[MS-TSRAP-Diff]:

Telnet Server Remote Administration Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
4/8/2008	1.0	New	Version 1.0 release
5/16/2008	1.0.1	Editorial	Changed language and formatting in the technical content.
6/20/2008	1.0.2	Editorial	Changed language and formatting in the technical content.
7/25/2008	1.0.3	Editorial	Changed language and formatting in the technical content.
8/29/2008	1.0.4	Editorial	Changed language and formatting in the technical content.
10/24/2008	1.0.5	Editorial	Changed language and formatting in the technical content.
12/5/2008	1.1	Minor	Clarified the meaning of the technical content.
1/16/2009	2.0	Major	Updated and revised the technical content.
2/27/2009	2.0.1	Editorial	Changed language and formatting in the technical content.
4/10/2009	2.0.2	Editorial	Changed language and formatting in the technical content.
5/22/2009	3.0	Major	Updated and revised the technical content.
7/2/2009	3.0.1	Editorial	Changed language and formatting in the technical content.
8/14/2009	3.0.2	Editorial	Changed language and formatting in the technical content.
9/25/2009	3.1	Minor	Clarified the meaning of the technical content.
11/6/2009	3.1.1	Editorial	Changed language and formatting in the technical content.
12/18/2009	4.0	Major	Updated and revised the technical content.
1/29/2010	4.0.1	Editorial	Changed language and formatting in the technical content.
3/12/2010	4.0.2	Editorial	Changed language and formatting in the technical content.
4/23/2010	5.0	Major	Updated and revised the technical content.
6/4/2010	5.0.1	Editorial	Changed language and formatting in the technical content.
7/16/2010	5.0.1	None	No changes to the meaning, language, or formatting of the technical content.
8/27/2010	5.0.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2010	5.0.1	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	5.0.1	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	5.0.1	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	5.0.1	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	5.0.1	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
5/6/2011	5.0.1	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	5.1	Minor	Clarified the meaning of the technical content.
9/23/2011	5.1	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	6.0	Major	Updated and revised the technical content.
3/30/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	6.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	7.0	Major	Updated and revised the technical content.
11/14/2013	7.0	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	7.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	7.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	7.0	~~No Change~~None	No changes to the meaning, language, or formatting of the technical content.
10/16/2015	7.0	~~No Change~~None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the Telnet Server Remote Administration Protocol. Telnet Server Remote Administration Protocol provides a [MS-DCOM] interface used for performing management tasks on telnet server. Telnet Server Remote Administration Protocol specifies an interface that:

- Get information regarding all the telnet sessions handled by telnet server at any given instance.
- Send message to a session.
- Terminate a session.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative ~~and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms~~. All other sections and examples in this specification are informative.

1.1 Glossary

~~The~~This document uses the following terms ~~are specific to this document~~:

activation: In the DCOM protocol, a mechanism by which a client provides the **CLSID** of an object class (4) and obtains an object (4), either from that object class or a class factory that is able to create such objects. For more information, see [MS-DCOM].

authentication level: A numeric value indicating the level of authentication or message protection that **remote procedure call (RPC)** will apply to a specific message exchange. For more information, see [C706] section 13.1.2.1 and [MS-RPCE].

class identifier (CLSID): A **GUID** that identifies a software component; for instance, a DCOM object class (4) or a COM class.

globally unique identifier (GUID): A term used interchangeably with **universally unique identifier (UUID)** in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also **universally unique identifier (UUID)**.

Interface Definition Language (IDL): The International Standards Organization (ISO) standard language for specifying the interface for remote procedure calls. For more information, see [C706] section 4.

Network Data Representation (NDR): A specification that defines a mapping from **Interface Definition Language (IDL)** data types onto octet streams. **NDR** also refers to the runtime environment that implements the mapping facilities (for example, data provided to **NDR**). For more information, see [MS-RPCE] and [C706] section 14.

remote procedure call (RPC): A context-dependent term commonly overloaded with three meanings. Note that much of the industry literature concerning RPC technologies uses this term interchangeably for any of the three meanings. Following are the three definitions: (*) The runtime environment providing remote procedure call facilities. The preferred usage for this meaning is "RPC runtime". (*) The pattern of request and response message exchange between two parties (typically, a client and a server). The preferred usage for this meaning is "RPC exchange". (*) A single message from an exchange as defined in the previous definition. The preferred usage for this term is "RPC message". For more information about RPC, see [C706].

telnet server: An implementation of the server side of Telnet Protocol [RFC854].

telnet session: An active telnet connection between a telnet client and a telnet server.

universally unique identifier (UUID): A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager entry-point vectors, and **RPC** objects. UUIDs are highly likely to be unique. UUIDs are also known as **globally unique identifiers (GUIDs)** and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the UUID.

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

1.2 References

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

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, https://www2.opengroup.org/ogsys/catalog/c706

[MS-DCOM] Microsoft Corporation, "Distributed Component Object Model (DCOM) Remote Protocol".

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-OAUT] Microsoft Corporation, "OLE Automation Protocol".

[MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions".

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

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

1.2.2 Informative References

None.

1.3 Overview

The Telnet Server Remote Administration Protocol is a Distributed Component Object Model (DCOM) Protocol [MS-DCOM] interface that is exposed by a DCOM server and consumed by a DCOM client. A client uses the Telnet Server Remote Administration Protocol by invoking DCOM method calls on the interface exposed by the DCOM server that implements the protocol.

Telnet Server Remote Administration Protocol is a stateless protocol. An implementation can call any of the methods any number of times and in any order. Each call to a method in the DCOM/COM interface is independent of any other call to the same or different method.

1.4 Relationship to Other Protocols

This protocol depends on the DCOM Remote Protocol, as specified in [MS-DCOM]. The DCOM Remote Protocol implementation ~~MUST~~is required to provide and ~~MUST~~ use all underlying protocols, as specified in [MS-RPCE], [MS-DCOM], and [C706].

1.5 Prerequisites/Preconditions

The client using the protocol is required to have available valid credentials recognized by the server accepting the client requests. The client is required to use security providers that recognize such credentials to authenticate to the remote server by using SSPI supported by the Remote Procedure Call Protocol.

The server system is required to start the DCOM Remote Protocol. The DCOM **activation** service is required to be fully initialized before the activation request. See section 1.3.1 of [MS-DCOM].

1.6 Applicability Statement

The Telnet Remote Server Administration Protocol is designed for administering a **telnet server** on remote clients and servers.

1.7 Versioning and Capability Negotiation

The Telnet Server Remote Administration protocol does not support negotiation of the interface version to use.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

There are no standards assignments for this protocol. This protocol uses the following **CLSIDs** (as specified in [MS-DCOM] section 1.9):

CLSID_EnumTelnetClientsSvr = ({FE9E48A4-A014-11D1-855C-00A0C944138C})

The following **GUID** is used for the interface:

IID_IManageTelnetSessions= ({034634FD-BA3F-11D1-856A-00A0C944138C});

2 Messages

2.1 Transport

Message transport in the Telnet Server Remote Administration protocol uses the Distributed Component Object Model (DCOM) protocol [MS-DCOM], which uses **RPC** [C706] as its transport.

2.2 Common Data Types

In addition to the RPC base types and definitions specified in [C706] and [MS-DTYP], additional data types are defined in the following sections.

2.2.1 PSZSESSIONDATA

pszSessionData is a string field with the below syntax (in ABNF representation, as specified in [RFC4234]).

```
Start-rule = NumberofSessions SEP1 *(SessionInformation SEP1)

NumberofSessions = 1*UNICODEDIGIT ;

SEP1 = NULL ","; comma

NULL = ""; null
UNICODECHAR =
(%x01-FF %x00-FF) / (NULL (%x00-2B / %x2D-5B / %x5D-FF))   ;Unicode            character
other than comma and back slash

UNICODEDIGIT = NULL %x30-39; Unicode digit

SessionInformation =
ID SEP2 Userdomain SEP2 username SEP2 computername SEP2 year SEP2 month   SEP2 dayofweek SEP2
day SEP2 hour SEP2 minute SEP2 second SEP2 milliseconds SEP2 idletime SEP2

SEP2 = NULL "\"; back slash

ID = 1*UNICODEDIGIT;

Userdomain = *UNICODECHAR;

username = *UNICODECHAR ;

computername = *UNICODECHAR ;

year = 4*5UNICODECHAR ;

month = 1*2UNICODECHAR ;

dayofweek = 1*2UNICODECHAR ;

day = 1*2UNICODECHAR ;

hour = 1*2UNICODECHAR ;

minute = 1*2UNICODECHAR ;

second = 1*2UNICODECHAR ;

milliseconds = 1*3UNICODECHAR ;

idletime = 1*UNICODECHAR ;
```

NumberofSessions: A string that specifies the number of current active telnet sessions on the server.

Userdomain: A string that specifies the domain of which the user that established the telnet session is a member.

UserName: A string that specifies the user name of the user that established the telnet session.

Computername: A string that specifies the name of the client computer.

Year: A string that specifies the year component of time at which the telnet session was established. The valid values for this field are 1601 through 30827.

Month: A string that specifies the month component of time at which the telnet session was established. The valid values for this filed are as below:

Value	Meaning
1	January
2	February
3	March
4	April
5	May
6	June
7	July
8	August
9	September
10	October
11	November
12	December

Dayofweek: A string that specifies the day of week component of time at which the telnet session was established. The valid values for this field are as below:

Value	Meaning
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

Day: A string that specifies the day component of time at which the telnet session was established. The valid values for this field are 1 through 31.

Hour: A string that specifies the hour component of time at which the telnet session was established. The valid values for this field are 0 through 23.

Minute: A string that specifies the minute component of time at which the telnet session was established. The valid values for this field are 0 through 59.

Second: A string that specifies the second component of time at which the telnet session was established. The valid values for this field are 0 through 59.

Milliseconds: A string that specifies the millisecond component of time at which the telnet session was established. The valid values for this field are 0 through 999.

Idletime: A string that specifies the idle time (represented in seconds). Idle time is the time for which there has been no exchange of any communication between telnet client and telnet server.

3 Protocol Details

The client side of this protocol is simply a pass-through. No additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

3.1 Client and Server Details

A client in the context of this specification is a machine issuing a Telnet Server Remote Administration Protocol request. The request is issued against a Telnet Server Remote Administration Protocol server. In this context, a server is a machine handling the request issued by the client.

This protocol MUST instruct the RPC runtime to perform a strict **NDR** data consistency check at target level 5.0, as specified in section 2.2.5.3.3.1 of [MS-RPCE].

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation (server side) maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with what is described in this document.

The following variables should be maintained by the telnet server for each active telnet session, and the Telnet Server Remote Administration Protocol server should be able to fetch these from the telnet server.

ID: An integer identifier that uniquely identifies a telnet session. Telnet Server Remote Administration Protocol uses the ID to uniquely identify a session.

TimeOfLogon: Stores the time at which the telnet session was established.

IdleTime: Stores the time for which there has been no user activity in the telnet session.

3.1.2 Timers

None.

3.1.3 Initialization

The client MUST instantiate an object using CLSID_EnumTelnetClientsSvr on the server machine using DCOM Remote Protocol activation. The client then MUST initialize interface IID_IManageTelnetSessions on the server machine using DCOM Remote Protocol activation.

3.1.4 Message Processing Events and Sequencing Rules

On each interface, the server MUST support multiple outstanding calls. This protocol does not enforce an upper limit to the number of parallel invocations or outstanding calls that the server must support. Changed name to that of parent section.

The IManageTelnetSessions interface MUST be uniquely identified by **UUID** 034634FD-BA3F-11D1-856A-00A0C944138C.

This interface includes the following methods:

Methods in RPC Opnum Order

Method	Description
GetTelnetSessions	Opnum: 7
TerminateSession	Opnum: 8
SendMsgToASession	Opnum: 9

Opnums 0, 1, and 2 are reserved for the IDispatch interface.

All methods MUST NOT throw exceptions.

3.1.4.1 GetTelnetSessions (Opnum 7)

The GetTelnetSessions method is used to query the telnet server for information about all active **telnet sessions**.

```
HRESULT GetTelnetSessions(
  [out, retval] BSTR* pszSessionData
);
```

pszSessionData: A string pointer to PSZSESSIONDATA string that contains information about telnet sessions in the server. <1>

The server must fill various fields of PSZSESSIONDATA as below.

NumberofSessions: The server MUST set the value for this field to the number of current active telnet session in Telnet Server. The server MUST ensure that the value of this field matches the number of instances of SessionInformation strings.

SessionInformation: The server MUST have one SessionInformation string per active session and the number of SessionInformation strings MUST be the same as the value of NumberofSessions field.

ID: ID of the session. The server must fill this filed with the unique identifier of the session. The server can reuse unique identifiers assigned to a session, but the server MUST ensure that at any given point in time only one telnet session exists with a particular ID. Refer to Section 3.1.1 for an abstract data model that the server can maintain.

Userdomain: The server MUST set the value for this field to the domain of which the user that established the telnet session is a member. If the user account is not a member of any domain then the server MUST fill this field with the computer name of the server.

UserName: The server MUST set the value for this field to the user name of the user that established the telnet session.

Computername: The server MUST set the value for this field to either the IPv4 or IPv6 address of the client. <2>

Year: The server MUST determine the value for this field from the time of logon expressed in Coordinated Universal Time (UTC).

Month: The server MUST determine the value for this field from the time of logon expressed in Coordinated Universal Time (UTC).

Dayofweek: The server MUST determine the value for this field from the time of logon expressed in Coordinated Universal Time (UTC).

Day: The server MUST determine the value for this field from the time of logon expressed in Coordinated Universal Time (UTC).

Hour: The server MUST determine the value for this field from the time of logon expressed in Coordinated Universal Time (UTC).

Minute: The server MUST determine the value for this field from the time of logon expressed in Coordinated Universal Time (UTC).

Second: The server MUST determine the value for this field from the time of logon expressed in Coordinated Universal Time (UTC).

Milliseconds: The server MUST determine the value for this field from the time of logon expressed in Coordinated Universal Time (UTC).

Idletime: The server MUST set the value of this field to the time for which there has been no exchange of any communication between telnet client and server.

Return Values: The server MUST return zero if the method is successful. The server MUST return 0x01 if processing fails and set output parameters to NULL. These are in addition to the values that can be returned by the underlying [MS-DCOM] implementation.

Exceptions Thrown: No exceptions are thrown beyond those thrown by the underlying DCOM protocol [MS-DCOM].

3.1.4.2 TerminateSession (Opnum 8)

The TerminateSession method terminates a telnet session.

```
HRESULT TerminateSession(
  [in] DWORD dwUniqueId
);
```

dwUniqueId: The ID of the session. The ID of a session can be obtained by calling the GetTelnetSessions method or can be user provided. The server MUST ensure that at any given point in time only one telnet session exists with a particular ID. Refer to Section 3.1.1 for an abstract data model that the server can maintain.

Return Values: The server MUST return zero if the method is successful. The server MUST return 0x01 if processing fails. These are in addition to the values that can be returned by the underlying [MS-DCOM] implementation.

Exceptions Thrown: No exceptions are thrown beyond those thrown by the underlying DCOM protocol [MS-DCOM].

3.1.4.3 SendMsgToASession (Opnum 9)

The SendMsgToASession method directs the telnet server to send a text message to the telnet client that initiated the session.

```
HRESULT SendMsgToASession(
  [in] DWORD dwUniqueId,
  [in] BSTR szMsg
);
```

dwUniqueId: The ID of the session. The ID of a session can be obtained using the GetTelnetSessions method or can be user provided. The server MUST ensure that at any given point in time only one

telnet session exists with a particular ID. Refer to Section 3.1.1 for an abstract data model that the server can maintain.

szMsg: The string text that has to be sent.

Return Values: The server MUST return zero if the method is successful. The server MUST return 0x01 if processing fails. These are in addition to the values that can be returned by the underlying [MS-DCOM] implementation.

Exceptions Thrown: No exceptions are thrown beyond those thrown by the underlying DCOM protocol [MS-DCOM].

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

1. Get telnet session information.

    1. Client calls GetTelnetSessions.

    2. Server returns *pszSessionData* filled with session information.

```
Netmon capture of pszSessionData:

00 31 00 2C 00 34 00 32 00 30 00 5C 00 43 00 4F 00 4E 00 54 00 4F 00 53 00 4F 00 5C 00 41 00
64 00 6D 00 69 00 6E 00 69 00 73 00 74 00 72 00 61 00 74 00 6F 00 72 00 5C 00 3A 00 3A 00 66
00 66 00 66 00 66 00 3A 00 31 00 39 00 32 00 2E 00 31 00 36 00 38 00 2E 00 30 00 2E 00 31 00
30 00 31 00 5C 00 32 00 30 00 30 00 38 00 5C 00 31 00 31 00 5C 00 33 00 5C 00 31 00 32 00 5C
00 39 00 5C 00 33 00 37 00 5C 00 39 00 5C 00 34 00 38 00 32 00 5C 00 31 00 31 00 36 00 5C 00
2C 00

PSZSessionData string:
1,420\CONTOSO\Administrator\::ffff:1921680101\2008\11\3\12\9\37\9\482\116\,

where
NumberofSessions = 1
SessionInformation = 420\CONTOSO\Administrator\::ffff:1921680101\2008\11\3\12\9\37\9\482\116\
Userdomain= CONTOSO
UserName = Administrator
Computername = ::ffff:1921680101
Year= 2008
Month = 11
Dayofweek = 3
Day = 12
Hour = 9
Minute = 37
Second = 9
Milliseconds = 482
Idletime = 116
```

2. Terminate a telnet session.

    1. The client calls the TerminateSession with *dwUniqueId* set to the ID of the session to be terminated. ID can either be supplied by the user to Client or client uses an ID obtained in example 1.

    2. The server terminates the session identified by *dwUniqueId* and returns zero.

3. Send a message to a session.

    1. The client calls SendMsgToASession with *dwUniqueId* and *szMsz*. ID can either be supplied by the user to Client or client uses an ID obtained in example 1.

        Sample input passed by client:

```
dwUniqueId = 101
szMsg = ""test"
```

    2. The server sends the message "test" to the telnet session with ID 101 and returns zero.

5 Security

5.1 Security Considerations for Implementers

For all methods, the server is required to evaluate the **authentication level** and the security principal rights to invoke that method, and the server is required to fail the operation if the security requirements are not met.<3>

5.2 Index of Security Parameters

None.

6 Appendix A: Full IDL

For ease of implementation the full **IDL** is provided below, where "ms-oaut.idl" refers to the IDL found in [MS-OAUT] Appendix A.

```
import "ms-oaut.idl";

[
object,
uuid(034634FD-BA3F-11D1-856A-00A0C944138C),
dual,
pointer_default(unique)
]

interface IManageTelnetSessions : IDispatch
{
HRESULT GetTelnetSessions( [ out, retval ] BSTR *pszSessionData );
HRESULT TerminateSession([in] DWORD dwUniqueId );
HRESULT SendMsgToASession([in] DWORD dwUniqueId, [in] BSTR szMsg );
};
```

7 Appendix B: Product Behavior

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

- Windows XP operating system
- Windows Server 2003 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
- Windows Server 2012 R2 operating system

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

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

<1> Section 3.1.4.1: The size of pszSessionData returned by Windows Telnet Server is more than what a client will expect based on the ABNF specification. A client must ignore characters that ~~are after~~follow the part that can be interpreted based on the ABNF representation for pszSessionData. The additional data is spurious and has no meaning.

<2> Section 3.1.4.1: Windows XP and Windows Server 2003 set the ComputerName field to IPV4 address. Windows Vista, Windows Server 2008, Windows 7, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 set the ComputerName field to the IPV6 address of the client.

<3> Section 5.1: Windows telnet server enforces that the client provides identity that has administrative privileges in the server.

8 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

9 Index

T

V