

[MS-RASA]: Remote Access Server Advertisement (RASADV) Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

- This protocol 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 make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.
- Microsoft does not claim any trade secret rights in this documentation.
- Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. If you are interested in obtaining a patent license, please contact protocol@microsoft.com.
- The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

This protocol documentation is intended for use in conjunction with publicly available standard specifications, network programming art, and Microsoft Windows distributed systems concepts, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

A protocol specification 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.

Revision Summary

Date	Revision History	Revision Class	Comments
07/20/2007	0.1	Major	MCCP Milestone 5 Initial Availability
09/28/2007	0.1.1	Editorial	Revised and edited the technical content.
10/23/2007	0.1.2	Editorial	Revised and edited the technical content.
11/30/2007	0.1.3	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
01/25/2008	0.1.4	Editorial	Revised and edited the technical content.

Table of Contents

1	Introduction	4
1.1	Glossary	4
1.2	References	4
1.2.1	Normative References	4
1.2.2	Informative References.....	4
1.3	Protocol Overview (Synopsis).....	4
1.4	Relationship to Other Protocols.....	5
1.5	Prerequisites/Preconditions	5
1.6	Applicability Statement	5
1.7	Versioning and Capability Negotiation.....	5
1.8	Vendor-Extensible Fields	5
1.9	Standards Assignments.....	5
2	Messages	6
2.1	Transport	6
2.2	Message Syntax	6
2.2.1	Server Advertisement.....	6
3	Protocol Details	7
3.1	RAS Server Details	7
3.1.1	Abstract Data Model	7
3.1.2	Timers	7
3.1.3	Initialization	7
3.1.4	Higher-Layer Triggered Events.....	7
3.1.5	Message Processing Events and Sequencing Rules	7
3.1.6	Timer Events.....	7
3.1.7	Other Local Events	7
3.2	Listener Details	7
3.2.1	Abstract Data Model	7
3.2.2	Timers	7
3.2.3	Initialization	8
3.2.4	Higher-Layer Triggered Events.....	8
3.2.5	Message Processing Events and Sequencing Rules	8
3.2.6	Timer Events.....	8
3.2.7	Other Local Events	8
4	Protocol Examples	9
5	Security	10
5.1	Security Considerations for Implementers	10
5.2	Index of Security Parameters	10
6	Appendix A: Windows Behavior	11
7	Index.....	12

1 Introduction

This document specifies the Remote Access Server Advertisement (RASADV) Protocol, by which **Remote Access Service (RAS) servers** advertise their presence within a local network, which allows network administrators to detect nonmalicious configuration and deployment of gateways providing external access to their network.

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

Domain
Remote Access Service (RAS) Server

The following terms are specific to this document:

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

1.2 References

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. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[IANAIMA] Internet Assigned Numbers Authority, "Internet Multicast Addresses", March 2007, <http://www.iana.org/assignments/multicast-addresses>

[IANAPORT] Internet Assigned Numbers Authority, "Port Numbers", November 2006, <http://www.iana.org/assignments/port-numbers>

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980, <http://www.ietf.org/rfc/rfc768.txt>

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981, <http://www.ietf.org/rfc/rfc791.txt>

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

1.2.2 Informative References

There are no informative references.

1.3 Protocol Overview (Synopsis)

The RASADV Protocol is a simple, unidirectional, multicast protocol. The sender is a RAS server, which puts its machine name in a message and periodically multicasts it on its local network. [<1>](#) The receiver listens for periodic messages and passes the message content (the sender's machine

name) and the source address to an application.<2> A typical application using the listener side of this protocol displays the information to the user.

1.4 Relationship to Other Protocols

The RASADV Protocol depends on the User Datagram Protocol (UDP), as specified in [\[RFC7681\]](#), as a transport. No other protocols depend on the RASADV Protocol.

1.5 Prerequisites/Preconditions

1.6 Applicability Statement

The RASADV Protocol applies only to detecting intentional or accidental configuration and deployment of servers over time. It does not apply to detecting malicious configuration and deployment, nor does it apply if an administrator requires the ability to quickly detect such servers on demand, as it relies on a periodic broadcast mechanism.

1.7 Versioning and Capability Negotiation

The RASADV Protocol has no versioning or capability negotiation capabilities.

1.8 Vendor-Extensible Fields

This protocol has no vendor-extensible fields.

1.9 Standards Assignments

The RASADV Protocol uses the following standards assignments.

Parameter	Value	Reference
UDP port number	9753	[IANAPORT]
IP version 4 (IPv4) multicast address	239.255.2.2	[IANAIMA]

2 Messages

The following sections specify how RASADV Protocol messages are transported and common RASADV Protocol data types.

2.1 Transport

All messages MUST be sent over UDP, as specified in [\[RFC768\]](#), with the UDP destination port set to 9753 and the IP destination address set to 239.255.2.2. The IP Time to Live (TTL), as specified in [\[RFC791\]](#) section 3.1, SHOULD be set to 15.

2.2 Message Syntax

The RASADV Protocol has a single message type.

2.2.1 Server Advertisement

The UDP message MUST be the ASCII string "Hostname=<hostname>", where <hostname> is the host name of the sender, followed by a line feed (0x0A) and a null character (0x00). The host name SHOULD [<3>](#) be a fully-qualified domain name, but MAY instead be a non-fully-qualified domain name.

3 Protocol Details

The following sections specify details of the RASADV Protocol, including abstract data models and message processing rules.

3.1 RAS Server Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation 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 that described in this document.

Host Name: The host name of the RAS server.

3.1.2 Timers

Advertisement Timer: A periodic timer used to multicast a server advertisement. It SHOULD have a period of 1 hour.

3.1.3 Initialization

When a RAS server starts, it MUST immediately send a server advertisement, as specified in section [2.2.1](#), and start its Advertisement Timer.

3.1.4 Higher-Layer Triggered Events

This protocol has no higher-layer triggered events.

3.1.5 Message Processing Events and Sequencing Rules

The RAS server role has no message processing events or sequencing rules.

3.1.6 Timer Events

When the Advertisement Timer expires, the RAS server MUST send a server advertisement, as specified in section [2.2.1](#), and restart its Advertisement Timer.

3.1.7 Other Local Events

This protocol has no other local events.

3.2 Listener Details

3.2.1 Abstract Data Model

The listener role has no abstract data model.

3.2.2 Timers

The listener role has no timers.

3.2.3 Initialization

When a listener starts, it MUST start listening for UDP messages on port 9753 and join the IPv4 multicast group 239.255.2.2.

3.2.4 Higher-Layer Triggered Events

This protocol has no higher-layer triggered events.

3.2.5 Message Processing Events and Sequencing Rules

When a message arrives, the listener MUST deliver the message text and the source IP address to the application.

3.2.6 Timer Events

The listener role has no timer events.

3.2.7 Other Local Events

This protocol has no other local events.

4 Protocol Examples

In this example, a network administrator wants to detect accidental deployments of RAS servers on the local network.

1. The network administrator starts a listener tool. The tool begins listening on the RASADV port and the multicast address.
2. Later, someone configures a RAS server on the local network, on a computer named "myserver.example.com" with an IP address of 10.10.10.10. When the RAS server starts, it sends a server advertisement message containing the string "Hostname=myserver.example.com".
3. The listener receives the message and passes 10.10.10.10 and the string to the application. The application displays (for example) "10.10.10.10 Hostname=myserver.example.com".

5 Security

The following sections specify security considerations for implementers of the RASADV Protocol.

5.1 Security Considerations for Implementers

RASADV assumes that servers advertise themselves to be detected, and therefore it does not provide any security or the ability to detect malicious servers.

5.2 Index of Security Parameters

This protocol has no security parameters.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows Server 2008
- Windows Vista
- Windows Server 2003
- Windows XP
- Windows 2000

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

[<1> Section 1.3:](#) Windows 2000 Server, Windows Server 2003, and Windows Server 2008 support the sender role in the Routing and Remote Access Service (RRAS), which is an add-on service that combines remote access support with network routing functionality.

[<2> Section 1.3:](#) The listener is implemented in Windows by the command "netsh ras show activeservers", which displays the messages as they arrive.

[<3> Section 2.2.1:](#) Windows uses a non-fully-qualified domain name if the sender is not a member of a domain.

7 Index

A

Abstract data model
[listener](#)
[server](#)
[Applicability](#)

D

Data model - abstract
[listener](#)
[server](#)

E

[Examples](#)

F

[Fields - vendor-extensible](#)

G

[Glossary](#)

H

Higher-layer triggered events
[listener](#)
[server](#)

I

[Implementer - security considerations](#)
[Index of security parameters](#)
[Informative references](#)
Initialization
[listener](#)
[server](#)
[Introduction](#)

L

Listener
[abstract data model](#)
[higher-layer triggered events](#)
[initialization](#)
[local events](#)
[message processing](#)
[overview](#)
[sequencing rules](#)
[timer events](#)
[timers](#)
Local events
[listener](#)
[server](#)

M

Message processing
[listener](#)
[server](#)
Messages
[overview](#)
[syntax](#)
[transport](#)

N

[Normative references](#)

O

[Overview](#)

P

[Parameters - security index](#)
[Preconditions](#)
[Prerequisites](#)

R

References
[informative](#)
[normative](#)
[overview](#)
[Relationship to other protocols](#)

S

Security
[implementer considerations](#)
[overview](#)
[parameter index](#)
Sequencing rules
[listener](#)
[server](#)
Server
[abstract data model](#)
[higher-layer triggered events](#)
[initialization](#)
[local events](#)
[message processing](#)
[overview](#)
[sequencing rules](#)
[timer events](#)
[timers](#)
[Server advertisement](#)
[Standards assignments](#)
[Syntax](#)

T

Timer events

[listener](#)

[server](#)

Timers

[listener](#)

[server](#)

[Transport](#)

Triggered events - higher-layer

[listener](#)

[server](#)

V

[Vendor-extensible fields](#)

[Versioning](#)

W

[Windows behavior](#)