[MS-DSPA]:

Device Session Property Access Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Support. For questions and support, please contact dochelp@microsoft.com.

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.

Date	Revision History	Revision Class	Comments
5/15/2014	4.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	4.0	None	No changes to the meaning, language, or formatting of the technical content.
10/16/2015	4.0	None	No changes to the meaning, language, or formatting of the technical content.
7/14/2016	4.0	None	No changes to the meaning, language, or formatting of the technical content.
6/1/2017	4.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 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] to enable the exchange.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

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**, clients have access to an object through interfaces implemented on the object. For more information, see [MS-DCOM].

payload: Tag-specific data sent as part of each DSLR message ([MS-DSLR]). 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] section 2.1.2.2. A **stub** that is used on the client is called a "client **stub**", and a **stub** that is used on the server is called a "server **stub**".

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] section 3.9.

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

1.2 References

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

1.2.1 Normative References

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

[MS-DSLR] Microsoft Corporation, "Device Services Lightweight Remoting Protocol".

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

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

1.2.2 Informative References

[UPNPAV] UPnP Forum, "UPnP ConnectionManager Service v2", May 2006, http://upnp.org/specs/av/UPnP-av-ConnectionManager-v2-Service-20060531.pdf

1.3 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] 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) messages, are implemented and offered by the device (acting in this case as the **stub**) while the host computer acts as the **proxy**, 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): This function is used to get a string value for the given property described by the property name.
- GetDWORDProperty (section 2.2.1.3): This function is used to get a DWORD value for the given property described by the property name.
- SetDWORDProperty (section 2.2.1.2): 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 1: Device Session Property Access Protocol block diagram

1.4 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] for more details on this protocol.

1.4.1 Device Services Lightweight Remoting (DSLR) Protocol

The Device Services Lightweight Remoting (DSLR) Protocol is a **Component Object Model (COM)**-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 expects 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].

1.5 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] 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.

1.6 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.

1.7 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.

1.8 Vendor-Extensible Fields

This protocol uses HRESULT values as defined in [MS-DSLR] section 2.2.2.5, as well as specific HRESULT values defined in section 2.2 of this document.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

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

2.2 Message Syntax

This protocol references commonly used data types as defined in [MS-DTYP].

DSLR uses a tag-based formatting for its messages. See [MS-DSLR] 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** 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 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**.

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

2.2.1 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.

2.2.1.1 GetStringProperty

The GetStringProperty is a two-way request message.

2.2.1.1.1 GetStringProperty (request)

The *CallingConvention* parameter in the Dispatch Request tag MUST be dslrRequest (0x00000001), as specified in [MS-DSLR] 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	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	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 possible names and the meaning of each of these property names can be found in section 2.2.1.1.1.1 and section 2.2.1.1.1.2.

2.2.1.1.1.1 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.

2.2.1.1.1.2 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.

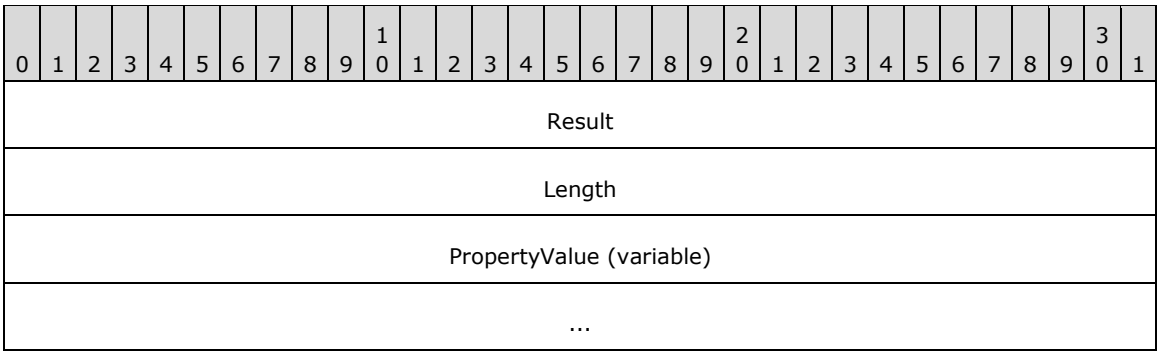
XTY: As mentioned earlier, this property can be any UTF-8 string with a maximum size of 2048 and MUST not begin with X.

PBV: The value for the build version is arbitrary.

2.2.1.1.2 GetStringProperty (response)

The *CallingConvention* parameter in the Dispatch Response tag MUST be dslrResponse (0x00000002), as specified in [MS-DSLR] section 2.2.2.2.

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



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> Constraints for these values are described in section 2.2.1.1.1.1 and section 2.2.1.1.1.2.

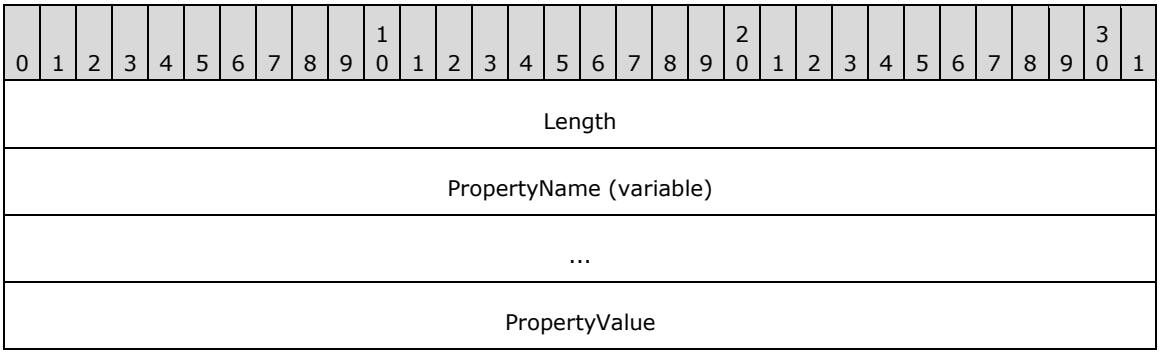
2.2.1.2 SetDWORDProperty

SetDWORDProperty is a two-way request message.

2.2.1.2.1 SetDWORDProperty (request)

The *CallingConvention* parameter in the Dispatch Request tag MUST be dslrRequest (0x00000001), as specified in [MS-DSLR] section 2.2.2.1. The function handle for the Dispatch Request tag for SetDWORDProperty (section 2.2.1.2) MUST be 0x00000003.

The Request payload (input parameters) is as follows.



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.

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.

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

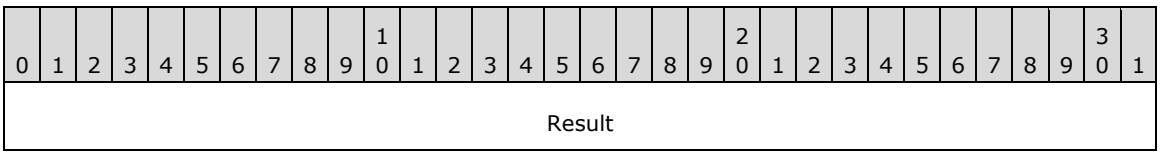
2.2.1.2.1.2 Device Capabilities PropertyBag

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

2.2.1.2.2 SetDWORDProperty (response)

The callingConvention parameter in the Dispatch Response tag MUST be dslrResponse (0x00000002), as specified in [MS-DSLR] section 2.2.2.2.

The Response payload (result) is as follows.



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.

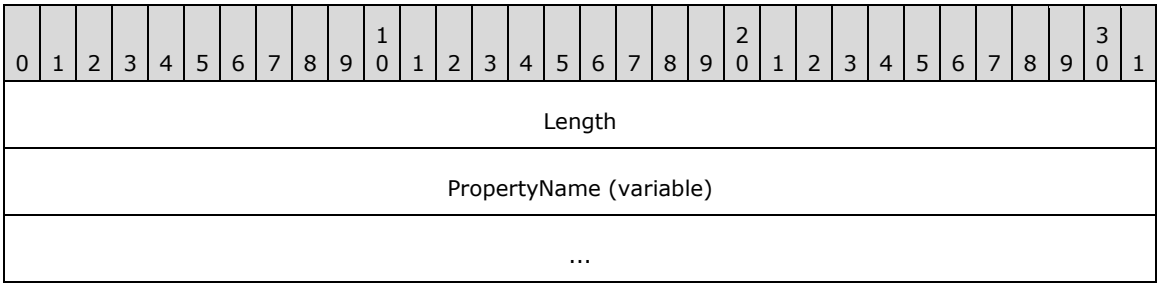
2.2.1.3 GetDWORDProperty

GetDWORDProperty is a two-way request message.

2.2.1.3.1 GetDWORDProperty (request)

The *CallingConvention* parameter in the Dispatch Request tag MUST be dslrRequest (0x00000001), as specified in [MS-DSLR] section 2.2.2.1. The function handle for the Dispatch Request tag for GetDWORDProperty (section 2.2.1.3) MUST be 0x00000002.

The Request payload (input parameters) is as follows.



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 and section 2.2.1.3.1.2.

2.2.1.3.1.1 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

2.2.1.3.1.2 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

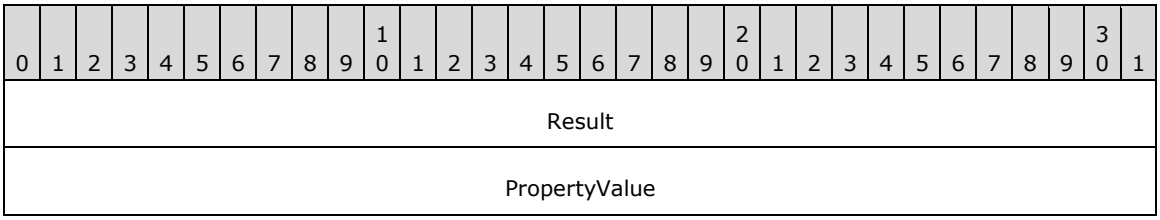
Property Name	Meaning	Valid Values
H10	Is 10 feet help allowed?	True or False
WEB	Is 10 feet web content allowed	True or False
H02	Is 2 feet help allowed?	True or False
WE2	Is 2 feet web content allowed?	True or False
AUD	Is audio allowed?	True or False
AUR	Is audio Non WMP?	True or False
ARA	Is auto 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

Property Name	Meaning	Valid Values
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

2.2.1.3.2 GetDWORDProperty (response)

The *CallingConvention* parameter in the Dispatch Response tag MUST be dslrResponse (0x00000002), as specified in [MS-DSLR] section 2.2.2.2.

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



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> Constraints for these values are described in section 2.2.1.3.1.1 and section 2.2.1.3.1.2.

3 Protocol Details

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

3.1 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

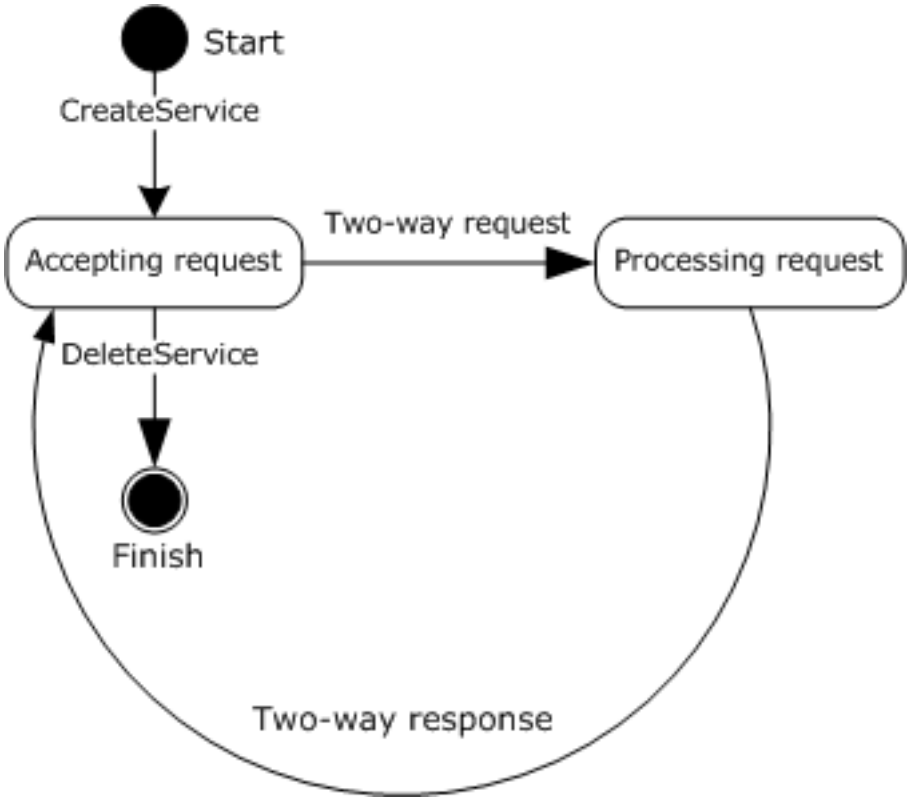


Figure 2: Device state machine

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.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 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.

3.1.4 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.

3.1.5 Processing Events and Sequencing Rules

The following sections describe the states and events outlined in 3.1.1.

3.1.5.1 Create Service

This event is described in [MS-DSLR].

3.1.5.2 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.

3.1.5.3 Delete Service

This event is described in [MS-DSLR].

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 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.

4 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.

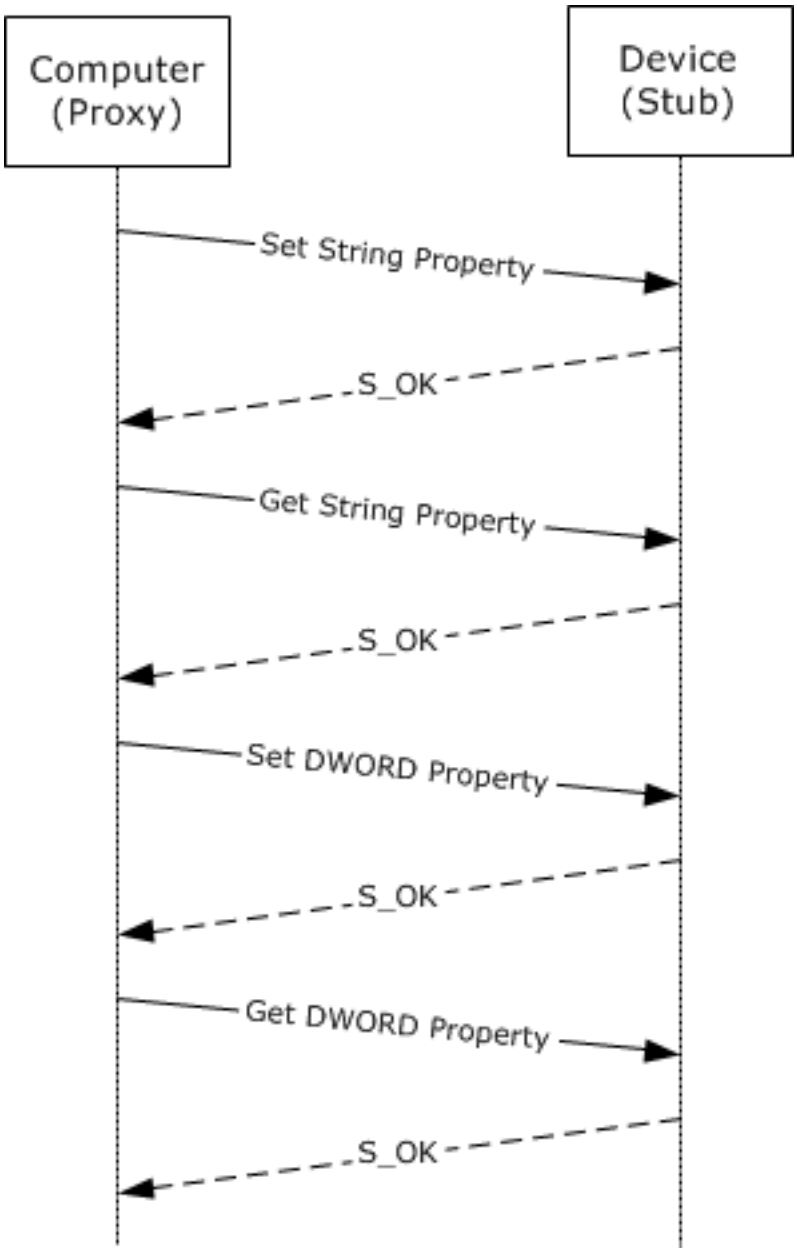


Figure 3: DSPA Protocol sequence diagram

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.

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 updates to those products.

- 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 in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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: Media Center on Windows Vista does not support the following properties for the GetStringProperty 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: 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

7 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].

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].

8 Change Tracking

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

9 Index