
RADAConfig:2 Service

For UPnP Version 1.0

Status: Standardized DCP

Date: April 30, 2011

Document Version: 1.0

Service Template Version: 2.00

This Standardized DCP has been adopted as a Standardized DCP by the Steering Committee of the UPnP Forum, pursuant to Section 2.1(c)(ii) of the UPnP Forum Membership Agreement. UPnP Forum Members have rights and licenses defined by Section 3 of the UPnP Forum Membership Agreement to use and reproduce the Standardized DCP in UPnP Compliant Devices. All such use is subject to all of the provisions of the UPnP Forum Membership Agreement.

THE UPNP FORUM TAKES NO POSITION AS TO WHETHER ANY INTELLECTUAL PROPERTY RIGHTS EXIST IN THE PROPOSED SERVICES, IMPLEMENTATIONS OR IN ANY ASSOCIATED TEST SUITES. THE PROPOSED SERVICES, STANDARDIZED SERVICES, IMPLEMENTATIONS AND ANY ASSOCIATED TEST SUITES ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE PROPOSED SERVICES, STANDARDIZED SERVICES, IMPLEMENTATIONS AND ASSOCIATED TEST SUITES INCLUDING BUT NOT LIMITED TO ALL IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OF REASONABLE CARE OR WORKMANLIKE EFFORT, OR RESULTS OR OF LACK OF NEGLIGENCE.

Copyright UPnP Forum © 2011. All rights reserved.

Authors	Company
Bich Nguyen (Co-chair)	Cisco
Sridhar Ramaswamy	Cisco
Ayodele Damola	Ericsson
Bryan Roe	Intel
Vlad Stirbu	Nokia
Cathy Chan	Nokia
Jan Brands	NXP
Jeffrey Kang	Philips
Naveen Rudrappa	Philips
Wouter van der Beek	Philips
Shrikant Kanaparti	Samsung
Se-Hee Han	Samsung
Mahfuzur Rahman (Co-chair)	Samsung

Authors	Company
Sanjeev Verma	Samsung

UPnP Forum in no way guarantees the accuracy or completeness of this author list and in no way implies any rights for or support from those members listed. This list is not the specifications' contributor list that is kept on the UPnP Forum's website.

Contents

Contents.....	3
List of Tables.....	5
List of Figures	6
1 Overview and Scope.....	7
1.1 Introduction.....	7
1.2 Vendor-defined Extensions.....	8
1.3 References.....	8
1.3.1 Normative References.....	8
1.3.2 Informative References.....	8
2 Service Modeling Definitions.....	9
2.1 Service Type	9
2.2 Terms and Abbreviations	9
2.2.1 Abbreviations.....	9
2.2.2 Terms	9
2.3 <i>RADAConfig</i> Service Architecture.....	10
2.4 State Variables	10
2.4.1 State Variable Overview	10
2.4.2 <i>SystemInfo</i>	10
2.4.3 <i>SystemInfoUpdateID</i>	11
2.4.4 <i>A ARG TYPE FilterList</i>	11
2.4.5 <i>A ARG TYPE UUID</i>	11
2.4.6 <i>A ARG TYPE VirtualDeviceDescr</i>	11
2.4.7 <i>A ARG TYPE VirtualDeviceID</i>	11
2.4.8 <i>A ARG TYPE VirtualDeviceList</i>	11
2.5 Eventing and Moderation.....	12
2.5.1 Relationships Between State Variables.....	12
2.6 Actions	12
2.6.1 <i>GetSystemInfo()</i>	12
2.6.2 <i>EditFilter()</i>	13
2.6.3 <i>CreateVirtualDevice()</i>	15
2.6.4 <i>GetVirtualDevices()</i>	16
2.6.5 <i>DestroyVirtualDevice()</i>	16
2.6.6 Relationships Between Actions	17
2.6.7 Error Code Summary	17
2.7 Theory of Operation.....	18
2.7.1 Getting Partial SystemInfo.....	18
2.7.2 Filter Editing	18
2.7.3 Effects of <i>EditFilter()</i> on <i>SystemInfo</i>	19
2.7.4 Virtual Device.....	27
3 XML Service Description	29

4	Test	32
Appendix A.	RemoteAccessDiscoveryAgent Structures (Normative)	33
A.1	FilterList Template	33
A.2	Virtual Device Description Template	34
A.3	Virtual Device List Template	35
Appendix B.	Enhanced Filtering (Informative)	38

List of Tables

Table 2-1:	Abbreviations.....	9
Table 2-2:	State Variables	10
Table 2-3:	Eventing and Moderation.....	12
Table 2-4:	Actions	12
Table 2-5:	Arguments for <u>GetSystemInfo()</u>	13
Table 2-6:	Error Codes for <u>GetSystemInfo()</u>	13
Table 2-7:	Arguments for <u>EditFilter()</u>	14
Table 2-8:	Error Codes for <u>EditFilter()</u>	14
Table 2-9:	Arguments for <u>CreateVirtualDevice()</u>	15
Table 2-10:	Error Codes for <u>CreateVirtualDevice()</u>	15
Table 2-11:	Arguments for <u>GetVirtualDevices()</u>	16
Table 2-12:	Error Codes for <u>GetVirtualDevices()</u>	16
Table 2-13:	Arguments for <u>DestroyVirtualDevice()</u>	17
Table 2-14:	Error Codes for <u>DestroyVirtualDevice ()</u>	17
Table 2-15:	Error Code Summary	17

List of Figures

Figure 1-1:	SSDP Aggregation.....	7
Figure 2-1:	RADA Synchronization tree in each home network on startup of RAS.....	19
Figure 2-2:	RADA Synchronisation tree in each home after “edit”	21
Figure 2-3	Synchronisation trees in each network after adding a filter on device1	23
Figure 2-4	Remote synchronization trees in each network after removing a filter.....	25
Figure 2-5:	Virtual Device.....	27
Figure 4-1.	An example of the implementation of the enhanced filtering.....	38

1 Overview and Scope

This service definition is compliant with the UPnP Device Architecture version 1.0. It defines a service type referred to herein as *RADAConfig* service. This service type enables configuration of the in-band synchronization mechanism between Remote Access Clients and Remote Access Server.

1.1 Introduction

A Remote Access Discovery Agent aggregates information about UPnP devices and services from two primary sources, depending if the devices are located in the local network or they are located in a remote device. For aggregating the devices and services available in the local network, the Remote Access Discovery Agent is constantly monitoring the SSDP traffic, which enables the RADA to have an up-to-date image of the UPnP network. The RADA finds information about remote UPnP devices and services by synchronizing with remote RADAs.

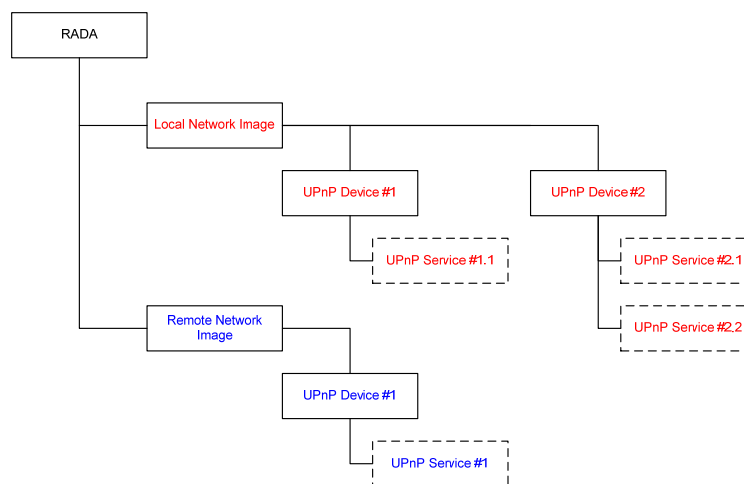


Figure 1-1: SSDP Aggregation.

The main purpose in maintaining this aggregate view of available devices, is to alleviate the timing dependencies inherent in the UPnP Discovery mechanism. When a UPnP Control Point issues a search request, the request packet contains a parameter that specifies the maximum number of seconds a device can wait before sending the response. In remote scenarios, it is possible that this value will be exceeded with normal network traffic delay.

This aggregate view also serves to minimize the amount of SSDP traffic that needs to flow across the remote transport, as some remote scenarios may be cost-sensitive in regards to the amount of data that is transferred, since SSDP is often described as a “chatty” protocol.

The aggregate view could be used to restrict the visibility of local UPnP devices and services from remote devices and of UPnP devices and services hosted by remote devices to your local network.

This service does not address the actual transport protocol used to facilitate Remote Access.

The *RADAConfig* service is a UPnP service that provides control points with the following functionality:

- Customize which local devices and services are visible in remote networks
- Customize which remote devices and services are visible in the local network

This service does not address:

- Aggregation of the the local view of the UPnP network.
- Relaying discovery messages in the local network on behalf of remote devices

- Transport protocol used to facilitate Remote Access or its configuration.

1.2 Vendor-defined Extensions

Whenever vendors create additional vendor-defined state variables, actions or properties, their assigned names and XML representation MUST follow the naming conventions and XML rules as specified in [DEVICE], Section 2.5, “Description: Non-standard vendor extensions”.

1.3 References

1.3.1 Normative References

This section lists the normative references used in this specification and includes the tag inside square brackets that is used for each such reference:

[DEVICE] – UPnP Device Architecture, version 1.0.

Available at: <http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0-20080424.pdf>.

Latest version available at: <http://www.upnp.org/specs/arch/UPnP-DeviceArchitecture-v1.0.pdf>.

[DADS-XSD] – XML Schema for UPnP RA Discovery Agent XML Data Structures

Available at: <http://www.upnp.org/schemas/ra/dads-v1-20090930.xsd>.

Latest version available at: <http://www.upnp.org/schemas/ra/dads-v1.xsd>.

[RAServer] – RAServer:1, UPnP Forum,

Available at: <http://www.upnp.org/specs/ra/UPnP-ra-RAServer-v1-Device-20090930.pdf>.

Latest version available at: <http://www.upnp.org/specs/ra/UPnP-ra-RAServer-v1-Device.pdf>.

[RADASync] – RADASync:1, UPnP Forum,

Available at: <http://www.upnp.org/specs/ra/UPnP-ra-RADASync-v1-Service-20090930.pdf>.

Latest version available at: <http://www.upnp.org/specs/ra/UPnP-ra-RADASync-v1-Service.pdf>.

[RFC 2119] – IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, March 1997.

Available at: <http://www.ietf.org/rfc/rfc2119.txt>.

[XML] – “Extensible Markup Language (XML) 1.0 (Third Edition)”, François Yergeau, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, eds., W3C Recommendation, February 4, 2004.

Available at: <http://www.w3.org/TR/2004/REC-xml-20040204/>.

1.3.2 Informative References

This section lists the informative references that are provided as information in helping understand this specification:

[RAARCH] – RAArchitecture:1, UPnP Forum,

Available at: <http://www.upnp.org/specs/ra/UPnP-ra-RAArchitecture-v1-20090930.pdf>.

Latest version available at: <http://www.upnp.org/specs/ra/UPnP-ra-RAArchitecture-v1.pdf>.

2 Service Modeling Definitions

2.1 Service Type

The following service type identifies a service that is compliant with this specification:

urn:schemas-upnp-org:service:RADAConfig:2

RADAConfig service is used herein to refer to this service type.

2.2 Terms and Abbreviations

2.2.1 Abbreviations

Table 2-1: Abbreviations

Definition	Description
RAC	Remote Access Client
RADA	Remote Access Discovery Agent
RAS	Remote Access Server

2.2.2 Terms

2.2.2.1 Local device

A local device is a UPnP device that is attached to the physical network where the RADA is located.

2.2.2.2 Remote Access Client

The Remote Access Client (RAC) is the peer physical device that is not part of the physical home network. The RAC is exposing only the UPnP devices and services that are embedded in the physical device.

2.2.2.3 Remote Access Server

The Remote Access Server (RAS) is the peer physical device located in the home network. RAS is exposing to the RAC the UPnP devices and services available in the physical home network as well as any embedded in the physical RAS device.

2.2.2.4 Remote device

A remote device is a UPnP device that is not attached to the physical network where the RADA is located.

2.2.2.5 SystemInfo Change

The SystemInfo change happens when one of the following occurs:

- A Remote Access connection is established
- A Remote Access connection is ended
- A Remote Access Transport profile is added/modified/deleted
- A device joins a remote network

- A device leaves a remote network
- A filter is added/modified/deleted

2.2.2.6 UpdateID

An unsigned integer associated with *SystemInfo*. The value is incremented each time the *SystemInfo* is modified. Upon reaching the value $2^{32}-1$, the next update rolls the value back to zero.

2.3 **RADAConfig** Service Architecture

The Remote Access Architecture defines management interfaces that allow the configuration of the ACLs that will restrict the visibility of the home devices from the remote control points and of remote devices from local control points.

2.4 State Variables

Reader Note: For a first-time reader, it may be more helpful to read the action definitions before reading the state variable definitions.

2.4.1 State Variable Overview

Table 2-2: State Variables

Variable Name	R/O ¹	Data Type	Allowed Values	Eng. Units
<i>SystemInfo</i>	<u>R</u>	string	See Section 2.4.2	
<i>SystemInfoUpdateID</i>	<u>R</u>	ui4	See Section 2.4.3	
<i>A_ARG_TYPE_FilterList</i>	<u>R</u>	string	See Section 2.4.4	
<i>A_ARG_TYPE_UUID</i>	<u>R</u>	string	See Section 2.4.5	
<i>A_ARG_TYPE_VirtualDeviceDescr</i>	<u>O</u>	string	See Section 2.4.6	
<i>A_ARG_TYPE_VirtualDeviceID</i>	<u>O</u>	string	See Section 2.4.7	
<i>A_ARG_TYPE_VirtualDeviceList</i>	<u>O</u>	string	See Section 2.4.8	

¹ R = Required, O = Optional, X = Non-standard.

2.4.2 **SystemInfo**

This state variable contains the snapshot of all devices that are reachable/visible from the RADA, which are grouped according to the network in which they are present. The state variable contains also the identity of the remote networks, and filters that determine how the synchronization process will be performed.

The structure of the *SystemInfo* state variable is a DADS XML Document:

- <systemInfo> is the root element.
- See the DADS schema [DADS-XSD] for more details on the structure. The available properties and their names are described in Appendix A.1 of [RADASync].

Note that since the value of *SystemInfo* is XML, it needs to be escaped (using the normal XML rules: [XML] Section 2.4 Character Data and Markup) before embedding in a SOAP response message.

Note: implementers must be aware that this state variable is shared with the RADASync service, which updates the information about remote devices, and with the RATAConfig service, which maintains information about Remote Access Transport profiles. This state variable **MUST** be updated by the device and propagated internally to those other services (if present on the same device). Each modification in [SystemInfo](#) **MUST** be signalled by the device through the [SystemInfoUpdateID](#) evented state variable (see Section 2.4.3).

2.4.3 SystemInfoUpdateID

This state variable contains the value of the [updateID](#) attribute of the <systemInfo> element in the [SystemInfo](#) state variable XML Document. Its value is updated each time a SystemInfo Change occurs (see Section 2.2.2.5). Note that the [SystemInfo](#) state variable may also be changed by the RADASync and RATAConfig services. Therefore, the RADAConfig service **MUST** event the [SystemInfoUpdateID](#) state variable, even if one of the other services (on the same device) is responsible for changing the [SystemInfo](#).

2.4.4 A_ARG_TYPE FilterList

The structure of the [A_ARG_TYPE FilterList](#) state variable is a DADS XML Document:

- <filterList> is the root element.
- See the DADS schema for more details on the structure. The available properties and their names are described in Appendix A.1.

Note that since the value of [A_ARG_TYPE FilterList](#) is XML, it needs to be escaped (using the normal XML rules: [XML] Section 2.4 Character Data and Markup) before embedding in a SOAP response message.

2.4.5 A_ARG_TYPE UUID

This state variable is introduced to provide type information for action arguments used for identifying a particular local or remote network.

2.4.6 A_ARG_TYPE VirtualDeviceDescr

This state variable describes the devices and services included in a virtual device. The device type for the virtual device can be any existing valid UPnP device types as specified by various UPnP DCPs. Please see Appendix A.2 for the template of the virtual device description.

2.4.7 A_ARG_TYPE VirtualDeviceID

This state variable is introduced to provide type information for action argument used for identifying a virtual device. The identification of a virtual device is unique with this service.

2.4.8 A_ARG_TYPE VirtualDeviceList

The structure of the [A_ARG_TYPE VirtualDeviceList](#) state variable is a DADS XML Document:

- <virtualDeviceList> is the root element.
- See the DADS schema for more details on the structure. The available properties and their names are described in Appendix A.3.

Note that since the value of [A_ARG_TYPE VirtualDeviceList](#) is XML, it needs to be escaped (using the normal XML rules: [XML] Section 2.4 Character Data and Markup) before embedding in a SOAP response message.

2.5 Eventing and Moderation

Table 2-3: Eventing and Moderation

Variable Name	Evented	Moderated Event	Max Event Rate ¹	Logical Combination	Min Delta per Event ²
<u>SystemInfo</u>	<u>NO</u>	<u>NO</u>			
<u>SystemInfoUpdateID</u>	<u>YES</u>	<u>NO</u>			
<u>A_ARG_TYPE_FilterList</u>	<u>NO</u>	<u>NO</u>			
<u>A_ARG_TYPE_UUID</u>	<u>NO</u>	<u>NO</u>			
<u>A_ARG_TYPE_VirtualDeviceDescr</u>	<u>NO</u>	<u>NO</u>			
<u>A_ARG_TYPE_VirtualDeviceID</u>	<u>NO</u>	<u>NO</u>			
<u>A_ARG_TYPE_VirtualDeviceList</u>	<u>NO</u>	<u>NO</u>			

¹ Determined by N, where Rate = (Event)/(N secs).

² (N) * (allowedValueRange Step).

2.5.1 Relationships Between State Variables

The SystemInfoUpdateID state variable is updated when the SystemInfo state variable is changed. Note that the SystemInfo state variable may also be changed by the RADASync and RATAConfig services.

2.6 Actions

Table 2-4: Actions

Name	R/O ¹
<u>GetSystemInfo()</u>	<u>R</u>
<u>EditFilter()</u>	<u>R</u>
<u>CreateVirtualDevice()</u>	<u>O</u>
<u>GetVirtualDevices()</u>	<u>O</u>
<u>DestroyVirtualDevice()</u>	<u>O</u>

¹ R = REQUIRED, O = OPTIONAL, X = Non-standard.

2.6.1 GetSystemInfo()

This action exposes information about all device(s) in the local network and devices exposed by remote networks. A Control Point may invoke this action to determine the list of devices visible for a credential, or to view the list of credentials that have access to a particular device.

2.6.1.1 Arguments

Table 2-5: Arguments for **GetSystemInfo()**

Argument	Direction	relatedStateVariable
<u>ID</u>	<u>IN</u>	<u>A_ARG_TYPE_UUID</u>
<u>SystemInfo</u>	<u>OUT</u>	<u>SystemInfo</u>

2.6.1.1.1 ID

The UUID of the RADA of a network or the UUID of a device for which the SystemInfo is requested. If the value is “*” then the full SystemInfo is returned.

2.6.1.1.2 SystemInfo

The entire or a subset of the SystemInfo tree specific to the RADA indicated by the UUID.

2.6.1.2 Dependency on State

None.

2.6.1.3 Effect on State

None.

2.6.1.4 Control Point Requirements

None.

2.6.1.5 Errors

Table 2-6: Error Codes for **GetSystemInfo()**

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
701	ID Not Found	The UUID corresponding to the ID specified does not match any devices found on the local or remote network.

2.6.2 EditFilter()

This action allows a Control Point to modify the accessibility settings for the specified device(s) on the local network. Only those devices which meet the criteria specified by the Filters in the system will be exposed to the Remote Access Client Device (Export) or the Home Network (Import).

2.6.2.1 Arguments

Table 2-7: Arguments for EditFilter()

Argument	Direction	relatedStateVariable
<u>Filter</u>	<u>IN</u>	<u>A_ARG_TYPE_FilterList</u>

2.6.2.1.1 Filter

The Filter settings for one or more networks, to insert into, remove from, or edit inside the filter table. The type of operation (add, delete, edit) is indicated by the updateType attribute in the filterList XML document.

2.6.2.2 Dependency on State

If the updateType is “add” or “delete” then if the controlMode of the specified filter argument is different from the exportControlMode or the importControlMode of the corresponding RADA listed in the SystemInfo state variable and there is at least one filter defined for that RADA in the SystemInfo state variable, then the action MUST fail with an error code 702 (Invalid Filter).

2.6.2.3 Effect on State

The Filter is used to modify the appropriate accessControl fields on the SystemInfo state variable. The modification of the SystemInfo state variable results in an update of the SystemInfoUpdateID evented state variable.

Note: Since the SystemInfo state variable is shared with the RATAConfig and RADASync services, the implementation MUST propagate the modification of its value internally to those services, if present on the same device.

2.6.2.4 Control Point Requirements

Prior to modifying the filters using the EditFilter() action, a Control Point SHOULD examine the currently set filters. This can be done by invoking the GetSystemInfo() action. A Control Point SHOULD use the SystemInfoUpdateID evented state variable to determine whether the SystemInfo state variable has changed since the previous time it invoked the GetSystemInfo() action.

To modify the exportControlMode or the importControlMode of a RADA, a Control Point SHOULD first invoke the EditFilter() action to delete the corresponding filter and then invoke the EditFilter() action again to add the desired filter.

2.6.2.5 Errors

Table 2-8: Error Codes for EditFilter()

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
702	Invalid Filter	The provided Filter is not valid.

2.6.3 CreateVirtualDevice()

This action enables a control point to create a virtual device. A virtual device represents a unified view of services and devices to be exposed by the RADAConfig to the remote network under a single device description. To the remote network the virtual device is represented as a regular device with its own device description, however, in reality the device description may include a composition of services and devices from multiple devices into the local network that are transparent to the remote network. When this action succeeds the device dynamically creates a device description for the remote network based on the information provided by the input argument. The device creates a URL for the device description. When the virtual device is exposed to the remote network, the URL can be used to retrieve the virtual device description.

Table 2-9: Arguments for CreateVirtualDevice()

Argument	Direction	relatedStateVariable
<u>VirtualDeviceDescr</u>	<u>IN</u>	<u>A_ARG_TYPE_VirtualDeviceDescr</u>
<u>VirtualDeviceID</u>	<u>OUT</u>	<u>A_ARG_TYPE_VirtualDeviceID</u>

2.6.3.1.1 VirtualDeviceDescr

The input argument VirtualDeviceDescr contains the information related to the virtual device which including the type of the virtual device, the devices and services that are part of the virtual device etc.

2.6.3.1.2 VirtualDeviceID

The output argument VirtualDeviceID contains the unique identification of the virtual device created by this action.

2.6.3.2 Dependency on State

None.

2.6.3.3 Effect on State

None.

2.6.3.4 Control Point Requirements

None.

2.6.3.5 Errors

Table 2-10: Error Codes for CreateVirtualDevice()

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
703	Virtual Device Type not supported	The device does not support the requested virtual device type.
704	Invalid UDN	One of the physical device's UDN invalid

ErrorCode	errorDescription	Description
705	Unable to process device description	The device is unable to create the virtual device.

2.6.4 GetVirtualDevices()

This action enables a control point to retrieve the list of all currently configured virtual devices.. If this action succeeds then the action returns a list of virtual devices including the virtual device id, and information about all the devices that are part of the virtual device. Arguments

Table 2-11: Arguments for GetVirtualDevices()

Argument	Direction	relatedStateVariable
<u>VirtualDeviceList</u>	<u>OUT</u>	<u>A_ARG_TYPE_VirtualDeviceList</u>

2.6.4.1.1 VirtualDeviceList

This output argument includes a list of virtual devices that are currently configured in the remote access server. The list includes the identification of each virtual device and the information about each device that are part of the virtual device.

2.6.4.2 Dependency on State

None.

2.6.4.3 Effect on State

None.

2.6.4.4 Control Point Requirements

None.

2.6.4.5 Errors

Table 2-12: Error Codes for GetVirtualDevices()

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
706	Virtual Device Not Found	The virtual device does not exist.

2.6.5 DestroyVirtualDevice()

This action enables a control point to delete a previously configured virtual device, which includes the device description of the virtual device. The input argument VirtualDeviceID identifies the virtual device to be removed. If the virtual device does not exist when this action is invoked then the action returns an error.

2.6.5.1 Arguments

Table 2-13: Arguments for *DestroyVirtualDevice()*

Argument	Direction	relatedStateVariable
<i>VirtualDeviceID</i>	<i>IN</i>	<i>A_ARG_TYPE_VirtualDeviceID</i>

2.6.5.1.1 *VirtualDeviceID*

This input argument includes the identification of the virtual device that is to be removed.

2.6.5.2 Dependency on State

None.

2.6.5.3 Effect on State

None.

2.6.5.4 Control Point Requirements

None.

2.6.5.5 Errors

Table 2-14: Error Codes for *DestroyVirtualDevice ()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
706	Virtual Device Not Found	The virtual device does not exist.

2.6.6 Relationships Between Actions

The *GetSystemInfo* action is used to fetch filter information that can be updated using the *EditFilter* action. Complete filter tables can be swapped into and out of the service.

2.6.7 Error Code Summary

The following table lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error should be returned.

Table 2-15: Error Code Summary

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
700		Reserved for future extensions.

ErrorCode	errorDescription	Description
701	ID Not Found	The UUID corresponding to the ID specified does not match any devices found on the local or remote network.
702	Invalid Filter	The provided Filter is not valid.

Note: 800-899 Error Codes are not permitted for standard actions. See UPnP Device Architecture section on Control for more details.

2.7 Theory of Operation

2.7.1 Getting Partial SystemInfo

A control point can request the full [SystemInfo](#) structure by including the "*" as the value of the [ID](#) argument of the [GetSystemInfo\(\)](#) action.

A control point may request a subset of the SystemInfo tree by including specific UUID as the value of the ID argument of the [GetSystemInfo\(\)](#) action. For example if the control point indicates the UUID of a remote RADA then the returned value will be the list of the devices that are visible from the remote network the RADA is attached to. For getting information about a particular device, the control point needs to indicate the UUID of the device itself as the value of the [ID](#) argument of the [GetSystemInfo\(\)](#) action. The return value will include the remote network that exposes the device but only the particular device will be listed as available from that network.

2.7.2 Filter Editing

The RADAConfig service's primary role is to configure the filters that are used by the RADASync service, during its synchronization process between Remote Access Servers and Remote Access Clients.

There are two types of filters that the RADAConfig service defines: allowList and denyList filters. When allowlist filtering is specified, without any filters defined by the RADAConfig service, the RADASync would not expose local devices when synchronizing with a remote network, respectively it would not allow discovery of remote devices from a remote network on the local network. If denyList filtering is specified, without any filters defined by the RADAConfig service, the RADASync would expose all the local devices when synchronizing with a remote network, and it would allow the discovery of remote devices from that remote network on the local network.

The default control mode for the export filter is "allowList" and for the import filter is "denyList". A particular implementation may change this behavior.

When a Remote Access Client established a secure tunnel to the Remote Access Server, the RADASync service on the RAS establishes a synchronization session with the RADASync on the RAC. The RADASync service refers to the filter policy defined by the RADAConfig service to determine how devices should be added to the local/remote branches of a synchronization tree.

If one would like to access some of the local devices remotely, all one would need to do is to invoke the [EditFilter\(\)](#) action. The simplest type of filter, is simply a pass-thru filter, where the specified filter allows all devices to be exposed outside the home. Such a filter would look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">
  <filterList>
```

```

<filter
  target="31638B2A-AC78-4766-82D5-D53C6480638A"
  controlMode="denyList"
  updateType="edit">
</filter>
</filterList>
</dads>

```

The filter specified above, would allow all the devices on the local network to be discoverable on the remote network, regardless of which transport credential was used to establish the secure tunnel.

A more typical filter would be one that more tightly scopes the devices and credentials that are discoverable on the remote network. Such a filter may look like the following:

```

<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">
  <filterList>
    <filter
      target="31638B2A-AC78-4766-82D5-D53C6480638A"
      controlMode="allowList"
      updateType="add">
      <deviceInfo
        uuid="12338B2A-AC78-4766-82D5-D53C64806ABC"
        <accessControl>
          <access credentialID="David MP3 Player"></access>
        </accessControl>
      </deviceInfo>
    </filter>
  </filterList>
</dads>

```

The filter specified above, only allows the Remote Access Client that uses the credential identified by “David MP3 Player” to discover the media server with the specified UUID.

2.7.3 Effects of EditFilter() on SystemInfo

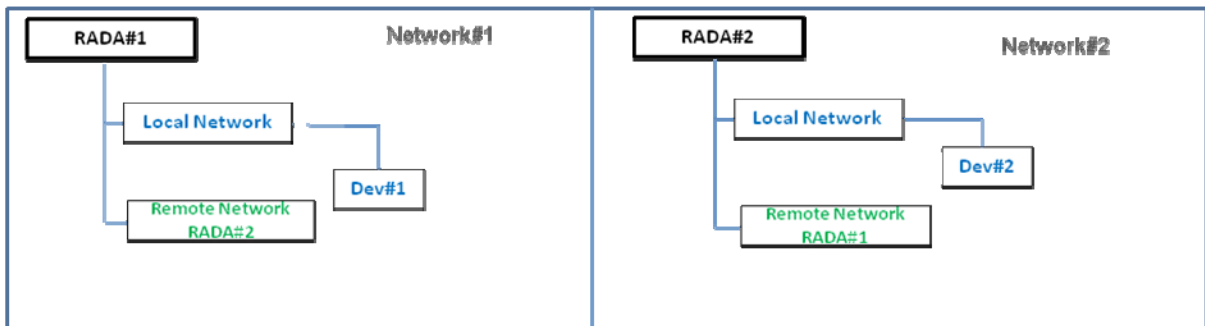


Figure 2-1: RADA Synchronization tree in each home network on startup of RAS

The above figure shows two home networks and synchronization tree (visibility) of devices in each network (Network#1 and Network#2). Both networks have their export control mode set to “allowList” and import control mode set to “denyList”. [GetSystemInfo\(\)](#) response of RADA#1 is as below.

GetSystemInfo response of RADA#1

```
<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">
  <systemInfo
    updateID="1">
    <localNetwork>
      <rada
        uuid="RADA_NETWORK1"
        exportControlMode="allowList">
        <deviceInfo
          uuid="DEVICE1_NETWORK1_ABCDEF"
          cache-control="1200"
          descriptionDocument="server.xml"
          server="My media server">
        </deviceInfo>
      </rada>
    </localNetwork>
    <remoteNetwork>
      <rata
        credentialID="network#2_CredentialID_12345"
        connected="true"></rata>
      <rada
        uuid="RADA_NETWORK2"
        dddLocation="http://10.8.0.2:1900"
        importControlMode="denyList"
        heartbeat="">
      </rada>
    </remoteNetwork>
  </systemInfo>
</dads>
```

GetSystemInfo() response of RADA#2

```
<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">
  <systemInfo
    updateID="1">
    <localNetwork>
      <rada
        uuid="RADA_NETWORK2 "
        exportControlMode="allowList">
        <deviceInfo
          uuid="DEVICE2_NETWORK2_ABCDEF"
          cache-control="1000"
          descriptionDocument="render.xml"
          server="My media renderer">
        </deviceInfo>
      </rada>
    </localNetwork>
    <remoteNetwork>
```

```

<rata
  credentialID="Network#1_CredentialID_12345"
  connected="true"></rata>
<rada
  uuid="RADA_NETWORK_1_ABCDEF"
  dddLocation="http://10.8.0.1:1900"
  importControlMode="denyList"
  heartbeat="">
</rada>
</remoteNetwork>
</systemInfo>
</dads>

```

Since exportControlMode of both RADA is set to “allowList” and there are no filters active, Device#1 is not propagated to RADA#2 and Device#3 is not propagated to RADA#1.

EditFilter() operation on RADA#1

```

<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">
  <filterList>
    <filter
      target="RADA_NETWORK1"
      controlMode="denyList"
      updateType="edit">
    </filter>
  </filterList>
</dads>

```

The below diagram shows the devices visible in each network after EditFilter() action. “updateType” specified is “edit” which means all the filters defined will be deleted and the controlMode of RADA is set to “denyList”. Since there are no filters defined, there will be no filter deletion and exportControlMode of RADA#1 is directly set to “denyList”. RADA#1 will add the devices of its local network to RADA#2 using AddRemoteDevices().

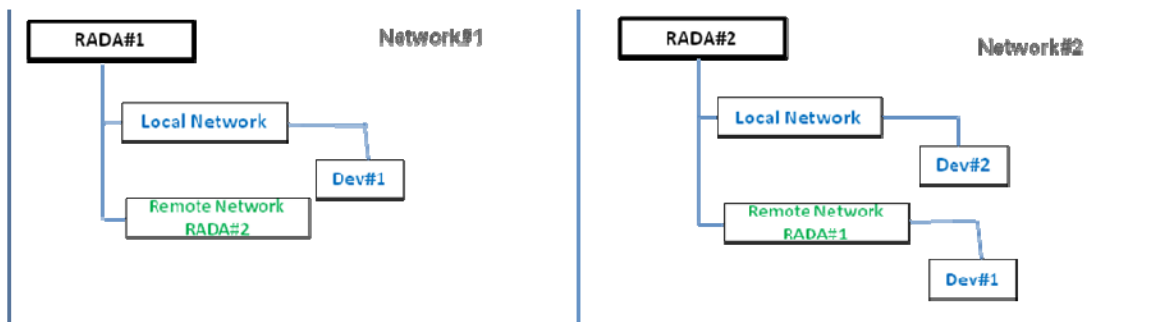


Figure 2-2: RADA Synchronisation tree in each home after “edit”

```

<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">

```

```

<systemInfo
  updateID="2">
  <localNetwork>
    <rada
      uuid="RADA_NETWORK1 "
      exportControlMode="denyList">
      <deviceInfo
        uuid="DEVICE1_NETWORK1_ABCDEF"
        cache-control="1200"
        descriptionDocument="server.xml"
        server="My media server">
      </deviceInfo>
    </rada>
  </localNetwork>
  <remoteNetwork>
    <rata
      credentialID="network#2_CredentialID_12345"
      connected="true"></rata>
    <rada
      uuid="RADA_NETWORK_2_ABCDEF"
      dddLocation="http://10.8.0.1:1900"
      importControlMode="denyList"
      heartbeat="">
    </rada>
  </remoteNetwork>
</systemInfo>
</dads>

```

***GetSystemInfo()* response of RADA#2**

```

<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">
  <systemInfo
    updateID="2">
    <localNetwork>
      <rada
        uuid="RADA_NETWORK2_ABCDEF"
        exportControlMode="allowList">
        <deviceInfo
          uuid="DEVICE2_NETWORK2_ABCDEF"
          cache-control="1000"
          descriptionDocument="render.xml"
          server="My media renderer">
        </deviceInfo>
      </rada>
    </localNetwork>
    <remoteNetwork>
      <rata
        credentialID="Network#1_CredentialID_12345"
        connected="true"></rata>
      <rada
        uuid="RADA_NETWORK_1_ABCDEF"
        dddLocation="http://10.8.0.1:1900"
        importControlMode="denyList"
        heartbeat="1200">
        <deviceInfo
          uuid="RADA_NETWORK1_ABCDEF"
          cache-control="1200"

```

```

        descriptionDocument="server.xml"
        server="My media server">
    </deviceInfo>
</rada>
</remoteNetwork>
</systemInfo>
</dads>

```

As RADA#2 is aware of Device#1, it is reflected in the [GetSystemInfo\(\)](#) response as shown above.

Adding a filter on RADA#1

Adding a filter for Device#1 in RADA#1 such that the device is not seen on Network#2 will look like as below.

```

<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">
  <filterList>
    <filter
      target="RADA_NETWORK1_ABCDEF"
      controlMode="denyList"
      updateType="add">
      <deviceInfo
        uuid="DEVICE1_NETWORK1_ABCDEF" >
        <accessControl>
          <access
            credentialID="network#2_CredentialID_12345"></access>
          </accessControl>
        </deviceInfo>
      </filter>
    </filterList>
  </dads>

```

As the exportControlMode for RADA#1 is denyList, adding a filter for a device would lead to removing of Device#1 from Network#2. The snapshot of the devices visible in each network after the filter operation will look like as below.

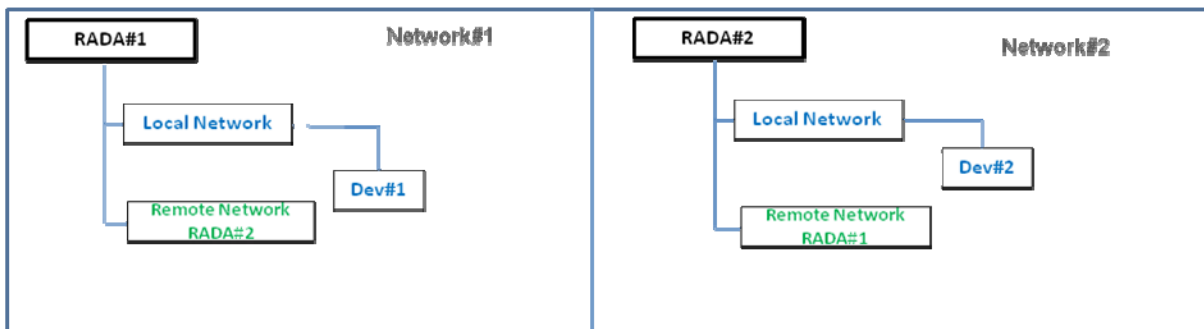


Figure 2-3 Synchronisation trees in each network after adding a filter on device1

[GetSystemInfo\(\)](#) response of RADA#1

```

<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
http://www.upnp.org/schemas/ra/dads-v1.xsd">
<systemInfo
  updateID="3">
  <localNetwork>
    <rada
      uuid="RADA_NETWORK1 "
      exportControlMode="denyList">
      <deviceInfo
        uuid="DEVICE1_NETWORK1_ABCDEF"
        cache-control="1200"
        descriptionDocument="server.xml"
        server="My media server">
      </deviceInfo>
      <accessControl>
        <access
credentialID="Network#1_CredentialID_12345"></access>
        </accessControl>
      </rada>
    </localNetwork>
    <remoteNetwork>
      <rata
        credentialID="network#2_CredentialID_12345"
        connected="true"></rata>
      <rada
        uuid="RADA_NETWORK_2_ABCDEF"
        dddLocation="http://10.8.0.1:1900"
        importControlMode="denyList"
        heartbeat="">
      </rada>
    </remoteNetwork>
  </systemInfo>
</dads>

```

GetSystemInfo response of RADA#2

```

<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
http://www.upnp.org/schemas/ra/dads-v1.xsd">
<systemInfo
  updateID="3">
  <localNetwork>
    <rada
      uuid="RADA_NETWORK2_ABCDEF"
      exportControlMode="allowList">
      <deviceInfo
        uuid="DEVICE2_NETWORK2_ABCDEF"
        cache-control="1000"
        descriptionDocument="render.xml"
        server="My media renderer">
      </deviceInfo>
    </rada>
  </localNetwork>
  <remoteNetwork>
    <rata
      credentialID="Network#1_CredentialID_12345"
      connected="true"></rata>
    <rada

```



```

        uuid="RADA_NETWORK_1_ABCDEF"
        dddLocation="http://10.8.0.1:1900"
        importControlMode="denyList"
        heartbeat="" >
    </rada>
</remoteNetwork>
</systemInfo>
</dads>

```

Removing a filter in RADA#1

Removing the filter added to Device#1 in RADA#1 such that the device will be seen on Network#2 will look like as below.

```

<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">
  <filterList>
    <filter
      target="RADA_NETWORK1_ABCDEF"
      controlMode="denyList"
      updateType="delete">
      <deviceInfo
        uuid="DEVICE1_NETWORK1_ABCDEF" >
        <accessControl>
          <access
            credentialID="network#2_CredentialID_12345"></access>
          </accessControl>
        </deviceInfo>
      </filter>
    </filterList>
  </dads>

```

As the exportControlMode for RADA#1 is “denyList”, removing a filter for device would lead to adding of Device#1 to Network#2. The snapshot of the devices visible in each network after the filter operation will look like as below.

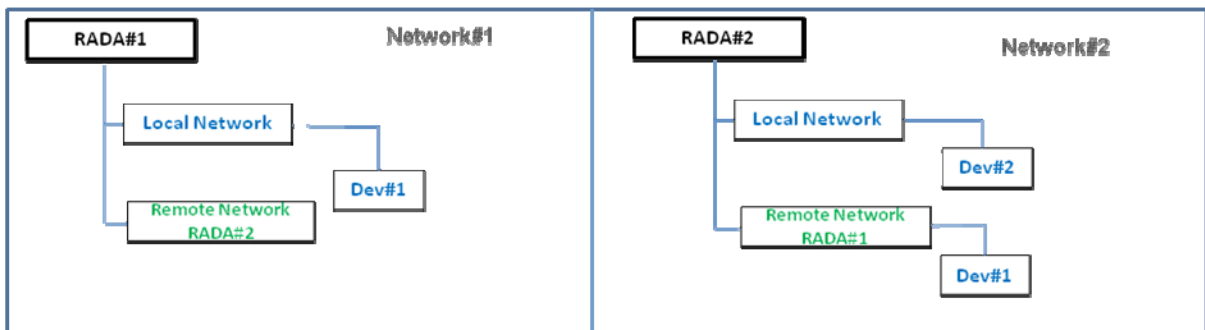


Figure 2-4 Remote synchronization trees in each network after removing a filter

GetSystemInfo() response of RADA#1

```

<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">

```

```

<systemInfo
  updateID="4">
  <localNetwork>
    <rada
      uuid="RADA_NETWORK1"
      exportControlMode="denyList">
      <deviceInfo
        uuid="DEVICE1_NETWORK1_ABCDEF"
        cache-control="1200"
        descriptionDocument="server.xml"
        server="My media server">
      </deviceInfo>
    </rada>
  </localNetwork>
  <remoteNetwork>
    <rata
      credentialID="network#2_CredentialID_12345"
      connected="true"></rata>
    <rada
      uuid="RADA_NETWORK_2_ABCDEF"
      dddLocation="http://10.8.0.1:1900"
      importControlMode="denyList"
      heartbeat="">
    </rada>
  </remoteNetwork>
</systemInfo>
</dads>

```

GetSystemInfo() response of RADA#2

```

<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">
  <systemInfo
    updateID="4">
    <localNetwork>
      <rada
        uuid="RADA_NETWORK2_ABCDEF"
        exportControlMode="allowList">
        <deviceInfo
          uuid="DEVICE2_NETWORK2_ABCDEF"
          cache-control="1000"
          descriptionDocument="render.xml"
          server="My media renderer">
        </deviceInfo>
      </rada>
    </localNetwork>
    <remoteNetwork>
      <rata
        credentialID="Network#1_CredentialID_12345"
        connected="true"></rata>
      <rada
        uuid="RADA_NETWORK_1_ABCDEF"
        dddLocation="http://10.8.0.1:1900"
        importControlMode="denyList"
        heartbeat="1200">
        <deviceInfo
          uuid="RADA_NETWORK1_ABCDEF"
          cache-control="1200"

```

```

descriptionDocument="server.xml"
  server="My media server">
    </deviceInfo>
  </rada>
</remoteNetwork>
</systemInfo>
</dads>

```

2.7.4 Virtual Device

The RADAConfig service allows a control point to configure a virtual device that can be exposed to the remote network. A virtual device represents a unified view of services and devices to be exposed by the RADAConfig to the remote network under a single device description. To the remote network the virtual device is represented as a regular device with its own device description, however, in reality the device description includes a composition of services and devices from multiple devices in the local network that are transparent to the remote network. In contrast to the filtering mechanism which allows only to configure devices or services for each device to be exposed to the remote network with its own device description, the RADAConfig can be used to expose only the virtual device to the remote network without allowing the remote network to comprehend actual devices in the local network. In the virtual device scenario, the remote network only sees a single device (i.e., the virtual device) with some services that the remote device can access.

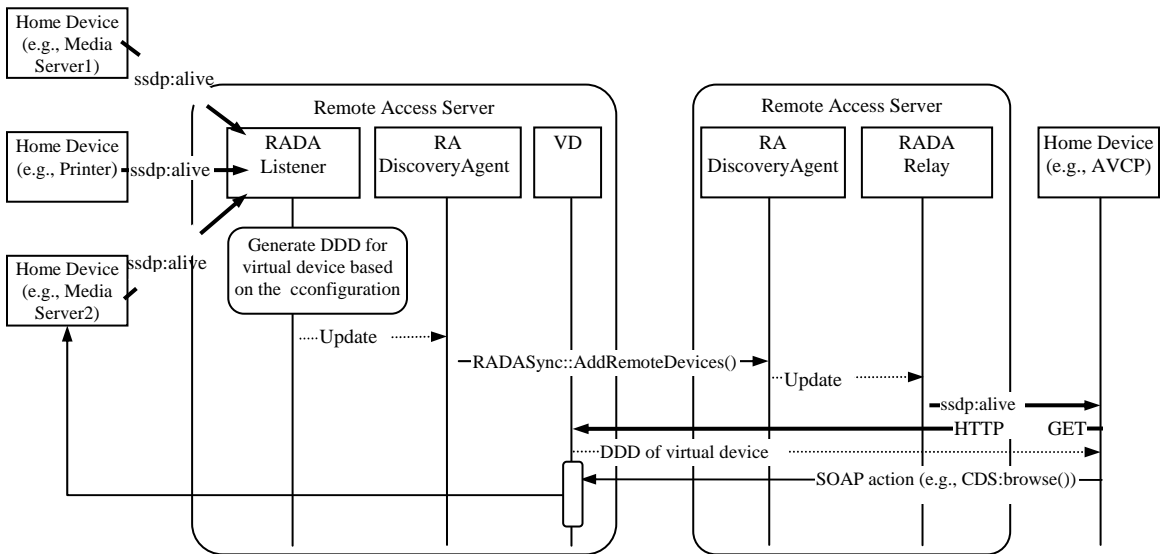


Figure 2-5: Virtual Device.

The virtual device as shown in the figure above consists of services from three different devices in the local network. However, only the virtual device is exposed to the remote network without exposing each individual devices in the local network. The virtual device includes a subset of services to be exposed from each device in the local network. When a remote device access a service described in the virtual device, the request is proxided through the VD (Virtual Device) entity in the RAS to the actual device and the response is also proxided through the VD. The URL to access services defined in the virtual device points to the URL of the RAS and hence all the requests from remote home goes through the RAS. The device type to identify the virtual device can be any valid existing devices types. If a control point would like to configure the virtual device, the control point need to invoke the *CreateVirtualDevice()* action. A control point can also retrieve the current configuration of the virtual device by invoking the *GetVirtualDevices()* action. The filtering and the virtual device features supported by the RADAConfig can co-exist in the RAS.

The virtual device can be used to support finer grained CDS filtering. This would be an implementation specific feature which can configured locally through internal interface.

3 XML Service Description

```

<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">

  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>

  <actionList>

    <action>
      <name>GetSystemInfo</name>
      <argumentList>
        <argument>
          <name>ID</name>
          <direction>in</direction>
          <relatedStateVariable>
            A_ARG_TYPE_UUID
          </relatedStateVariable>
        </argument>
        <argument>
          <name>SystemInfo</name>
          <direction>out</direction>
          <relatedStateVariable>
            SystemInfo
          </relatedStateVariable>
        </argument>
      </argumentList>
    </action>

    <action>
      <name>EditFilter</name>
      <argumentList>
        <argument>
          <name>Filter</name>
          <direction>in</direction>
          <relatedStateVariable>
            A_ARG_TYPE_FilterList
          </relatedStateVariable>
        </argument>
      </argumentList>
    </action>

    <action>
      <name>CreateVirtualDevice</name>
      <argumentList>
        <argument>
          <name>VirtualDeviceDescr</name>
          <direction>in</direction>
          <relatedStateVariable>
            A_ARG_TYPE_VirtualDeviceDescr
          </relatedStateVariable>
        </argument>
      </argumentList>
    </action>
  </actionList>
</scpd>

```

```

        <name>VirtualDeviceID</name>
        <direction>out</direction>
        <relatedStateVariable>
            A_ARG_TYPE_VirtualDeviceID
        </relatedStateVariable>
    </argument>

<action>
    <name>GetVirtualDevices</name>
    <argumentList>
        <argument>
            <name>VirtualDeviceList</name>
            <direction>out</direction>
            <relatedStateVariable>
                A_ARG_TYPE_VirtualDeviceList
            </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>DestroyVirtualDevice</name>
    <argumentList>
        <argument>
            <name>VirtualDeviceID</name>
            <direction>in</direction>
            <relatedStateVariable>
                A_ARG_TYPE_VirtualDeviceID
            </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<!-- Declarations for other actions defined by UPnP vendor
    (if any)go here. -->

</actionList>

<serviceStateTable>

    <stateVariable sendEvents="no">
        <name>SystemInfo</name>
        <dataType>string</dataType>
    </stateVariable>

    <stateVariable sendEvents="yes">
        <name>SystemInfoUpdateID</name>
        <dataType>ui4</dataType>
    </stateVariable>

    <stateVariable sendEvents="no">
        <name>A_ARG_TYPE_FilterList</name>
        <dataType>string</dataType>
    </stateVariable>

    <stateVariable sendEvents="no">

```

```
<name>A_ARG_TYPE_UUID</name>
<dataType>string</dataType>
</stateVariable>

<stateVariable sendEvents="no">
  <name>A_ARG_TYPE_VirtualDeviceDescr</name>
  <dataType>string</dataType>
</stateVariable>

<stateVariable sendEvents="no">
  <name>A_ARG_TYPE_VirtualDeviceID</name>
  <dataType>string</dataType>
</stateVariable>

<stateVariable sendEvents="no">
  <name>A_ARG_TYPE_VirtualDeviceList</name>
  <dataType>string</dataType>
</stateVariable>

<!-- Declarations for other state variables defined by
      UPnP vendor (if any)go here. -->

</serviceStateTable>
</scpd>
```

4 Test

No semantic tests have been specified for this service.

Appendix A. RemoteAccessDiscoveryAgent Structures (Normative)

A.1 FilterList Template

The following shows the generalized layout of a FilterList Template. More elements and/or attributes MAY be added in future versions of FilterList templates.

The *forum* character style is used to indicate names defined by the RAWC. Implementations need to fill out the parts that are printed in *vendor* character style.

```
<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v1.xsd">
  <filterList>
    <filter
      target=""
      controlMode=""
      updateType="">
      <deviceInfo
        uuid="">
        <accessControl>
          <access credentialID=""></access>
          <!-- Any other access (if any) go here.-->
        </accessControl>
      </deviceInfo>
      <!--Any other deviceInfo (if any) go here.-->
    </filter>
    <!-- Any other filter (if any) go here.-->
  </filterList>
</dads>
```

xml

REQUIRED for all XML documents. Case sensitive.

RADA

REQUIRED. Must have "urn:schemas-upnp-org:ra:dads" as the value for the xmlns attribute; this references the UPnP Remote Access Working Committee RADA Datastructure Template Schema. As long as the same xmlns is used, the data structure template MUST be backward compatible, i.e. usable by legacy implementations.

filterList

REQUIRED. Contains the list of filters to be added to the SystemInfo. Contains the following sub elements:

filter

REQUIRED. Contains the filter that will be applied to a single network. Contains the following sub elements:

@target

REQUIRED. xs:string. Contains the UUID of the RADA device on which the filter will be applied. This could be the local RADA or a remote RADA.

@controlMode

REQUIRED. xs:token. Contains the access control mode for advertising devices. Possible values are "allowList" or "denyList".

@updateType

REQUIRED. xs:token. Determines if a filter will be created, edited or deleted. The corresponding values of @updateType are "add", "edit", and "delete".

deviceInfo

OPTIONAL. Contains information about a single device in the network. MUST be omitted when the filter@target represents a remote RADA. When the filter@target represents the local RADA, deviceInfo MUST be present for each device present in the network. Contains the following sub elements:

@uuid

REQUIRED. xs:string. Contains the UUID of the root device.

accessControl

OPTIONAL. Contains the list of remote RADAs that can or cannot see this local device. Contains the following sub elements:

access

OPTIONAL. Describes the remote RADA that can or cannot see the local device. Contains the following attribute:

@credentialID

REQUIRED. xs:string. Contains the identity of the remote RADA that is affected by this entry. If filter@controlMode is "allowList", the remote RADA has visibility rights on this local device. If filter@controlMode is "denyList", the remote RADA does not have visibility rights on this local device.

A.2 Virtual Device Description Template

```
<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
  http://www.upnp.org/schemas/ra/dads-v2.xsd">
  <virtualDevice>
    <virtualDeviceType>
      "MERGE"/"/AGGREGATE"/"/FILTER"
    </virtualDeviceType>
    <rootDeviceType>
      Root device type of the virtual device
    </rootDeviceType>
    <physicalDevice>
      <UDN>
        UDN of the first physical device proxied by this virtual
device
      </UDN>
      <allowedUSN>
        Unique Service Name of embedded device or service to be
exposed to the remote network. ONLY used for Filter type virtual device
      </allowedUSN>
      ...
    </physicalDevice>
    <physicalDevice>
      <UDN>
        UDN of the second physical device proxied by this virtual
device
      </UDN>
    </physicalDevice>
    ...
  </virtualDevice>
```

```
</dads>
```

xml

REQUIRED for all XML documents. Case sensitive.

dads

REQUIRED. Must have "urn:schemas-upnp-org:ra:dads" as the value for the xmlns attribute; this references the UPnP Remote Access Working Committee RADA Datastructure Template Schema. As long as the same xmlns is used, the data structure template MUST be backward compatible, i.e. usable by legacy implementations.

virtualDevice

REQUIRED. Contains a list of devices to be included in the virtual device. Contains the following sub elements:

virtualDeviceType

REQUIRED. Contains the type of the virtual device. There are three types of virtual devices: **MERGE**, **AGGREGATE**, and **FILTER**.

MERGE: indicates that the virtual device contains only devices of the same type (e.g., media server devices) and the remote network only sees a single device description. A **MERGE** virtual device with only one physical device is allowed and can act as a proxy device. For a **MERGE** virtual device, physical device list MAY be empty at the creation of virtual device. The behavior of such a virtual device is that it will merge all the physical devices of the specified type at the creation time or later when those devices join the local network.

AGGREGATE: indicates that the virtual device includes devices of different types and the remote network may see multiple (embedded) devices. An **AGGREGATE** virtual device with only one physical device is allowed and can act as a proxy device. For an **AGGREGATE** virtual device physical device list MAY be empty at the creation of virtual device. The behavior of such a virtual device is that it will forward action requests to the designated physical device.

FILTER: Indicates that the virtual device allows only specific services or embedded devices in a device to be exposed to the remote network. Please see Appendix B (Enhanced Filtering) for details. For a **FILTER** virtual device, physical devices MUST be present in the local network at the time of creation otherwise the virtual device would not know what to filter.

rootDeviceType

REQUIRED. Contains the type of the root device for this virtual device (e.g., media server device, telephony server device etc.) The allowed values are standardized device types as defined by various UPnP DCPs.

physicalDevice

REQUIRED. This is a multi-valued property and defines a physical device. A virtual device may contain multiple physical devices. It contains the following sub-elements:

UDN

REQUIRED. Contains the UDN of the physical device to be included in the virtual device. This is a single-valued property.

allowedUSN

REQUIRED. Contains the unique service name of the embedded device or service that are to be exposed in the remote network. This is a multi-valued property. This property is only valid when the virtual device is used to filter services from different devices and the value of the virtualDeviceType is "**FILTER**".

A.3 Virtual Device List Template

```
<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads">
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
http://www.upnp.org/schemas/ra/dads-v2.xsd">
< virtualDeviceList >
  <virtualDevice>
    <virtualDeviceID>
      ID of the Virtual Device
    </virtualDeviceID >
    <DDD>
      Device Description Document (according to UDA) that will be
      exposed to the remote network
    </DDD>
    <virtualDeviceType>
      "MERGE"/"AGGREGATE"/"FILTER"
    </virtualDeviceType>
    <rootDeviceType>
      Root device type of the virtual device
    </rootDeviceType>
    <physicalDevice>
      <UDN>
        UDN of the first physical device proxied by this virtual
        device
      </UDN>
      <allowedUSN>
        Unique Service Name of embedded device or service to be
        exposed to the remote network. ONLY used for Filter type virtual
        device
      </allowedUSN>
      ...
    </physicalDevice>
    <physicalDevice>
      UDN of the second physical device proxied by this virtual
      device
    </physicalDevice>
    ...
  </virtualDevice>
  ...
</virtualDeviceList>
</dads>

```

xml

REQUIRED for all XML documents. Case sensitive.

dads

REQUIRED. Must have "urn:schemas-upnp-org:ra:dads" as the value for the xmlns attribute; this references the UPnP Remote Access Working Committee RADA Datastructure Template Schema. As long as the same xmlns is used, the data structure template MUST be backward compatible, i.e. usable by legacy implementations.

virtualDeviceList

REQUIRED. Contains a list of virtual devices currently configured in the remote access server. Contains zero or more of the following sub-elements:

virtualDevice

REQUIRED. Contains a list of devices to be included in the virtual device. Contains the following sub-elements:

virtualDeviceType

REQUIRED. Contains the type of the virtual device. There are three types of virtual devices: MERGE, AGGREGATE, and FILTER.

MERGE: indicates that the virtual device contains only devices of the same type (e.g., media server devices) and the remote network only sees a single device description. A **MERGE** virtual device with only one physical device is allowed and can act as a proxy device. For a **MERGE** virtual device, physical device list MAY be empty at the creation of virtual device. The behavior of such a virtual device is that it will merge all the physical devices of the specified type at the creation time or later when those devices join the local network.

AGGREGATE: indicates that the virtual device includes devices of different types and the remote network may see multiple (embedded) devices. An **AGGREGATE** virtual device with only one physical device is allowed and can act as a proxy device. For an **AGGREGATE** virtual device physical device list MAY be empty at the creation of virtual device. The behavior of such a virtual device is that it will forward action requests to the designated physical device.

FILTER: Indicates that the virtual device allows only specific services or embedded devices in a device to be exposed to the remote network. Please see Appendix B(Enhanced Filtering) for details. For a **FILTER** virtual device, physical devices MUST be present in the local network at the time of creation otherwise the virtual device would not know what to filter.

rootDeviceType

REQUIRED. Contains the type of the root device for this virtual device (e.g., media server device, telephony server device etc.)The allowed values are standardized device types as defined by various UPnP DCPs.

physicalDevice

REQUIRED. This is a multi-valued property and defines a physical device. A virtual device may contain multiple physical devices. It contains the following sub-elements:

UDN

REQUIRED. Contains the UDN of the physical device to be included in the virtual device. This is a single-valued property.

allowedUSN

REQUIRED. Contains the unique service name of the embedded device or service that are to be exposed in the remote network. This is a multi-valued property. This property is only valid when the virtual device is used to filter services from different devices and the value of the virtualDeviceType is "**FILTER**".

Appendix B. Enhanced Filtering (Informative)

The enhanced filtering mechanism solves the problem of exposing all the devices and services under the root device to the remote network. If the root device is exposed to the remote network, then all the embedded devices or services under the root device are exposed to the remote network. However, sometimes it is required that some of the embedded devices or services (e.g., add-on services such as Quality of Service, Low Power, Security, etc) under a device should not be exposed to the remote network even though the root device is exposed. This requires a finer grained filtering mechanism which allows filtering of devices and services under a root device. Figure 4-1 shows an example of a remote access server that implements finer grained filtering mechanism where devices and services under a device are filtered as required.

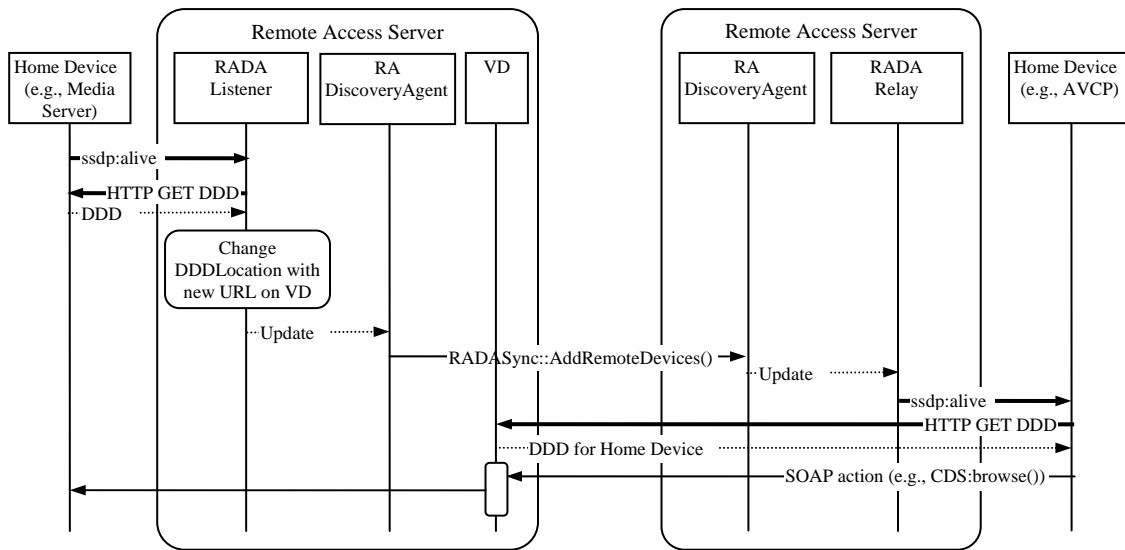


Figure 4-1. An example of the implementation of the enhanced filtering

The creation of a ***FILTER*** type virtual device can achieve enhanced filtering. During the creation of a ***FILTER*** type virtual device the RADAListener receives all the device description and generates a new device description based on the configuration that specifies which devices or services will be exposed to the remote network. The location of the device description is also changed from the actual URL of the device to a newly created URL on the VD (Virtual Device) entity in the RAS. The URL of the DDD of a devices is in the newly created URL on the VD and hence when a device in the remote network wants to retrieve a DDD, it does HTTP GET on the newly created URL on the VD. All the URLs in a DDD usually are relative to the URL that retrieves the DDD. However, in the enhanced filtering case these URLs need to be changed to the absolute URLs. When a device in the remote network, for example, invokes a SOAP action on a service, the control URL of the SOAP action still is for the actual device that implements the service for the action rather than the newly created URL on the RAS.