
RemoteUIClientDevice:1 Device Template Version 1.01

For UPnP Version 1.0

Status: Standardized DCP

Date: April 16, 2014

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 STANDARDIZED DCPS. THE STANDARDIZED DCPS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE STANDARDIZED DCPS, 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.

© 2014 UPnP Forum. All Rights Reserved.

Authors	Company
Ylian Saint-Hilaire	Intel Corporation
Mark Walker	Intel Corporation
Ku Bong Min	LG Electronics
Richard Bardini	Intel Corporation

Contents

1. OVERVIEW AND SCOPE	3
2. DEVICE DEFINITIONS	4
2.1. DEVICE TYPE.....	4
2.2. DEVICE MODEL	4
2.2.1. <i>Description of Device Requirements</i>	4
2.2.2. <i>Relationships Between Services</i>	6
2.3. THEORY OF OPERATION.....	6
2.3.1. <i>Secure Remote UI Clients (if DeviceSecurity implemented in Remote UI client device)</i>	6
3. XML DEVICE DESCRIPTION	7
4. TEST	8
APPENDIX A: ACCESS CONTROL DEFINITIONS (IF <i>DEVICESECURITY</i> SERVICE IS IMPLEMENTED)	9
A1 PERMISSIONS	9
A2 PROFILES	10
A3 ACCESS CONTROL LIST (ACL) ENTRY	10

List of Tables

Table 1: <i>RemoteUIClientDevice</i> Service Descriptions	4
Table 2: Device Requirements for stand-alone <i>RemoteUIClientDevice</i>	4
Table 3: Device Requirements for embedded <i>RemoteUIClientDevice</i>	5
Table 4: Defined permissions for <i>RemoteUIClient</i> Service	9

1. Overview and Scope

This device template is compliant with the UPnP Architecture, Version 1.0.

This document defines the device

urn:schemas-upnp-org:device:RemoteUIClientDevice:1.

This device can be a UPnP root device, or embedded within a different device.

The *RemoteUIClientDevice* encapsulates all services for the Remote UI Client Device Control Protocol (DCP).

2. Device Definitions

2.1. Device Type

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

<urn:schemas-upnp-org:device:RemoteUIClientDevice:1>

2.2. Device Model

It is recommended that *RemoteUIClientDevice* be implemented with support for securing UPnP™ actions. It is also recommended that securing of UPnP™ action is done using the *DeviceSecurity* service as defined by the UPnP™ security working committee. If implemented, the *DeviceSecurity* service must be contained either inside *RemoteUIClientDevice* implementation or in a device that encompasses the *RemoteUIClientDevice*. These two models are described below.

2.2.1. Description of Device Requirements

The following table briefly describes the services used in *RemoteUIClientDevice*.

Table 1: RemoteUIClientDevice Service Descriptions

Service Name	Service Description
<i>RemoteUIClient</i>	Allows for basic operations on a Remote UI client including: user interface connection management, optionally user interface availability management and optionally basic user interaction.
<i>DeviceSecurity</i>	Actions for taking ownership, configuring access control, establishing secure sessions, and invoking secure actions.

2.2.1.1. DeviceSecurity within RemoteUIClientDevice

This model is typically applicable to physical devices that need *DeviceSecurity* functionality (including device ownership and access control) to be used only by the *RemoteUIClientDevice*. In this case, products that expose devices of the type <urn:schemas-upnp-org:device:RemoteUIClientDevice:1> must implement minimum version numbers of the required service specified in the table below.

Table 2: Device Requirements for stand-alone RemoteUIClientDevice

DeviceType	Root	Req. or Opt. ¹	ServiceType	Req. or Opt. ¹	Service ID ²
RemoteUIClientDevice:1	<i>Yes</i>	<i>R</i>	RemoteUIClient:1	<i>R</i>	RemoteUIClient
			DeviceSecurity:1	<i>O</i>	DeviceSecurity
			<i>Non-standard services embedded by an UPnP vendor go here.</i>	<i>X</i>	<i>To be defined by vendor</i>

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

² Prefixed by urn:[upnp-org:serviceId:](urn:schemas-upnp-org:serviceId:) .

Relationship between Services

Figure 1 shows the logical structure of the device and services defined by the working group for UPnP™ technology enabled Remote UI clients.

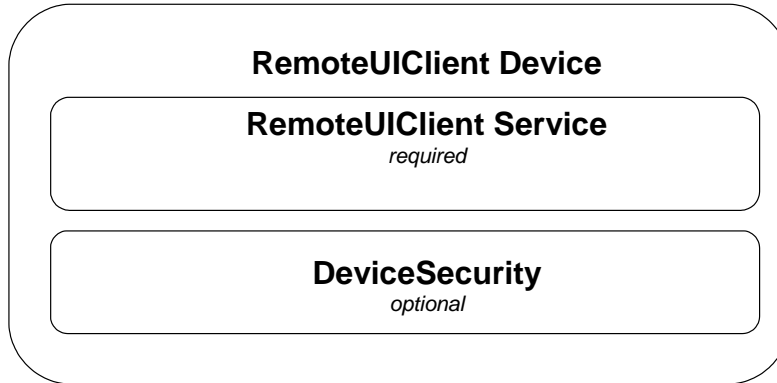


Figure 1: DeviceSecurity within RemoteUIClientDevice

2.2.1.2. DeviceSecurity outside RemoteUIClientDevice

This model is typically applicable to physical devices that implement Remote UI Client functionality, but the *RemoteUIClientDevice* may use *DeviceSecurity* that is already part of another device. An example of this would be where **urn:schemas-upnp-org:device:RemoteUIClientDevice:1** is implemented inside a device of the type **urn:schemas-upnp-org:device:BasicDevice:1**. The *BasicDevice* in this case contains the *DeviceSecurity* service that may be used by another UPnP™ device e.g., *MediaRenderer*. The implementation of *RemoteUIClientDevice* must contain the minimum version number of the service specified in the table below.

Table 3: Device Requirements for embedded RemoteUIClientDevice

DeviceType	Root	Req. or Opt. ¹	ServiceType	Req. or Opt. ¹	Service ID ²
<i>RemoteUIClientDevice:1</i>	<i>Yes</i>	<i>R</i>	<i>RemoteUIClient:1</i>	<i>R</i>	<i>RemoteUIClient</i>
			<i>Non-standard services embedded by an UPnP vendor go here.</i>	<i>X</i>	<i>To be defined by vendor</i>

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

² Prefixed by urn:[upnp-org:serviceId:](#) .

Relationships between Services

Figure 2 shows the logical structure of the device and services defined by the working group for UPnP™ technology enabled Remote UI clients that may use the *DeviceSecurity* service for other UPnP™ devices contained in the same physical device. *RemoteUIClient* service may be dependent on the *DeviceSecurity* service for providing access control to the actions defined in the services.

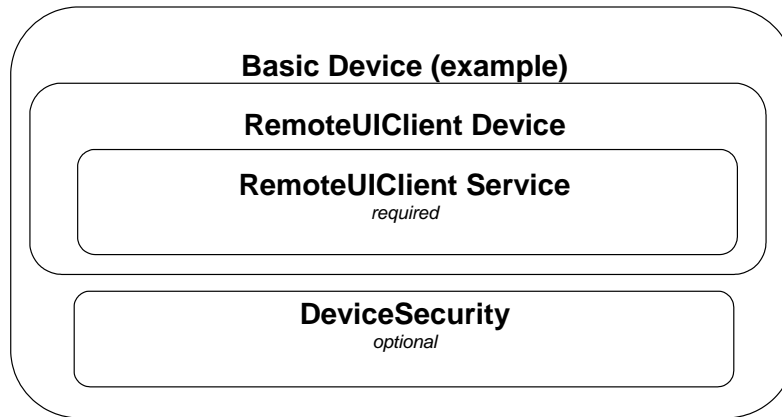


Figure 2: Example of *DeviceSecurity* outside *RemoteUIClientDevice*

2.2.2. Relationships Between Services

The dependencies between the services are listed in the above section under the possible models of implementing services in *RemoteUIClientDevice*.

2.3. Theory of Operation

A Remote UI client can be a fully autonomous device that runs its own user interface in addition to executing remotored UIs, or it may be a fully remotored device with no local UI capability. In both cases, the *RemoteUIClientDevice:1* UPnP device is implemented, allowing anyone on the network to discover and control the Remote UI client. Since many different types of user interaction device exist with many different form factors and usages, the behavior of the Remote UI client device may change.

In any case, it is highly recommended for the Remote UI client to employ the *DeviceSecurity* service to secure specific UPnP™ Remote UI client actions. This section assumes that the reader has an overall understanding of UPnP™ Security. Please refer to the *DeviceSecurity:1* Service Control Specification for detailed description of a secure UPnP™ device.

2.3.1. Secure Remote UI Clients (if *DeviceSecurity* implemented in Remote UI client device)

RemoteUIClient service provides a set of actions to manage remote connections and process user's input data. The actions in this service that change the device state should be authenticated via UPnP™ Security. Some actions in *RemoteUIClient* service can carry critical information such as password in their argument field. By using *DecryptAndExecute* action defined in *DeviceSecurity* service, security sensitive information can be protected. A control point that accesses the secure actions on the service has to be initially authenticated via a Security Console application as described in UPnP™ Security DCP. Access control definitions such as Permissions, Profiles and Access Control List (ACL) for Remote UI client device are described in Appendix A.

3. XML Device Description

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <URLBase>base URL for all relative URLs</URLBase>
  <device>
    <deviceType>urn:schemas-upnp-org:device:RemoteUIClientDevice:1
  </deviceType>
  <friendlyName>short user-friendly title</friendlyName>
  <manufacturer>manufacturer name</manufacturer>
  <manufacturerURL>URL to manufacturer site</manufacturerURL>
  <modelDescription>long user-friendly title</modelDescription>
  <modelName>model name</modelName>
  <modelName>model number</modelName>
  <modelURL>URL to model site</modelURL>
  <serialNumber>manufacturer's serial number</serialNumber>
  <UDN>uuid:UUID</UDN>
  <UPC>Universal Product Code</UPC>
  <iconList>
    <icon>
      <mimetype>image/format</mimetype>
      <width>horizontal pixels</width>
      <height>vertical pixels</height>
      <depth>color depth</depth>
      <url>URL to icon</url>
    </icon>
  </iconList>
  <serviceList>
    <service>
      <serviceType>urn:schemas-upnp-org:service:RemoteUIClient:1</serviceType>
      <serviceId>urn:upnp-org:serviceId:RemoteUIClient</serviceId>
      <SCPDURL>URL to service description</SCPDURL>
      <controlURL>URL for control</controlURL>
      <eventSubURL>URL for eventing</eventSubURL>
    </service>
  </serviceList>
  <presentationURL>URL for presentation</presentationURL>
</device>
</root>
```

4. Test

No semantic tests are defined for this device.

Appendix A: Access Control Definitions (if *DeviceSecurity* service is implemented)

This section specifies the Permissions, Profiles and Access Control List (ACL) entry to be implemented in the *DeviceSecurity* service that can optionally be used by the *RemoteUIClientDevice*. This is used by the Security Console to assign access control of secure actions on the Remote UI client device to control point applications. Please refer to the *DeviceSecurity1.0* service specification for more details about Security Console, Permissions, Profiles and ACLs.

A1 Permissions

The following table describes the permissions to perform access control to the secure actions of the services embedded in the Remote UI client device. The *RUICDeviceAll* is a required permission which can securely access all the actions in the *RemoteUIClient* service. The other permissions are optional. Vendors may define additional set of permissions to perform access control on the Remote UI client device. For example, they may provide separate master and guest permissions for finer granularity of access. However for maximum interoperability, vendors should use the optional permissions presented in this document rather than implementing their own security permissions.

Table 4: Defined permissions for *RemoteUIClient* Service

Permission	Allowed Actions
<i>RUICDeviceAll</i> ¹	<i>All actions in RemoteUIClient Service.</i>
<i>RUICDeviceInfo</i>	<i>GetCurrentConnections, GetDeviceProfile, GetUIListing</i>
<i>RUICDeviceChangeConnection</i>	<i>Connect, Disconnect</i>
<i>RUICDeviceChangeStatus</i>	<i>AddUIListing, RemoveUIListing, DisplayMessage, ProcessInput</i>

¹ *RUICDeviceAll* must be implemented.

When implementing only the required *RUICDeviceAll* permission, the following XML format is used:

```
<Permission>
  <UName>RUICDeviceControl</UName>
  <ACLEntry>
    <RUIWG:RUICDeviceAll/>
  </ACLEntry>
  <FullDescriptionURL></FullDescriptionURL>
  <ShortDescription>
    This permission allows the control point to set and get all secure actions
    of all the services of the Remote UI client device.
  </ShortDescription>
</Permission>
```

XML element tags *UName*, *ACLEntry*, *FullDescription*, *ShortDescription* and *Permission* are defined in the *DeviceSecurity1.0* service specification.

The above defined permission is returned by the Remote UI client device in the “*DefinedPermissions*” argument of *DeviceSecurity*’s *GetDefinedPermission* action

If the *DeviceSecurity* service resides **inside** the *RemoteUIClientDevice*, it will contain only the defined permissions of the Remote UI client device (as mentioned above). The “DefinedPermissions” argument of GetDefinedPermission action returned by the *DeviceSecurity* in this case would be:

```
<DefinedPermissions>
  <Permission>
    <UName>RUICDeviceControl</UName>
    <ACLEntry>
      <RUIWG:RUICDeviceAll/>
    </ACLEntry>
    <FullDescriptionURL></FullDescriptionURL>
    <ShortDescription>
      Allow this application to complete control of the Remote UI client device.
    </ShortDescription>
  </Permission>
</DefinedPermissions>
```

If the *DeviceSecurity* service resides **outside** of the *RemoteUIClientDevice* and the *RemoteUIClientDevice* is embedded in a container device with other devices such as *MediaRenderer*, the “DefinedPermissions” argument of GetDefinedPermission action returned by the *DeviceSecurity* service in this case would be:

```
<DefinedPermissions>
  <Permission>
    <UName>RUICDeviceControl</UName>
    <ACLEntry>
      <RUIWG:RUICDeviceAll/>
    </ACLEntry>
    <FullDescriptionURL></FullDescriptionURL>
    <ShortDescription>
      Allow this application to complete control of the Remote UI client device.
    </ShortDescription>
  </Permission>
  <Permission>
    e.g., Permission defined by MediaRenderer Device
  </Permission>
  ...
</DefinedPermissions>
```

A2 Profiles

There is no profile specified to be used for the Remote UI client device. However, vendors may define profiles of their own. Please refer to *DeviceSecurity*1.0 service specification for more details.

A3 Access Control List (ACL) entry

If *DeviceSecurity* service is implemented in the UPnP™ Remote UI client device, *RemoteUIClient* would have the “<RUIWG:RUICDeviceAll>” defined permission for access control. Following XML shows an example ACL entry granting this defined permission to the control point specified in the subject element. The string value “dRDPBgZz...” under the <hash> tag denotes the public key hash of the control point for which this ACL is defined as an example.

```
<acl>
```

```
<entry>
  <subject>
    <hash>
      <algorithm>SHA1</algorithm>
      <value>dRDPBgZzTFq7Jl2Q2N/YNghcfj8=</value>
    </hash>
  </subject>
  <access>
    <RUIWG:RUICDeviceAll/>
  </access>
  <valid>
    <not-before>2002-10-23_05:17:32</not-before>
    <not-after>2004-12-31_23:59:59</not-after>
  </valid>
</entry>
</acl>
```