState contains SMB2_LEASE_HANDLE_CACHING, Open.OplockState is equal to Held, and Open.IsDurable is TRUE. ▪ **Open.IsPersistent is TRUE.**
2014/10/13	V46.0 – 2014/05/15	In Section 3.3.5.9.11, Handling the SMB2_CREATE_REQUEST_LEASE_V2 Create Context, clarified what happens when a lease is found. Text added is shown in **bold**. Changed to: This section applies only to servers that implement the SMB 3.x dialect family. If both SMB2_CREATE_DURABLE_HANDLE_RECONNECT and SMB2_CREATE_REQUEST_LEASE_V2 create contexts are present in the request, they are processed as specified in section 3.3.5.9.7, and this section does not apply. If the server does not support leasing, the server MUST ignore the SMB2_CREATE_REQUEST_LEASE_V2 Create Context request.If Connection.Dialect does not belong to the SMB 3.x dialect family or if RequestedOplockLevel is not SMB2_OPLOCK_LEVEL_LEASE, the server SHOULD<282> ignore the SMB2_CREATE_REQUEST_LEASE_V2 Create Context request. By specifying a RequestedOplockLevel of SMB2_OPLOCK_LEVEL_LEASE, the client is requesting that a lease be acquired for this open. If the request does not provide an SMB2_CREATE_REQUEST_LEASE_V2 Create Context, the lease request MUST be ignored and Open.OplockLevel MUST be set to

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		SMB2_OPLOCK_LEVEL_NONE. The processing changes involved in acquiring the lease are: In the "Path Name Validation" phase, the server MUST attempt to locate a Lease Table by performing a lookup in GlobalLeaseTableList using Connection.ClientGuid as the lookup key. If no LeaseTable is found, one MUST be allocated and the following values set: ▪ LeaseTable.ClientGuid is set to Connection.ClientGuid. ▪ LeaseTable.LeaseList is set to an empty list. If the allocation fails, the create request MUST be failed with STATUS_INSUFFICIENT_RESOURCES. The server MUST attempt to locate a Lease by performing a lookup in the LeaseTable.LeaseList using the LeaseKey in the SMB2_CREATE_REQUEST_LEASE_V2 as the lookup key. If a lease is found but Lease.Filename does not match the file name for the incoming request, the request MUST be failed with STATUS_INVALID_PARAMETER. **If a lease is found, the server MUST construct an SMB2_CREATE_RESPONSE_LEASE_V2 response create context as specified below.**

2014/10/13	V46.0 – 2014/05/15	In Section 3.3.5.9.12, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, clarified the handling of various fields (DesiredAccess, ShareAccess, and CreateOptions) when reconnecting to a durable handle v2. Text added is shown in **bold**; text deleted is shown in *italics*.
		Changed to:
		This section applies only to servers that implement the SMB 3.x dialect family.
		There is no processing done for "Path Name Validation" or "Open Execution" as listed in section 3.3.5.9.
		The processing changes involved for this create context are:
		▪ The server MUST look up an existing Open in the GlobalOpenTable by doing a lookup with the FileId.Persistent portion of the create context.
		▪ If the lookup fails, the server SHOULD<284> fail the request with STATUS_OBJECT_NAME_NOT_FOUND and proceed as specified in "Failed Open Handling" in section 3.3.5.9.
		▪ If any of the following conditions is TRUE, the server MUST fail the request with STATUS_OBJECT_NAME_NOT_FOUND:
		  ▪ Open.Lease is not NULL and Open.ClientGuid is not equal to the ClientGuid of the connection that received this request.
		  ▪ If Open.IsPersistent is TRUE and the SMB2_DHANDLE_FLAG_PERSISTENT bit is not set in the Flags field of the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, the server SHOULD<285> fail the request with STATUS_OBJECT_NAME_NOT_FOUND.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ Open.CreateGuid is not equal to the CreateGuid in the request. ▪ Open.IsDurable is FALSE and Open.IsResilient is FALSE or unimplemented. ▪ Open.Session is not NULL. ▪ Open.Lease is NULL and the SMB2_CREATE_REQUEST_LEASE or SMB2_CREATE_REQUEST_LEASE_V2 create context is present. ▪ Open.Lease is NOT NULL and the SMB2_CREATE_REQUEST_LEASE or SMB2_CREATE_REQUEST_LEASE_V2 create context is not present. ▪ The SMB2_CREATE_REQUEST_LEASE_V2 create context is also present in the request, the server supports directory leasing, and Open.Lease.LeaseKey does not match the LeaseKey provided in the SMB2_CREATE_REQUEST_LEASE_V2 create context. ▪ The SMB2_CREATE_REQUEST_LEASE create context is also present in the request, the server supports leasing, and Open.Lease.LeaseKey does not match the LeaseKey provided in the SMB2_CREATE_REQUEST_LEASE create context. ▪ If Open.Lease is not NULL, the server supports leasing, Lease.Version is 1, and the request does not contain the SMB2_CREATE_REQUEST_LEASE create context, or if Lease.Version is 2 and the request does not contain the SMB2_CREATE_REQUEST_LEASE_V2 create context, the server SHOULD<286> fail the request with STATUS_OBJECT_NAME_NOT_FOUND. ▪ If any of the following conditions is TRUE, the server MUST fail the request with STATUS_INVALID_PARAMETER: ▪ The CREATE request also contains the SMB2_CREATE_DURABLE_HANDLE_REQUEST context, the SMB2_CREATE_DURABLE_HANDLE_RECONNECT context, or the SMB2_CREATE_DURABLE_HANDLE_REQUEST_V2 context. ▪ Open.Lease is not NULL and Open.FileName does not match the file name specified in the Buffer field of the SMB2 CREATE request. ▪ **The server MUST ignore the DesiredAccess, ShareAccess and CreateOptions fields in the request.** ▪ *The DesiredAccess field of the SMB2 CREATE request is nonzero and does not match Open.DesiredAccess.* ▪ *The ShareAccess field of the SMB2 CREATE request does not match Open.ShareMode.* ▪ *All the bits in the CreateOptions field of the SMB2 CREATE request, with the exception of the FILE_COMPLETE_IF_OPLOCKED bit, do not match the corresponding bits in Open.CreateOptions.*
2014/09/16	V46.0 – 2014/05/15	In Section 3.3.5.2.7.2, Handling Compounded Related Requests, updated the behavior note for when the previous request fails with an error: Changed from:

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		<209> Section 3.3.5.2.7.2: If the parent request failed to create FileId or the compounded request does not contain a FileId, Windows Vista SP1, Windows Server 2008, Windows 7, and Windows Server 2008 R2 fail the compounded request with STATUS_INVALID_PARAMETER. If the previous session expired, 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 servers fail the next request in the compounded chain with STATUS_NETWORK_SESSION_EXPIRED, and the subsequent requests in the compounded chain will be failed with STATUS_INVALID_PARAMETER. Changed to: <209> Section 3.3.5.2.7.2: If the previous session expired, Windows Vista SP1, Windows Server 2008, Windows 7, and Windows Server 2008 R2 servers fail the next request in the compounded chain with STATUS_NETWORK_SESSION_EXPIRED, and the subsequent requests in the compounded chain will be failed with STATUS_INVALID_PARAMETER.
2014/09/16	V46.0 – 2014/05/15	In Section 2.2.3, SMB2 NEGOTIATE Request, updated the ClientGuid field description: Changed from: ClientGuid (16 bytes): It MUST be a GUID (as specified in [MS-DTYP] section 2.3.4.3) generated by the client, if the Dialects field contains a value other than 0x0202. Otherwise, the client MUST set this to 0. Changed to: ClientGuid (16 bytes): It MUST be a GUID (as specified in [MS-DTYP] section 2.3.4.2) generated by the client.
2014/09/16	V46.0 – 2014/05/15	In Section 3.3.5.9.12, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, updated the processing rules for the Open and Lease objects. For details of these changes, see [MS-SMB2-Diff]: Server Message Block (SMB) Protocol Versions 2 and 3. This PDF shows the differences between the May 2014 release and the current Preview document.
2014/09/16	V46.0 – 2014/05/15	In the sections listed below, made changes to clarify the compounded responses size limit. For details of these changes, see [MS-SMB2-Diff]: Server Message Block (SMB) Protocol Versions 2 and 3. This PDF shows the differences between the May 2014 release and the current Preview document. In Section 1.2.2, added references [RFC1001] and [RFC1002]. In Section 1.3, Overview, added references to section 2.1, [RFC1001], and [RFC1002]. In Section 1.9 Standards Assignments, updated content for Direct TCP Transport and NetBIOS-over-TCP port standards assignments. In Section 2.1 Transport, added transport packet header structure. Removed NetBIOS-over-TCP transport reference. In Section 3.2.4.1, Sending Any Outgoing Message, updated the processing rules for the Connection object. In Section 3.2.4.1.4, Sending Compounded Requests, added product behavior note for compounded CREATE + READ/WRITE requests. In Section 3.2.5.1, Receiving Any Message, updated the processing rules that apply when the message size received exceeds Connection.MaxTransactSize.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		In Section 3.3.4.1.3, Sending Compounded Responses, updated the processing rules that apply when the response message is greater than Connection.MaxTransactSize+256. Added a product behavior note for calculation of the response message. In Section 3.3.5.2, Receiving Any Message, updated the processing rules for when the length of the message exceeds Connection.MaxTransactSize+256.
2014/09/16	V46.0 – 2014/05/15	In the product behavior note <294> for Section 3.3.5.12, Receiving an SMB2 READ Request, and product behavior note <301> for Section 3.3.5.13, Receiving an SMB2 WRITE Request, added Unbuffered row to the Object Store Parameter table. For details of these changes, see [MS-SMB2-Diff]: Server Message Block (SMB) Protocol Versions 2 and 3. This PDF shows the differences between the May 2014 release and the current Preview document.
2014/09/16	V46.0 – 2014/05/15	In Section 3.3.5.9, Receiving an SMB2 CREATE Request, updated the conditional initialization rule for the SMB 3.x dialect family. For details of these changes, see [MS-SMB2-Diff]: Server Message Block (SMB) Protocol Versions 2 and 3. This PDF shows the differences between the May 2014 release and the current Preview document.
2014/09/16	V46.0 – 2014/05/15	In Section 3.3.5.9.7, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT Create Context, updated the processing rules. For details of these changes, see [MS-SMB2-Diff]: Server Message Block (SMB) Protocol Versions 2 and 3. This PDF shows the differences between the May 2014 release and the current Preview document.
2014/09/16	V46.0 – 2014/05/15	In Sections 3.3.5.9.10, Handling the SMB2_CREATE_DURABLE_HANDLE_REQUEST_V2 Create Context, and 3.3.5.9.12, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, updated the processing rules that apply when Open.IsPersistent is TRUE. For details of these changes, see [MS-SMB2-Diff]: Server Message Block (SMB) Protocol Versions 2 and 3. This PDF shows the differences between the May 2014 release and the current Preview document.
2014/08/21	V46.0 – 2014/05/15	The following product behavior notes have been added: In Section 3.1.4.3, Encrypting the Message: Windows clients and servers do not encrypt the message if the connection is NetBIOS over TCP. In Section 3.2.5.1.1, Decrypting the Message: Windows clients discard the message if it is encrypted and the connection is NetBIOS over TCP. In Section 3.3.5.2.1, Decrypting the Message: Windows server will discard the message if it is encrypted and the connection is NetBIOS over TCP. The preceding changes are supported in Windows Server 2012 R2 with [MSKB-2975719].
2014/08/21	V46.0 – 2014/05/15	In Section 3.3.4.1.1, Signing the Message, the second bullet point of the first list has been changed to: ▪ If the request was signed by the client, the response message being sent contains a nonzero SessionId, and a nonzero TreeId in the SMB2 header, and the session identified by SessionId has Session.SigningRequired equal to TRUE, if either global EncryptData is FALSE or Connection.ClientCapabilities does not include the SMB2_GLOBAL_CAP_ENCRYPTION bit. The preceding changes are supported in Windows Server 2012 R2 with [MSKB-

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		2975719].
2014/08/21	V46.0 – 2014/05/15	In Section 3.3.4.1.1, Signing the Message, updated the following product behavior note: <185> Windows servers always sign the final session setup response when the user is neither anonymous nor guest. Windows 8 operating system, Windows Server 2012 operating system, Windows 8.1 operating system without [MSKB-2976995], and Windows Server 2012 R2 operating system without [MSKB-2976995] servers fail to sign responses other than SMB2_NEGOTIATE, SMB2_SESSION_SETUP, and SMB2_TREE_CONNECT when Session.SigningRequired is TRUE, global EncryptData is TRUE, RejectUnencryptedAccess is FALSE, and either Connection.Dialect is "2.002" or "2.100" or Connection.ClientCapabilities does not include SMB2_GLOBAL_CAP_ENCRYPTION.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/01/19	V7.0 – 2014/05/15	In Section 3.1.5.8, Receiving a Data Transfer Message, clarified Connection.SendImmediate when a peer receives a Data Transfer Message and has an empty Connection.Send queue. Changed from: If Connection.SendQueue is empty, 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. 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 set Connection.SendImmediate to TRUE and MUST promptly send a Data Transfer message on the Connection, as specified in section 3.1.5.1.
2014/12/22	V7.0 – 2014/05/15	In Section 3.1.5.8, Receiving a Data Transfer Message, clarified the handling of Connection.KeepaliveRequested when the SMBD_DIRECT_RESPONSE_REQUESTED flag is set. Changed from: If the SMB_DIRECT_RESPONSE_REQUESTED flag is set in the received Flags field, then Connection.KeepaliveRequested MUST be set to "PENDING". If the Connection.SendQueue is empty, then the receiver MUST promptly send a Data Transfer message passing an empty buffer on the Connection, as specified in section 3.1.5.1. Changed to: If the SMB_DIRECT_RESPONSE_REQUESTED flag is set in the received Flags field, then Connection.KeepaliveRequested MUST be set to "PENDING". The receiver MUST set Connection.SendImmediate to TRUE and promptly send a Data Transfer message on the

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Connection, as specified in section 3.1.5.1.
2014/11/10	V7.0 – 2014/05/15	In Section 3, Protocol Details, references to Connection.MaxFragmentedSize have been updated to clarify the exact behavior for validation when a data transfer is received. In Section 3.1.1.1, Per RDMA Transport Connection, changed from: Connection.MaxFragmentedSize: The maximum fragmented upper-layer payload receive size supported by the remote peer for this connection. Changed to: Connection.MaxFragmentedSendSize: The maximum fragmented upper-layer payload receive size supported by the remote peer for this connection. Connection.MaxFragmentedRecvSize: The maximum fragmented upper-layer payload receive size supported by the local peer for this connection. In Section 3.1.4.1, Connecting to the Peer, changed from: Determine an initial value for Connection.ReceiveCreditMax, Connection.SendCreditTarget, Connection.MaxSendSize, Connection.MaxFragmentedSize, Connection.MaxReceiveSize and Connection.KeepaliveInterval.<2> Changed to: Determine an initial value for Connection.ReceiveCreditMax, Connection.SendCreditTarget, Connection.MaxSendSize, Connection.MaxFragmentedRecvSize, Connection.MaxReceiveSize, and Connection.KeepaliveInterval.<2> In Section 3.1.4.2, Send Message, changed from: The sender MUST determine if the buffer contains a message that is of a length less than or equal to Connection.MaxFragmentedSize. If not, the message cannot be sent and an implementation-specific local error MUST be returned. Changed to: The sender MUST determine if the buffer contains a message that is of a length less than or equal to Connection.MaxFragmentedSendSize. If not, the message cannot be sent and an implementation-specific local error MUST be returned. In Section 3.1.4.7, Query Connection Parameters, changed from: The Connection.MaxSendSize, Connection.MaxFragmentedSize, Connection.MaxReceiveSize, Connection.MaxReadWriteSize, and Connection.KeepaliveInterval for this connection MUST be returned. Changed to: The Connection.MaxSendSize, Connection.MaxFragmentedSendSize, Connection.MaxReceiveSize, Connection.MaxReadWriteSize, and Connection.KeepaliveInterval for this connection MUST be returned. In Sections 3.1.5.2, Sending a Negotiate Request Message, and 3.1.5.3, Sending a Negotiate Response Message, changed from: MaxFragmentedSize MUST be set to Connection.MaxFragmentedSize. Changed to: MaxFragmentedSize MUST be set to

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Connection.MaxFragmentedRecvSize. In Section 3.1.5.6, Receiving a Negotiate Request Message, changed from: The receiver MUST: … ▪ Set Connection.MaxSendSize to the smaller of Connection.MaxSendSize and the value of the received MaxReceiveSize field. Changed to: The receiver MUST: … ▪ Set Connection.MaxSendSize to the smaller of Connection.MaxSendSize and the value of the received MaxReceiveSize field. ▪ Set Connection.MaxFragmentedSendSize to MaxFragmentedSize. In Section 3.1.5.7, Receiving a Negotiate Response Message, changed from: If any of the preceding conditions are not satisfied, the receiver MUST terminate the connection and return a failure status to the caller of section 3.1.4.1.Otherwise, the receiver MUST: … ▪ Set Connection.SendCredits to the value of the received the CreditsGranted field. ▪ Set Connection.MaxFragmentedSendSize to MaxFragmentedSize. In Section 3.1.7.2, Connection Arrival, changed from: Determine an initial value for Connection.ReceiveCreditMax, Connection.SendCreditTarget, Connection.MaxSendSize, Connection.MaxFragmentedSize, Connection.MaxReceiveSize, Connection.MaxReadWriteSize and Connection.KeepaliveInterval.<7> Changed to: Determine an initial value for Connection.ReceiveCreditMax, Connection.SendCreditTarget, Connection.MaxSendSize, Connection.MaxFragmentedRecvSize, Connection.MaxReceiveSize, Connection.MaxReadWriteSize and Connection.KeepaliveInterval.<7>
2014/11/10	V7.0 – 2014/05/15	In Section 3.1.5.7, Receiving a Negotiate Response Message, updated a processing rule to set Connection.MaxReadWriteSize to the value of the received MaxReadWriteSize field instead of the MaxReceiveSize field. Changed from: Set Connection.MaxReadWriteSize to the smaller of an implementation-specific value<6> and the value of the received MaxReceiveSize field. Changed to:

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Set Connection.MaxReadWriteSize to the smaller of an implementation-specific value<6> and the value of the received MaxReadWriteSize field.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/09/16	V12.1 – 2014/05/15	Updated the product behavior note <7> for Section 3.1.5.1, mechListMIC Processing, to further specify the case where a MIC is included in AUTHENTICATE_MESSAGE. Changed from: On Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2, if AES Kerberos ciphers are negotiated by Kerberos, the signature in the SPNEGO mechListMIC field MUST be processed by the recipient. Changed to: On all other product versions shown in the applicability list at the beginning of this section, the following processing is used for the mechListMIC field: -- If AES Kerberos ciphers are negotiated by Kerberos, the signature in the SPNEGO mechListMIC field MUST be processed by the recipient. -- If NTLM authentication is most preferred by the client and the server, and the client includes a MIC in AUTHNTICATE_MESSAGE, then the mechListMIC field becomes mandatory in order for the authentication to succeed. Windows clients in this case send an NTLM token instead of an SPNEGO token.

Return to top of page

[MS-TDS]: Tabular Data Stream Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/06/30	V18.0 – 2015/06/30	In section 6, Appendix A: Product Behavior, product behavior notes <11>, <13>, <17>, <22>, <27>, <31>, and <41> are updated as follows. Note <11> is changed from: <11> Section 2.2.3.1.1: Only legacy clients that support SQL Server versions that were released prior to sql_server 7.0 can use Pre-TDS7 Login. Changed to: <11> Section 2.2.3.1.1: Only legacy clients that support SQL Server

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
versions that were released prior to SQL Server 7.0 can use Pre-TDS7 Login.

Note <13> is changed from:
<13> Section 2.2.3.1.1: Only clients that support sql_server 7.0 or later can use TDS7 Login.
Changed to:
<13> Section 2.2.3.1.1: Only clients that support SQL Server 7.0 or later can use TDS7 Login.

In note <17>, the second paragraph is changed from:
Version can be of value 0, 1, or 2. A value of 0 denotes collations in SQL Server 2000. A value of 1 denotes collations in SQL Server 2005. A value of 2 denotes collations in SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, and SQL Server 2014.
Changed to:
Version can be of value 0, 1, or 2. A value of 0 denotes collations in SQL Server 2000. A value of 1 denotes collations in SQL Server 2005. A value of 2 denotes collations in SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, SQL Server 2014, and SQL Server 2016 CTP2.

Note <22> is changed from:
<22> Section 2.2.6.4: The version numbers used by clients are as follows.
<table>
<tr><th>SQL Server version</th><th>Version sent from client to server</th></tr>
<tr><td>SQL Server 7.0</td><td>0x00000070</td></tr>
<tr><td>SQL Server 2000</td><td>0x00000071</td></tr>
<tr><td>SQL Server 2000 SP1</td><td>0x01000071</td></tr>
<tr><td>SQL Server 2005</td><td>0x02000972</td></tr>
<tr><td>SQL Server 2008</td><td>0x03000A73</td></tr>
<tr><td>SQL Server 2008 R2</td><td>0x03000B73</td></tr>
<tr><td>SQL Server 2012
SQL Server 2014</td><td>0x04000074</td></tr>
</table>
Changed to:
<22> Section 2.2.6.4: The version numbers used by clients are as follows.
<table>
<tr><th>SQL Server version</th><th>Version sent from client to server</th></tr>
<tr><td>SQL Server 7.0</td><td>0x00000070</td></tr>
<tr><td>SQL Server 2000</td><td>0x00000071</td></tr>
</table>
</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		<table><tr><td>SQL Server 2000 SP1</td><td>0x01000071</td></tr><tr><td>SQL Server 2005</td><td>0x02000972</td></tr><tr><td>SQL Server 2008</td><td>0x03000A73</td></tr><tr><td>SQL Server 2008 R2</td><td>0x03000B73</td></tr><tr><td>SQL Server 2012 SQL Server 2014 SQL Server 2016 CTP2</td><td>0x04000074</td></tr></table> Note <27> is changed from: <27> Section 2.2.6.5: In SQL Server 2012, and SQL Server 2014, the server always sends the value 0 for the INSTOPT option when the string specified in the client's INSTOPT option is "MSSQLServer". The reason for this is that "MSSQLServer" is the name of a default instance, and "MSSQLServer" may be provided by the client even in the absence of an explicit instance name. SQL Server 2000, SQL Server 2005, SQL Server 2008, and SQL Server 2008 R2, which support the INSTOPT field always validate the client-specified string against the server's instance name. Changed to: <27> Section 2.2.6.5: In SQL Server 2012, SQL Server 2014, and SQL Server 2016 CTP2, the server always sends the value 0 for the INSTOPT option when the string specified in the client's INSTOPT option is "MSSQLServer". The reason for this is that "MSSQLServer" is the name of a default instance, and "MSSQLServer" may be provided by the client even in the absence of an explicit instance name. SQL Server 2000, SQL Server 2005, SQL Server 2008, and SQL Server 2008 R2, which support the INSTOPT field always validate the client-specified string against the server's instance name. In note <31>, the fourth paragraph is changed from: SNAC [MSDN-SNAC] and SQLClient use the VERSION option in the Pre-Login Response message to detect whether DoneRowCount is LONG or ULONGLONG. It is ULONGLONG if VERSION in the Pre-Login Response message indicates that the server is SQL Server 2005, SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, or SQL Server 2014. Otherwise, DoneRowCount is LONG. Changed to: SNAC [MSDN-SNAC] and SQLClient use the VERSION option in the Pre-Login Response message to detect whether DoneRowCount is LONG or ULONGLONG. It is ULONGLONG if VERSION in the Pre-Login Response message indicates that the server is SQL Server 2005, SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, SQL Server 2014, or SQL Server 2016 CTP2. Otherwise, DoneRowCount is LONG. Note <41> is changed from: <41> Section 2.2.7.13: The following table shows the values in

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>network transfer format.
<table>
<tr><th>SQL Server</th><th>Client to server</th><th>Server to client</th></tr>
<tr><td>SQL Server 7.0</td><td>0x00000070</td><td>0x07000000</td></tr>
<tr><td>SQL Server 2000</td><td>0x00000071</td><td>0x07010000</td></tr>
<tr><td>SQL Server 2000 SP1</td><td>0x01000071</td><td>0x71000001</td></tr>
<tr><td>SQL Server 2005</td><td>0x02000972</td><td>0x72090002</td></tr>
<tr><td>SQL Server 2008*</td><td>0x03000A73</td><td>0x730A0003</td></tr>
<tr><td>SQL Server 2008 R2</td><td>0x03000B73</td><td>0x730B0003</td></tr>
<tr><td>SQL Server 2012
SQL Server 2014</td><td>0x04000074</td><td>0x74000004</td></tr>
</table>
*SQL Server 2008 TDS version 0x03000A73 does not include support for NBCROW and fSparseColumnSet.
Changed to:
<41> Section 2.2.7.13: The following table shows the values in network transfer format.
<table>
<tr><th>SQL Server</th><th>Client to server</th><th>Server to client</th></tr>
<tr><td>SQL Server 7.0</td><td>0x00000070</td><td>0x07000000</td></tr>
<tr><td>SQL Server 2000</td><td>0x00000071</td><td>0x07010000</td></tr>
<tr><td>SQL Server 2000 SP1</td><td>0x01000071</td><td>0x71000001</td></tr>
<tr><td>SQL Server 2005</td><td>0x02000972</td><td>0x72090002</td></tr>
<tr><td>SQL Server 2008*</td><td>0x03000A73</td><td>0x730A0003</td></tr>
<tr><td>SQL Server 2008 R2</td><td>0x03000B73</td><td>0x730B0003</td></tr>
<tr><td>SQL Server 2012
SQL Server 2014
SQL Server 2016 CTP2</td><td>0x04000074</td><td>0x74000004</td></tr>
</table></td></tr>
<tr><td>2014/09/16</td><td>V17.0 – 2014/05/15</td><td>In Section 2.2.7.4, COLMETADATA, added "Count" to the Token Stream Definition to read:
<pre>COLMETADATA = TokenType
Count
NoMetaData / (1
*ColumnData)</pre>
In Section 2.2.7.4, COLMETADATA, added the following language in the "Flags" row of the Token Stream Parameter Details table to clarify</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		the size of the flags: The size of the Flags parameter is always fixed at 16 bits regardless of the TDS version. Each of the 16 bits of the Flags parameter is interpreted based on the TDS version negotiated during login.
2014/09/16	V17.0 – 2014/05/15	In Section 2.2.7.1 ALTMETADATA, in the "Flags" row of the Token Stream Parameter Details table, updated "fCaseSens" to read "fCaseSen" and changed the order of fCaseSen and fNullable in the bullet list to read: These bit flags are described in least significant bit order. With the exception of fNullable, all of these bit flags SHOULD be set to zero. For a description of each bit flag, see section 2.2.7.4: ▪ fNullable is a bit flag, 1 if the column is nullable. ▪ fCaseSen In Section 2.2.7.4 COLMETADATA, in the "Flags" row of the Token Stream Parameter Details table, changed the order of fCaseSen and fNullable in the bullet list to read: The size of the Flags parameter is always fixed at 16 bits regardless of the TDS version. Each of the 16 bits of the Flags parameter is interpreted based on the TDS version negotiated during login. Bit flags in least significant bit order: ▪ fNullable is a bit flag. Its value is 1 if the column is nullable. ▪ fCaseSen is a bit flag. Set to 1 for string columns with binary collation and always for the XML data type. Set to 0 otherwise. In Section 2.2.7.17 RETURNVALUE, in the "Flags" row of the Token Stream Parameter Details table, changed the order of fCaseSen and fNullable in the bullet list to read: These bit flags are described in least significant bit order. All of these bit flags SHOULD be set to zero. For a description of each bit flag, see section 2.2.7.4. ▪ fNullable ▪ fCaseSen

Return to top of page

[MS-TSGU]: Terminal Services Gateway Server Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/06/22	V34.0 – 2014/05/15	In various sections, added information about the CONNECT_PKT_FRAGMENT structure. In Section 2.2.5.4.1, UdpPktType Enumeration, added the

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
PKT_TYPE_CONNECT_REQ_FRAGMENT constant and value.
Changed from:
<table>
<tr><th>Constant/value</th><th>Description</th></tr>
<tr><td>PKT_TYPE_DISCONNECT/4</td><td>This constant represents the DISCONNECT packet type sent either by the RDG client or RDG server during the Shutdown Phase (section 1.3.1.1.3).</td></tr>
</table>
Changed to:
<table>
<tr><th>Constant/value</th><th>Description</th></tr>
<tr><td>PKT_TYPE_DISCONNECT/4</td><td>This constant represents the DISCONNECT packet type sent either by the RDG client or RDG server during the Shutdown Phase (section 1.3.1.1.3).</td></tr>
<tr><td>PKT_TYPE_CONNECT_REQ_FRAGMENT /5</td><td>This constant represents the fragment of CONNECT_REQUEST packet type sent by the client. The RDG client MUST use the PKT_TYPE_CONNECT_REQ_FRAGMENT packet type to send connection request to the RDP server. It MUST do so by splitting a CONNECT_PKT request into one or more fragments of type CONNECT_PKT_FRAGMENT (section 2.2.11.10).<8></td></tr>
</table>
<8> Section 2.2.5.4.1: In Windows implementations, the maximum size of each CONNECT_PKT_FRAGMENT fragment is 1000 bytes.

Added 2 new sections - 2.2.11.10, CONNECT_PKT_FRAGMENT Structure, and 3.8.3, Establishing a Connection
2.2.11.10 CONNECT_PKT_FRAGMENT Structure
The RDG client MUST use the PKT_TYPE_CONNECT_REQ_FRAGMENT packet type to send connection requests to the RDP server. It MUST do so by splitting a CONNECT_PKT request into one or more fragments of type CONNECT_PKT_FRAGMENT.<22> Multi-byte values in this structure are transmitted in little-endian byte order.

<pre>
typedef struct _CONNECT_PKT_FRAGMENT
{
    UDP_PACKET_HEADER hdr;
    USHORT usFragmentID;
    USHORT usNoOfFragments;
    USHORT cbFragmentLength;
    BYTE fragment[0];
} CONNECT_PKT_FRAGMENT, *PCONNECT_PKT_FRAGMENT;
hdr (4 bytes): A UDP_PACKET_HEADER structure (section 2.2.11.7).
usFragmentID (2 bytes): Identifies the fragment number. The
</pre>
</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		first fragment starts with 0. usNoOfFragments (2 bytes): Total number of fragments. cbFragmentLength (2 bytes): Length of this fragment. fragment (variable): An array of bytes representing a portion of the CONNECT_PKT request <22> Section 2.2.11.10: In Windows implementations, the maximum size of each CONNECT_PKT_FRAGMENT fragment is 1000 bytes. 3.8.3 Establishing a Connection The client MUST transmit one or more CONNECT_PKT_FRAGMENT structures, as specified in section 2.2.11.10, to the server to establish the connection. The following is a list of constants and variables that hold the state temporarily: ▪ connectReqBufferLen is the length of the connect request buffer connectPktBuff. ▪ reqLen is the actual length of the request in connectPktBuff. ▪ authCookieLen is the length of Authentication Cookie sent from the RDP server. ▪ MAX_DTLS_HDR_TRLR is the maximum length of the DTLS header and trailer bits. It is 96 bytes. ▪ Size of UDP_PACKET_HEADER is 4 bytes. ▪ LAYER_2_OVERHEAD is 100 bytes, which is MAX_DTLS_HDR_TRLR_SIZE + UDP header size. ▪ MAX_CONNECT_REQ_FRAGMENT_SIZE is the maximum size of each connect request fragment. It MUST be set to 1000 bytes. Before transmitting a CONNECT_PKT_FRAMEMENT, the client MUST do the following: ▪ Set connectReqBufferLen to sizeof(CONNECT_PKT) + authCookieLen + MAX_DTLS_HDR_TRLR. ▪ Allocate a buffer for connectPktBuff of size connectReqBufferLen for the CONNECT_PKT structure and set values for each of its fields. ▪ Set reqLen to the connect request buffer's hdr.pktLen + size of UDP_PACKET_HEADER. ▪ Set MaxUdpPacketSize = (uUpStreamMtu from the connect request's SyncData) - LAYER_2_OVERHEAD. ▪ Set fragmentCount = reqLen / MAX_CONNECT_REQ_FRAGMENT_SIZE ▪ If the remainder after the division of reqLen by MAX_CONNECT_REQ_FRAGMENT_SIZE is not zero, increase the fragment count by 1 to completely account for all of the bytes of the request.

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
Split the CONNECT_PKT buffer into fragmentCount fragments, meaning multiple buffers of type CONNECT_PKT_FRAGMENT.
Each fragment’s CONNECT_PKT_FRAGMENT fields MUST be set as follows:

Set UdpPktType to PKT_TYPE_CONNECT_REQ_FRAGMENT.
Set usNoOfFragments to fragmentCount, meaning the total number of fragments calculated.
Set usFragmentID to the Current Fragment number.
Set cbFragmentLength to MAX_CONNECT_REQ_FRAGMENT_SIZE or to the actual number of bytes remaining in the connect request buffer.
Set pktLen to (sizeof(CONNECT_PKT_FRAGMENT) - sizeof(UDP_PACKET_HEADER)) + cbFragmentLength of the Current Fragment.
Set the current fragment's length, fragmentLen, to cbFragmentLength of Current Fragment + sizeof(UDP_PACKET_HEADER).
If the very first fragment's fragmentLen < MaxUdpPacketSize, set fragmentLen to MaxUdpPacketSize.Finally, DTLS encrypts the fragments and sends them to the RDP server.

</td></tr>
<tr><td>2015/06/22</td><td>V34.0 – 2014/05/15</td><td>
In Section 2.2.6.1, Common Return Codes, updated the E_PROXY_TS_CONNECTFAILED return code value.
Changed from:
<table>
<tr><th>Return code/value</th><th>Description</th></tr>
<tr><td>0x000059E6
HRESULT_CODE(E_PROXY_TS_CONNECTFAILED)</td><td>Returned by RDG server when the RDG server fails to connect to the target server.</td></tr>
</table>
Changed to:
<table>
<tr><th>Return code/value</th><th>Description</th></tr>
<tr><td>0x000059DD
HRESULT_CODE(E_PROXY_TS_CONNECTFAILED)</td><td>Returned by RDG server when the RDG server fails to connect to the target server.</td></tr>
</table>
</td></tr>
<tr><td>2015/06/08</td><td>V34.0 – 2014/05/15</td><td>
In Section 2.2.9.2.1.9.1.1, TSG_PACKET_STRING_MESSAGE, updated that the value of the msgBytes field in the TSG_PACKET_STRING_MESSAGE structure SHOULD be determined by the serverCert field size.
In Section 2.2.9.2.1.9.1.1, TSG_PACKET_STRING_MESSAGE, changed from:
The TSG_PACKET_STRING_MESSAGE structure contains either the Consent Signing Message or the Administrative Message that is being sent from the RDG server to the client.
</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description

		<pre>typedef struct TSG_PACKET_STRING_MESSAGE {
		  long isDisplayMandatory;
		  long isConsentMandatory;
		  [range(0,65536)] unsigned long msgBytes;
		  [size_is(msgBytes)] wchar_t* msgBuffer;
		} TSG_PACKET_STRING_MESSAGE,
		 *PTSG_PACKET_STRING_MESSAGE;</pre>
		isDisplayMandatory: A Boolean that specifies whether the client needs to display this message.
		isConsentMandatory: A Boolean that specifies whether the user needs to give its consent before the connection can proceed.
		msgBytes: An unsigned long specifying the number of characters in msgBuffer, including the terminating null character. The value MUST be within 0 to 65536.
		
		Changed to:
		The TSG_PACKET_STRING_MESSAGE structure contains either the Consent Signing Message or the Administrative Message that is being sent from the RDG server to the client.
		
		<pre>typedef struct TSG_PACKET_STRING_MESSAGE {
		  long isDisplayMandatory;
		  long isConsentMandatory;
		  [range(0,n)] unsigned long msgBytes;
		  [size_is(msgBytes)] wchar_t* msgBuffer;
		} TSG_PACKET_STRING_MESSAGE,
		 *PTSG_PACKET_STRING_MESSAGE;</pre>
		isDisplayMandatory: A Boolean that specifies whether the client needs to display this message.
		isConsentMandatory: A Boolean that specifies whether the user needs to give its consent before the connection can proceed.
		msgBytes: An unsigned long specifying the number of characters in msgBuffer, including the terminating null character. The size of the message SHOULD<21> be determined by the serverCert field in the HTTP_TUNNEL_RESPONSE_OPTIONAL Structure (section 2.2.10.21). The consent message is embedded in the HTTP_TUNNEL_RESPONSE as part of the HTTP_TUNNEL_RESPONSE_OPTIONAL structure. When the HTTP_TUNNEL_RESPONSE_FIELD_CONSENT_MSG flag is set in the HTTP_TUNNEL_RESPONSE_FIELDS_PRESENT_FLAGS (section 2.2.5.3.8), the HTTP_TUNNEL_RESPONSE_OPTIONAL data structure contains a consent message in the HTTP_UNICODE_STRING format (section 2.2.10.22).
		<21> Section 2.2.9.2.1.9.1.1: The maximum number of characters in the constant message depends on the serverCert field size in the HTTP_TUNNEL_REPONSE_OPTIONAL structure. (The serverCert is used for SoH encryption.) The following table is a guideline for determining the maximum number of characters in the msgBytes field:

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th></th><th>Windows 8.1, Windows Server 2012 R2</th><th>Windows Server 2008, Windows Server 2008 R2, Windows Server 2012</th></tr>
<tr><td>MAX of HTTP_TUNNEL_RESPONSE size</td><td>22528</td><td>65536</td></tr>
<tr><td>Required HTTP_TUNNEL_RESPONSE</td><td>18</td><td>18</td></tr>
<tr><td>Optional HTTP_TUNNEL_RESPONSE_OPTIONAL header</td><td>24</td><td>24</td></tr>
<tr><td>Allow server cert size
The size of the certificate depends on the key size</td><td>~1500</td><td>~1500</td></tr>
<tr><td>Max consent message (in bytes)</td><td>20986</td><td>63994</td></tr>
<tr><td>Max consent message (in character size, including spaces, carriage return and the ending 0 string)</td><td>~10493</td><td>~31997</td></tr>
</table>

In Section 4.1.1, Normal Scenario, Section 4.1.2, Pluggable Authentication Scenario with Consent Message Returned, and Section 6, Appendix A: Full IDL, updated the examples to reflect the changes in Section 2.2.9.2.1.9.1.1, TSG_PACKET_STRING_MESSAGE. For example, in Section 4.1.1, Normal Scenario, changed from:
…
Where the servicemessage is set as follows.

<pre>
typedef struct _TSG_PACKET_STRING_MESSAGE
    {
        long isDisplayMandatory = 1;
        long isConsentMandatory = 1;
        [range(0, 65536)] unsigned long msgBytes = 4;
        [size_is(msgBytes)] wchar_t* msgBuffer = "Test";
    } TSG_PACKET_STRING_MESSAGE;
</pre>

Changed to:
…
Where the servicemessage is set as follows.

<pre>
typedef struct _TSG_PACKET_STRING_MESSAGE
    {
        long isDisplayMandatory = 1;
        long isConsentMandatory = 1;
</pre>
</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		[range(0, 12288)] unsigned long msgBytes = 4; [size_is(msgBytes)] wchar_t* msgBuffer = "Test"; } TSG_PACKET_STRING_MESSAGE;
2014/09/16	V34.0 – 2014/05/15	In Section 2.2.9.1, TSENDPOINTINFO, updated the minimum valid value for the field numResourceNames from 0 to 1.
2014/09/16	V34.0 – 2014/05/15	In Section 2.2.9.1, TSENDPOINTINFO, updated the description for the field Port. Changed from: Port: Specifies the protocol ID and TCP port number for the target server endpoint to which the RDG server connects. The protocol ID is in the low order 16 bits of this field and port number is in the high order 16 bits. These values can be ignored. The value of the protocol id is protocol-dependent. For example, RDP uses 3. Changed to: Port: Specifies the protocol ID and TCP port number for the target server endpoint to which the RDG server connects. The protocol ID is in the low order 16 bits of this field and port number is in the high order 16 bits. The value of the protocol ID must be set to 3. In Section 2.2.10.2, HTTP_CHANNEL_PACKET Structure, updated the description for field protocol. Changed from: protocol (2 bytes): An unsigned short that represents the protocol number used for connection with the target server. Changed to: protocol (2 bytes): An unsigned short that represents the protocol number used for connection with the target server. The value MUST be set to 3.
2014/09/16	V34.0 – 2014/05/15	In Section 3.3.3.1, Keep-alive Timer, added a new product behavior note <49>. <49> Section 3.3.3.1: In the following TSGU clients the default timer value on the client is 8 minutes. ▪ Windows 7 with RDP 8.0/8.1 Client Update ▪ Windows Server 2008 R2 with RDP 8.0/8.1 Client Update ▪ Windows 8 ▪ Windows Server 2012 ▪ Windows 8.1 ▪ Windows Server 2012 R2 In newer versions of TSGU client, beginning with RDP 8.1, with the updates in the following KBs installed, the default time period is 1 minute.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ Windows 8.1/Windows Server 2012 R2: KB 2921855 ▪ RDP 8.1 for Windows 7/Windows Server 2008 R2: KB 2923545 This timer is not supported in the following versions of Windows: ▪ Windows XP SP2 ▪ Windows Server 2003 with SP1 ▪ Windows Vista ▪ Windows Server 2008

Return to top of page

[MS-UCODEREF]: Windows Protocols Unicode Reference

Errata Published YYYY/MM/DD	Protocol Document Version	Description

2015/04/27	V10.0 – 2014/05/15	In Section 3.1.5, Message Processing Events and Sequencing Rules, added a new Section 3.1.5.5, Comparing UTF-16 Strings Ordinally:
		3.1.5.5 Comparing UTF-16 Strings Ordinally
		To do a case sensitive ordinal comparison of strings, a binary comparison of the UTF-16 code points of the strings is done. To do a case insensitive ordinal string comparison, ToUpperCase (3.1.5.3.1) is done on each string before doing the ordinal comparison.
		3.1.5.5.1 CompareStringOrdinal Algorithm
		This algorithm compares two UTF-16 strings by doing an ordinal, e.g.: binary, comparison. Optionally the caller can request that the comparison be done on the upper case form of the string.
		`COMMENT CompareStringOrdinal`
		`COMMENT On Entry: StringA – A UTF-16 string to be compared`
		`COMMENT On Entry: StringB – Second UTF-16 string to compare`
		`COMMENT On Entry: IgnoreCaseFlag – TRUE to ignore case when comparing`
		`COMMENT`
		`COMMENT On Exit: Result – A value indicating if StringA is less than,`
		`COMMENT equal to, or greater than StringB`
		`PROCEDURE CompareStringOrdinal`
		`IF IgnoreCaseFlag is TRUE THEN`

Errata Published YYYY/MM/DD	Protocol Document Version	Description

		SET StringA TO ToUpperCase(StringA)
		SET StringB TO ToUpperCase(StringB)
		ENDIF
		
		SET index TO 0
		
		WHILE index is less than Length(StringA) and
		index is also less than Length(StringB)
		
		IF StringA[index] is less than StringB[index] THEN
		SET Result TO "StringA is less than StringB"
		RETURN
		ENDIF
		IF StringA[index] is greater than StringB[index]
		THEN
		SET Result TO "StringA is greater than StringB"
		RETURN
		ENDIF
		INCREMENT index
		ENDWHILE
		
		IF Length(StringA) is equal to Length(StringB) THEN
		SET Result TO "StringA is equal to StringB"
		ELSE IF Length(StringA) is less than Length(StringB)
		THEN
		SET Result TO "StringA is less than StringB"
		ELSE
		Assert Length(StringA) must be greater than
		Length(StringB)
		SET Result TO "StringA is greater than StringB"
		ENDIF
		RETURN

Return to top of page

[MS-WCCE]: Windows Client Certificate Enrollment Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/12/08	V38.0 – 2014/05/15	In Section 3.2.1.4.3.2.1, PropID = 0x00000001 (CR_PROP_FILEVERSION) "CA File Version", and Section 3.2.1.4.3.2.2, PropID = 0x00000002 (CR_PROP_PRODUCTVERSION) "CA Product Version", clarified the description of the format of CR_PROP_FILEVERSION and CR_PROP_PRODUCVERSION for versions of Windows later than Windows Server 2003. In Section 3.2.1.4.3.2.1, PropID = 0x00000001 (CR_PROP_FILEVERSION) "CA File Version", changed from: The client has requested the CA file version property. If the CA implements the Config_File_Version datum, the CA SHOULD construct a [UNICODE] string of the form "w.x.y.z", where w, x, y, and z MUST be numeric values indicating the version of the CA. The

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		CA MAY construct a [UNICODE] string of the form "w.x:y.z".<79> <79> Section 3.2.1.4.3.2.1: In Windows Server 2003 and Windows 2000 Server, the format of the string is "w.x:y.z". Changed to: The client has requested the CA file version property. If the CA implements the Config_File_Version datum, the CA constructs a [UNICODE] string of the form "w.x.y.z" or "w.x:y.z",<79> where w, x, y, and z MUST be numeric values indicating the version of the CA. <79> Section 3.2.1.4.3.2.1: In Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2, the format of the string is "w.x:y.z". . In Section 3.2.1.4.3.2.2, PropID = 0x00000002 (CR_PROP_PRODUCTVERSION) "CA Product Version", changed from: The client has requested the CA product version property. If the CA implements the Config_Product_Version datum, the CA SHOULD construct a [UNICODE] string of the form "w.x.y.z", where w, x, y, and z MUST be numeric values indicating the version of the server hosting the CA, which might or might not match the version of the CA returned for the previous property. The CA MAY construct a [UNICODE] string of the form "w.x:y.z". <81> <81> Section 3.2.1.4.3.2.2: In Windows Server 2003 and Windows 2000 Server, the format of the string is "w.x:y.z". Changed to: The client has requested the CA product version property. If the CA implements the Config_Product_Version datum, the CA constructs a [UNICODE] string of the form "w.x.y.z" or "w.x:y.z",<81> where w, x, y, and z MUST be numeric values indicating the version of the server hosting the CA, which might or might not match the version of the CA returned for the previous property. <81> Section 3.2.1.4.3.2.2: In Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2, the format of the string is "w.x:y.z".

Return to top of page

[MS-WCFESAN]: WCF-Based Encrypted Server Administration and Notification Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/08/12	V5.0 – 2014/05/15	In Section 2.2.5.9.2, PasswordChangeStatus, a new value, InvalidPassword, has been added for Azure Active Directory. InvalidPassword is supported in Windows Server 2012 R2 with [MSKB-2975719].

Return to top of page

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td>2015/03/16</td><td>V4.0 – 2014/05/15</td><td>
In several places in Section 2, Messages, corrected syntax values.
In Section 2.2.3, Session Header, changed the value of WDSMCTP_OP_NACK from 0x0999 to 0x09.
Changed from:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>WDSMCTP_OP_NACK
0x0999</td><td>Section 2.2.18</td></tr>
</table>
Changed to:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>WDSMCTP_OP_NACK
0x09</td><td>Section 2.2.18</td></tr>
</table>

In Section 2.2.11, LEAVE Packet, changed the values for WDSMCTP_LEAVE_REASON_COMPLETE, WDSMCTP_LEAVE_REASON_CANCELLED, and WDSMCTP_LEAVE_REASON_INACTIVE.
Changed from:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>WDSMCTP_LEAVE_REASON_COMPLETE
0x01</td><td>MUST be set when the client has fully downloaded the content and no longer requires access to the multicast session.</td></tr>
<tr><td>WDSMCTP_LEAVE_REASON_CANCELLED
0x02</td><td>MUST be set when a user or administrative action on the client caused the client to remove itself from the multicast session before it could fully download the content.</td></tr>
<tr><td>WDSMCTP_LEAVE_REASON_INACTIVE
0x03</td><td>MUST be set if the client is leaving the multicast session because the client failed to receive any packets.</td></tr>
</table>
Changed to:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>WDSMCTP_LEAVE_REASON_COMPLETE
0x00</td><td>MUST be set when the client has fully downloaded the content and no longer requires access to the multicast session.</td></tr>
<tr><td>WDSMCTP_LEAVE_REASON_CANCELLED
0x01</td><td>MUST be set when a user or administrative action on the client caused the client to remove itself from the multicast session before it could fully download the content.</td></tr>
<tr><td>WDSMCTP_LEAVE_REASON_INACTIVE
0x02</td><td>MUST be set if the client is leaving the multicast session because the client failed to receive any packets.</td></tr>
</table>

In Section 2.2.18, NACK Packet, corrected the packet diagram order and changed the RangeCount size from 8 bytes to 2 bytes.
Changed from:
<table>
<tr><td>RangeCount</td></tr>
<tr><td>…</td></tr>
<tr><td>RangeList (variable)</td></tr>
</table>
RangeCount (8 bytes): Specifies the number of ranges for ODATA packets that the client did not receive. Each range is specified using a pair of (StartSeqNo, EndSeqNo) fields.
Changed to:
<table>
<tr><td>RangeCount</td><td>RangeList (variable)</td></tr>
</table>
RangeCount (2 bytes): Specifies the number of ranges for ODATA packets that the client did not receive. Each range is specified using a pair of (StartSeqNo, EndSeqNo) fields.
</td></tr>
<tr><td>2015/03/16</td><td>V4.0 – 2014/05/15</td><td>In Section 3.1.1.4, Protocol Parameters, updated the description for the MaxJoinAckSends parameter to state that the QCR packet is received from the client instead of the server.
Changed from:
MaxJoinAckSends: Specifies the maximum number of JOINACK packets that the server MUST send when no QCR packet is received from server. The default value is 3.
Changed to:
MaxJoinAckSends: Specifies the maximum number of JOINACK</td></tr>
</table>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		packets that the server MUST send when no QCR packet is received from the client. The default value is 3.

[MS-WFDAA]: Wi-Fi Direct (WFD) Application to Application Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/11/10	V1.0 – 2014/05/15	In Section 2.2.2, AppWFDConnectionIE Message, four fields which do not appear on the wire-VendorExtensionIE, cbLength, OUI, and OUIType-have been removed, and the packet diagram has been updated accordingly. View this Word document with Track Changes turned on to see the information added to Section 2 MS-WFDAA_Section_2_2_2_Diff.
2014/10/27	V1.0 – 2014/05/15	Sections 2.2.1, AppWFDAcceptHeader Message, and 2.2.2, AppWFDConnectionIE Message, have been updated to clarify the role of the TCP transport in passing information. In Section 2.2.1, AppWFDAcceptHeader Message: Changed from: The AppWFDAcceptHeader message is sent by the client to the server after the L3 connection is established to confirm the connection. Changed to: The AppWFDAcceptHeader message is sent by the client to the server after the TCP connection is established using the port and IP information sent in the AppWFDConnectionIE message, to confirm the connection. In Section 2.2.2, AppWFDConnectionIE Message: Changed from: PortAndIPAddr (variable): This field contains the port and the IP address in the following format: Changed to: PortAndIPAddr (variable): This field contains the TCP port and the IP address in the following format:

Return to top of page

[MS-WPO]: Windows Protocols Overview

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/10/13	V4.0 – 2014/05/15	In Section 6.4, File Access Services Protocols, added the following paragraph about IOCLTs: Network captures of SMB 2 protocol traffic can include input and output control codes (IOCTLs) relating to device-specific behavior, such as failover clustering, that are not part of the SMB 2 Protocol. For more information on these IOCTLs, which are defined in the

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Windows SDK header file, winioctl.h, see [MSDN-DevInOutCtrl].

Return to top of page

[MS-WMF]: Windows Metafile Format

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/09/16	V11.1 – 2014/05/15	In Section 2.3.2.3, META_PLACEABLE Record, updated the BoundingBox description. Changed from: BoundingBox (8 bytes): The destination rectangle, measured in logical units, for displaying the metafile. The size of a logical unit is specified by the Inch field. Changed to: BoundingBox (8 bytes): The rectangle in the playback context (or simply the destination rectangle), measured in logical units, for displaying the metafile. The size of a logical unit is specified by the Inch field.

Return to top of page

[MS-WSMV]: Web Services Management Protocol Extensions for Windows Vista

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/02/02	V27.0 – 2014/05/15	Removed Section 4.5, Compression Example, because the included example request and response do not contain any compressed data, and added compression details to Section 3.2.4.1.19, Remote Shell Compression. Section 3.2.4.1.19, Remote Shell Compression, changed from: If this SOAP header is sent<134>, Web Services Management Protocol Extensions for Windows Vista clients MUST compress any data that is sent in a Send message by using the specified compression algorithm.<135> <135> Section 3.2.4.1.19: Windows Server 2003 R2 with KB968930, Windows Vista SP1 with KB968930, Windows Server 2008 with KB968930, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 clients send the CompressionType SOAP header as part of the wst:Create message with the value "xpress" to request the encoding according to the compression algorithm COMP_ALG_W2K3, specified in[MS-DRSR] section 4.1.10.6.19. Changed to: If this SOAP header is sent<134>, Web Services Management Protocol Extensions for Windows Vista clients MUST compress any data that is sent in a Send message by using the specified

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		compression algorithm.<135> <135> Section 3.2.4.1.19: Windows Server 2003 R2 with KB968930, Windows Vista SP1 with KB968930, Windows Server 2008 with KB968930, Windows 7, Windows 8, and Windows 8.1 clients send the CompressionType SOAP header as part of the wst:Create message with the value "xpress" to request the encoding according to the LZ77+Huffman compression algorithm described in [MS-XCA] section 3.2. Specifically, each stream block is encoded using the following algorithm: ▪ The first 4 bytes represent the length of the original message -1 and the length of the compressed message – 1. Anything smaller than 266 bytes is not compressed. ▪ If the size of the original data is 266 bytes or larger, the LZ77+Huffman algorithm is used to compress data. ▪ If the total compressed size is more than input data/chunk, the data is written as-is. In this case, the original message length equals the compressed message length. Likewise, such an encoded block is read as-is (instead of processing through a decompressor). ▪ The stream is compressed in 64 KB chunks; for example, a 100 KB input data is compressed in two separate blocks.
2015/02/02	V27.0 – 2014/05/15	In Section 2.2.9.1.3.1.1, HTTP Headers, corrected the Authorization description. Changed from: Authorization: Contains the credentials as defined according to the framework as specified in [RFC2616] section 14.8. Authorization = "Authorization" ":" credentials credentials = "CredSSP" auth-data2 auth-data2 = 1#(gssapi-data) where gssapi-data is the base64 encoding of the InitializeContextToken, as specified in [RFC4559] section 4.2. The client MUST include the Authorization field in the request until the Web Services Management Protocol Extensions for Windows Vista service responds with a "200 OK" response, indicating that the security context is complete. Changed to: Authorization: Contains the CredSSP messages as defined according to the CredSSP protocol specified in [MS-CSSP] section 2. Authorization = "Authorization" ":" credentials credentials = "CredSSP" auth-data2 auth-data2 = 1#(CredSSP-Protocol-Data)

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		where CredSSP-Protocol-Data is the base64 encoding of TLS-encrypted CredSSP protocol messages specified in [MS-CSSP]. A sample message exchange is described in [MS-CSSP] section 4. The client MUST include the Authorization field in the request until the Web Services Management Protocol Extensions for Windows Vista service responds with a "200 OK" response, indicating that the security context is complete.
2014/12/22	V27.0 – 2014/05/15	In Section 3.2.4.1.19, Remote Shell Compression, corrected the reference for the COMP_ALG_W2K3 compression algorithm in the product behavior note <135>. Changed from: If this SOAP header is sent<134>, Web Services Management Protocol Extensions for Windows Vista clients MUST compress any data that is sent in a Send message by using the specified compression algorithm.<135> <135> Section 3.2.4.1.19: Windows Server 2003 R2 with KB968930, Windows Vista SP1 with KB968930, Windows Server 2008 with KB968930, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 clients send the CompressionType SOAP header as part of the wst:Create message with the value "xpress" to request the encoding according to the compression algorithm "COMP_ALG_W2K3", as specified in [MS-XCA]. Changed to: If this SOAP header is sent<134>, Web Services Management Protocol Extensions for Windows Vista clients MUST compress any data that is sent in a Send message by using the specified compression algorithm.<135> <135> Section 3.2.4.1.19: Windows Server 2003 R2 with KB968930, Windows Vista SP1 with KB968930, Windows Server 2008 with KB968930, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 clients send the CompressionType SOAP header as part of the wst:Create message with the value "xpress" to request the encoding according to the compression algorithm COMP_ALG_W2K3, specified in [MS-DRSR] section 4.1.10.6.18.
2014/09/16	V27.0 – 2014/05/15	In Section 2.2.4.15, DisconnectType, changed IdleTimeout to IdleTimeOut. In Section 2.2.4.37, Shell, changed MaxIdleTimeout to MaxIdleTimeOut and changed IdleTimeout to IdleTimeOut.

Return to top of page

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

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2014/10/13	V23.0 –	In Section 5.1, Security Considerations, added the following

Errata Published YYYY/MM/DD	Protocol Document Version	Description
	2014/05/15	consideration and product behavior note: If the server allows downloading content via SSL, then additional checks are performed on the certificates to verify trust and confirm authenticity of the content.<40> <40> Section 5.1: WUA uses CAPI to perform certificate revocation checks to ensure that the certificates can be trusted. There are three revocation check methods: CRL (cert revocation list), CTL (cert trust list), and OCSP stapling. ▪ For servers that provide non-Microsoft rooted certificates, such as WSUS supporting SSL, CAPI uses the method based on the certificates it encounters. Typically it is the CRL method, where the CRL URL is embedded onto the certificates. ▪ For servers that provide Microsoft rooted certificates, CAPI uses the disallow trust list (CTL) instead of the embedded CRL URLs. See [MSKB-2813430] for more information. WUA ignores any "revocation server offline/unreachable error"; that is, CAPI cannot refresh the CRL/CTL cache. If the server is reachable, WUA will honor any revocation errors that occur and fail the operation if the certificate is found to be revoked.
2014/09/16	V23.0 – 2014/05/15	Added the following 2 new sections for the SyncPrinterCatalog() method. **2.2.2.2.9 SyncPrinterCatalog** This method is invoked to synchronize metadata describing the best matching printer drivers for the client. The syntax of this method refers to the following concepts as specified in sections 3.1.1 and 3.2.1. ▪ The integer-valued revision ID used to identify an update revision. ▪ The string-valued HardwareID, which identifies a hardware device installed on the client machine. ▪ The integer-valued deployment ID, which identifies a deployment. ▪ The prerequisite relationship between updates. ▪ The client metadata cache. <wsdl:operation name=" SyncPrinterCatalog" /> The SOAP operation is defined as follows: <soap:operation soapAction="http://www.microsoft.com/ SoftwareDistribution/Server/ClientWebService/SyncUpdates" style="document" /> Request: <s:element name=" SyncPrinterCatalog">

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		<s:complexType> <s:sequence> <s:element minOccurs="0" maxOccurs="1" name="cookie" type="tns:Cookie" /> <s:element minOccurs="0" maxOccurs="1" name="installedNonLeafUpdateIDs" type="tns:ArrayOfInt" /> <s:element minOccurs="0" maxOccurs="1" name="printerUpdateIDs" type="tns:ArrayOfInt" /> </s:sequence> </s:complexType> </s:element> **cookie:** Specifies a cookie that MUST have been obtained from a previous call to GetCookie (section 2.2.2.2.2), GetFileLocations (section 2.2.2.2.7), or SyncUpdates (section 2.2.2.2.4). This element MUST be present. **InstalledNonLeafUpdateIDs:** Contains an array of revision IDs of all nonleaf (in the prerequisite graph) revisions in the client cache that are installed on the client. These IDs MUST have been obtained from the UpdateInfo.ID returned from a previous call to SyncUpdates. **printerUpdateIDs:** Contains an array of revision IDs of all printer driver updates in the client cache. These IDs MUST have been obtained from the UpdateInfo.ID returned from a previous call to SyncUpdates. Response: <s:element name=" SyncPrinterCatalogResponse"> <s:complexType> <s:sequence> <s:element minOccurs="0" maxOccurs="1" name="SyncPrinterCatalogResult" type="s1:SyncInfo" /> </s:sequence> </s:complexType> </s:element> **SyncUpdatesResult:** Upon successful completion of this operation, this element MUST be returned. The client SHOULD interpret this result, as specified in section 3.1.5.7. The format is the same as the one defined in the Response section of 2.2.2.2.4. **3.1.5.12 SyncPrinterCatalog** **Synopsis**: This call supports the synchronrverization of printer driver updatemetadata to client computers and is used only by the Add Printer Wizard on the client.

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		Request Validation: **Parameter** / **Validation conditions** / **Error code** cookie / MUST be a valid cookie, issued by this server, that has not expired. / InvalidCookie, ServerChanged, or CookieExpired **Data Processing**: The data processing specified in this section references most of the elements of the abstract data model, as specified in section 3.1.1. The server MUST check whether the configuration data returned from GetConfig (section 3.1.5.2) has changed since the last time the client synchronized and, if so, throw a ConfigChanged ErrorCode fault. The server SHOULD check whether client registration is required but the client is not yet registered. If so, it SHOULD throw a RegistrationRequired ErrorCode. The next step is for the server to compute the NeededRevisions list for the client. The server MUST do so as follows: 1. Restrict the set of revisions to those that are deployed to the client computer's target group, combined with any dependencies (prerequisite or bundle) of such updates. 2. Restrict the resulting set further to those revisions whose prerequisites are satisfied by the updates whose revision IDs are specified in Parameters.InstalledNonLeafUpdateIDs. 3. Restrict the resulting set further to either: ▪ Revisions for which all the following conditions hold: ▪ UpdateType = Driver ▪ DriverClass = Printer ▪ If there is already a driver installed on the device: ▪ The revision has an entry in the driver table that MUST be a "better" match than the installed driver. ▪ The revision MUST have an entry in the driver table that matches the Provider and Manufacturer for the installed driver. Next, the server MUST generate the list of CachedRevisions by accepting the revisions listed in Parameters.CachedDriverIDs. **Results**: If no faults occur during the operation, the server MUST return a SyncPrinterCatalogResponse message to the client. It MUST generate the response as follows: ▪ **SyncPrinterCatalogResponse.NewUpdates**: Populated with entries for revision in the NeededRevisions list that are

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		not in the CachedRevisions list: **ID:** The revision ID. **Deployment:** Information about the deployment to this revision. If this revision was not explicitly deployed to the client by an administrator (for example, it was included in the NeededRevisions list because it was a dependency of an explicitly deployed revision), the DeploymentAction MUST be set to "Evaluate". For driver updates (UpdateType = driver), when the client reports a protocolVersion of "1.6" or higher in the GetCookie call, the server SHOULD include all the HardwareIDs that are selected as "best" matches associated with this revision from the driver table.<28> **IsLeaf:** Specifies whether the revision is a leaf on the prerequisite graph. That is, no entries in the abstract data model prerequisite table (as specified in section 3.1.1) have this revision's UpdateID specified as a PrerequisiteUpdateID. **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. ▪ **SyncPrinterCatalogResponse.ChangedUpdates**: Populated with entries for revisions in the NeededRevisions list that are also in the CachedRevisions list, but for which Deployment or IsLeaf data has changed since the last time the client synchronized with the server. The fields of these entries are populated according to the server's abstract data model (as specified in section 3.1.1) as follows: **Deployment:** The entry in the deployment table that specifies how the revision is deployed to the client's target group. **IsLeaf:** The entry in the Revision table that specifies whether the revision is a leaf in the prerequisite graph. ▪ **SyncPrinterCatalogResponse.Truncated**: The server MAY choose to return a subset of the updates that would normally be returned in the <NewUpdates> collection to reduce the processing overhead incurred by a single call to the server. In such cases, the server MUST set Truncated = TRUE.<29>

Errata Published YYYY/MM/DD	Protocol Document Version	Description
		▪ **SyncPrinterCatalogResponse.NewCookie**: The server MUST return a new cookie for the client to use on subsequent SyncUpdates calls. The server SHOULD update the cookie with the highest (most recent) LastChangeTime stored in the deployment table. This allows the server to determine, on future calls to SyncUpdates, whether a revision that stays in scope for the client needs to have its deployment returned in the ChangedUpdates list; if the current deployment's LastChangeTime is less than the value stored in this cookie, then the deployment need not be returned in the ChannelUpdates list, because the deployment data will already be cached on the client. <28>WSUS 3.0 Service Pack 1 includes the HardwareIDs when the deployment is associated with a driver update and the protocolVersion reported by the client in the GetCookie call is "1.6". <29>WSUS truncates responses at 200 NewUpdates.

Return to top of page

Office Protocols Errata

No errata are available for the Office Protocols documents.

SharePoint Protocols Errata

No errata are available for the SharePoint Protocols documents.

Exchange Server Protocols Errata

No errata are available for the Exchange Server Protocols documents.

SQL Server Protocols Errata

This topic lists the Errata found in the SQL Server 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.

[MS-DTSX2]: Data Transformation Services Package XML Version 2 File Format

[MS-ODATAJSON]: OData Protocol JSON Format Standards Support Document

[MS-SSAS]: SQL Server Analysis Services Protocol

[MS-DTSX2]: Data Transformation Services Package XML Version 2 File Format

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td>2014/12/22</td><td>V2.0 – 2014/05/20</td><td>In Section 2.4, ExecutableTypePackage, added a description of the PackageParameters element to the table as follows:
<table>
<tr><th>Element</th><th>Constraints</th><th>Type definition</th><th>Description</th></tr>
<tr><td>PackageParameters</td><td>None</td><td>PackageParametersType</td><td>Specifies a collection of elements of type PackageParameterType. Each such element defines a package parameter that is available to the package.</td></tr>
</table></td></tr>
<tr><td>2014/12/22</td><td>V2.0 – 2014/05/20</td><td>In Section 2.4.4.1.2.1.1, ConnectionManagerConnectionManagerAttributeGroup, corrected the description of the UseEncryption attribute as follows:
Changed from:
A value of type DTS:BooleanStringCap that specifies whether to use encryption for data access.
"True" specifies not to use encryption for data access.
"False" specifies to use encryption for data access.
Changed to:
A value of type DTS:BooleanStringCap that specifies whether to use encryption</td></tr>
</table>

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>for data access.
"True" specifies to use encryption for data access.
"False" specifies not to use encryption for data access.</td></tr>
<tr><td>2014/12/22</td><td>V2.0 – 2014/05/20</td><td>In Section 2.4.4.1.2.1.5.2, HttpConnectionAttributeGroup, added a description of the Certificate enumeration value to the table as follows:
<table>
<tr><th>Enumeration value</th><th>Description</th></tr>
<tr><td>Certificate</td><td>A string that specifies the client certificate to the HTTP server in base64.</td></tr>
</table></td></tr>
<tr><td>2014/12/22</td><td>V2.0 – 2014/05/20</td><td>In Section 2.7.1.1.1.1.5, PipelineComponentComponentClassIDEnum, updated the schema for SQL Server Compact Destination and for Raw File Destination.
Changed from:
<pre><!--Script Component, ADO.Net source, XML Source,ADO.Net Destination, DataReader Destination, SQL ServerCompact Destination--></pre>Changed to:
<pre><!--Script Component, ADO.Net source, XML Source,ADO.Net Destination, DataReader Destination, SQL Server Compact Destination--></pre>Changed from:
<pre><!--Raw File--></pre>Changed to:
<pre><!--Raw File Destination--></pre>In the table in this section, added the Raw File Destination Component enumeration value and description, and removed individual row enumeration value entries for ADO.Net Destination Component and SQL Server Compact Edition Destination Component.
Note ADO.Net Destination Component and SQL Server Compact Destination Component remain in the description for enumeration value {874F7595-FB5F-40FF-96AF-FBFF8250E3EF}.
<table>
<tr><th>Enumeration value</th><th>Description</th></tr>
<tr><td>{874F7595-FB5F-40FF-96AF-FBFF8250E3EF}<60></td><td>▪ Script Component
▪ ADO.Net Source Component</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td>▪ XML Source Component
▪ ADO.Net Destination Component
▪ DataReader Destination Component
▪ SQL Server Compact Destination Component</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>{04762BB6-892F-4EE6-AD46-9CEB0A7EC7A2}</td><td>Raw File Destination Component</td></tr>
</table>
</td></tr>
<tr><td>2014/12/22</td><td>V2.0 – 2014/05/20</td><td>In Section 2.7.1.1.2.1, PipelinePathType, added the id attribute to the schema as follows:

<xs:attribute name="id" type="xs:int" form="unqualified"/></td></tr>
<tr><td>2014/12/22</td><td>V2.0 – 2014/05/20</td><td>In Sections 2.7.1.11.1.1.1, SqlTaskDataType, and 5.8, SQLTask XSD, removed the TSQLExecuteTaskAttributeGroup attribute group from the schema.</td></tr>
<tr><td>2014/12/22</td><td>V2.0 – 2014/05/20</td><td>In Section 2.7.1.11.1.1.1.10.5, BackupCompressionActionEnum, updated the schema.
Changed from:

<xs:simpleType name="BackupActionForExistingBackupsEnum">
<xs:restriction base="xs:int">
<xs:minInclusive value="0"/>
<xs:maxInclusive value="1"/>
</xs:restriction>
</xs:simpleType>

Changed to:

<xs:simpleType name="BackupCompressionActionEnum">
<xs:restriction base="xs:int">
<xs:minInclusive value="0"/>
<xs:maxInclusive value="2"/>
</xs:restriction>
</xs:simpleType></td></tr>
<tr><td>2014/12/22</td><td>V2.0 – 2014/05/20</td><td>In Section 2.7.1.26.1, XMLTaskOperationTypeEnum, updated the simpleType name in the schema.
Changed from:

<xs:simpleType name="XMLTaskDataOperationTypeEnum"></td></tr>
</table>

<table>
<tr><th>Errata Published YYYY/MM/DD</th><th>Protocol Document Version</th><th>Description</th></tr>
<tr><td></td><td></td><td>Changed to:

<xs:simpleType name="XMLTaskOperationTypeEnum"></td></tr>
<tr><td>2014/12/22</td><td>V2.0 – 2014/05/20</td><td>In Section 2.9.2, BaseExecutablePropertyAttributeGroup, added a description of the ExecValueType attribute to the table as follows:
<table><tr><th>Attribute</th><th>Description</th></tr><tr><td>ExecValueType</td><td>A value of type xs:DtsDataTypeEnum that specifies the type of ExecValue if the value of the ForceExecValue attribute is "True".</td></tr></table></td></tr>
</table>

Return to top of page

[MS-ODATAJSON]: OData Protocol JSON Format Standards Support Document

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/03/30	V2.0 – 2014/05/20	In Section 1.3, Microsoft Implementations, the following products were added: Microsoft SharePoint Foundation 2013 Service Pack 1 (SP1) Microsoft SharePoint Server 2013 Service Pack 1 (SP1)

[MS-SSAS]: SQL Server Analysis Services Protocol

Errata Published YYYY/MM/DD	Protocol Document Version	Description
2015/03/30	V16.0 – 2014/05/20	After Section 2.2.4.2.2.17, Annotation, the following three new sections and related Product Behavior Notes were added: **2.2.4.2.2.17.1 LinguisticSchemas Annotation** The **LinguisticSchemas** annotation<80> is a predefined **Annotation** that is available on the **Database** object with Name = LinguisticSchemas. The value of this annotation is the element **LinguisticSchemas** with the following complex type. <xsd:complexType name="LinguisticSchemas"> <xsd:sequence> <xsd:element name="LinguisticSchema" type="LinguisticSchema" minOccurs="1" /> </xsd:sequence> </xsd:complexType>

Errata Published YYYY/MM/DD	Protocol Document Version	Description

Element	Read-Only	Default value	Description
LinguisticSchemas		Empty	A collection of **LinguisticSchema** objects.

The following is an example of a **LinguisticSchemas** annotation.

```
  <Annotation>
    <Name>LinguisticSchemas</Name>
    <Value>
      <LinguisticSchemas xmlns="">
        <LinguisticSchema Language="en-US"
xmlns="http://schemas.microsoft.com
/sqlserver/2014/01/linguisticschema">
          <Entities>
            <Entity Name="my_table"
EdmEntitySet="Sandbox.Table1">
              <Words>
                <Word>my table</Word>
              </Words>
            </Entity>
            <Entity Name="my_table.A"
EdmEntitySet="Sandbox.Table1" EdmProperty="A">
              <Words>
                <Word>A</Word>
                <Word>my table a</Word>
              </Words>
            </Entity>
            <Entity Name="my_table.B"
EdmEntitySet="Sandbox.Table1" EdmProperty="B">
              <Words>
                <Word>B</Word>
              </Words>
            </Entity>
            <Entity Name="my_table.XL_row_number"
EdmEntitySet="Sandbox.Table1"
EdmProperty="v__XL_RowNumber">
              <Words>
                <Word>XL row number</Word>
              </Words>
            </Entity>
          </Entities>
        </LinguisticSchema>
      </LinguisticSchemas>
    </Value>
  </Annotation>
```

2.2.4.2.2.17.1.1 LinguisticSchema

This complex type represents a **LinguisticSchema**.<81>

Errata Published YYYY/MM/DD	Protocol Document Version	Description

```
  <xsd:complexType name="LinguisticSchema">
    <xsd:sequence>
      <xsd:element name="Entities" type="EntitiesType"
minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="Language" type="xsd:language"
use="required" />
  </xsd:complexType>

  <xsd:complexType name="EntitiesType">
    <xsd:sequence>
      <xsd:element name="Entity" type="EntityType"
minOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
```

Element	Read-Only	Default value	Description
Entities		Empty	A collection of **Entity** objects.

Attribute	Description
Language	Language code for the language of the linguistic schema. The language MUST comply with [HTML] section 8.1.1.

2.2.4.2.2.17.1.1.1 Entity

This complex type represents an **Entity**.<82>

```
  <xsd:complexType name="EntityType">
    <xsd:sequence>
      <xsd:element name="Words" type="WordsType"
minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="Name" type="Name" use="required"
/>
    <xsd:attribute name=EdmEntitySet"
type="EdmQualifiedName" use="required" />
    <xsd:attribute name="EdmProperty"
type="EdmSimpleName" use=optional" />
  </xsd:complexType>

  <xsd:complexType name="WordsType">
    <xsd:sequence>
      <xsd:element name="Word" type="xsd:token"
minOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
```

Errata Published YYYY/MM/DD	Protocol Document Version	Description

Element	Read-Only	Default value	Description
Words		Empty	A collection of string objects. Each **Word** represents a term that can be used to refer to the **Entity**.

Attribute	Description
Name	Name of the entity.
EdmEntitySet	Name of the **EdmEntitySet** that represents the entity or contains the **EdmProperty** that represents the entity.
EdmProperty	Name of the **EdmProperty** that represents the entity.

<80> Section 2.2.4.2.2.17.1: The **LinguisticSchemas** annotation is not supported by SQL Server.

<81> Section 2.2.4.2.2.17.1.1: The **LinguisticSchema** complex type is not supported by SQL Server.

<82> Section 2.2.4.2.2.17.1.1.1: The **EntityType** complex type is not supported by SQL Server.