**[MS-DSPA]:**

**Device Session Property Access Protocol**

**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](http://go.microsoft.com/fwlink/?LinkId=214445) or the [Community Promise](http://go.microsoft.com/fwlink/?LinkId=214448). 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](mailto: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](http://www.microsoft.com/trademarks).
* **Fictitious Names**. The example companies, organizations, products, domain names, e-mail 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 |
| --- | --- | --- | --- |
| 11/6/2009 | 0.1 | Major | First Release. |
| 12/18/2009 | 0.1.1 | Editorial | Changed language and formatting in the technical content. |
| 1/29/2010 | 0.1.2 | Editorial | Changed language and formatting in the technical content. |
| 3/12/2010 | 0.1.3 | Editorial | Changed language and formatting in the technical content. |
| 4/23/2010 | 0.1.4 | Editorial | Changed language and formatting in the technical content. |
| 6/4/2010 | 1.0 | Major | Updated and revised the technical content. |
| 7/16/2010 | 1.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 8/27/2010 | 1.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/8/2010 | 1.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 11/19/2010 | 1.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 1/7/2011 | 1.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/11/2011 | 1.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 3/25/2011 | 1.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 5/6/2011 | 1.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/17/2011 | 1.1 | Minor | Clarified the meaning of the technical content. |
| 9/23/2011 | 1.2 | Minor | Clarified the meaning of the technical content. |
| 12/16/2011 | 2.0 | Major | Updated and revised the technical content. |
| 3/30/2012 | 2.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/12/2012 | 2.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/25/2012 | 2.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 1/31/2013 | 2.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 8/8/2013 | 3.0 | Major | Updated and revised the technical content. |
| 11/14/2013 | 4.0 | Major | Updated and revised the technical content. |
| 2/13/2014 | 4.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 5/15/2014 | 4.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/30/2015 | 4.0 | No Change | No changes to the meaning, language, or formatting of the technical content. |
| 10/16/2015 | 4.0 | No Change | No changes to the meaning, language, or formatting of the technical content. |

Table of Contents

[1 Introduction 6](#_Toc432489238)

[1.1 Glossary 6](#_Toc432489239)

[1.2 References 6](#_Toc432489240)

[1.2.1 Normative References 6](#_Toc432489241)

[1.2.2 Informative References 7](#_Toc432489242)

[1.3 Overview 7](#_Toc432489243)

[1.4 Relationship to Other Protocols 8](#_Toc432489244)

[1.4.1 Device Services Lightweight Remoting (DSLR) Protocol 8](#_Toc432489245)

[1.5 Prerequisites/Preconditions 9](#_Toc432489246)

[1.6 Applicability Statement 9](#_Toc432489247)

[1.7 Versioning and Capability Negotiation 9](#_Toc432489248)

[1.8 Vendor-Extensible Fields 9](#_Toc432489249)

[1.9 Standards Assignments 9](#_Toc432489250)

[2 Messages 10](#_Toc432489251)

[2.1 Transport 10](#_Toc432489252)

[2.2 Message Syntax 10](#_Toc432489253)

[2.2.1 Property Bag Service 10](#_Toc432489254)

[2.2.1.1 GetStringProperty 10](#_Toc432489255)

[2.2.1.1.1 GetStringProperty (request) 10](#_Toc432489256)

[2.2.1.1.1.1 AV Property Bag 11](#_Toc432489257)

[2.2.1.1.1.2 Device Capabilities PropertyBag 11](#_Toc432489258)

[2.2.1.1.2 GetStringProperty (response) 11](#_Toc432489259)

[2.2.1.2 SetDWORDProperty 12](#_Toc432489260)

[2.2.1.2.1 SetDWORDProperty (request) 12](#_Toc432489261)

[2.2.1.2.1.1 AV Property Bag 13](#_Toc432489262)

[2.2.1.2.1.2 Device Capabilities PropertyBag 13](#_Toc432489263)

[2.2.1.2.2 SetDWORDProperty (response) 13](#_Toc432489264)

[2.2.1.3 GetDWORDProperty 13](#_Toc432489265)

[2.2.1.3.1 GetDWORDProperty (request) 13](#_Toc432489266)

[2.2.1.3.1.1 AV Property Bag 14](#_Toc432489267)

[2.2.1.3.1.2 Device Capabilities PropertyBag 14](#_Toc432489268)

[2.2.1.3.2 GetDWORDProperty (response) 16](#_Toc432489269)

[3 Protocol Details 17](#_Toc432489270)

[3.1 Device Details 17](#_Toc432489271)

[3.1.1 Abstract Data Model 18](#_Toc432489272)

[3.1.2 Timers 18](#_Toc432489273)

[3.1.3 Initialization 18](#_Toc432489274)

[3.1.4 Higher-Layer Triggered Events 18](#_Toc432489275)

[3.1.5 Processing Events and Sequencing Rules 18](#_Toc432489276)

[3.1.5.1 Create Service 18](#_Toc432489277)

[3.1.5.2 Two-Way Requests 18](#_Toc432489278)

[3.1.5.3 Delete Service 19](#_Toc432489279)

[3.1.6 Timer Events 19](#_Toc432489280)

[3.1.7 Other Local Events 19](#_Toc432489281)

[3.2 Host Computer Details 19](#_Toc432489282)

[4 Protocol Examples 20](#_Toc432489283)

[5 Security 22](#_Toc432489284)

[5.1 Security Considerations for Implementers 22](#_Toc432489285)

[5.2 Index of Security Parameters 22](#_Toc432489286)

[6 Appendix A: Product Behavior 23](#_Toc432489287)

[7 Appendix B: Protocol Information String (PRT) 24](#_Toc432489288)

[8 Change Tracking 28](#_Toc432489289)

[9 Index 29](#_Toc432489290)

# Introduction

This document describes the Device Session Property Access Protocol. This protocol enables a computer to exchange name-value pairs with a device in an active device session. The Device Session Property Access Protocol uses the Device Services Lightweight Remoting (DSLR) Protocol as specified in [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) to enable the exchange.

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 [[RFC2119]](http://go.microsoft.com/fwlink/?LinkId=90317). 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.

## Glossary

The following terms are specific to this document:

**big-endian**: Multiple-byte values that are byte-ordered with the most significant byte stored in the memory location with the lowest address.

**Component Object Model (COM)**: An object-oriented programming model that defines how objects interact within a single process or between processes. In [**COM**](#gt_ef2ebebc-1760-407a-9ace-af48f9050e02), clients have access to an object through interfaces implemented on the object. For more information, see [[MS-DCOM]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DCOM%5d.pdf).

**payload**: Tag-specific data sent as part of each DSLR message ([[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf)). Each DSLR tag contains one payload. Examples include Dispatcher Request tag payload ([MS-DSLR] section 2.2.2.1) (data identifying the type of request being made on the remote service), dispenser CreateService message payload ([MS-DSLR] section 2.2.2.3) (the parameters for the CreateService function), service-specific function payloads (the parameters for the service-specific functions), and so on.

**proxy**: A network node that accepts network traffic originating from one network agent and transmits it to another network agent.

**stub**: Used as specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824) section 2.1.2.2. A [**stub**](#gt_4cef864e-5108-43a7-a631-68ffc7967546) that is used on the client is called a "client [**stub**](#gt_4cef864e-5108-43a7-a631-68ffc7967546)", and a [**stub**](#gt_4cef864e-5108-43a7-a631-68ffc7967546) that is used on the server is called a "server [**stub**](#gt_4cef864e-5108-43a7-a631-68ffc7967546)".

**UTF-8**: A byte-oriented standard for encoding Unicode characters, defined in the Unicode standard. Unless specified otherwise, this term refers to the UTF-8 encoding form specified in [[UNICODE5.0.0/2007]](http://go.microsoft.com/fwlink/?LinkId=154659) section 3.9.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [[RFC2119]](http://go.microsoft.com/fwlink/?LinkId=90317). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 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](http://msdn.microsoft.com/en-us/library/dn781092.aspx).

### 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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[MS-DSLR] Microsoft Corporation, "[Device Services Lightweight Remoting Protocol](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf)".

[MS-DTYP] Microsoft Corporation, "[Windows Data Types](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DTYP%5d.pdf)".

[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](http://go.microsoft.com/fwlink/?LinkId=90317)

### Informative References

[UPNPAV] UPnP Forum, "UPnP ConnectionManager Service v2", May 2006, [http://upnp.org/specs/av/UPnP-av-ConnectionManager-v2-Service-20060531.pdf](http://go.microsoft.com/fwlink/?LinkId=166314)

## Overview

The Device Session Property Access (DSPA) Protocol is used to exchange name-value pairs between the host computer and the device for Audio-Visual (A/V) and device-capability properties.

This protocol uses the Device Services Lightweight Remoting (DSLR) Protocol, specified in [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) to enable the remoting of services between the two devices over a reliable point-to-point channel.

The [Property Bag service (section 2.2.1)](#Section_1189415e2c9d407fabaa494f29bbd15a) messages, must be implemented and offered by the device (acting in this case as the [**stub**](#gt_4cef864e-5108-43a7-a631-68ffc7967546)) while the host computer acts as the [**proxy**](#gt_2b529701-3e64-4bf8-97ec-15afbba18b73), in DSLR nomenclatures. For a more detailed definition of these roles, please refer to [MS-DSLR]). The Property Bag service contains the following messages or functions:

* [GetStringProperty (section 2.2.1.1)](#Section_4e4a9ef0580f4494a1c6d69d75778117): This function is used to get a string value for the given property described by the property name.
* [GetDWORDProperty (section 2.2.1.3)](#Section_f73db6a243494d28b06f046a296d2373): This function is used to get a DWORD value for the given property described by the property name.
* [SetDWORDProperty (section 2.2.1.2)](#Section_7bafab58c4684322aafaccc7a9f25023): This function is used to set a DWORD value for the given property described by the property name.

As described previously, at any given time, the host computer will act as a client (proxy in DSLR terminology, which invokes the service remotely) and client device will act as a server (stub, which performs the request). In this document, we will refer to the client device as the stub and the host computer as the proxy, but for the sake of simplicity and consistency, in general cases we will always refer the host computer as the "host" and the client device as the "client".

The following block diagram shows the relationship between the host device (that is, the host computer) and the extender device (client).

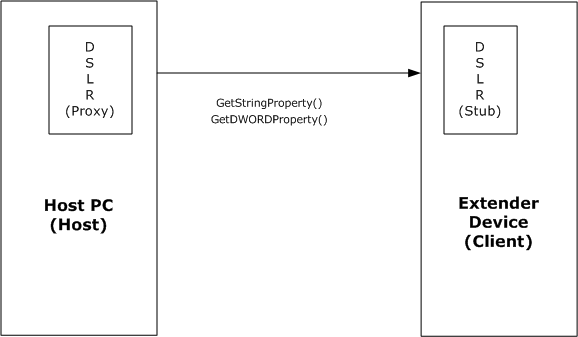


Figure : Device Session Property Access Protocol block diagram

## Relationship to Other Protocols

The Device Session Property Access Protocol uses the Device Services Lightweight Remoting (DSLR) Protocol to exchange name value pairs between the host computer and the device. See [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) for more details on this protocol.

### Device Services Lightweight Remoting (DSLR) Protocol

The Device Services Lightweight Remoting (DSLR) Protocol is a [**Component Object Model (COM)**](#gt_ef2ebebc-1760-407a-9ace-af48f9050e02)-like protocol that enables remoting of services (for example, function calls, events, and so on) over a reliable point-to-point connection. It enables an application to call functions on and/or send events to a remote device over the established channel. The service itself is implemented on the local/stub side of the connection, and the remote side creates a proxy for that service. DSLR is direction agnostic, that is, each side of the connection can act as both a proxy for a remote service and a stub that manages calls into a local service. Both the stub and proxy are implemented by the DSLR consumer; each side has knowledge of the functions/events exposed by the service, as well as the in/out parameters for each. By convention, the request/response calling convention follows COM rules:

* The function returns an HRESULT.
* All [in] parameters are serialized in the request tag.
* The returned HRESULT is serialized in the response tag, followed by the [out] parameters, if successful.
* The caller should expect the returned HRESULT to be either one of the values returned by the function, or one of the DSLR failure values.
* The caller is not required to evaluate any of the [out] parameters if the call returned a failure.

For more information about this protocol, please refer to [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf).

## Prerequisites/Preconditions

For the Device Session Property Access Protocol to function properly, the following requirements must be met:

* A network connection has been established between the host (host computer) and the client device.
* The DSLR modules have been initialized and started on both devices. Once completed, the proxy side calls the **CreateService** request to instantiate the service on the stub side, and creates a proxy for that service (that is, an object that implements the proxied service's interfaces). As part of the **CreateService** request, it allocates a service handle that is sent to the stub side. This handle will subsequently be used when calling functions on the service and to terminate the service via **DeleteService**. See [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) for more information about this process.
  + The following class GUID is passed in the CreateService ([MS-DSLR] section 2.2.2.3) messages for the Property Bag Service:

**ServiceID GUID**: 1EEEDA73-2B68-4d6f-8041-52336CF46072.

* + The following class GUID is passed in the CreateService messages for the Property Bag Service for the AV Property Bag:

**ClassID GUID**: 077bfd3a-7028-4913-bd14-53963dc37754.

* + The following class GUID is passed in the CreateService messages for the Property Bag Service for the Device Capabilities Property Bag:

**ClassID GUID**: EF22F459-6B7E-48ba-8838-E2BEF821DF3C.

## Applicability Statement

The Device Session Property Access Protocol provides the mechanism by which a host computer and a client device can exchange name/value pairs describing the device capabilities and the AV properties.

## Versioning and Capability Negotiation

This protocol has no specific capability negotiation or versioning aspects, aside from the following considerations:

* DSLR extensibility is achieved by:
  + **Adding functions:** Backwards compatible as long as the old functions are kept.

## Vendor-Extensible Fields

This protocol uses HRESULT values as defined in [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) section 2.2.2.5, as well as specific HRESULT values defined in section [2.2](#Section_6486c3c7b9e144da99eb8f4ebf5959fa) of this document.

## Standards Assignments

None.

# Messages

## Transport

Messages are transported over DSLR, which can be implemented on top of any stream-based or message-based reliable transport.

## Message Syntax

This protocol references commonly used data types as defined in [[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DTYP%5d.pdf).

DSLR uses a tag-based formatting for its messages. See [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) for details of the tag formats.

The Device Session Property Access Protocol messages MUST follow the DSLR message syntax for requests and responses, as specified in [MS-DSLR] section 2.2.

The DSLR [**payload**](#gt_ba5b8e95-9bae-4562-af24-dca6e860bc38) for a request is defined by the DSLR Dispatcher Request tag payload, followed by the child payload for a given message (that is, the function parameters for the given message). The Request tag payload includes: the service handle for the specific service (see section [1.5](#Section_a0b840105a3f4c7aaa4aa964cc25eaf5) for how this service handle is obtained), the function handle for the specific function being called on that service (defined by the service), the calling convention for that function, and a one-time request handle allocated by the client for each request. See [MS-DSLR] section 2.2.1 for the format of the DSLR Dispatcher Request tag payload.

The DSLR payload for a response is defined by the DSLR Dispatcher Response tag payload, followed by the child payload of a given message (that is, the result and return parameters for the given message). The Response tag payload includes the *CallingConvention* and the matching one-time request handle to which this response corresponds. See [MS-DSLR] section 2.2.1 for the format of the DSLR Dispatcher Response tag payload.

The format of the data types for input and output parameters for the following functions are defined in [MS-DSLR]. See section 2.2.2.6 for valid input/output parameters and how they are formatted on the wire as [**big-endian**](#gt_6f6f9e8e-5966-4727-8527-7e02fb864e7e).

For more details on the DSLR message syntax, please refer to [MS-DSLR].

### Property Bag Service

The host PC uses this service to exchange property name/value pairs with the client device. In this scenario the host PC has the proxy code to send the messages and the client devices have the stub to receive the messages. After finishing the request, the stub returns the result specified for each message type.

#### GetStringProperty

The GetStringProperty is a two-way request message.

##### GetStringProperty (request)

The *CallingConvention* parameter in the Dispatch Request tag MUST be dslrRequest (0x00000001), as specified in [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) section 2.2.2.1. The function handle for the Dispatch Request tag for GetStringProperty MUST be 0x00000000.

The Request payload (input parameters) is as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PropertyName (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Length (4 bytes):** An unsigned 32-bit integer. The size of the variable represented by **PropertyName**.

**PropertyName (variable):** A [**UTF-8**](#gt_409411c4-b4ed-4ab6-b0ee-6d7815f85a35) string. A generic parameter describing the name for the property. The possible names and the meaning of each of these property names can be found in section [2.2.1.1.1.1](#Section_37642cf690ae408793b9bceb4bf30a25) and section [2.2.1.1.1.2](#Section_de407e7348e94213ab0c2f9f1ad9c5df).

###### AV Property Bag

The property name specific to AV and its respective value types for GetStringProperty are shown in the following table.

| Property name | Meaning | Valid values |
| --- | --- | --- |
| XspHostAddress | XSP host IP address | The numeric host address string is a dotted-decimal IPv4 address or an IPv6 hex address. |

###### Device Capabilities PropertyBag

The property name specific to Device capabilities and their respective value types for GetStringProperty are shown in the following table.

| Property name | Meaning | Valid values |
| --- | --- | --- |
| NAM | Client name | A UTF-8 string with a value of "McxClient". |
| PRT | Protocol information | A UTF-8 string with a maximum size of 2048. |
| XTY | Device type | A UTF-8 string with a maximum size of 2048. |
| PBV | Device build version | A UTF-8 string with a maximum size of 2048. |

Additional Property Descriptions:

**PRT:** The protocol information specifies the media types supported by the device. The formatting of the string describing the protocol information can be found in [7](#Section_481c7afda00f4450b8b0bdc07af4354a).

**XTY:** As mentioned earlier, this property can be any UTF-8 string with a maximum size of 2048 and MUST not begin with X.

**PBV:** The value for the build version is arbitrary.

##### GetStringProperty (response)

The *CallingConvention* parameter in the Dispatch Response tag MUST be dslrResponse (0x00000002), as specified in [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) section 2.2.2.2.

The Response payload (result and output parameters) is as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Result | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PropertyValue (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Result (4 bytes):** An unsigned 32-bit integer. HRESULT is returned from the function call. See [MS-DSLR] for the definitions of error codes. The following return values are specific to the Device Session Property Access Protocol.

| Condition | HRESULT |
| --- | --- |
| Property exists | S\_OK |
| Property does not exist | S\_FALSE |
| Interface is not implemented | E\_NOTIMPL |

**Length (4 bytes):** An unsigned 32-bit integer. The size of the variable represented by PropertyValue.

**PropertyValue (variable):** A UTF8 String. An appropriate property value based on the property name is returned with the result.[<1>](#Appendix_A_1) Constraints for these values are described in section [2.2.1.1.1.1](#Section_37642cf690ae408793b9bceb4bf30a25) and section [2.2.1.1.1.2](#Section_de407e7348e94213ab0c2f9f1ad9c5df).

#### SetDWORDProperty

SetDWORDProperty is a two-way request message.

##### SetDWORDProperty (request)

The *CallingConvention* parameter in the Dispatch Request tag MUST be dslrRequest (0x00000001), as specified in [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) section 2.2.2.1. The function handle for the Dispatch Request tag for [SetDWORDProperty (section 2.2.1.2)](#Section_7bafab58c4684322aafaccc7a9f25023) MUST be 0x00000003.

The Request payload (input parameters) is as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PropertyName (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PropertyValue | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Length (4 bytes):** An unsigned 32-bit integer. The size of the variable represented by PropertyName.

**PropertyName (variable):** A UTF-8 string. A generic parameter describing the name for the property. The name and meaning of each of these property names is shown in section [2.2.1.2.1.1](#Section_8e6ee7238bba40cdb1e794d59b91c06b).

**PropertyValue (4 bytes):** An unsigned 32-bit integer. An appropriate property value based on the **PropertyName** is set. Constraints for these values are mentioned in section 2.2.1.2.1.1.

###### AV Property Bag

The property name specific to AV and the respective value types for SetDWORDProperty are shown in the following table.

| Property Name | Meaning | Valid Values |
| --- | --- | --- |
| IsMuted | Boolean value showing if the volume is mute | 0 or 1 |
| Volume | Value representing the volume level | 0 - 65535 |

###### Device Capabilities PropertyBag

There are no property names specific to the Device capabilities for SetDWORDProperty.

##### SetDWORDProperty (response)

The callingConvention parameter in the Dispatch Response tag MUST be dslrResponse (0x00000002), as specified in [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) section 2.2.2.2.

The Response payload (result) is as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Result | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Result (4 bytes):** An unsigned 32-bit integer. HRESULT is returned from the function call. See [MS-DSLR] for the definitions of possible error codes. The following return values are specific to the Device Session Property Access Protocol.

| Condition | HRESULT |
| --- | --- |
| Property exists | S\_OK |
| Property does not exist | S\_FALSE |
| Interface is not implemented | E\_NOTIMPL |

There are no output parameters for this request.

#### GetDWORDProperty

GetDWORDProperty is a two-way request message.

##### GetDWORDProperty (request)

The *CallingConvention* parameter in the Dispatch Request tag MUST be dslrRequest (0x00000001), as specified in [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) section 2.2.2.1. The function handle for the Dispatch Request tag for [GetDWORDProperty (section 2.2.1.3)](#Section_f73db6a243494d28b06f046a296d2373) MUST be 0x00000002.

The Request payload (input parameters) is as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PropertyName (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Length (4 bytes):** An unsigned 32-bit integer. The size of the variable represented by **PropertyName**.

**PropertyName (variable):** A UTF-8 string. A generic parameter describing the name for the property. The name and meaning of each of these property names is described in section [2.2.1.3.1.1](#Section_78432675d2dc4577843607b0bf01392c) and section [2.2.1.3.1.2](#Section_0f625fd427e4407fbcda4e3197aaa994).

###### AV Property Bag

The property name specific to AV and the respective value types for GetDWORDProperty are shown in the following table.

| Property Name | Meaning | Valid Values |
| --- | --- | --- |
| IsMuted | Boolean value showing if the volume is mute | 0 or 1 |
| WmvTrickModesSupported | Boolean value specifying if the Trick mode is supported | 0 or 1 |
| Volume | Value representing the volume level | 0 - 65535 |

###### Device Capabilities PropertyBag

The property names specific to the device capabilities and their respective value types for GetDWORDProperty are shown in the following table.

| Property Name | Meaning | Valid Values |
| --- | --- | --- |
| PHO | Are advanced photo features allowed? | True or False |
| EXT | Are Extender Settings allowed? | True or False |
| MAR | Are over-scan margins needed? | True or False |
| POP | Are pop ups allowed? | True or False |
| ZOM | Is video zoom mode allowed? | True or False |
| NLZ | Is nonlinear zoom supported? | True or False |
| RSZ | Is raw stretched zoom supported? | True or False |
| WID | Is wide screen enabled? | True or False |
| H10 | Is 10 feet help allowed? | True or False |
| WEB | Is 10 feet web content allowed | True or False |
| H02 | Is 2 feet help allowed? | True or False |
| WE2 | Is 2 feet web content allowed? | True or False |
| AUD | Is audio allowed? | True or False |
| AUR | Is audio Non WMP? | True or False |
| ARA | Is auto restart allowed? | True or False |
| BLB | Is black letters box needed? | True or False |
| CCC | Is CC rendered by the client? | True or False |
| CRC | Is CD burning allowed? | True or False |
| CPY | Is CD copying allowed? | True or False |
| CDA | Is CD playback allowed? | True or False |
| CLO | Is the close button shown? | True or False |
| DRC | Is DVD burning allowed? | True or False |
| DVD | Is DVD playback allowed? | True or False |
| FPD | Is FPD allowed? | True or False |
| GDI | Is GDI renderer used? | True or False |
| HDV | Is HD content allowed? | True or False |
| HDN | Is HD content allowed by the network? | True or False |
| SDN | Is SD content allowed by the network | True or False |
| REM | Is input treated as if from a remote? | True or False |
| ANI | Is intensive animation allowed? | True or False |
| 2DA | Is 2-D animation allowed? | True or False |
| HTM | Is HTML supported? | True or False |
| DES | Is MCE a Windows shell? | True or False |
| DOC | Is My Documents populated? | True or False |
| SCR | Is a native screensaver required? | True or False |
| ONS | Is an online spotlight allowed? | True or False |
| SUP | Is RDP super blt allowed? | True or False |
| BIG | Is remote UI renderer big-endian? | True or False |
| RUI | Is remote UI rendering supported? | True or False |
| SDM | Is a screen data mode workaround needed? | True or False |
| TBA | Is a Toolbar allowed? | True or False |
| SYN | Is transfer to a device allowed? | True or False |
| APP | Is tray applet allowed? | True or False |
| TVS | Is a TV skin used? | True or False |
| SOU | Is UI sound supported? | True or False |
| VID | Is video allowed? | True or False |
| W32 | Is Win32 content allowed? | True or False |
| WIN | Is window mode allowed? | True or False |
| VIZ | Is WMP visualization allowed? | True or False |
| VOL | Is volume UI allowed? | True or False |
| MUT | Is mute UI allowed? | True or False |

##### GetDWORDProperty (response)

The *CallingConvention* parameter in the Dispatch Response tag MUST be dslrResponse (0x00000002), as specified in [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf) section 2.2.2.2.

The Response payload (result and output parameters) is as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Result | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PropertyValue | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Result (4 bytes):** An unsigned 32-bit integer. An HRESULT is returned from the function call. See [MS-DSLR] section 2.2.2.5 for the definitions of possible error codes. The following return values are specific to the Device Session Property Access Protocol.

| Condition | HRESULT |
| --- | --- |
| Property exists | S\_OK |
| Property does not exist | S\_FALSE |
| Interface is not implemented | E\_NOTIMPL |

**PropertyValue (4 bytes):** An unsigned 32-bit integer. An appropriate property value based on the property name is returned with the result.[<2>](#Appendix_A_2) Constraints for these values are described in section [2.2.1.3.1.1](#Section_78432675d2dc4577843607b0bf01392c) and section [2.2.1.3.1.2](#Section_0f625fd427e4407fbcda4e3197aaa994).

# Protocol Details

For the Device Session Property Access Protocol, the client device is the stub and the host computer is the proxy.

## Device Details

The device is the stub waiting to receive service messages. Upon receiving the service messages it processes them and returns the responses to the proxy.

The Device Session Property Access Protocol device has the following states, as illustrated in the following figure:

1. Start
2. Accepting request
3. Processing request
4. Finish

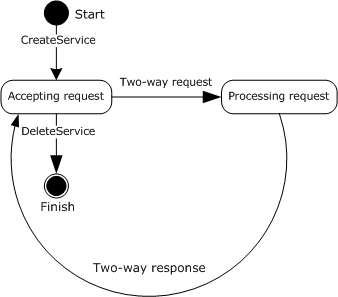


Figure : Device state machine

**Start State**

The device is ready to instantiate services. The following message is processed in this state:

* CreateService

**Accepting Messages**

The device has received the CreateService message to instantiate the service, and is ready to accept requests on that service. The following events are processed in this state:

* Two-way request
* DeleteService

**Processing Request**

The device is executing a two-way request received from the host, including sending the response for two-way requests. The following event is processed in this state:

* Two-way response

**Finish State**

The device has received the DeleteService message and cleaned up the remote service. No events are processed in this state.

### Abstract Data Model

None.

### Timers

None.

### Initialization

Before the PropertyBag service can function, DSLR MUST be started and initialized on the client device. Furthermore, the device has to call CreateService on itself to instantiate all PropertyBag services between the host computer and the device.

### Higher-Layer Triggered Events

The client side of this protocol is simply a pass-through. That is, 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.

### Processing Events and Sequencing Rules

The following sections describe the states and events outlined in [3.1.1](#Section_25054dac99ff41079982d341a84ec0fd).

#### Create Service

This event is described in [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf).

#### Two-Way Requests

The following two-way requests are possible for Device Session Property Access Protocol.

* GetStringProperty
* GetDWORDProperty
* SetDWORDProperty

Further explanation for each of these requests is described in section [2.2.1](#Section_1189415e2c9d407fabaa494f29bbd15a).

#### Delete Service

This event is described in [[MS-DSLR]](file:///E:\Target\Windows\Published\Books\MS-DSPA\%5bMS-DSLR%5d.pdf).

### Timer Events

None.

### Other Local Events

None.

## Host Computer Details

The host computer is the client side of this protocol and is simply a pass-through. That is, 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.

# Protocol Examples

The following list shows the sequence of Device Session Property Access Protocol messages that pass over the wire after the host computer and client device have established a connection.

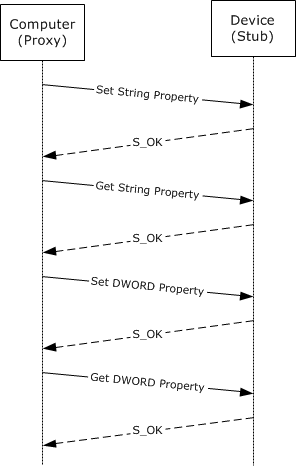


Figure : DSPA Protocol sequence diagram

1. The host sends the SetStringProperty message.
2. The client responds with S\_OK when the property exists.
3. The host sends the GetStringProperty message.
4. The client responds with S\_OK when the property exists.
5. The host sends the SetDWORDProperty message.
6. The client responds with S\_OK when the property exists.
7. The host sends the GetDWORDProperty message.
8. The client responds with S\_OK when the property exists.

# Security

## Security Considerations for Implementers

None.

## Index of Security Parameters

None.

# 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.

* Extenders for Windows Media Center
* Windows Vista operating system
* Windows 7 operating system
* Windows 8 operating system
* Windows 8.1 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 2.2.1.1.2](#Appendix_A_Target_1): Media Center on Windows Vista does not support the following properties for the [GetStringProperty](#Section_4e4a9ef0580f4494a1c6d69d75778117) function.

| Property name | Meaning | Valid values |
| --- | --- | --- |
| PRT | Protocol information | UTF-8 string with a maximum size of 2048. |
| XTY | Device type | UTF-8 string with a maximum size of 2048. |
| PBV | Device build version | UTF-8 string with a maximum size of 2048. |

[<2> Section 2.2.1.3.2](#Appendix_A_Target_2): Media Center on Windows Vista does not support the following property for the GetDWORDProperty function.

| Property name | Meaning | Valid values |
| --- | --- | --- |
| RSZ | Is raw stretched zoom supported? | True or False |

Media Center on Windows 7, Windows 8, and Windows 8.1 does not support the following property for the GetDWORDProperty function.

| Property name | Meaning | Valid values |
| --- | --- | --- |
| SDM | Is screen data mode workaround needed? | True or False |

# Appendix B: Protocol Information String (PRT)

Partner applications can query Media Center Extender devices to determine the supported Audio Visual (AV) media file formats supported by those devices. Devices use the PRT string property to expose the supported AV formats of the specific device.

The format for the protocol information string is mentioned in <section 2.5.2> ProtocolInfo Concept in [[UPNPAV]](http://go.microsoft.com/fwlink/?LinkId=166314).

The <additional info> field is a name value pair separated by ";". The name value pairs follow the format of <org-name>\_<token-name>=<value>. Additional information for the same can be found in <section 2.5.2.1> 4th Field - <additionalInfo> [UPNPAV].

The <org-name>\_<token-name> values used are:

* DLNA.ORG\_PN
* MICROSOFT.COM\_PN

The following table shows all the supported media types for <org-name>="MICROSOFT.COM" and <org-name>="DLNA.ORG" respectively. The terms used in the supported column are further described in subsequent tables.

| <org-name>="MICROSOFT.COM" | Video media type | Audio media type |
| --- | --- | --- |
| WMALSL | N/A | MTG\_WMA\_LOSSLESS |
| WAV\_PCM | N/A | MTG\_PCM |
| DVRMS\_MPEG2 | MTG\_MPV | MTG\_MPA |
| DVRMS\_MPEG2 | N/A | MTG\_AC3 |
| VC1\_APL2\_FULL | MTG\_VC1 | MTG\_WMA\_STD |
| VC1\_APL2\_PRO | MTG\_VC1 | MTG\_WMA\_PRO |
| VC1\_APL3\_FULL | MTG\_VC1 | MTG\_WMA\_STD |
| VC1\_APL3\_PRO | MTG\_VC1 | MTG\_WMA\_PRO |
| MPEG4\_P2\_MP4\_ASP\_L5\_MPEG1\_L3 | MTG\_MPEG4P2 | MTG\_MP3 |
| MPEG4\_P2\_AVI\_ASP\_L5\_MPEG1\_L3 | MTG\_MPEG4P2 | MTG\_MP3 |
| MPEG4\_P2\_MP4\_ASP\_L5\_AC3 | MTG\_MPEG4P2 | MTG\_AC3 |
| MPEG4\_P2\_AVI\_ASP\_L5\_AC3 | MTG\_MPEG4P2 | MTG\_AC3 |
| AVC\_AVI\_MP\_HD\_L4\_1\_MPEG1\_L3 | MTG\_MPEG4P10 | MTG\_MP3 |
| AVC\_MP4\_MP\_HD\_MPEG1\_L3 | MTG\_MPEG4P10 | MTG\_MP3 |
| AVC\_MP4\_MP\_HD\_AC3 | MTG\_MPEG4P10 | MTG\_AC3 |
| AVC\_AVI\_MP\_HD\_L4\_1\_AC3 | MTG\_MPEG4P10 | MTG\_AC3 |

| <org-name>="DLNA.ORG" | Video media type | Audio media type |
| --- | --- | --- |
| WMABASE | N/A | MTG\_WMA\_STD |
| WMAFULL | N/A | MTG\_WMA\_STD |
| WMAPRO | N/A | MTG\_WMA\_PRO |
| MP3 | N/A | MTG\_MP3 |
| AC3 | N/A | MTG\_AC3 |
| LPCM | N/A | MTG\_PCM |
| MPEG\_ES\_PAL | MTG\_MPV | N/A |
| MPEG\_ES\_NTSC | MTG\_MPV | N/A |
| MPEG\_ES\_PAL\_XAC3 | MTG\_MPV | MTG\_AC3 |
| MPEG\_ES\_NTSC\_XAC3 | MTG\_MPV | MTG\_AC3 |
| WMVMED\_BASE | MTG\_WMV | MTG\_WMA\_STD |
| WMVMED\_FULL | MTG\_WMV | MTG\_WMA\_STD |
| WMVMED\_PRO | MTG\_WMV | MTG\_WMA\_PRO |
| WMVHIGH\_FULL | MTG\_WMV | MTG\_WMA\_STD |
| WMVHIGH\_PRO | MTG\_WMV | MTG\_WMA\_PRO |
| WMVSPLL\_BASE | MTG\_WMV | MTG\_WMA\_STD |
| WMVSPML\_BASE | MTG\_WMV | MTG\_WMA\_STD |
| WMVSPML\_MP3 | MTG\_WMV | MTG\_MP3 |
| MPEG1 | MTG\_MPV | MTG\_MPA |
| MPEG\_PS\_NTSC | MTG\_MPV | MTG\_AC3 |
| MPEG\_PS\_NTSC | N/A | MTG\_PCM |
| MPEG\_PS\_NTSC | N/A | MTG\_MPA |
| MPEG\_PS\_PAL | MTG\_MPV | MTG\_AC3 |
| MPEG\_PS\_PAL | MTG\_MPV | MTG\_PCM |
| MPEG\_PS\_PAL | MTG\_MPV | MTG\_MPA |
| MPEG4\_P2\_TS\_ASP\_MPEG1\_L3 | MTG\_MPEG4P2 | MTG\_MP3 |
| MPEG4\_P2\_TS\_ASP\_AC3 | MTG\_MPEG4P2 | MTG\_AC3 |
| AVC\_MP4\_MP\_SD\_MPEG1\_L3 | MTG\_MPEG4P10 | MTG\_MP3 |
| AVC\_TS\_MP\_HD\_MPEG1\_L3 | MTG\_MPEG4P10 | MTG\_MP3 |
| AVC\_MP4\_MP\_HD\_AC3 | MTG\_MPEG4P10 | MTG\_AC3 |
| AVC\_MP4\_MP\_SD\_AC3 | MTG\_MPEG4P10 | MTG\_AC3 |
| AVC\_TS\_MP\_HD\_AC3 | MTG\_MPEG4P10 | MTG\_AC3 |

The following tables shows the media type mappings from the previous table in the supported video and audio column based on the protocol used.

HTTP

| Media type | Meaning | Default value |
| --- | --- | --- |
| MTG\_MPA | MPEG Audio | Yes |
| MTG\_AC3 | AC3 Audio | Yes |
| MTG\_AAC | AAC Audio | Yes |
| MTG\_HE\_AAC | AAC HE Audio | Yes |
| MTG\_PCM | PCM Audio | Yes |
| MTG\_MP3 | MP3 Audio | Yes |
| MTG\_MPV | MPEG 1/2 Video | Yes |
| MTG\_WMV | WMV Video | Yes |
| MTG\_VC1 | VC-1 video | Yes |
| MTG\_MPEG4P10 | MPEG 4 Part 10 Video | Yes |
| MTG\_MPEG4P2 | MPEG 4 Part 2 Video | Yes |

RTP

| Media type | Meaning | Default value |
| --- | --- | --- |
| MTG\_MPA | MPEG Audio | Yes |
| MTG\_AC3 | AC3 Audio | No |
| MTG\_PCM | PCM Audio | No |
| MTG\_WMA\_STD | WMA Audio Std | Yes |
| MTG\_WMA\_PRO | WMA Audio Pro | Yes |
| MTG\_WMA\_LOSSLESS | WMA-Lossless Audio | Yes |
| MTG\_MP3 | MP3 Audio | Yes |
| MTG\_MPV | MPEG 1 and MPEG 2 Video | Yes |
| MTG\_WMV | WMV Video | Yes |
| MTG\_VC1 | VC-1 video | Yes |
| MTG\_MPEG4P10 | MPEG 4 Part 10 Video | No |
| MTG\_MPEG4P2 | MPEG 4 Part 2 Video | No |

The "Default Value" in the previous tables is supported when the PRT string query returns a "null".

The protocol information is used to describe the string formatted as:< protocol >" :"< network >" :"< contentFormat >" :"< additional Info >

The sample protocol information string based on the format above is shown here:

rtsp-rtp-udp:\*:audio/x-ms-wma:DLNA.ORG\_PN=WMAFULL;

DLNA.ORG\_PN=WMAPRO;MICROSOFT.COM\_PN=WMALSL

rtsp-rtp-udp:\*:audio/mpeg:DLNA.ORG\_PN=MP3

http-get:\*:audio/L16:MICROSOFT.COM\_PN=WAV\_PCM

rtsp-rtp-udp:\*:video/mpeg:MICROSOFT.COM\_PN=DVRMS\_MPEG2

rtsp-rtp-udp:\*:video/x-ms-wmv:DLNA.ORG\_PN=WMVHIGH\_PRO;

MICROSOFT.COM\_PN=WMVHIGH\_LSL;DLNA.ORG\_PN=WMVHIGH\_FULL;

MICROSOFT.COM\_PN=VC1\_APL2\_FULL;MICROSOFT.COM\_PN=VC1\_APL2\_PRO;

MICROSOFT.COM\_PN=VC1\_APL2\_LSL;MICROSOFT.COM\_PN=WMVIMAGE1\_MED;

MICROSOFT.COM\_PN=WMVIMAGE2\_MED

http-get:\*:video/mpeg:DLNA.ORG\_PN=MPEG1;

DLNA.ORG\_PN=MPEG\_PS\_NTSC;DLNA.ORG\_PN=MPEG\_PS\_PAL

For more information about the formatting of the protocol information string and the naming convention please refer to [UPNPAV].

# Change Tracking

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

# Index

A

[Abstract data model](#section_25054dac99ff41079982d341a84ec0fd) 18

[Applicability](#section_1cc532161c1c41b887c7fb35851cd3ec) 9

C

[Capability negotiation](#section_3fa6ac01a20e454d8deeeb248e4d2bef) 9

[Change tracking](#section_15c587139c4f4a199c75dfa2e515ff1f) 28

CreateService

[event](#section_153fee4a35a94fe89790bd821fab4a70) 18

D

DeleteService

[event](#section_8ae4f7919b404d23bc2dd2313dfff7de) 19

[Device - overview](#section_da24d1d3a9274821aea80ffae1378254) 17

[Device Services Lightweight Remoting (DSLR) Protocol - relationship to other protocols](#section_8700195c2f144ef38ebc04000c22207f) 8

E

[Examples - overview](#section_06f7db27d7564834b489001179319ceb) 20

F

[Fields - vendor-extensible](#section_7a0e464b12bf4202b66f65faba173e40) 9

G

[GetDWORDProperty message](#section_f73db6a243494d28b06f046a296d2373) 13

[GetDWORDProperty\_request packet](#section_0642b5dbf9cf49cb9be2325794e4defa) 13

[GetDWORDProperty\_response packet](#section_860ddfc121d647739ead84a4adc9de22) 16

[GetStringProperty message](#section_4e4a9ef0580f4494a1c6d69d75778117) 10

[GetStringProperty\_request packet](#section_bc52a4ae62044f5e9f1ebc00cd6a781e) 10

[GetStringProperty\_response packet](#section_89f9878b4cd0455cad4f7cea7691f63f) 11

[Glossary](#section_8d4e582115864c69ba35a71b407b8220) 6

H

[Higher-layer triggered events](#section_cb1c7b5c83ea4bf793bbe59ab8b58405) 18

I

[Implementer - security considerations](#section_4ab2c07a11f944e2b152350157f240dc) 22

[Index of security parameters](#section_9d63e578b61d483e986f8911ae9ef834) 22

[Informative references](#section_b80641cce0a6420ca80f84d88c1da545) 7

[Initialization](#section_08fb3507a7ba4876b7b74bb1c618eb4b) 18

[Introduction](#section_37c4ce519bc84df0a993d86d3c66b205) 6

L

[Local events](#section_a4398e0f3db844399e19a381b9224ba0) 19

M

[Message processing - overview](#section_52226d954d8f4e1e964c44839c28e2b6) 18

Messages

[GetDWORDProperty message](#section_f73db6a243494d28b06f046a296d2373) 13

[GetStringProperty message](#section_4e4a9ef0580f4494a1c6d69d75778117) 10

[Property Bag Service](#section_1189415e2c9d407fabaa494f29bbd15a) 10

[SetDWORDProperty message](#section_7bafab58c4684322aafaccc7a9f25023) 12

[transport](#section_452d070d71ac428886ccd6cefd9b413b) 10

N

[Normative references](#section_3f04349a70234ab89846a0ba9c8ea576) 6

O

[Overview (synopsis)](#section_deda8d7de6b0431e8b96afce5909f10c) 7

P

[Parameters - security index](#section_9d63e578b61d483e986f8911ae9ef834) 22

[Preconditions](#section_a0b840105a3f4c7aaa4aa964cc25eaf5) 9

[Prerequisites](#section_a0b840105a3f4c7aaa4aa964cc25eaf5) 9

[Product behavior](#section_878d6bb415eb40dabad47317c4d835ed) 23

[Property Bag Service](#section_1189415e2c9d407fabaa494f29bbd15a) 10

[Property Bag Service message](#section_1189415e2c9d407fabaa494f29bbd15a) 10

Protocol Details

[overview](#section_50254b01a0e54d72a6b76267ecc39699) 17

[Protocol Information String (PRT)](#section_481c7afda00f4450b8b0bdc07af4354a) 24

R

[References](#section_0e8394a65d784f608547791b0e81045d) 6

[informative](#section_b80641cce0a6420ca80f84d88c1da545) 7

[normative](#section_3f04349a70234ab89846a0ba9c8ea576) 6

[Relationship to other protocols](#section_43d0df08205143028d52cdd549ab7640) 8

[Device Services Lightweight Remoting (DSLR) Protocol](#section_8700195c2f144ef38ebc04000c22207f) 8

[overview](#section_43d0df08205143028d52cdd549ab7640) 8

S

Security

[implementer considerations](#section_4ab2c07a11f944e2b152350157f240dc) 22

[parameter index](#section_9d63e578b61d483e986f8911ae9ef834) 22

[Sequencing rules - overview](#section_52226d954d8f4e1e964c44839c28e2b6) 18

[SetDWORDProperty message](#section_7bafab58c4684322aafaccc7a9f25023) 12

[SetDWORDProperty\_request packet](#section_aca77a94985543568c5f34d6b4221e35) 12

[SetDWORDProperty\_response packet](#section_1a2d837c222843b68165d868fc2ef269) 13

[Standards assignments](#section_450587f270a14f8b9d33fa2d3720e071) 9

T

[Timer events](#section_803769d9d8364e92b1b2d2ad39ae4b3f) 19

[Timers](#section_c7b97436eee5457f85172f9446e0de84) 18

[Tracking changes](#section_15c587139c4f4a199c75dfa2e515ff1f) 28

[Transport](#section_452d070d71ac428886ccd6cefd9b413b) 10

[Triggered events - higher-layer](#section_cb1c7b5c83ea4bf793bbe59ab8b58405) 18

[Two-way requests](#section_e8a05db179ce45958d21370d9bbf4f27) 18

V

[Vendor-extensible fields](#section_7a0e464b12bf4202b66f65faba173e40) 9

[Versioning](#section_3fa6ac01a20e454d8deeeb248e4d2bef) 9