

Intel® SDK for UPnP™ Devices

Programming Guide

Intel® SDK for UPnP™ Devices Version 1.2.1

November 2002

Contents

1

Devices don't have to wait for a control point to search for their services. They can advertise their device availability by means of the SSDP NOTIFY command on the 239.255.255.250:1900 multicast address.

On the server side, an event server waits for subscribe and unsubscribe requests. The event server is an HTTP-

1.2 SDK Architecture

The following diagram shows the architecture of the SDK:

The API layer also maintains the table of control point and device handles registered with the SDK. On each call to an API function, the SDK will validate the handle is known by checking in the handle table. Currently, there is a limit of one control point and one device handle that can be allocated in one process at a time. In other words, an application can register once as a device and once as a control point. Any further attempts to register by that application will fail.

For information about the API, refer to the *Intel® SDK for UPnP™ Devices v1.2 API Reference*.

1.2.3 SSDP

The SSDP module implements the Simple Service Discovery Protocol, providing the discovery phase of UPnP. This module allows control points to send multicast searches for services and devices on the network and receives the replies to those searches. It also notifies them when new services are announced on the network.

associated with the device. Using Virtual Directories, described in section 1.3, device applications can request the SDK to generate callbacks into the application to handler4nhe C.3

1.3 Virtual Directories

The integrated Mini Web Server inside the SDK supports a concept known as Virtual Directories. A Virtual

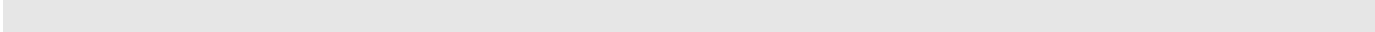
SDK, it should be located off of the directory specified in `UpnpSetWebServerRootDir()`. The second parameter registers a callback with the SDK. The prototype

There are three requests handled by a device:

- subscription requests
- ~~ge0 11.32~~ variable requests

```
int TvDeviceHandleSubscriptionRequest( IN struct Upnp_Subscription_Request *
```

Note that the UPnP Forum has deprecated this capability. For clients to access state variables, the preferable





- an optional cookie to pass to the callback function when invoked (NULL)

`UpnpDownloadXmlDoc()` returns a completely parsed DOM document of the description document, ready to be consumed by the control point application. This function can take a lot of memory if the description document is large, since it is downloaded, parsed, and returned in one chunk. An alternate API that the SDK offers allows much larger HTTP transfers by breaking the transfer down into chunks. Unlike `UpnpDownloadXmlDoc()`, it requires

```
int TvCtrlPointCallbackEventHandler( Upnp_EventType EventType,  
                                     void *Event,  
                                     void *Cookie )  
{
```



```
case UPNP_CONTROL_ACTION_COMPLETE:
{
    struct Upnp_Action_Complete *a_event =
        (struct Upnp_Action_Complete *) Event;

    if (a_event-
```