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

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **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 make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications 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 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, email addresses, logos, people, places, and events 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 specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do 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 are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
08/08/2013	1.0	New	Released new document.
11/14/2013	2.0	Major	Significantly changed the technical content.
02/13/2014	3.0	Major	Significantly changed the technical content.
05/15/2014	3.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1   Introduction

This document specifies the Remote Desktop Protocol: Display Control Channel Extension to the Remote Desktop Protocol: Basic Connectivity and Graphics Remoting, as specified in [MS-RDPBCGR] sections 1 to 5. This control protocol is used by the server to request display configuration changes in a remote session. Display configuration changes include the addition, removal and repositioning of monitors, resolution updates, and orientation updates.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1   Glossary

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

Windows-1252 character set

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

virtual channel
dynamic virtual channel

The following terms are specific to this document:

ANSI character: An 8-bit Windows-1252 character set unit.

Desktop Scale Factor: The scale factor (as a percentage) applied to Windows Desktop Applications.

Device Scale Factor: The scale factor (as a percentage) applied to Windows Store Applications.

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.

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

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

[MS-RDPEDYC] Microsoft Corporation, "Remote Desktop Protocol: Dynamic Channel Virtual Channel Extension".

[MS-RDPEGFX] Microsoft Corporation, "Remote Desktop Protocol: Graphics Pipeline Extension".

Release: Thursday, May 15, 2014

[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

1.2.2 Informative References

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

1.3 Protocol Overview (Synopsis)

The sequence of messages exchanged by the Remote Desktop Protocol: Display Control Virtual Channel Extension is described in the following figure. The messages exchanged in this diagram are strictly sequential.
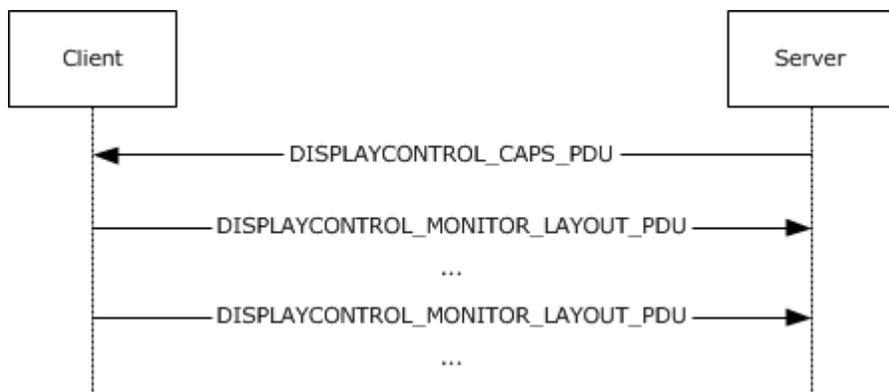
Figure 1: The Display Control message sequence

After the Display Control dynamic virtual channel has been opened, the server communicates display capabilities to the client by sending the **DISPLAYCONTROL_CAPS_PDU** (section 2.2.2.1) message. This message specifies a set of parameters to which the client is required to adhere when sending the **DISPLAYCONTROL_MONITOR_LAYOUT_PDU** (section 2.2.2.2) message.

To request a display configuration change on the server (such as the addition of a monitor or a new resolution for an existing monitor), the client sends the **DISPLAYCONTROL_MONITOR_LAYOUT_PDU** message to the server. If the requested configuration is not possible, or is invalid, the server will not update the remote session with the requested parameters.

Changes in the server-side display configuration occur out of band to the Remote Desktop Protocol: Display Control Virtual Channel Extension. If the requested graphics configuration is valid and can be configured on the server, then the server will either:

- Initiate a Deactivation-Reactivation Sequence (as specified in [MS-RDPBCGR] section 1.3.1.3) if the Remote Desktop Protocol: Graphics Pipeline Extension is not being used to remote session graphics.

- Restart the graphics pipeline using the surface management commands (specified in [MS-RDPEGFX] section 1.3) if the Remote Desktop Protocol: Graphics Pipeline Extension is being used to remote session graphics.

The **DISPLAYCONTROL_MONITOR_LAYOUT_PDU** message can be sent whenever a display configuration change is required.

1.4 Relationship to Other Protocols

The Remote Desktop Protocol: Display Control Virtual Channel Extension is embedded in a dynamic virtual channel transport, as specified in [MS-RDPEDYC] sections 1 to 3.

1.5 Prerequisites/Preconditions

The Remote Desktop Protocol: Display Control Virtual Channel Extension operates only after the dynamic virtual channel transport is fully established. If the dynamic virtual channel transport is terminated, the Remote Desktop Protocol: Display Control Virtual Channel Extension is also terminated. The protocol is terminated by closing the underlying virtual channel. For details about closing the dynamic virtual channel, refer to [MS-RDPEDYC] section 3.2.5.2.

If the RemoteFX codec ([MS-RDPRFX] sections 2.2.2 and 3.1.8) is being used to encode graphics data from a remote session, then the Remote Desktop Protocol: Display Control Virtual Channel Extension SHOULD NOT be used to request display configuration changes.

1.6 Applicability Statement

The Remote Desktop Protocol: Display Control Virtual Channel Extension is applicable in scenarios where a mechanism to request display configuration changes in a remote session without disconnecting and reconnecting is required.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The Remote Desktop Protocol: Display Control Virtual Channel Extension is designed to operate over a dynamic virtual channel, as specified in [MS-RDPEDYC] sections 1 to 3. The dynamic virtual channel name is the null-terminated **ANSI character** string "Microsoft::Windows::RDS::DisplayControl". The usage of channel names in the context of opening a dynamic virtual channel is specified in [MS-RDPEDYC] section 2.2.2.1.
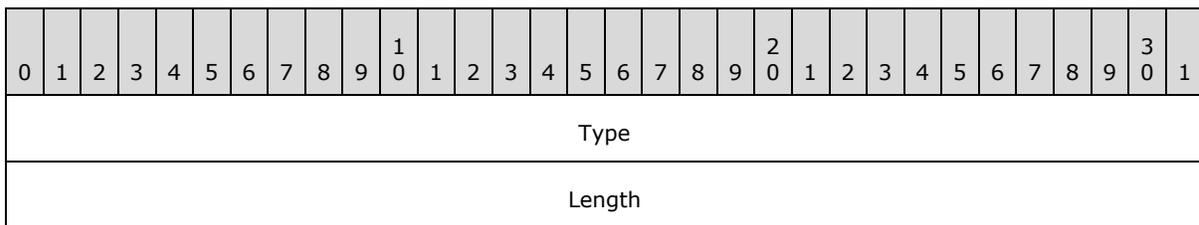
2.2 Message Syntax

The following sections specify the Remote Desktop Protocol: Display Virtual Channel Extension message syntax.

2.2.1 Common Data Types

2.2.1.1 DISPLAYCONTROL_HEADER

The **DISPLAYCONTROL_HEADER** structure is included in all display control PDUs and specifies the PDU type and the length of the PDU.

0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
Type																															
Length																															

Type (4 bytes): A 32-bit unsigned integer that specifies the display control PDU type.

Value	Meaning
DISPLAYCONTROL_PDU_TYPE_CAPS 0x00000005	**DISPLAYCONTROL_CAPS_PDU** (section 2.2.2.1)
DISPLAYCONTROL_PDU_TYPE_MONITOR_LAYOUT 0x00000002	**DISPLAYCONTROL_MONITOR_LAYOUT_PDU** (section 2.2.2.2)

Length (4 bytes): A 32-bit unsigned integer that specifies the length of the display control PDU, in bytes. This value MUST include the length of the DISPLAYCONTROL_HEADER (8 bytes).

2.2.2 Display Control Messages

2.2.2.1 DISPLAYCONTROL_CAPS_PDU

The DISPLAYCONTROL_CAPS_PDU message is a server-to-client PDU that is used to specify a set of parameters which the client must adhere to when sending the **DISPLAYCONTROL_MONITOR_LAYOUT_PDU** (section 2.2.2.2) message.

0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
Header																															
...																															
MaxNumMonitors																															
MaxMonitorAreaFactorA																															
MaxMonitorAreaFactorB																															

Header (8 bytes): A **DISPLAYCONTROL_HEADER** (section 2.2.1.1) structure. The **Type** field MUST be set to DISPLAYCONTROL_PDU_TYPE_CAPS (0x00000005).

MaxNumMonitors (4 bytes): A 32-bit unsigned integer that specifies the maximum number of monitors supported by the server.

MaxMonitorAreaFactorA (4 bytes): A 32-bit unsigned integer that is used to specify the maximum monitor area supported by the server. The maximum supported monitor area (in square pixels) is given by **MaxNumMonitors** * **MaxMonitorAreaFactorA** * **MaxMonitorAreaFactorB**.

MaxMonitorAreaFactorB (4 bytes): A 32-bit unsigned integer that is used to specify the maximum monitor area supported by the server. The maximum supported monitor area (in square pixels) is given by **MaxNumMonitors** * **MaxMonitorAreaFactorA** * **MaxMonitorAreaFactorB**.

2.2.2.2   DISPLAYCONTROL_MONITOR_LAYOUT_PDU

The **DISPLAYCONTROL_MONITOR_LAYOUT_PDU** message is a client-to-server PDU that is used to request a display configuration change on the server, such as the addition of a monitor or a new resolution for an existing monitor. Note that the entire monitor layout MUST be included in the **Monitors** field even if the configuration of only a single monitor is updated.

0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
Header																															
...																															
MonitorLayoutSize																															
NumMonitors																															
Monitors (variable)																															
...																															

Header (8 bytes): A **DISPLAYCONTROL_HEADER** (section 2.2.1.1) structure. The **Type** field MUST be set to DISPLAYCONTROL_PDU_TYPE_MONITOR_LAYOUT (0x00000002).

MonitorLayoutSize (4 bytes): A 32-bit unsigned integer that specifies the size, in bytes, of a single element in the **Monitors** field. This field MUST be set to 40 bytes, the size of the **DISPLAYCONTROL_MONITOR_LAYOUT** structure (section 2.2.2.2.1).

NumMonitors (4 bytes): A 32-bit unsigned integer that specifies the number of display monitor definitions in the **Monitors** field. The maximum number of monitor definitions allowed is specified in the **MaxNumMonitors** field of the **DISPLAYCONTROL_CAPS_PDU** (section 2.2.2.1) message.

Monitors (variable): A variable-length array containing a series of **DISPLAYCONTROL_MONITOR_LAYOUT** structures that specify the display monitor layout of the client. The number of **DISPLAYCONTROL_MONITOR_LAYOUT** structures is specified by the **NumMonitors** field. The area (in square pixels) of the layout specified by the **DISPLAYCONTROL_MONITOR_LAYOUT** structures MUST NOT exceed the maximum monitor area defined by the server in the **DISPLAYCONTROL_CAPS_PDU** message.

2.2.2.2.1   DISPLAYCONTROL_MONITOR_LAYOUT

The **DISPLAYCONTROL_MONITOR_LAYOUT** structure is used to specify the characteristics of a monitor. The coordinates used to describe the monitor position MUST be relative to the upper-left corner of the monitor designated as the "primary display monitor". The upper-left corner of the primary monitor is always (0, 0).

0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
Flags																															
Left																															
Top																															
Width																															
Height																															
PhysicalWidth																															
PhysicalHeight																															
Orientation																															
DesktopScaleFactor																															
DeviceScaleFactor																															

Flags (4 bytes): A 32-bit unsigned integer that specifies monitor configuration flags.

Flag	Meaning
DISPLAYCONTROL_MONITOR_PRIMARY 0x00000001	The monitor specified by this structure is the primary monitor.

Left (4 bytes): A 32-bit signed integer that specifies the x-coordinate of the upper-left corner of the display monitor.

Top (4 bytes): A 32-bit signed integer that specifies the y-coordinate of the upper-left corner of the display monitor.

Width (4 bytes): A 32-bit unsigned integer that specifies the width of the monitor in pixels. The width MUST be greater than or equal to 200 pixels and less than or equal to 8192 pixels, and MUST NOT be an odd value.

Height (4 bytes): A 32-bit unsigned integer that specifies the height of the monitor in pixels. The height MUST be greater than or equal to 200 pixels and less than or equal to 8192 pixels.

PhysicalWidth (4 bytes): A 32-bit unsigned integer that specifies the physical width of the monitor, in millimeters (mm). This value MUST be ignored if it is less than 10 mm or greater than 10,000 mm or the **PhysicalHeight** field is less than 10 mm or greater than 10,000 mm.

PhysicalHeight (4 bytes): A 32-bit unsigned integer that specifies the physical height of the monitor, in millimeters. This value MUST be ignored if it is less than 10 mm or greater than 10,000 mm or **PhysicalWidth** is less than 10 mm or greater than 10,000 mm.

Orientation (4 bytes): A 32-bit unsigned integer that specifies the orientation of the monitor in degrees. Valid values are 0, 90, 180 or 270. This value MUST be ignored if it is not set to one of these values.

Value	Meaning
ORIENTATION_LANDSCAPE 0	The desktop is not rotated.
ORIENTATION_PORTRAIT 90	The desktop is rotated clockwise by 90 degrees.
ORIENTATION_LANDSCAPE_FLIPPED 180	The desktop is rotated clockwise by 180 degrees.
ORIENTATION_PORTRAIT_FLIPPED 270	The desktop is rotated clockwise by 270 degrees.

DesktopScaleFactor (4 bytes): A 32-bit, unsigned integer that specifies the **desktop scale factor** of the monitor. This value MUST be ignored if it is less than 100 percent or greater than 500 percent, or if **DeviceScaleFactor** is not 100 percent, 140 percent, or 180 percent.

DeviceScaleFactor (4 bytes): A 32-bit, unsigned integer that specifies the **device scale factor** of the monitor. This value MUST be ignored if it is not set to 100 percent, 140 percent, or 180 percent, or if **DesktopScaleFactor** is less than 100 percent or greater than 500 percent.

3   Protocol Details

3.1   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.

Note It is possible to implement the following conceptual data by using a variety of techniques as long as the implementation produces external behavior that is consistent with that described in this document.

3.1.2   Timers

None.

3.1.3   Initialization

None.

3.1.4   Higher-Layer Triggered Events

None.

3.1.5   Processing Events and Sequencing Rules

3.1.5.1   Sending DISPLAYCONTROL_CAPS_PDU

The structure and fields of the **DISPLAYCONTROL_CAPS_PDU** message are specified in section 2.2.2.1. The message fields MUST be populated in accordance with this description.

3.1.5.2   Processing DISPLAYCONTROL_MONITOR_LAYOUT_PDU

The structure and fields of the **DISPLAYCONTROL_MONITOR_LAYOUT_PDU** message are specified in section 2.2.2.2. The server SHOULD trigger a display configuration update in the remote session using the parameters specified by the client if all of the following conditions are true:

- All of the fields specified in the **DISPLAYCONTROL_MONITOR_LAYOUT_PDU** message are valid, consistent and within range.

- None of the specified monitors overlap.

- Each monitor is adjacent to at least one other monitor (even if only at a single point).

3.1.6   Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Client Details

3.2.1 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.

Note It is possible to implement the following conceptual data by using a variety of techniques as long as the implementation produces external behavior that is consistent with that described in this document.

3.2.1.1 Maximum Monitor Count

The Maximum Monitor Count store contains the contents of the **MaxNumMonitors** field of the **DISPLAYCONTROL_CAPS_PDU** (section 2.2.2.1) message.

3.2.1.2 Maximum Monitor Area Factor A

The Maximum Monitor Area Factor A store contains the contents of the **MaxMonitorAreaFactorA** field of the **DISPLAYCONTROL_CAPS_PDU** (section 2.2.2.1) message.

3.2.1.3 Maximum Monitor Area Factor B

The Maximum Monitor Area Factor B store contains the contents of the **MaxMonitorAreaFactorB** field of the **DISPLAYCONTROL_CAPS_PDU** (section 2.2.2.1) message.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Processing Events and Sequencing Rules

3.2.5.1 Processing DISPLAYCONTROL_CAPS_PDU

The structure and fields of the **DISPLAYCONTROL_CAPS_PDU** message are specified in section 2.2.2.1. Upon receiving the **DISPLAYCONTROL_CAPS_PDU** message, the client MUST store the contents of the **MaxNumMonitors**, **MaxMonitorAreaFactorA** and **MaxMonitorAreaFactorB** fields in the **Maximum Monitor Count** (section 3.2.1.1), **Maximum Monitor Area Factor A**

(section 3.2.1.2), and **Maximum Monitor Area Factor B** (section 3.2.1.3) ADM elements respectively.

3.2.5.2 Sending DISPLAYCONTROL_MONITOR_LAYOUT_PDU

The structure and fields of the **DISPLAYCONTROL_MONITOR_LAYOUT_PDU** message are specified in section 2.2.2.2. The message fields MUST be populated in accordance with that description. The maximum area of the monitor layout MUST NOT exceed product of the **Maximum Monitor Count** (section 3.2.1.1), **Maximum Monitor Area Factor A** (section 3.2.1.2), and **Maximum Monitor Area Factor B** (section 3.2.1.3) ADM elements.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4   Protocol Examples

None.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6   Appendix A: 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 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.

7 Change Tracking

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

8 Index