

---

# **RadiusClient:1**

## **Service Template Version 1.01**

**For UPnP™ Version 1.0**  
**Status: Standardized DCP**  
**Date: October 17, 2003**

---

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.

© 2000-2003 Contributing Members of the UPnP™ Forum All rights Reserved

<b>Authors</b>	<b>Company</b>
Trevor Freeman	Microsoft Corporation
Ajay Garg	Intel Corporation

## Contents

<b>1. OVERVIEW AND SCOPE .....</b>	<b>3</b>
<b>2. SERVICE MODELING DEFINITIONS .....</b>	<b>3</b>
2.1. SERVICE TYPE .....	3
2.2. STATE VARIABLES .....	3
2.2.1. NumberOfAuthServerEntries .....	4
2.2.2. AuthServerIPAddress .....	4
2.2.3. AuthServerPortNumber .....	4
2.2.4. AuthServerSharedSecret .....	4
2.3. EVENTING AND MODERATION .....	4
2.3.1. Event Model .....	4
2.4. ACTIONS .....	5
2.4.1. GetGenericAuthServerEntry .....	5
2.4.2. GetSpecificAuthServerEntry .....	6
2.4.3. AddAuthServerEntry .....	6
2.4.4. DeleteAuthServerEntry .....	7
2.4.5. FactoryDefaultReset .....	8
2.4.6. ResetAuthentication .....	8
2.4.7. Non-Standard Actions Implemented by a UPnP™ Device Vendor .....	9
2.4.8. Common Error Codes .....	9
2.5. THEORY OF OPERATION .....	9
2.5.1. Operation of the RADIUS Client .....	9
<b>3. XML SERVICE DESCRIPTION .....</b>	<b>9</b>
<b>4. TEST .....</b>	<b>12</b>

## List of Tables

Table 1: State Variables .....	3
Table 2: Event Moderation .....	4
Table 3: Actions .....	5
Table 4: Arguments for GetGenericAuthServerEntry .....	5
Table 5: Arguments for GetSpecificAuthServerEntry .....	6
Table 6: Arguments for AddAuthServerEntry .....	7
Table 7: Arguments for DeleteAuthServerEntry .....	7
Table 8: Common Error Codes .....	9

## 1. Overview and Scope

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

This service enables the **control and configuration of radius component of IEEE 802.11 Wireless Access Points for the unmanaged network space, namely residential and small office LANs.**

Its intent is to simplify the setup experience, secure wireless networks and provide the framework for diagnosing and monitoring problems on wireless networks.

- This service-type enables remote setup and configuration of the **RADIUS related parameters of a wireless access point**

## 2. Service Modeling Definitions

### 2.1. ServiceType

The service is OPTIONAL as specified in  
**urn:schemas-upnp-org:device:WLANAccessPointDevice:1**

The following service type identifies a service that is compliant with this template:  
**urn:schemas-upnp-org:service:RadiusClient:1**

This service does not support the QueryStateVariable action.

### 2.2. State Variables

Table 1 shows all the state variables of RadiusClient service.

**Table 1: State Variables**

Variable Name	Req. or Opt. <sup>1</sup>	Data Type	Allowed Value <sup>2</sup>	Default Value <sup>2</sup>	Eng. Units
NumberOfAuthServerEntries	R	ui2	>=0	0	N/A
AuthServerIPAddress	R	String	IP Address, <= 32 char	Empty String	N/A
AuthServerPortNumber	R	ui2	Between 1 and 65535 inclusive	0	N/A
AuthServerSharedSecret	R	String	Shared Secret, <= 128 char	Empty String	N/A
<i>Non-standard state variables implemented by an UPnP™ device vendor go here.</i>	<i>X</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>

<sup>1</sup> R = Required, O = Optional, X = Non-standard.

<sup>2</sup> 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 from the appropriate table below.

### 2.2.1. NumberOfAuthServerEntries

This variable indicates the number of Authentication Server entries (number of elements in the array) configured for this access point. The access point will attempt authentication with the authentication servers listed in the array in order. This variable is read/write and is evented.

### 2.2.2. AuthServerIPAddress

This variable is an IPv4 or IPv6 address of the Authentication Server, such as a RADIUS Server, for 802.1x-based Authentication.

### 2.2.3. AuthServerPortNumber

This variable is the port number (such as 1645 or 1812 for RADIUS) of the Authentication Server, such as a RADIUS Server, for EAP-based Authentication. This variable is read/write.

### 2.2.4. AuthServerSharedSecret

This variable is a string that represents the password in plain text for the access point to authenticate into the Authentication Server, such as a Radius Server, for EAP-based Authentication. This variable is read/write.

## 2.3. Eventing and Moderation

Table 2: Event Moderation

Variable Name	Evented	Moderated Event	Max Event Rate <sup>1</sup>	Logical Combination	Min Delta per Event <sup>2</sup>
NumberOfAuthServerEntries	Yes	No	N/A	N/A	N/A
<i>Non-standard state variables implemented by an UPnP™ device vendor go here.</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>

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

<sup>2</sup> (N) \* (allowedValueRange Step).

### 2.3.1. Event Model

Only one state variable of the *RadiusClient* service is evented:

**NumberOfAuthServerEntries:** This state variable event helps keep the client's authentication server list synchronized with the authentication server list maintained at the AP device.

None of the events are moderated.

## 2.4. Actions

Table 3 lists the required and optional actions for the UPnP™ AP device. This is followed by 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.

Securing UPnP™ actions in this service is optional but strongly recommended, using UPnP™ security protocols as defined by UPnP™ Security working group. If the AP implements security for UPnP™ actions, Table 3 indicates the actions that **MUST** be secure. The others may be implemented as secure or open. Secure actions **MUST** be protected for both confidentiality and integrity.

Access permissions will be inherited from the containing device (e.g., *WLANAccessPointDevice*).

**Table 3: Actions**

Name	Secure or Open*	Req. or Opt.
<b>GetGenericAuthServerEntry</b>	S	R
<b>GetSpecificAuthServerEntry</b>	S	R
<b>AddAuthServerEntry</b>	S	R
<b>DeleteAuthServerEntry</b>	S	R
<b>FactoryDefaultReset</b>	S	R
<b>ResetAuthentication</b>	S	R

<sup>1</sup> R = Required, O = Optional, X = Non-standard.

\* This column is relevant if DeviceSecurity service is present in the container device

### 2.4.1. GetGenericAuthServerEntry

This action retrieves Authentication Server Entries one entry at a time. Control points can call this action with an incrementing array index until no more entries are found on the gateway. If **NumberOfAuthServerEntries** is updated during a call, the process may have to start over. Entries in the array are contiguous. As entries are deleted, the array is compacted, and the evented variable **NumberOfAuthServerEntries** is decremented. Authentication Server Entries are logically stored as an array on the AP and retrieved using an array index ranging from 0 to **NumberOfAuthServerEntries** – 1.

#### 2.4.1.1. Arguments

**Table 4: Arguments for GetGenericAuthServerEntry**

Argument	Direction	relatedStateVariable
<b>NewAuthServerIndex</b>	IN	NumberOfAuthServerEntries
<b>NewAuthServerIPAddress</b>	OUT	AuthServerIPAddress
<b>NewAuthServerPortNumber</b>	OUT	AuthServerPortNumber

Argument	Direction	relatedStateVariable
NewAuthServerSharedSecret	OUT	AuthServerSharedSecret

#### 2.4.1.2. Dependency on State (if any)

#### 2.4.1.3. Effect on State (if any)

#### 2.4.1.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP™ Device Architecture section on Control
713	InvalidIndex	The specified array index is out of bounds

### 2.4.2. GetSpecificAuthServerEntry

This action retrieves the Authentication Server Entry for a specified {address, port} combination.

#### 2.4.2.1. Arguments

Table 5: Arguments for GetSpecificAuthServerEntry

Argument	Direction	relatedStateVariable
NewAuthServerIPAddress	IN	AuthServerIPAddress
NewAuthServerPortNumber	IN	AuthServerPortNumber
NewAuthServerSharedSecret	OUT	AuthServerSharedSecret

#### 2.4.2.2. Dependency on State (if any)

#### 2.4.2.3. Effect on State (if any)

#### 2.4.2.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP™ Device Architecture section on Control
714	NoSuchEntryInArray	The specified values of AuthServerIPAddress and AuthServerPortNumber combination does not exist in the array

### 2.4.3. AddAuthServerEntry

This action creates a new authentication server entry in the Authentication server list.

### 2.4.3.1. Arguments

**Table 6: Arguments for AddAuthServerEntry**

Argument	Direction	relatedStateVariable
NewAuthServerIP	IN	AuthServerIPAddress
NewAuthServerPortNumber	IN	AuthServerPortNumber
NewAuthServerSharedSecret	IN	AuthServerSharedSecret

### 2.4.3.2. Dependency on State (if any)

### 2.4.3.3. Effect on State (if any)

### 2.4.3.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP™ Device Architecture section on Control
701	AlreadyHaveAuthenticationServerEntry	The specified server entry is already present in the authentication server entry this happen when ip,port are same, but different shared secret

## 2.4.4. DeleteAuthServerEntry

This action deletes the existing authentication server entry from the authentication server list. The authentication server entry is identified using authentication server IP address and port number.

### 2.4.4.1. Arguments

**Table 7: Arguments for DeleteAuthServerEntry**

Argument	Direction	relatedStateVariable
NewAuthServerIPAddress	IN	AuthServerIPAddress
NewAuthServerPortNumber	IN	AuthServerPortNumber

### 2.4.4.2. Dependency on State (if any)

### 2.4.4.3. Effect on State (if any)

### 2.4.4.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP™ Device Architecture section on Control

714	NoSuchEntryInArray	The specified values of AuthServerIPAddress and AuthServerPortNumber combination does not exist in the array
-----	--------------------	--

### 2.4.5. FactoryDefaultReset

This action resets all the state variables of the *RadiusClient* service to their factory default settings. It will remove all the radius server entries. This action also resets all the wireless sessions that were authenticated using radius server.

This action must be invoked internally if the control point calls **FactoryDefaultReset** action of the *DeviceSecurity* (or *WLANConfiguration* service if this service resides in the AP device), whereas vice versa is not true i.e., resetting this service will not invoke the *DeviceSecurity* reset (or *WLANConfiguration* reset).

#### 2.4.5.1. Arguments

None

#### 2.4.5.2. Dependency on State (if any)

#### 2.4.5.3. Effect on State (if any)

#### 2.4.5.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP™ Device Architecture section on Control

### 2.4.6. ResetAuthentication

This action resets all wireless stations that were authenticated via the RADIUS server.

This action must be invoked internally if the control point calls **ResetAuthentication** action of the *WLANConfiguration* service if RadiusClient service resides in the AP device, whereas vice versa is not true.

#### 2.4.6.1. Arguments

None

#### 2.4.6.2. Dependency on State (if any)

#### 2.4.6.3. Effect on State (if any)

#### 2.4.6.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP™ Device Architecture section on Control



### 2.4.7. Non-Standard Actions Implemented by a UPnP™ Device Vendor

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

### 2.4.8. 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 8: Common Error Codes**

ErrorCode	errorDescription	Description
401	Invalid Action	See UPnP™ Device Architecture section on Control
402	Invalid Args	See UPnP™ Device Architecture section on Control
404	Invalid Var	See UPnP™ Device Architecture section on Control
501	Action Failed	See UPnP™ Device Architecture section on Control
600-699	TBD	Common action errors. Defined by UPnP™ Forum Technical Committee
701-799		Common action errors defined by the UPnP™ Forum working committees
800-899	<i>TBD</i>	<i>(Specified by UPnP™ device vendor)</i>

## 2.5. Theory of Operation

### 2.5.1. Operation of the RADIUS Client

An AP enabled with UPnP™ technology may maintain the list of remote authentication (RADIUS) servers for EAP authentication. This list can be updated by the control points. The AP device may also have the capability to run the authentication server locally. The AP device will pick an authentication server from the authentication server list in the order they have been added to the list.

## 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>GetGenericAuthServerEntry</name>
      <argumentList>
        <argument>
          <name>NewAuthServerIndex</name>
          <direction>in</direction>
        </argument>
        <relatedStateVariable>NumberOfAuthServerEntries</relatedStateVariable>
      </argument>
      <argument>
        <name>NewAuthServerIPAddress</name>
        <direction>out</direction>
        <relatedStateVariable>AuthServerIPAddress</relatedStateVariable>
      </argument>
    </action>
  </actionList>
</scpd>
```

```

        </argument>
        <argument>
            <name>NewAuthServerPortNumber</name>
            <direction>out</direction>
        </relatedStateVariable>AuthServerPortNumber</relatedStateVariable>
        </argument>
        <argument>
            <name>NewAuthServerSharedSecret</name>
            <direction>out</direction>
        </relatedStateVariable>AuthServerSharedSecret</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>GetSpecificAuthServerEntry</name>
    <argumentList>
        <argument>
            <name>NewAuthServerIPAddress</name>
            <direction>in</direction>
        </relatedStateVariable>AuthServerIPAddress</relatedStateVariable>
        </argument>
        <argument>
            <name>NewAuthServerPortNumber</name>
            <direction>in</direction>
        </relatedStateVariable>AuthServerPortNumber</relatedStateVariable>
        </argument>
        <argument>
            <name>NewAuthServerSharedSecret</name>
            <direction>out</direction>
        </relatedStateVariable>AuthServerSharedSecret</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>AddAuthServerEntry</name>
    <argumentList>
        <argument>
            <name>NewAuthServerIPAddress</name>
            <direction>in</direction>
        </relatedStateVariable>AuthServerIPAddress</relatedStateVariable>
        </argument>
        <argument>
            <name>NewAuthServerPortNumber</name>
            <direction>in</direction>
        </relatedStateVariable>AuthServerPortNumber</relatedStateVariable>
        </argument>
        <argument>
            <name>NewAuthServerSharedSecret</name>
            <direction>in</direction>
        </relatedStateVariable>AuthServerSharedSecret</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>DeleteAuthServerEntry</name>
    <argumentList>
        <argument>
            <name>NewAuthServerIPAddress</name>
            <direction>in</direction>
        </relatedStateVariable>AuthServerIPAddress</relatedStateVariable>
        </argument>
        <argument>
            <name>NewAuthServerPortNumber</name>
            <direction>in</direction>
        </relatedStateVariable>AuthServerPortNumber</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>FactoryDefaultReset</name>
</action>
<action>
    <name>ResetAuthentication</name>
</action>
</actionList>

```

```
<serviceStateTable>
  <stateVariable sendEvents="yes">
    <name>NumberOfAuthServerEntries</name>
    <dataType>ui2</dataType>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>AuthServerIPAddress</name>
    <dataType>string</dataType>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>AuthServerPortNumber</name>
    <dataType>ui2</dataType>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>AuthServerSharedSecret</name>
    <dataType>string</dataType>
  </stateVariable>
</serviceStateTable>
</scpd>
```

## 4. Test

*No semantic tests have been defined for this service..*