

INTERNATIONAL STANDARD

ISO/IEC 29341-12-11

Second edition
2015-06-15

Information technology — UPnP Device Architecture —

Part 12-11: Remote User Interface Device Control Protocol - Remote User Interface Server Service

Technologies de l'information — Architecture de dispositif UPnP —

*Partie 12-11: Protocole de contrôle de dispositif d'interface utilisateur
à distance — Service serveur d'interface utilisateur à distance*



Reference number
ISO/IEC 29341-12-11:2015(E)

© ISO/IEC 2015



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2015

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

CONTENTS

Foreword	iv
Introduction.....	v
1. Scope.....	1
2. Normative References	1
3. Service Modeling Definitions	2
3.1. ServiceType	2
3.2. State Variables	2
3.2.1. UIListingUpdate	2
3.2.2. A_ARG_TYPE_DeviceProfile	2
3.2.3. A_ARG_TYPE_URI	3
3.2.4. A_ARG_TYPE_CompatibleUls	3
3.2.5. A_ARG_TYPE_String	3
3.2.6. A_ARG_TYPE_Int	3
3.3. Eventing and Moderation	3
3.4. Actions.....	4
3.4.1. GetCompatibleUls.....	4
3.4.2. SetUILifetime	5
3.4.3. Non-Standard Actions Implemented by a UPnP Vendor	6
3.4.4. Relationships Between Actions.....	6
3.4.5. Common Error Codes.....	6
4. Theory of Operation	7
4.1. Example Values of State Variables.....	7
4.1.1. A_ARG_TYPE_URI	7
4.1.2. A_ARG_TYPE_CompatibleUls.....	7
4.1.3. A_ARG_TYPE_DeviceProfile	11
4.1.4. UIListingUpdate	12
4.1.5. UIFilter Examples.....	12
5. A_ARG_TYPE_CompatibleUls XSD Schema	14
6. DeviceProfile XSD Schema	15
7. XML Service Description	16

LIST OF TABLES

Table 1: Service State Variables	2
Table 2: Event moderation.....	3
Table 3: Actions	4
Table 4: Arguments for GetCompatibleUIs()	4
Table 5: Arguments for SetUILifetime()	5
Table 6: Common Error Codes.....	6
Table 7: shortName values for known remoting protocols	10

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see <http://www.iso.org/directives>).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of the ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: [Foreword – Supplementary information](#)

ISO/IEC 29341-12-11 was prepared by UPnP Implementers Corporation and adopted, under the PAS procedure, by joint technical committee ISO/IEC JTC 1. Information technology, in parallel with its approval by national bodies of ISO and IEC.

This second edition replaces the first edition (ISO/IEC 29341-12-11:2008), which has been technically revised.

The list of all currently available parts of ISO/IEC 29341 series, under the general title *Information technology — UPnP Device Architecture*, can be found on the [ISO web site](#).

Introduction

ISO and IEC draw attention to the fact that it is claimed that compliance with this document may involve the use of patents as indicated below.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights. The holders of - these patent rights have assured ISO and IEC that they are willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC.

Intel Corporation has informed ISO and IEC that it has patent applications or granted patents.

Information may be obtained from:

Intel Corporation
Standards Licensing Department
5200 NE Elam Young Parkway
MS: JFS-98
USA – Hillsboro, Oregon 97124

Microsoft Corporation has informed ISO and IEC that it has patent applications or granted patents as listed below:

6101499 / US; 6687755 / US; 6910068 / US; 7130895 / US; 6725281 / US; 7089307 / US; 7069312 / US; 10/783 524 / US

Information may be obtained from:

Microsoft Corporation
One Microsoft Way
USA – Redmond WA 98052

Philips International B.V. has informed ISO and IEC that it has patent applications or granted patents.

Information may be obtained from:

Philips International B.V. – IP&S
High Tech campus, building 44 3A21
NL – 5656 Eindhoven

NXP B.V. (NL) has informed ISO and IEC that it has patent applications or granted patents.

Information may be obtained from:

NXP B.V. (NL)
High Tech campus 60
NL – 5656 AG Eindhoven

Matsushita Electric Industrial Co. Ltd. has informed ISO and IEC that it has patent applications or granted patents.

Information may be obtained from:

Matsushita Electric Industrial Co. Ltd.
1-3-7 Shiromi, Chuoh-ku
JP – Osaka 540-6139

Hewlett Packard Company has informed ISO and IEC that it has patent applications or granted patents as listed below:

5 956 487 / US; 6 170 007 / US; 6 139 177 / US; 6 529 936 / US; 6 470 339 / US; 6 571 388 / US; 6 205 466 / US

Information may be obtained from:

Hewlett Packard Company
1501 Page Mill Road
USA – Palo Alto, CA 94304

Samsung Electronics Co. Ltd. has informed ISO and IEC that it has patent applications or granted patents.

Information may be obtained from:

Digital Media Business, Samsung Electronics Co. Ltd.
416 Maetan-3 Dong, Yeongtong-Gu,
KR – Suwon City 443-742

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Original UPnP Document

Reference may be made in this document to original UPnP documents. These references are retained in order to maintain consistency between the specifications as published by ISO/IEC and by UPnP Implementers Corporation. The following table indicates the original UPnP document titles and the corresponding part of ISO/IEC 29341:

UPnP Document Title	ISO/IEC 29341 Part
UPnP Device Architecture 1.0	ISO/IEC 29341-1
UPnP Basic:1 Device	ISO/IEC 29341-2
UPnP AV Architecture:1	ISO/IEC 29341-3-1
UPnP MediaRenderer:1 Device	ISO/IEC 29341-3-2
UPnP MediaServer:1 Device	ISO/IEC 29341-3-3
UPnP AVTransport:1 Service	ISO/IEC 29341-3-10
UPnP ConnectionManager:1 Service	ISO/IEC 29341-3-11
UPnP ContentDirectory:1 Service	ISO/IEC 29341-3-12
UPnP RenderingControl:1 Service	ISO/IEC 29341-3-13
UPnP MediaRenderer:2 Device	ISO/IEC 29341-4-2
UPnP MediaServer:2 Device	ISO/IEC 29341-4-3
UPnP AV Datastructure Template:1	ISO/IEC 29341-4-4
UPnP AVTransport:2 Service	ISO/IEC 29341-4-10
UPnP ConnectionManager:2 Service	ISO/IEC 29341-4-11
UPnP ContentDirectory:2 Service	ISO/IEC 29341-4-12
UPnP RenderingControl:2 Service	ISO/IEC 29341-4-13
UPnP ScheduledRecording:1	ISO/IEC 29341-4-14
UPnP DigitalSecurityCamera:1 Device	ISO/IEC 29341-5-1
UPnP DigitalSecurityCameraMotionImage:1 Service	ISO/IEC 29341-5-10
UPnP DigitalSecurityCameraSettings:1 Service	ISO/IEC 29341-5-11
UPnP DigitalSecurityCameraStillImage:1 Service	ISO/IEC 29341-5-12
UPnP HVAC_System:1 Device	ISO/IEC 29341-6-1
UPnP HVAC_ZoneThermostat:1 Device	ISO/IEC 29341-6-2
UPnP ControlValve:1 Service	ISO/IEC 29341-6-10
UPnP HVAC_FanOperatingMode:1 Service	ISO/IEC 29341-6-11
UPnP FanSpeed:1 Service	ISO/IEC 29341-6-12
UPnP HouseStatus:1 Service	ISO/IEC 29341-6-13
UPnP HVAC_SetpointSchedule:1 Service	ISO/IEC 29341-6-14
UPnP TemperatureSensor:1 Service	ISO/IEC 29341-6-15
UPnP TemperatureSetpoint:1 Service	ISO/IEC 29341-6-16
UPnP HVAC_UserOperatingMode:1 Service	ISO/IEC 29341-6-17
UPnP BinaryLight:1 Device	ISO/IEC 29341-7-1
UPnP DimmableLight:1 Device	ISO/IEC 29341-7-2
UPnP Dimming:1 Service	ISO/IEC 29341-7-10
UPnP SwitchPower:1 Service	ISO/IEC 29341-7-11
UPnP InternetGatewayDevice:1 Device	ISO/IEC 29341-8-1
UPnP LANDevice:1 Device	ISO/IEC 29341-8-2
UPnP WANDevice:1 Device	ISO/IEC 29341-8-3
UPnP WANConnectionDevice:1 Device	ISO/IEC 29341-8-4
UPnP WLANAccessPointDevice:1 Device	ISO/IEC 29341-8-5
UPnP LANHostConfigManagement:1 Service	ISO/IEC 29341-8-10
UPnP Layer3Forwarding:1 Service	ISO/IEC 29341-8-11
UPnP LinkAuthentication:1 Service	ISO/IEC 29341-8-12
UPnP RadiusClient:1 Service	ISO/IEC 29341-8-13
UPnP WANCableLinkConfig:1 Service	ISO/IEC 29341-8-14
UPnP WANCCommonInterfaceConfig:1 Service	ISO/IEC 29341-8-15
UPnP WANDSLLinkConfig:1 Service	ISO/IEC 29341-8-16
UPnP WANEthernetLinkConfig:1 Service	ISO/IEC 29341-8-17
UPnP WANIPConnection:1 Service	ISO/IEC 29341-8-18
UPnP WANPOTSLinkConfig:1 Service	ISO/IEC 29341-8-19
UPnP WANPPPConnection:1 Service	ISO/IEC 29341-8-20
UPnP WLANConfiguration:1 Service	ISO/IEC 29341-8-21
UPnP Printer:1 Device	ISO/IEC 29341-9-1
UPnP Scanner:1.0 Device	ISO/IEC 29341-9-2
UPnP ExternalActivity:1 Service	ISO/IEC 29341-9-10

UPnP Document Title	ISO/IEC 29341 Part
UPnP Feeder:1.0 Service	ISO/IEC 29341-9-11
UPnP PrintBasic:1 Service	ISO/IEC 29341-9-12
UPnP Scan:1 Service	ISO/IEC 29341-9-13
UPnP QoS Architecture:1.0	ISO/IEC 29341-10-1
UPnP QosDevice:1 Service	ISO/IEC 29341-10-10
UPnP QosManager:1 Service	ISO/IEC 29341-10-11
UPnP QosPolicyHolder:1 Service	ISO/IEC 29341-10-12
UPnP QoS Architecture:2	ISO/IEC 29341-11-1
UPnP QOS v2 Schema Files	ISO/IEC 29341-11-2
UPnP QosDevice:2 Service	ISO/IEC 29341-11-10
UPnP QosManager:2 Service	ISO/IEC 29341-11-11
UPnP QosPolicyHolder:2 Service	ISO/IEC 29341-11-12
UPnP RemoteUIClientDevice:1 Device	ISO/IEC 29341-12-1
UPnP RemoteUIServerDevice:1 Device	ISO/IEC 29341-12-2
UPnP RemoteUIClient:1 Service	ISO/IEC 29341-12-10
UPnP RemoteUIServer:1 Service	ISO/IEC 29341-12-11
UPnP DeviceSecurity:1 Service	ISO/IEC 29341-13-10
UPnP SecurityConsole:1 Service	ISO/IEC 29341-13-11

INFORMATION TECHNOLOGY – UPNP DEVICE ARCHITECTURE –

Part 12-11: Remote User Interface Device Control Protocol – Remote User Interface Server Service

1. Scope

This service definition is compliant with the UPnP Device Architecture version 1.0.

This service is required for all Remote UI server devices.

It is specified in: **urn:schemas-upnp-org:service:RemoteUIServerDevice**

2. Normative References

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 29341-4-12, *Information Technology – UPnP Device Architecture – Part 4-12: Audio video Device Control Protocol – Level 2 – Content Directory Service*

IETF RFC 1738, *Uniform Resource Locators (URL)*, Tim Berners-Lee, et. Al., December 1994.

Available at: <http://www.ietf.org/rfc/rfc1738.txt>.

IETF RFC 3986, *Uniform Resource Identifiers (URI): Generic Syntax*, Tim Berners-Lee, et al, 2005.

Available at: <http://www.ietf.org/rfc/rfc3986.txt>.

3. Service Modeling Definitions

3.1. ServiceType

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

urn:schemas-upnp-org:service:*RemoteUIServer:1*.

3.2. State Variables

Table 1: Service State Variables

Variable Name	Req. or Opt. ¹	Data Type	Allowed Value	Default Value ²	Eng. Units
UIListingUpdate	O	string	Undefined	Empty string	N/A
A_ARG_TYPE_DeviceProfile	R	string	Undefined	Empty string	N/A
A_ARG_TYPE-CompatibleUIs	R	string	Undefined	Empty string	N/A
A_ARG_TYPE_String	R	string	Undefined	Empty string	N/A
A_ARG_TYPE_URI	O	uri	Undefined	Empty string	N/A
A_ARG_TYPE_Int	O	int	Undefined	0	N/A
<i>Non-standard state variables implemented by a UPnP vendor go here.</i>	<i>X</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>

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

² Default values listed in this column are required. To specify standard optional values or to delegate assignment of values to the vendor, you must reference a specific instance of an appropriate table below.

3.2.1. UIListingUpdate

The optional, evented *UIListingUpdate* state variable informs control points of changes in the listing of user interfaces exposed by the Remote UI server. *UIListingUpdate* is composed of a list of unique identifiers corresponding to user interfaces that have changed since the last event. Each line in the listing, including the last line, must be terminated with a carriage-return character and a linefeed character. *UIListingUpdate* values may consist of an empty string when the event is issued by the Remote UI server for the first time. An example value of *UIListingUpdate* is shown in section 4.1.4.

3.2.2. A_ARG_TYPE_DeviceProfile

A_ARG_TYPE_DeviceProfile values are UTF-8 XML-formatted strings used by the UI client device to represent the list of all supported remoting protocols.

3.2.3. A_ARG_TYPE_URI

An *A_ARG_TYPE_URI* value is a string formatted as a URI. UI server devices must be able to support values of *A_ARG_TYPE_URI* that are 1024 bytes in length. UI server device support for longer values is optional. The URI value must be properly escaped URIs as described in IETF RFC 3986. In addition, URI values must be escaped according to the requirements of IETF RFC 1738.

3.2.4. A_ARG_TYPE-CompatibleUIs

An *A_ARG_TYPE-CompatibleUIs* value is a string formatted as UTF-8 XML representing the list of all UIs that are compatible with a designated Remote UI client device. Remote UI server devices must be able to support values of *A_ARG_TYPE-CompatibleUIs* that are 10k bytes in length. UI server support for longer values is optional.

3.2.5. A_ARG_TYPE_String

A simple string type.

3.2.6. A_ARG_TYPE_Int

A simple integer type.

3.3. Eventing and Moderation

Table 2: Event moderation

Variable Name	Evented	Moderated Event	Max Event Rate ¹	Logical Relation	Min Delta per Event ²
UIListingUpdate	Yes	Yes	0.5	N/A	N/A
A_ARG_TYPE_DeviceProfile	No	N/A	N/A	N/A	N/A
A_ARG_TYPE-CompatibleUIs	No	N/A	N/A	N/A	N/A
A_ARG_TYPE_String	No	N/A	N/A	N/A	N/A
A_ARG_TYPE_URI	No	N/A	N/A	N/A	N/A
A_ARG_TYPE_Int	No	N/A	N/A	N/A	N/A

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

² (N) * (allowedValueRange Step).

3.4. Actions

Immediately following this table is detailed information about these actions, including short descriptions of the actions, the effects of the actions on state variables, and error codes defined by the actions.

Table 3: Actions

Name	Req. or Opt. ¹
GetCompatibleUIs	<u>R</u>
SetUILifeTime	<u>O</u>
<i>Non-standard actions implemented by an UPnP vendor go here.</i>	<u>X</u>

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

3.4.1. GetCompatibleUIs

This action retrieves a list of user interfaces that are compatible with a given client device as specified by the client *A_ARG_TYPE_DeviceProfile*. It returns a (possibly empty) list of UIs compatible with the specified device profile.

3.4.1.1. Arguments

Table 4: Arguments for GetCompatibleUIs ()

Argument	Direction	relatedStateVariable
InputDeviceProfile	<u>IN</u>	A_ARG_TYPE_DeviceProfile
UIFilter	<u>IN</u>	A_ARG_TYPE_String
UIListing	<u>OUT</u>	A_ARG_TYPE_CompatibleUIs

- **UIFilter** is a string employed to request additional filtering of the **UIListing** output. **UIFilter** is employed to limit the **UIListing** result to UIs that possess specific values of *A_ARG_TYPE_CompatibleUIs* schema elements and attributes. **UIFilter** is composed of a comma-separated value list of *A_ARG_TYPE_CompatibleUIs* schema elements, attributes and their values (see section 3.1.5 for **UIFilter** examples). See 1.3.1 in ISO/IEC 29341-4-12 for detailed definition of comma-separated value list (CSV). The **InputDeviceProfile** argument always takes precedence over the **UIFilter** argument in generating the **UIListing** result.
- When the value of **InputDeviceProfile** is an empty string, *all* UI listings are returned in the **UIListing** output argument.
- When no UIs matching the **InputDeviceProfile** are found, **GetCompatibleUIs()** is successful and returns an empty string in the **UIListing** output argument..

3.4.1.2. Dependency on State (if any)**3.4.1.3. Effect on State****3.4.1.4. Errors**

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	Operation Rejected	The Remote UI server has rejected the operation.
702	Invalid Filter Argument	GetCompatibleUIs() failed because the UIFilter argument specifies an invalid filter.

3.4.2. SetUILifetime

Sets the lifetime of UIs for those UIs that permit the default lifetime setting to be modified. A control point may alter the default lifetime setting to change the policy for setting the persistence of the UI after it has been disconnected from a RUI client. After a UI has been disconnected (or was never connected) and the lifetime expires, a UI server will presumably destroy that UI and the state (if any) of the associated application.

For UIs that permit the default lifetime setting to be modified, the lifetime can be set to one of the following values:

- *-1*: the UI has no finite lifetime and can be assumed to persist indefinitely, regardless of whether or not it is connected to a RUI client.
- *0*: the lifetime of the UI has expired.
- *n*: where $n > 0$: the lifetime of the UI will expire *n* minutes from now.

3.4.2.1. Arguments**Table 5: Arguments for SetUILifetime ()**

Argument	Direction	relatedStateVariable
UI	<u>IN</u>	A_ARG_TYPE_URI
Lifetime	<u>IN</u>	A_ARG_TYPE_Int

3.4.2.2. Dependency on State (if any)

None

3.4.2.3. Effect on State (if any)

None

3.4.2.4. Errors

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	Operation Rejected	The Remote UI server has rejected the operation.
703	Unsupported Lifetime	SetUILifetime() failed because the lifetime specified by the <i>Lifetime</i> argument is not supported for the specified UI.

3.4.3. Non-Standard Actions Implemented by a UPnP Vendor

To facilitate certification, non-standard actions implemented by UPnP vendors should be included in this service template. The UPnP Device Architecture lists naming requirements for non-standard actions (see the section on Description).

3.4.4. Relationships Between Actions

All actions defined have no specific relationship between them.

3.4.5. Common Error Codes

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 6: Common Error Codes

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	Operation Rejected	The Remote UI server has rejected the operation.
702	Invalid Filter	The action failed because a filter specified as an input argument is invalid.
703	Unsupported Lifetime	The action failed because a requested UI lifetime value specified as an input argument is not supported for the specified UI.

4. Theory of Operation

4.1. Example Values of State Variables

4.1.1. A_ARG_TYPE_URI

An *A_ARG_TYPE_URI* value is a string formatted as a URI as described in 3.2.3.

4.1.1.1. UIs

UIs are *A_ARG_TYPE_URI* strings of the following form that conforms to the syntax defined in IETF RFC 3986:

$$\langle PI \rangle : // \langle SIP \rangle [: \langle LPT \rangle] [/ \langle AID \rangle],$$

where:

PI: A Protocol identifier string. A short string that identifies a peer-to-peer remoting protocol, e.g.: *RDP*, *VNC*, *XRT*, etc. The syntax corresponds to the “scheme” term in the ABNF notation defined in IETF RFC 3986 Appendix A.

SIP: An IP address of a server device which can be a fully qualified domain name or literal IPv4 or IPv6 address. The syntax corresponds to the “host” term in the ABNF notation defined in IETF RFC 3986 Appendix A.

LPT: A Server port number. The syntax corresponds to the “port” term in the ABNF notation defined in IETF RFC 3986 Appendix A.

AID: An Application ID. A string that identifies a user interface or a remote-capable application. The value of *AID* can include a session ID. This corresponds to the “segment” term in the ABNF notation defined in IETF RFC 3986 Appendix A.

UIs can contain spaces and tabs, but cannot begin with a white space character.

Example:

The following UI corresponds to a Remote UI-enabled DVD browser application available at IP address and port number *1.8.7.2:333*. The *XRT2* remoting protocol is used in this case:

XRT2://1.8.7.2:333/DVDui

4.1.2. A_ARG_TYPE-CompatibleUIs

The value of an *A_ARG_TYPE-CompatibleUIs* is an XML block corresponding to a list of applications that are able to generate remote user interfaces, as well as status information on applications associated with UIs.

Remote UI server devices must be able to support values of *A_ARG_TYPE-CompatibleUIs* that are 10k bytes in length. Remote UI server device support for longer values is optional.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<uilst xmlns="urn:schemas-upnp-org:remoteui:uilst-1-0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:schemas-upnp-
org:remoteui:uilst-1-0 CompatibleUIs.xsd">
  <ui>
    <uiID>4560-9876-1265-8758</uiID>
    <name>Music player</name>
```

```

<description>Music browsing and playback application</description>
<iconList>
  <icon>
    <mimetype>image/png</mimetype>
    <width>40</width>
    <height>40</height>
    <depth>8</depth>
    <url>/icon40.png</url>
  </icon>
  <icon>
    <mimetype>image/png</mimetype>
    <width>160</width>
    <height>160</height>
    <depth>8</depth>
    <url>/icon160.png</url>
  </icon>
  <icon>
    <mimetype>image/jpeg</mimetype>
    <width>40</width>
    <height>40</height>
    <depth>24</depth>
    <url>/icon40.jpeg</url>
  </icon>
  <icon>
    <mimetype>image/jpeg</mimetype>
    <width>160</width>
    <height>160</height>
    <depth>24</depth>
    <url>/icon160.jpeg</url>
  </icon>
</iconList>
<fork>true</fork>
<lifetime>-1</lifetime>
<protocol shortName="VNC">
  <uri>VNC://1.3.4.5:5910/</uri>
  <uri>VNC://1.5.6.7:5910/</uri>
  <protocolInfo>...opaque...</protocolInfo>
</protocol>
<protocol shortName="XRT2">
  <uri>XRT2://1.3.4.5:444/music_player</uri>
  <uri>XRT2://1.5.6.7:444/music_player</uri>
  <protocolInfo>...opaque...</protocolInfo>
</protocol>
</ui>
<ui>
  <uiID>6789-568</uiID>
  <name>DVD Browser</name>
  <protocol shortName="XRT2">
    <uri>XRT2://1.8.7.2:333/DVDui</uri>
    <protocolInfo>...opaque...</protocolInfo>
  </protocol>
  <protocol shortName="RDP">
    <uri>RDP://1.8.7.2:555</uri>
  </protocol>
</ui>
</uilib>

```

In the example above, the first remoted application is a music player application and the second is an application for DVD browsing.

Section 4 contains the XSD schema that can be used to validate *A_ARG_TYPE-CompatibleUIs* values.

All values within XML leaf elements are always XML escaped.

<ui>

This required element occurs once for every remote UI advertised by the Remote UI server. Every **<ui>** element contains a detailed description of a single UI.

<uiID>

This required element uniquely identifies a UI.

If two or more servers on the same network serve Remote UIs for the exact same application, the **<uiID>** values must be exactly the same in each Remote UI server *A_ARG_TYPE-CompatibleUIs* listing.

It is recommended that Remote UI server implementations initialize and assign a unique **<uiID>** value for each enumerated Remote UI-enabled application and persist each **<uiID>** value through power cycles, etc. A Remote UI server generating fresh **<uiID>** values after each power cycle event may prevent Remote UI client devices from using the **<uiID>** values to 'bookmark' favorite remote UIs.

A **<uiID>** value must never be longer than 256 bytes in length.

<uiID> values are UTF-8 encoded strings that must not contain: special XML characters “<”, “>”; quotes; and leading or trailing space or tab characters.

<name>

This required element is a human readable, UTF-8 encoded string no longer than 256 bytes. E.g.: “CompanyX Jukebox”. It is recommended that this friendly name not include the name or IP address of the computer that is serving the user interface.

The value of **<name>** must not contain leading or trailing space or tab characters.

<description>

This optional element contains a human readable, UTF-8 encoded string that is no more than 2048 bytes long. The value of this element defines what the Remote UI-enabled application does, e.g.: “Organize and play your latest and favorite music from a huge selection available from CompanyX”.

The value of **<description>** must not contain leading or trailing space or tab characters.

<iconList>

This optional element contains a sequence of **<icon>** elements that each describe the form of an icon available to represent this UI. It is recommended to include 40 by 40 and 160 by 160 icons in both the PNG and JPEG image formats.

<icon>

When **<iconList>** is present, it must include at least one **<icon>** child element. The **<icon>** element includes five optional attributes:

mimetype: MIME type of the icon,
width: (in pixels) of the icon,
height: (in pixels) of the icon,
depth: color depth (in bits) of the icon,
url

<fork>

This optional element indicates whether or not connecting to this UI will fork a new UI to which the connecting client is actually connected. When **<fork>** is *true*, a new UI is spawned by the application in response to a connection request. When **<fork>** is *true*, subsequent UI listings may (depending on the UI server implementation) expose new UIs within the *A_ARG_TYPE-CompatibleUIs* listing corresponding to the UIs that were spawned in response to individual connection requests. When **<fork>** is *false*, new connection requests do not cause a new UI to be spawned and no new UI listings will appear in

A_ARG_TYPE_CompatibleUIs. The value of **<fork>** must be set to *true* or *false*. The default value of **<fork>** is *false*.

<lifetime>

This optional element indicates the lifetime value set for this UI. A control point may alter the default lifetime setting to change the policy for setting the persistence of the UI after it has been disconnected from a RUI client using the **SetLifetime()** action. After a UI has been disconnected (or was never connected) and the lifetime expires, a UI server will presumably destroy that UI and the state (if any) of the associated application.

For UIs that permit the default lifetime setting to be modified, the lifetime value is set to one of the following values:

- *-1*: the UI has no finite lifetime and can be assumed to persist indefinitely, regardless of whether or not it is connected to a RUI client.
- *0*: the lifetime of the UI has expired.
- *n*: where $n > 0$. The lifetime of the UI will expire *n* minutes from now.

The default value of **<lifetime>** is *-1*.

<protocol>

This required element contains all of the information needed for a specific UI remoting protocol. A **<ui>** element may contain multiple child **<protocol>** elements, indicating support for more than one remoting protocols by a single application.

The **<protocol>** element must define a string value for the required *shortName* attribute. Values for *shortName* corresponding to known Remote UI protocols appear in table 7. Implementations employing one or more of the protocols listed in table 7 must use the *shortName* string values as they are shown in the table (all capital letters).

Table 7: shortName values for known remoting protocols

shortName	Description
HTTP/HTML	The application supports remoting with HTML using HTTP as transport.
RDP	The application supports remoting with Microsoft RDP protocol.
VNC	The application supports remoting with AT&T VNC protocol.
XRT2	The application supports remoting with Intel XRT2 protocol.
LRDP	The application supports remoting with Nokias' LRDP protocol.
XHT	The application supports remoting with Samsungs' XHT protocol.
SGXML	The application supports remoting with Siemens' Gigaset XML protocol.
UIF	The application supports remoting with Philips UI Fragments protocol.
<i>(Vendor-specific protocol name)</i>	<i>(Specified by UPnP vendor.)</i>

Implementations may also use *shortName* to expose support for vendor-specific protocols. Vendor-defined *shortName* values may use any combination of upper or lower case letters. Short name values must be UTF-8 encoded and no longer than 256 bytes.

<uri>

At least one <uri> element must be present within each <protocol> tag. This element is used to specify a URI that can be used by a Remote UI client to connect to this Remote UI-enabled application. The contents of the <uri> element must conform to the rules described in section 4.1.1.

Multiple <uri> elements are present for the same protocol if the Remote IU server has multiple connectable network interfaces available for that protocol.

All URIs appearing in the *A_ARG_TYPE-CompatibleUIs* listing must be unique.

<protocolInfo>

The optional <protocolInfo> tag contains a block of data that is specific to a given remoting protocol. This block of data may contain information that can help in establishing preferences or compatibility between a Remote UI server and a Remote UI client.

Some protocols negotiate all of the session parameters out-of-band upon establishment of the remoting session. **RDP** for example, requires no additional compatibility criteria to be provided by UPnP Remote UI. In these cases the <protocolInfo> block is not needed.

4.1.3. A_ARG_TYPE_DeviceProfile

A_ARG_TYPE_DeviceProfile values are XML-formatted strings used by the Remote UI client device to represent its list of all remoting protocols supported by the Remote UI client device. The format of the device profile is UTF-8 encoded XML.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<deviceprofile xmlns="urn:schemas-upnp-org:remoteui:devprofile-1-0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:schemas-upnp-
org:remoteui:devprofile-1-0 DeviceProfile.xsd">
  <maxHoldUI>5</maxHoldUI>
  <protocol shortName="LRDP">
    <protocolInfo>LRDP:image 1500 UDP beep:sendonly</protocolInfo>
  </protocol>
  <protocol shortName="XHT">
    <protocolInfo> (opaque) </protocolInfo>
  </protocol>
  <protocol shortName="RDP"/>
  <protocol shortName="XRT2">
    <protocolInfo>version=2.1,displayWidth=640,displayHeight=480,imageEncoding=JPEG&PNG,serverV
olumeControl=TRUE,videoViewPortRequired=TRUE</protocolInfo>
  </protocol>
</deviceprofile>
```

A_ARG_TYPE_DeviceProfile values are validated with the XSD schema in section 5.

<MaxHoldUI>

This optional element defines how many user interfaces can be put on hold or in the background. If the Remote UI client does not support placing user interfaces on hold, the <MaxHoldUI> value is set to 0.

The default value of <MaxHoldUI> is 0.

<protocol>

This required tag contains all the information needed for a specific UI remoting protocol. Multiple child **<protocol>** elements may be used to indicate support for more than one remoting protocol.

The **<protocol>** tag must define a string value for the required *shortName* attribute. Values for *shortName* corresponding to known Remote UI protocols appear in table 7. Implementations employing one or more of the protocols listed in table 7 must use the *shortName* string values as they are shown in the table (all capital letters).

Implementations may also use *shortName* to expose support for vendor-specific protocols. Vendor-defined *shortName* values may use any combination of upper or lower case letters. Short name values must be UTF-8 encoded and no longer than 256 bytes.

<protocolInfo>

The optional **<protocolInfo>** tag contains a block of data that is specific to a given remoting protocol. This block of data may contain information that can help in establishing preferences or compatibility between a Remote UI server and a Remote UI client.

Some protocols negotiate all of the session parameters out-of-band upon establishment of the remoting session. **RDP** for example, requires no additional compatibility criteria to be provided by UPnP Remote UI. In these cases the **<protocolInfo>** block is not needed.

4.1.4. UIListingUpdate

The optional *UIListingUpdate* state variable is composed of a list of unique identifiers corresponding to user interfaces that have changed since the last event. Each user interface is demarcated within the list by **<uiID>** tags. Each line in the listing, including the last line, must be terminated with a carriage-return character and a linefeed character. *UIListingUpdate* values may consist of an empty string when the event is issued by the Remote UI server for the first time.

Example:

```
<uiID>4560-9876-1265-8758</uiID>
<uiID>6294-2053-A34E-A432</uiID>
<uiID>6789-568</uiID>
```

4.1.5. UIFilter Examples

The **GetCompatibleUIs()** action returns UI listings that are a match for the *InputDeviceProfile* argument. Additional filtering of the UIs returned in the *UIListing* output argument can be accomplished through the use of the *UIFilter* input argument.

UIFilter is composed of a comma-separated value list of *A_ARG_TYPE_CompatibleUIs* schema elements, attributes and their values. See 1.3.1 in ISO/IEC 29341-4-12 for detailed definition of comma-separated value list (CSV).

Example 1:

The following *UIFilter* argument causes only UI listings with 'music' in the value of the required **<name>** element to be returned in the *UIListings* output argument. It also causes the optional **<description>** element to be included, regardless of its value:

```
"name="*music*",description="*",
```

where the wild-card '*' character generates a match on an unlimited number of UTF-8 characters.

Example 2:

The following *UIFilter* argument causes only UI listings with 'png' somewhere in the value of the optional **<icon>** element attribute **mimetype** to be returned. Note that a requested **mimetype** attribute is only returned if

there is an accompanying <icon> element actually present in the UI listings that match *InputDeviceProfile*. The *UIFilter* argument is ignored otherwise:

"icon@mimetype="*png*"

If the entire *UIFilter* parameter is equal to "*", all optional elements and attributes (when present) and their values, are returned in *UIListing* along with the mandatory elements.

If the entire *UIFilter* parameter value is empty (i.e. absent), all the mandatory elements and attributes and their values, are returned in *UIListing*.

If a value for an optional *A_ARG_TYPE_CompatibleUIs* schema element or attribute is specified in *UIFilter* that is not actually present in any of the UI listings that are a match for the *InputDeviceProfile* argument, then the optional element or attribute is ignored. *GetCompatibleUIs()* returns successfully in these cases.

The *InputDeviceProfile* argument always takes precedence over the *UIFilter* argument in generating the *UIListing* result.

Elements and attributes required by the *A_ARG_TYPE_CompatibleUIs* schema to be present in the *UIListing* output argument are always returned. UI server implementations do not return optional *A_ARG_TYPE_CompatibleUIs* schema elements and attributes unless explicitly requested in the *UIFilter* input argument.

5. A_ARG_TYPE_CompatibleUI s XSD Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:schemas-upnp-org:remoteui:uilst-1-0"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified" id="uilst">
  <xs:element name="uilst">
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="ui">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="uiID" type="xs:string"/>
              <xs:element name="name" type="xs:string"/>
              <xs:element name="description" type="xs:string" minOccurs="0"/>
              <xs:element name="iconList" minOccurs="0">
                <xs:complexType>
                  <xs:sequence maxOccurs="unbounded">
                    <xs:element name="icon">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="mimetype" type="xs:string"/>
                          <xs:element name="width" type="xs:positiveInteger"/>
                          <xs:element name="height" type="xs:positiveInteger"/>
                          <xs:element name="depth" type="xs:positiveInteger"/>
                          <xs:element name="url" type="xs:anyURI"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="fork" type="xs:boolean" default="false" minOccurs="0"/>
        <xs:element name="lifetime" type="xs:integer" default="-1" minOccurs="0"/>
        <xs:element name="protocol" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="uri" type="xs:anyURI" nillable="false" maxOccurs="unbounded"/>
              <xs:element name="protocolInfo" type="xs:anyType" nillable="true" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="shortName" type="xs:string" use="required" form="unqualified"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

6. DeviceProfile XSD Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:schemas-upnp-org:remoteui:devprofile-1-0"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified" id="deviceprofile">
  <xs:element name="deviceprofile">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="maxHoldUI" type="xs:unsignedInt" default="0" minOccurs="0"/>
        <xs:element name="protocol" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="protocolInfo" type="xs:anyType" nillable="true" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="shortName" type="xs:string" use="required" form="unqualified"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

7. XML Service Description

```

<?xml version="1.0" encoding="UTF-8"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
      <name>GetCompatibleUIs</name>
      <argumentList>
        <argument>
          <name>InputDeviceProfile</name>
          <direction>in</direction>
          <relatedStateVariable>A_ARG_TYPE_DeviceProfile</relatedStateVariable>
        </argument>
        <argument>
          <name>UIFilter</name>
          <direction>in</direction>
          <relatedStateVariable>A_ARG_TYPE_String</relatedStateVariable>
        </argument>
        <argument>
          <name>UIListing</name>
          <direction>out</direction>
          <relatedStateVariable>A_ARG_TYPE_CompatibleUIs</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>SetUILifetime</name>
      <argumentList>
        <argument>
          <name>UI</name>
          <direction>in</direction>
          <relatedStateVariable>A_ARG_TYPE_URI</relatedStateVariable>
        </argument>
        <argument>
          <name>Lifetime</name>
          <direction>in</direction>
          <relatedStateVariable>A_ARG_TYPE_Int</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
  </actionList>
  <serviceStateTable>
    <stateVariable sendEvents="yes">
      <name>UIListingUpdate</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_DeviceProfile</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_URI</name>
      <dataType>uri</dataType>
    </stateVariable>
  </serviceStateTable>
</scpd>

```



```
<stateVariable sendEvents="no">
  <name>A_ARG_TYPE_CompatibleUIs</name>
  <dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
  <name>A_ARG_TYPE_String</name>
  <dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
  <name>A_ARG_TYPE_Int</name>
  <dataType>int</dataType>
</stateVariable>
</serviceStateTable>
</scpd>
```

