[MS-RDPEPS]:
Remote Desktop Protocol:
Session Selection Extension

Revision Summary

Date	Revision History	Revision Class	Comments
04/03/2007	0.01		MCPP Milestone Longhorn Initial Availability
06/01/2007	1.0		MCPP Milestone 3 + 90
07/03/2007	2.0	Major	MLonghorn+90
07/20/2007	2.0.1	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
08/10/2007	2.0.2	Editorial	Revised and edited the technical content.
09/28/2007	2.0.3	Editorial	Revised and edited the technical content.
10/23/2007	2.0.4	Editorial	Revised and edited the technical content.
11/30/2007	3.0	Major	Updated and revised the technical content.
01/25/2008	3.0.1	Editorial	Revised and edited the technical content.

Table of Contents

1 Introduction

This protocol extension expands upon the original connectivity options specified in the Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification to address a wide range of new scenarios where the Remote Desktop Protocol (RDP) is used to send the user experience of an application.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Client
Protocol Data Unit (PDU)
Server
Terminal Server
Unicode

The following terms are specific to this document:

RDP Source: A process or other operating system entity that generates a remoting protocol conforming to the [MS-RDPBCGR] specification.

RDP Source ID: An identifier that uniquely identifies the source of an RDP stream. The identifier is frequently exchanged between **client** and **server** through a secure out-of-band mechanism. While the identifier could be as simple as a process ID, it is often obfuscated in some way.

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.

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-RDPBCGR] Microsoft Corporation, "Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification", June 2007.

[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 Remote Desktop Protocol: Session Selection Extension describes the messages exchanged between an RDP **client** and a **terminal server** to facilitate the precise targeting of an application sharing context. This extension allows for multiple processes to accept connections on the same TCP/IP port. Normally, the RDP client is used to connect to a particular user's session. However, in the case where RDP is used to send the output of system processes (such as a virtual **server**), the user's credentials are insufficient.

This extension allows further specification of the session by including additional information such as a process ID or any other method to uniquely identify the **RDP source**.

1.4 Relationship to Other Protocols

This protocol extension is a minor enhancement to the Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification.

1.5 Prerequisites/Preconditions

Either type of identifier to be used is exchanged between client and server via an out-of-band mechanism. The IP address and port of the server can also be exchanged via an out-of-band mechanism.

1.6 Applicability Statement

This protocol is applicable when there is a need to allow multiple processes to accept connections on the same TCP/IP port.

1.7 Versioning and Capability Negotiation

There are two versions of this protocol. For specifications about handling and sending the version information, see sections 2.2.1, 3.1.5.1, and 3.2.5.1.

- Version 1 is specified by the RDP_PRECONNECTION_PDU_V1 **PDU**
- Version 2 is specified by the RDP_PRECONNECTION_PDU_V2 PDU

A listening process that implements this extension can always expect the PDU and the client will always send it.

The client has to use the exact type of PDU format implemented by the server. If the server implements the version 1 of the PDU then the client has to send that version. If the server implements version 2 of the PDU the client has to send that version. The client also needs to send the information expected by the server implementation in the PDU id and wszPCB fields. Since there is no negotiation for the format and the required information these have to be established through convention based on the type of server the client is connecting too.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

The following sections specify how Remote Desktop Protocol: Session Selection Extension messages are transported and the message syntax.

2.1 Transport

This protocol uses TCP/IP as its transport mechanism.

2.2 Message Syntax

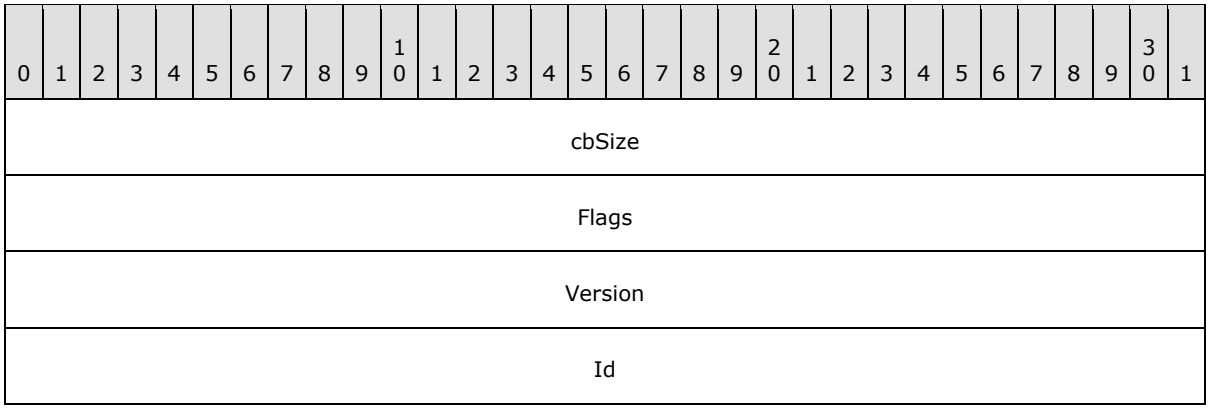
The following sections contain the Remote Desktop Protocol: Session Selection Extension message syntax.

2.2.1 Server RDP Preconnection PDU

The following sections contain Remote Desktop Protocol: Session Selection Extension message syntax for the RDP_PRECONNECTION_PDU_V1 and RDP_PRECONNECTION_PDU_V2 PDU's.

2.2.1.1 Version 1 (RDP_PRECONNECTION_PDU_V1)

The RDP_PRECONNECTION_PDU_V1 PDU is used by the client to let the listening process know which RDP source the connection is intended for.



cbSize (4 bytes): An unsigned 32-bit integer. This field identifies the total size, in bytes, of the RDP_PRECONNECTION_PDU_V1, or in the case of a version 2 packet, the size of a RDP_PRECONNECTION_PDU_V2 packet.

Flags (4 bytes): An unsigned 32-bit integer. MUST be set to zero when sending and ignored on receipt.

Version (4 bytes): An unsigned 32-bit integer. This field is one of the values in the table based on the version of the PDU that is sent. The field is intended for future extensibility. The field SHOULD be initialized by the client and SHOULD be ignored by the server, as specified in sections 3.1.5.1 and 3.2.5.1.

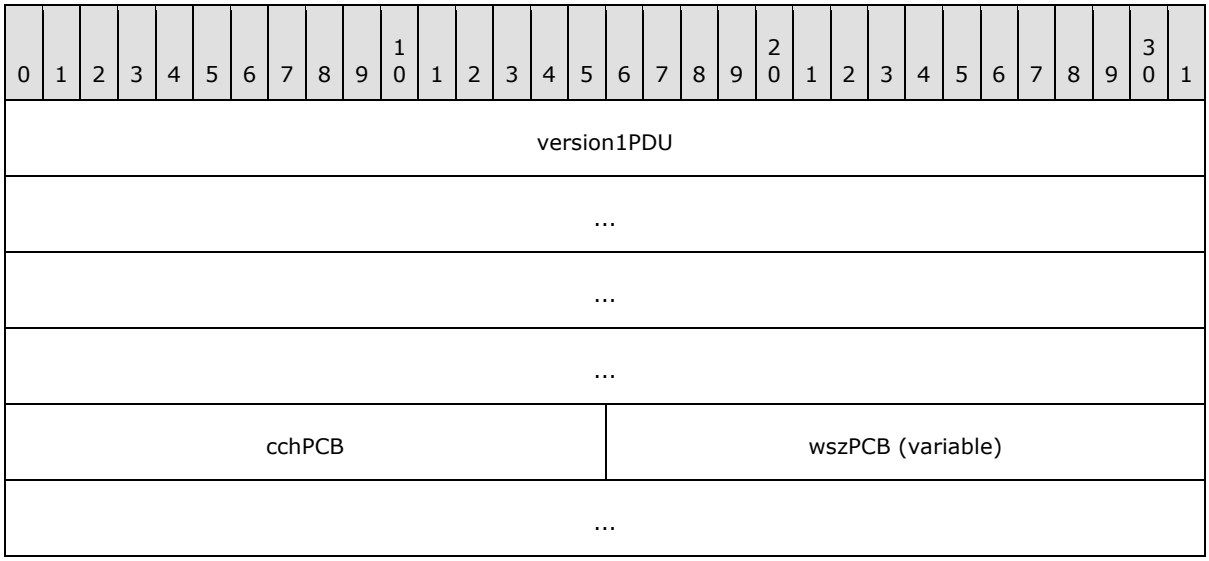
Value	Meaning
RDP_PRECONNECTION_PDU_V1 0x00000001	A version 1 connection PDU.

Value	Meaning
RDP_PRECONNECTION_PDU_V2 0x00000002	A version 2 connection PDU.

Id (4 bytes): An unsigned 32-bit integer. The Id is used to uniquely identify the RDP source. While the Id could be as simple as a process ID, it is often client or server specific and MAY be obfuscated.

2.2.1.2 Version 2 (RDP_PRECONNECTION_PDU_V2)

The RDP_PRECONNECTION_PDV_V2 extends the RDP_PRECONNECTION_PDU_V1 packet by adding a variable-size **Unicode** character string. The receiver of this PDU can use this string and the **Id** field of the RDP_PRECONNECTION_PDU_V1 packet to determine the RDP source. This string is opaque to the protocol.



version1PDU (16 bytes): The RDP_PRECONNECTION_PDU_V1 PDU header, as specified in section 2.2.1.1.

cchPCB (2 bytes): An unsigned 16-bit integer. The number of Unicode characters in the **wszPCB** field.

wszPCB (variable): A variable-length array of Unicode characters.

3 Protocol Details

The following sections specify protocol details of the Remote Desktop Protocol: Session Selection Extension, including abstract data models and message processing rules.

3.1 Client Details

This section contains client details including abstract data model, timers, initialization, higher-layer triggered events, and message processing events and sequencing rules.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

This protocol has a 10-second timer. It starts at the successful creation of the TCP connection between the client and server. The full length of the RDP_PRECONNECTION_PDU_V1 or RDP_PRECONNECTION_PDU_V2 MUST be sent by the client and received by the server within this window.

3.1.3 Initialization

The client MUST use the IP address and port already exchanged with the server via an out-of-band mechanism when making the connection.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The client MUST send either the RDP_PRECONNECTION_PDU_V1 or RDP_PRECONNECTION_PDU_V2 to the server before any other traffic.

3.1.5.1 Sending the RDP_PRECONNECTION_PDU_V1

The client MUST initialize the **cbSize** field with the total size, in bytes, of the PDU. The size of the RDP_PRECONNECTION_PDU_V1 PDU is always 16 bytes.

The client SHOULD initialize the version field with the RDP_PRECONNECTION_PDU_V1 value<1>.

The **Id** field SHOULD be set to the **RDP source ID** the connection is intended for. The value of this field can be determined by the client, either by convention or by communicating with the server.

3.1.5.2 Sending the RDP_PRECONNECTION_PDU_V2

The size sent in the RDP_PRECONNECTION_PDU_V2 PDU **cbSize** field is calculated by adding the size of the RDP_PRECONNECTION_PDU_V1 header, **cchPCB** field and **wszPCB** field. The size, in bytes, of the **wszPCB** field is calculated as the number of characters specified in the **cchPCB** field multiplied by 2. If the **cchPCB** field contains five characters, the **cbSize** will be 18 + (5 * 2) = 28. If the **cchPCB** field is 0, then the **wszPCB** field is not present, and the **cbSize** field MUST be initialized with a value of 18.

The client SHOULD initialize the **Version** field with the RDP_PRECONNECTION_PDU_V2 value<2>.

The client MUST initialize the **cchPCB** field with the number of characters in the **wszPCB** Unicode array.

The **wszPCB** field SHOULD be set to a value that identifies the server-side process the connection is intended for. The value can be determined either by convention or by communicating with the server through other mechanisms before the connection is established.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

This section contains server details including abstract data model, timers, initialization, higher-layer triggered events, and message processing events and sequencing rules.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

This protocol has a 10-second timer. It starts at the successful creation of the TCP connection between the client and server. The full length of the RDP_PRECONNECTION_PDU_V1 or RDP_PRECONNECTION_PDU_V2 MUST be sent by the client and received by the server within this window.

3.2.3 Initialization

The server MUST listen on the IP address and port already exchanged with the client via an out-of-band mechanism.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

This section includes information about the processing of RDP_PRECONNECTION_PDU_V1 and RDP_PRECONNECTION_PDU_V2.

3.2.5.1 Processing the RDP_PRECONNECTION PDU V1 and V2

When processing either the RDP_PRECONNECTION_PDU_V1 or RDP_PRECONNECTION_PDU_V2, the server MUST not make any assumption about the way the PDU is delivered by TCP/IP protocol. The server MUST read only the bytes that are part of this PDU. The server MUST NOT read more than the minimum size required. The server SHOULD wait and receive the whole PDU. After the whole PDU is received, the server MUST determine what process the connection is intended for. The server

MUST hand over the connection to the specified process. In case the information in the PDU does not map to any process, the server SHOULD disconnect the client.

In order to process this PDU, the server MUST first determine how long the PDU is. This is done by reading 4 bytes corresponding to the **cbSize** field of the PDU, as specified in section 2.2.1.1. If the **cbSize** field is 16 bytes, the server SHOULD consider the PDU a RDP_PRECONNECTION_PDU_V1. If the size is greater than or equal to 18 bytes, the server SHOULD consider this a RDP_PRECONNECTION_PDU_V2 PDU and check that the size is in the expected range based on the **cbSize** field, and disconnect the client if it is not.

Once the **cbSize** field is received, the server SHOULD read that number of bytes from the TCP/IP stream. If the number of bytes specified in **cbSize** is greater than the actual size of the payload, the server SHOULD ignore these bytes.

Upon receiving a RDP_PRECONNECTION_PDU_V2, the server MUST verify that the size of the **wszPCB** field is consistent with the PDU size expected by **cbSize** field. The **cbSize** MUST be greater than or equal to the size of the RDP_PRECONNECTION_PDU_V1 PDU plus the size of the **cchPCB** field and **wszPCB** field, calculated as **cchPCB** multiplied by 2. If **cbSize** does not meet this condition the server SHOULD disconnect the client<3>.

Once the whole PDU is received, the server SHOULD use the **Id** field and the **wszPCB** field string to identify the RDP source that the connection is intended for<4>. If there is no such entity, the server SHOULD terminate the connection<5>.

3.2.6 Timer Events

If the 10-second timer expires, the server MUST disconnect the client. Any resources associated with the connection are freed.

3.2.7 Other Local Events

None.

4 Protocol Examples

The following is an example of the RDP_PRECONNECTION_PDU_V1 PDU.

```
00000000  10 00 00 00 00 00 00 00-01 00 00 00 eb 99 c6 ee  ................
10 00 00 00 -> RDP_PRECONNECTION_PDU_V1::cbSize = 0x10 = 16 bytes
00 00 00 00 -> RDP_PRECONNECTION_PDU_V1::Flags = 0
01 00 00 00 -> RDP PRECONNECTION PDU V1::Version = 1
eb 99 c6 ee -> RDP PRECONNECTION PDU V1::Id = 0xEEC699EB = 4005992939
(random id)
```

The following is an example of the RDP_PRECONNECTION_PDU_V2 PDU, where the PDU is a RDP_PRECONNECTION_PDU_V1 extended with a variable-size Unicode string.

```
00000000  20 00 00 00 00 00 00 00-02 00 00 00 00 00 00 00   ................
00000010  07 00 54 00 65 00 73 00-74 00 56 00 4d 00 00 00  ..T.e.s.t.V.M...
20 00 00 00 -> RDP PRECONNECTION PDU V2::RDP PRECONNECTION PDU V1::cbSize = 0x20 = 32
bytes
00 00 00 00 -> RDP_PRECONNECTION_PDU_V2::RDP_PRECONNECTION_PDU_V1::Flags = 0
02 00 00 00 -> RDP_PRECONNECTION_PDU_V2::RDP_PRECONNECTION_PDU_V1::Version = 2
00 00 00 00 -> RDP_PRECONNECTION_PDU_V2::RDP_PRECONNECTION_PDU_V1::Id = 0
07 00       -> RDP PRECONNECTION PDU V2::cchPCB = 0x7 = 7 characters
54 00 65 00 73 00-74 00 56 00 4d 00 00 00 -> RDP PRECONNECTION PDU V2::wszPCB -> "TestVM"
(including null terminator)
```

5 Security

The following sections specify security considerations for implementers of the Remote Desktop Protocol: Session Selection Extension.

5.1 Security Considerations for Implementers

To avoid denial of service attacks, the server SHOULD not wait indefinitely for the PDU content to arrive, but disconnect the client if the timer specified in sections 3.1.2 and 3.2.2 has passed and the whole PDU has not been received.

5.2 Index of Security Parameters

None.

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 3.1.5.1: In Windows Vista, the client will initialize the **Version** field with the RDP_PRECONNECTION_PDU_V1 as the ID field is sufficient to identify the wanted RDP source. The **Id** is initialized to a value exchanged between the client and server via an out-of-band mechanism and appears as a random number.

<2> Section 3.1.5.2: In the Windows Server 2008 virtual machine implementation, the client will initialize the **Version** field to RDP_PRECONNECTION_PDU_V2 and set the ID field to zero. The string is used to identify the name of the wanted virtual machine. The string should be NULL-terminated; if not, the server will overwrite the last Unicode character with NULL prior to further processing.

The Windows Server 2008 virtual machine sends a GUID rather than a virtual machine name.

GUID formatting as follows: 00000000-0000-0000-0000-000000000000. The hex numbers are separated by '-' characters, but without any {} around it. The byte ordering is standard GUID byte ordering (same for all GUIDs represented as strings), case insensitive for the hex digits. No other form is allowed.

<3> Section 3.2.5.1: The Windows Vista and Windows Server 2008 virtual machine client is disconnected if the **cbSize** field verification fails.

<4> Section 3.2.5.1: The Windows Vista Collaboration API uses the RDP_PRECONNECTION_PDU_V1 PDU. The RDP_PRECONNECTION_PDU_V1 PDU carries a random number that was originally generated by the server and communicated to the client via some out-of-band mechanism. The server will compare the original generated number with the number carried by the RDP_PRECONNECTION_PDU_V1 PDU to locate the correct RDP source.

<5> Section 3.2.5.1: In Windows Server 2008 virtual machine, the RDP_PRECONNECTION_PDU_V2 PDU is used. The **Id** is ignored by the server. The server always expects a string. The string represents the name of the virtual machine the client tries to connect to. The server expects a null-terminated string at the end of the array. The server will not verify that the last character is a null terminator; instead, it will overwrite it with the null-terminated character.

7 Index

T

V

W