Linux SDK for UPnP Devices v1.4

Contents

4.2.9 Upnp_

2

License

3 About Callbacks

The Linux SDK for UPnP Devices contains functions that generate asynchronous callbacks. To simplify the application callback functions, these callbacks are executed on a thread owned by the SDK itself. The SDK executes the application's callback function in a thread context so the application can allocate memory and preserve the information it needs. The application can

4.1.3 UPNP_E_INVALID_PARAM [-101]

UPNP_E_INVALID_PARAM signifies that one or more of the parameters passed to the function is not

4.1.18 UPNP_E_SOCKET_READ [-202]

4.1.19 UPNP_E_SOCKET_BIND [-203]

UPNP_E_SOCKET_BIND signifies that the SDK had a problem binding a socket to a network interface. This occurs in any function that makes network connections, such as discovery (e.g. UpnpSearchAsync or UpnpSendAdvertisement), control (e.g. UpnpSendAction), eventing (e.g. UpnpNotify), and HTTP functions (e.g. UpnpDownloadXmlDoc).

4.1.20 UPNP_E_SOCKET_CONNECT [-204]

UPNP_E_SOCKET_READ signifies an error reading from a socket. This occurs in any function that makes network connections, such as discovery (e.g. UpnpSearchAsync or UpnpSendAdvertisement), control (e.g. UpnpSendAction), eventing (e.g. UpnpNotify), and HTTP functions (e.g. UpnpDownloadXmlDoc).

4.1.29

UPNP_E_INVALID_ARGUMENT [-501]

`UPNP_E_INVALID_ARGUMENT`

Received by a device when a control point issues a control request.

4.2.3.6

UPNP_DISCOVERY_ADVERTISEMENT_BYEBYE

Received by a control point when a device or service shuts down.

4.2.3.10

UPNP_EVENT_RECEIVED

Names

By specifying these different values to UpnpSearchAsync, the control point application can

4.2.6.2

UPNPREG_

4.2.7.10

IXML_Document* SoapHeader

4.2.8.7

struct in_addr CtrlPtIPAddr

IP address of sender requesting the state variable.

IP address of sender requesting the state variable.

4.2.8.8

DOMString CurrentVal

The current value of the variable.

4.2.9.2

char **CtrlUrl** [NAME_SIZE]

The control URL for the service.

The control URL for the service.

4.2.9.3

char **StateVarName** [NAME_SIZE]

4.2.9.4

DOM-String

4.2.12.1

```
Upnp_SID Sid
```

4.2.13

Returned along with a UPNP_EVENT_SUBSCRIPTION_REQSUBSCRIPTION

4.2.14

struct UpnpVirtualDirCallbacks

The UpnpVirtualDirCallbacks *functions a device application can register to virtualize URLs.*

Members

4.2.14.1 int

4.2.15

4.3.1

[illegible]

Initializes the LinuxDKPnP Devices.

Initializes the LinuxDKnPs. This functi

4.3.2

EXPORT_SPEC int UpnpFinish

FINISH

Return Value: `[int]` An integer representing one of the following:

- `UPNP_E_SUCCESS`: The operation completed successfully.
- `UPNP_E_FINISH`: The SDK is already terminated or is not initialized.
- `UPNP_E_INVALID_`

Parameters: `descriptionType`

Unregisters a root device registered with UpnpRegisterRootDevice or UpnpRegisterRootDevice2. After this call, the UpnpDevice

Parameters: `contentLength` The maximum permissible content length for incoming SO(t)-AP actions, in b(t)-ytes.

4.4 Discovery

Names

4.4.1 EXPORT_SPEC int
UpnpSearchAsync (IN UpnpClient_Handle Hnd, IN int Mx,
IN const char* Target,
IN const void* Cookie)
UpnpSearchAsync

Parameters:

`Hnd` The handle of the client performing the search.

`Mx` The time, in seconds, to wait Ior responses. If the time is greater than `MAX_SEARCH_TIME` then

UpnpSendActionExAsync (IN UpnpClient_Handle Hnd,

4.5.6

EXPORT_SPEC int **UpnpSendActi-nExAsync** (IN UpnpClient_Handle Hnd, IN c-nst char* ActionURL, INconst char* ServiceType, IN const char* DevUDN, IN IXML_Document* Header, IN IXML_Document* Acti-n, IN Upnp_FunPtr Fun, IN const void* Cookie)

UpnpSendActi-nExAsync *sends a message to change a state variable in a service, generating a callback when the operation is complete.*

UpnpSendActi-nExAsync sends a message to change a state variable in a service, generating a callback when the operati-n is c-mplete. See **UpnpSendActi-n** for c-mments -n positive return values. These p-sitive return values are sent in the event struct associated with the `UPNP_CONTROL_ACTION_COMPLETE` event.

Return Value: `[int]` An integer representing one of the following:

- `UPNP_E_SUCCESS`: The operation c-mpleted successfully.
- `UPNP_E_INVALID` pThe operation c-mpleted

[illegible]

UPNP

UpnpSubscribe

UpnpAcceptSubscription *accepts a subscription request and sends out the current state of the eventable variables for a service.*

UpnpAcceptSubscription accepts a subscription request and sends out the current state of the a DOM document for the variables to event rather than an array of strings. This function is sychronous and generates no callbacks.

UpnpAcceptSubscriptionExt

Return Value: `[int]`

UpnpRenewSubscriptionAsync renews a subscription that is about to expire, generating a callback when the operation is complete.

Note that many of the error codes for this function are returned in the Upnp_Event_Subscribe structure. In those cases, the function returns

request or renewal. The default value accepts the time-out set by the control point. If a control

UpnpSubscribeAsync performs the same operation as UpnpSubscribe, but returns immediately and calls the registered callback function when the operation is complete.

UpnpSubscribeAsync performs the same operation as UpnpSubscribe, but returns immediately and calls the registered callback function when the operation is complete.

Note that many of the error codes for this function are returned in the Upnp_Event_Subscribe structure. In those cases, the function returns UPNP_E_SUCCESS and the appropriate error code will be in the Upnp_Event_Subscribe.ErrCode field in the Event structure passed to the callback.

Return Value: `[int]`

Parameters:

`Hnd` The handle of the control point that is subscribing.

`PublisherUrl` The URL of the service to subscribe to.

`TimeOut` The requested subscription time. Upon return, it contains the actual subscription time returned from the service.

`Fun` Pointer to the callback function for this subscribe request.

`Cookie` A user data value passed to the callback function when invoked.

4.6.11

EXPORT_SPEC int **UpnpUnSubscribe** (IN UpnpClient_Handle Hnd, IN Upnp_SID SubsId)

Return Value: [int] An integer representing one of the following:

- UPNP_E_SUCCESS: The operation completed successfully.
- UPNP_E_INVALID_HANDLE: The handle is not a valid control point handle.
- UPNP_E_INVALID_SID: The SubsId is not a valid subscription ID.
- UPNP_E_NETWORK

Return Value: `[int]` An integer representing one of the following:

- `UPNP_E9990ng one of followi16091001304577125mDT0290f30795712`

Parameters:

`Hnd` The handle of the subscribed control point.

`SubsId` The ID returned when the control point subscribed to the service.

`Fun`

UpnpCloseHttpPost

Return Value: `[int]` An integer representing one of the following:

- UPNP_E_SUCCESS: The operation completed successfully.
- UPNP_E_INVALID_PARAM: Either **url**, **handle**, **contentType**, **contentLength** or **httpStatus** is not a valid pointer.
- UPNP_E_INVALID_URL: The **url** is not a valid URL.
- UPNP_E_OUTOF_MEMORY: Insufficient resources exist to download this file.
- UPNP_E_NETWORK_ERROR: A network error occurred.
- UPNP_E_SOCKET_WRITE: An error or timeout occurred writing to a socket.
- UPNP_E_SOCKET_READ: e8(or50378(or)-507(time)-1(ou)1(t)]TJ89.75982□11

UpnpOpenHttpGetProxy *gets a file specified in a URL through the specified proxy.*

UpnpOpenHttpGetProxy gets a file specified in a URL through the specified proxy. The SDK allocates the memory for **handle** and **contentType**, the application is responsible for freeing this memory.

Return Value: `[int]` An integer representing one of the following:

- `UPNP_E_SUCCESS`: The operation completed successfully.
- `UPNP_E_INVALID_PARAM`: Either **url**, **handle**, **contentType**, **contentLength** or **httpStatus** is not a valid

4.7.4

EXPORT_SPEC int **UpnpOpenHttpGetEx** (IN const char* url, IN OUT void** handle, IN OUT char** contentType, IN OUT int* contentLength, IN OUT int* httpStatus, IN int lowRange, IN int highRange, IN int timeout)

gets specified number of bytes from a file specified in the URL.

Parameters: `url`

4.7.6

EXPORT_SPEC int **UpnpHttpGetProgress** (IN void* handle, OUT unsigned int* length, OUT unsigned int* total)

UpnpHttpGetProgress *rettrieve progress information of a http-get transfer.*

UpnpHttpGetProgress rettrieve progress information of a http-get transfer.

Return Value: `[int]`

UpnpCloseHttpGet closes the connection and frees memory that was allocated for the

Parameters:		
	`url`	The URL in which to send the POST request.
	`handle`	

4.7.11

EXPORT_SPEC int **UpnpCloseHttpPost** (IN void* handle, IN OUT int* httpStatus, IN int timeout)

UpnpCloseHttpPost

Return Value: `[int]` An integer representing one of the following:

- UPNP_E_SUCCESS: The operation completed successfully.
- UPNP_E_INVALID_PARAM: Either **url** or **xmlDoc** is not a valid pointer.
- UPNP_E_INVALID_DESC: The XML document was not found or it does not contain a valid XML description.
- UPNP_E_INVALID_URL url The is not a valid URL.
- UPNP _E_OUTOF_MEMORY: There are insu□cient resources to download the XML document.
- UPNP_E_NETWORK_ERROR: A network error occurred.
- UPNP_E_SOCKET_WRITE: An error or timeout occurred writing to a socket.
- UPNP_E_SOCKET_READ: An error or timeout occurred reading from a s60□1(c)28(kn)29(t.)]TJ/F149□963□Tf-9.963-15.9
- UPNP

I

4.8.4

EXPORT_SPEC int **UpnpIsWebserverEnabled** ()

UpnpIsWebServerEnabled *returns* `TRUE`

Return Value: [int] An integer representing one of the following:

- UPPN_E_SUCCESS: The operation completed successfully.
- UPNP_E_INVALID_ARGUMENT: dirName is not valid.

Parameters: dirName The name of the virtual directory mappBTΩF to 4ETh]TJETQ001915.93575595975Tmq[]0d0.398w0.199m3.15201.99lSQQ

UpnpAddToActionResponse (IN OUT IXML_Document**
ActionResponse,
IN const char* ActionName,
IN const char* ServType,
IN const char* ArgName,
IN const char* ArgVal)

UpnpAddToActionResponse *creates an action response packet based on its out-*

an Pte-3oxSd IN const char* ArgName,

IN const char* ArgVal)

5 Optional Tool APIs

UpnpAddToAction

5.5

EXPORT