Application Note

WICED™ SSDP

WICED™ SSDP

(Simple Service Discovery Protocol)

5300 California Avenue • Irvine, California, 92677 • Phone: 949-926-5000 • Fax: 949-926-5203

WICED-SSDP
September 17, 2015

Revision History

Revision	*Date*	*Change Description*
WICED-SSDP-R 0.1	Septermber 17, 2015	Initial Revision

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

Table of Contents

1 About this Document

1.1 Purpose and Scope

This document provides instructions to use the WICED SSDP package and samples to provide SSDP functionality to your application. Using the sample Applications, API's and WICED SDK utilities, you will be able to add SSDP capability to your IoT device.

Note: This document applies to **WICED SDK 3.3.x** or higher.

1.2 Audience

This document is for software developers who are using the WICED Development System to create applications for secure embedded wireless networked devices.

2 Terminology

HTTP	Hyper-Text Transfer Protocol
SSDP	Simple Service Discovery Protocol is a network protocol for advertisement and discovery of network services and presence.

3 SSDP Daemon Description & Overview

This document describes the system software, utilities, and reference application(s) and snippets which demonstrate SSDP capability along with how to enable SSDP in your WICED application. Using this documentation, the developer will learn how to use the WICED libraries for setting up and using SSDP to advertise the device's presence and services.

There are two basic operations for SSDP:

1) After initialization, register a callback for receiving notifications. This allows your application to receive and respond to SSDP Notifications.

2) After initialization, send an M-SEARCH message and wait for responses.

4 SSDP Daemon API

The OTA Extraction Library has functions to write data to the download staging area, check download status, verify, and extract downloaded OTA Images.

```
/** Callback for notifications
 *
 * Prototype for the user-defined function. Function is called when we recive a NOTIFY packet.
 *
 * @param nofity_info    : [in] ptr to info about the NOTIFY packet.
 * @param data           : [in] opaque app data
 *
 *      NOTES: The event_info structure is stored on the stack!
 *              It will not be around after the callback returns!
 *              Make a copy of info you want to keep the info!
 *
 */
typedef void (*wiced_ssdp_notify_callback_t)(wiced_ssdp_notify_info_t* nofity_info, void* data);

/** Start the SSDP daemon
 *  You must either
 *      set the start flags in the params (start_server, start_multicast)
 *  or
 *      call one of the start functions below
 *
 *
 * @param   params    : pointer to the parameter structure
 * @param   ssdp_info : pointer to store instance to use in subsequent calls

 * @return  WICED_SUCCESS
 *          WICED_ERROR
 *          WICED_BADARG
 */
extern wiced_result_t wiced_ssdp_init( wiced_ssdp_params_t *ssdp_params, void** ssdp_info );

/** Shut down the SSDP daemon
 *  This stops both the server and multicast messages
 *
 * @param   ssdp_info   : pointer info structure returned from wiced_ssdp_server_start()

 * @return  WICED_SUCCESS
 *          WICED_ERROR
 *          WICED_BADARG
 */
extern wiced_result_t wiced_ssdp_deinit( void* ssdp_info );

/** Register SSDP Notify callback
 *  Register a callback so the application can be notified when we receive a NOTIFY packet
 *
 *  NOTE: to disable the callback, use wiced_ssdp_notify_register_callback(ssdp_info, NULL,
NULL);
 *
 * @param   ssdp_info   : pointer info structure returned from wiced_ssdp_init()
 * @param   callback    : callback to register (call with NULL to de-register)
 * @param   data        : returned in callback (opaque to ssdp support)
 *
 * @return  WICED_SUCCESS
 *          WICED_ERROR
 *          WICED_BADARG
 */
extern wiced_result_t wiced_ssdp_notify_register_callback( void *ssdp_info,
wiced_ssdp_notify_callback_t callback, void *data );


/** Send an M-Search message and wait for responses
 *
 *  NOTE: this is a blocking call
 *
```

```
 * @param   ssdp_info           : pointer info structure returned from wiced_ssdp_server_start()
 * @param   params              : pointer to m-search send parameters
 *
 * @return  WICED_SUCCESS
 *          WICED_ERROR
 *          WICED_BADARG
 */
wiced_result_t wiced_ssdp_send_msearch_wait_for_results(void *ssdp_info,
wiced_ssdp_msearch_params_t *params);
```

5 More information on SSDP

The SSDP message format is defined in the document “UPnP Device Architecture 1.1”.
http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf

Connecting everything®

BROADCOM

BROADCOM CORPORATION
5300 California Avenue
Irvine, California, 92677

Phone : +1-949-926-5000
Fax: +1-949-926-5203
E-mail: info@broadcom.com
Web: www.broadcom.com