

# User Manual

## DA16200 SoftAP User Provisioning

UM-WI-028

### **Abstract**

*This document describes the use of Provisioning specification of the DA16200.*

---

---

## Contents

Abstract .....	1
1 Terms and Definitions.....	3
2 References .....	3
3 Introduction.....	4
4 How to Create New F/W with Provisioning Function.....	4
5 Specification for DA16200 Provisioning .....	4
6 Sample Code for Application Client .....	7
Revision History .....	11

## 1 Terms and Definitions

TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
AP	Access Point

## 2 References

- [1] DA16200, SDK Programmer Guide, Dialog Semiconductor
- [2] DA16200, Android Provisioning Tool User Manual, Dialog Semiconductor

### 3 Introduction

This document describes how to implement a Client program that performs provisioning functions for the DA16200.

The DA16200 supports Soft-AP mode to set up Wi-Fi configuration using by mobile-APP or other kind of peer on-air device.

Before starting the work, user should check if the Provisioning Feature is enabled, and if not, to enable it in the SDK. After this, the DA16200 provisioning function runs automatically when the DA16200 starts the Soft-AP mode.

The DA16200 SDK provides mobile-APP for provisioning.

For more information, see DA16200 Android Provisioning Tool User Manual [2].

### 4 How to Create New F/W with Provisioning Function

The DA16200 SDK provides provisioning function as a compile feature.

```
~/SDK/src/customer/config_generic_sdk.h
#undef __SUPPORT_PROVISION__ // Support Soft-AP provision
#ifdef __SUPPORT_PROVISION__
#undef __SUPPORT_USER_PROVISION__
#endif /* __SUPPORT_PROVISION__ */
```

User can change the above options as *#define* to create new F/W.

Also, provisioning function in Soft-AP mode is provided in DA16200 SDK as TLS/TCP by default for encryption. To work with TCP, you can change the following shape:

```
~/SDK/src/common/inc/sys_common_features.h
#if defined ( __SUPPORT_PROVISION__ )
#define __SUPPORT_PROVISION_TLS__ // Soft-AP Provision : TLS
#endif // __SUPPORT_PROVISION__
```

### 5 Specification for DA16200 Provisioning

The DA16200 user provisioning function runs as TLS or TCP server with defined port number. The default protocol is TLS.

Customer/developer can change the proper port number in the user provisioning function.

- TLS/TCP Server : Gateway Address ( Default : 10.0.0.1 )
- Port Number : 9900 ( Default )
- Data structure format :

```

/*****
 *
 * softap_provision.h
 *
 * Copyright (c) 2014-2018 FCI Inc.
 * All Rights Reserved.
 *
 * Author: Bright Jeong
 *
 *****/

#define MAX_SSID_LEN 128
#define MAX_PASSKEY_LEN 128

```

```

#define MAX_WEP_KEY_LEN      16

#define DEFAULT_AUTH_TYPE    4      // WPA-PSK AUTO Mode

/* Provisioning structure */
typedef struct _prov_config {
    /// Auto reboot flag - 0: No reboot, 1: Auto reboot
    int    auto_restart_flag;

    /// SSID of Wi-Fi AP
    char    ssid[MAX_SSID_LEN + 1];
    /// Passphrase key of Wi-Fi AP security
    char    psk[MAX_PASSKEY_LEN + 1];

    /// Wi-Fi Security Mode - 0: OPEN, 1:WEP, 2:WPA-PSK, 3:WPA2-PSK,
    4:WPA-AUTO
    int    auth_type;

    /* For WEP-Key */
    int    wep_key_index;
    char    wep_key[4][MAX_WEP_KEY_LEN + 1];

    /*
    * Country Code List:
    *
    * AD AE AF AI AL AM AR AS AT AU AW AZ BA BB BD BE BF BG BH BL
    * BM BN BO BR BS BT BY BZ CA CF CH CI CL CN CO CR CU CX CY CZ
    * DE DK DM DO DZ EC EE EG ES ET EU FI FM FR GA GB GD GE GF GH
    * GL GP GR GT GU GY HK HN HR HT HU ID IE IL IN IR IS IT JM JO
    * JP KE KH KN KP KR KW KY KZ LB LC LI LK LS LT LU LV MA MC MD
    * ME MF MH MK MN MO MP MQ MR MT MU MV MW MX MY NG NI NL NO NP
    * NZ OM PA PE PF PG PH PK PL PM PR PT PW PY QA RE RO RS RU RW
    * SA SE SG SI SK SN SR SV SY TC TD TG TH TN TR TT TW TZ UA UG
    * UK US UY UZ VA VC VE VI VN VU WF WS YE YT ZA ZW
    */
    char    country[4];

    /// 0:DHCP Client, 1:STATIC
    int    ip_addr_mode;

    /// SNTP 0:Disable, 1:Enable
    int    sntp_flag;
    /// SNTP Server URI
    char    sntp_server[32];
    /// SNTP Client renew time
    int    sntp_period;

    /// DPM 0:Disable, 1:Enable
    int    dpm_mode;

    /// DPM keep-alive periodic time
    int    dpm_ka;

    /// DPM user-wakeup time
    int    dpm_user_wu;

    /// DPM TIM wakeup period time
    int    dpm_tim_wu;
} user_prov_config_t;

```

- Auto-reboot flag after provisioning setup
 

```
int auto_restart_flag;
```

  - No reboot, 1: Auto reboot
  - Default value is “No Reboot” mode
- SSID and PSK values
  - char ssid[MAX\_SSID\_LEN + 1]; // Max 128 + 1 Bytes
  - char psk[MAX\_PASSKEY\_LEN + 1]; // Max 128 + 1 Bytes
- Wi-Fi authentication type
 

```
int auth_type;
```

  - 0: OPEN, 1:WEP, 2:WPA-PSK, 3:WPA2-PSK, 4:WPA-AUTO  
/\* For WEP-Key \*/
 

```
int wep_key_index;
```

```
char wep_key[4][MAX_WEP_KEY_LEN]; // Max 16 Bytes
```
- Country code
 

```
char country[4];
```

  - Select Capital characters for country code as below defined:  
AD AE AF AI AL AM AR AS AT AU AW AZ BA BB BD BE BF BG BH BL  
BM BN BO BR BS BT BY BZ CA CF CH CI CL CN CO CR CU CX CY CZ  
DE DK DM DO DZ EC EE EG ES ET EU FI FM FR GA GB GD GE GF GH  
GL GP GR GT GU GY HK HN HR HT HU ID IE IL IN IR IS IT JM JO  
JP KE KH KN KP KR KW KY KZ LB LC LI LK LS LT LU LV MA MC MD  
ME MF MH MK MN MO MP MQ MR MT MU MV MW MX MY NG NI NL NO NP  
NZ OM PA PE PF PG PH PK PL PM PR PT PW PY QA RE RO RS RU RW  
SA SE SG SI SK SN SR SV SY TC TD TG TH TN TR TT TW TZ UA UG  
UK US UY UZ VA VC VE VI VN VU WF WS YE YT ZA ZW
- Network IP address mode
 

```
int ip_addr_mode; // 0:DHCP Client, 1:STATIC
```

  - Default value is DHCP Client mode.
- SNTP time-server
 

```
int sntp_flag; // 0: Disable, 1:Enable
```

```
char sntp_server[32]; // SNTP Server IP address
```

```
int sntp_period; // SNTP update period time
```
- DPM configuration
 

```
int dpm_mode; // 0:Disable, 1:Enable
```

```
int dpm_ka; // DPM keep-alive periodic time
```

```
int dpm_user_wu; // DPM user-wakeup time
```

```
int dpm_tim_wu; // DPM TIM wakeup period time
```

## 6 Sample Code for Application Client

This example shows how user can create their own standard that contains additional information to the basic information provided.

User can add additional informaton in "user\_prov\_config\_t" structure in "`~/SDK/src/application/inc/user_provision.h`".

In the provisioning tool, user should define the TLS/TCP server address as "default gateway address" and the port number.

- TCP/TLS Server Address : Default gateway IP address (Soft-AP gateway address)
- Port Number : 9900 (Default value)

User provisioning tool should send TCP data as shown in below sample code.

- Sample code for user provisioning (ex. TCP client):  
`~/SDK/src/application/system_apps/user_softap_provision.c`

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <time.h>
#include <unistd.h>

#define PROVISION_PORT_NO      9900
#define TX_BUF_SIZE           1024

#define MAX_SSID_LEN          128
#define MAX_PASSKEY_LEN      128
#define MAX_WEP_KEY_LEN      16

#define DEFAULT_AUTH_TYPE     4      // WPA-PSK Auto Mode

/* Local provisioning structure */
typedef struct _prov_config {
    /* 0: No reboot, 1: Auto Reboot */
    int    auto_restart_flag;

    char   ssid[MAX_SSID_LEN + 1];
    char   psk[MAX_PASSKEY_LEN + 1];

    /* 0: OPEN, 1:WEP, 2:WPA-PSK, 3:WPA2-PSK, 4:WPA-AUTO */
    int    auth_type;

    /* For WEP-Key */
    int    wep_key_index;
    char   wep_key[4][MAX_WEP_KEY_LEN + 1];

    /*
     * Country Code
     *
     * AD AE AF AI AL AM AR AS AT AU AW AZ BA BB BD BE BF BG BH BL
     * BM BN BO BR BS BT BY BZ CA CF CH CI CL CN CO CR CU CX CY CZ
     * DE DK DM DO DZ EC EE EG ES ET EU FI FM FR GA GB GD GE GF GH
     * GL GP GR GT GU GY HK HN HR HT HU ID IE IL IN IR IS IT JM JO
     * JP KE KH KN KP KR KW KY KZ LB LC LI LK LS LT LU LV MA MC MD
     */
};
```

```

* ME MF MH MK MN MO MP MQ MR MT MU MV MW MX MY NG NI NL NO NP
* NZ OM PA PE PF PG PH PK PL PM PR PT PW PY QA RE RO RS RU RW
* SA SE SG SI SK SN SR SV SY TC TD TG TH TN TR TT TW TZ UA UG
* UK US UY UZ VA VC VE VI VN VU WF WS YE YT ZA ZW
*/
char    country[4];

int     ip_addr_mode;           // 0:DHCP Client, 1:STATIC

int     sntp_flag;
char    sntp_server[32];
int     sntp_period;

int     dpm_mode;
int     dpm_ka;
int     dpm_user_wu;
int     dpm_tim_wu;
} user_prov_config_t;

#define TEST_SSID                "N_A1004_WPAx-PSK"
#define TEST_PSK                 "N12345678"
#define TEST_AUTH_TYPE          4      // WPA-PSK

#define TEST_COUNTRY            "KR"
#define TEST_IP_ADDR_TYPE      0      // DHCP Client

void make_prov_config_data(prov_config_t *config)
{
    memset(config, 0, sizeof(prov_config_t));

    /* Config Wi-Fi information : Simple */
    sprintf(config->ssid, "%s", TEST_SSID);
    config->auth_type = DEFAULT_AUTH_TYPE;
    if (config->auth_type > 0)
        sprintf(config->psk, "%s", TEST_PSK);

    /*
    * Additional information
    * - Default country code is "US"
    * - Default ip address mode is "DHCP Client"
    */
    sprintf(config->country, "%s", TEST_COUNTRY);
    config->ip_addr_mode = TEST_IP_ADDR_TYPE;      // DHCP Client

    config->auto_restart_flag = 1;
}

void print_config(prov_config_t *config)
{
    printf("\n--- Test configuration :\n");
    printf("\t- ssid\t\t: %s\n", config->ssid);
    printf("\t- psk\t\t: %s\n", config->psk);
    printf("\t- auth_type\t: %d (0:OPEN, 1:WPA-PSK, 2:WEP, 3:WPA2-PSK, 4:WPA-
AUTO)\n", config->auth_type);

    if (config->auth_type == 2) {
        printf("\t- wep_key_index\t: %d\n", config->wep_key_index);
        printf("\t\twep_key 0\t\t: %s\n", config->wep_key[0]);
    }
}

```



```

        printf("\t\twep_key_0\t\t: %s\n", config->wep_key[1]);
        printf("\t\twep_key_0\t\t: %s\n", config->wep_key[2]);
        printf("\t\twep_key_0\t\t: %s\n", config->wep_key[3]);
    }

    printf("\n");
    printf("\t- Country Code\t: %s\n", config->country);

    printf("\t- SNTP mode\t: %d\n", config->sntp_flag);
    if (config->sntp_flag == 1) {
        printf("\t\tSNTP Server\t\t: %s\n", config->sntp_server);
        printf("\t\tSNTP period\t\t: %d\n", config->sntp_period);
        printf("\n");
    }

    printf("\t- DPM mode\t: %d\n", config->dpm_mode);
    if (config->dpm_mode == 1) {
        printf("\t\tDPM Keep-Alive Time\t: %d\n", config->dpm_ka);
        printf("\t\tDPM User Wakeup Time\t: %d\n", config->dpm_user_wu);
        printf("\t\tDPM TIM Wakeup Time\t: %d\n", config->dpm_tim_wu);
    }
}

int main(int argc , char *argv[])
{
    int    sock;
    struct sockaddr_in server;
    char   peer_ip[16], peer_port[8];
    char   *input_str, *semi_col;
    prov_config_t *tx_msg;
    int    i = 0;
    int    length, result;
    int    port_no;

    if (argc < 2) {
        printf("Usage : tcpc_provision peer_ip:port\n");
        return 0;
    }

    memset(peer_ip, 0, 16);
    memset(peer_port, 0, 8);

    input_str = argv[1];
    length = strlen(input_str);
    while (length-- > 0 && input_str[i] != ':') {
        peer_ip[i] = input_str[i];
        i++;
    }

    /* Get port number */
    if (length > 0) {
        semi_col = strstr(argv[1], ":");
        strcpy(peer_port, semi_col+1);
        port_no = atoi(peer_port);
    } else {
        port_no = PROVISION_PORT_NO;
    }
}

```

```
/* Create socket */
sock = socket(AF_INET, SOCK_STREAM, 0);
if (sock == -1) {
    printf("Could not create socket");
    goto test_exit;
}
printf("Socket created ...");

server.sin_addr.s_addr = inet_addr(peer_ip);
server.sin_family = AF_INET;
server.sin_port = htons(port_no);

// Connect to remote server
if (connect(sock, (struct sockaddr *)&server, sizeof(server)) < 0) {
    printf("connect failed. Error");

    goto test_exit;
}
printf("Connected ...\\n");

tx_msg = (prov_config_t *)malloc(sizeof(prov_config_t));

make_prov_config_data(tx_msg);

/* Send some data */
if (result = send(sock, tx_msg, sizeof(prov_config_t), 0) < 0) {
    printf("!!! Send fail (result=0x%x)\\n", result);
    goto test_exit;
}

print_config(tx_msg);

/* For safe tcp transmission */
sleep(1);

test_exit :
close(sock);
free(tx_msg);

return 0;
}
```

## Revision History

Revision	Date	Description
1.0	27-Aug-2020	Initial version.

### Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

### Disclaimer

Unless otherwise agreed in writing, the Dialog Semiconductor products (and any associated software) referred to in this document are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of a Dialog Semiconductor product (or associated software) can reasonably be expected to result in personal injury, death or severe property or environmental damage. Dialog Semiconductor and its suppliers accept no liability for inclusion and/or use of Dialog Semiconductor products (and any associated software) in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Information in this document is believed to be accurate and reliable. However, Dialog Semiconductor does not give any representations or warranties, express or implied, as to the accuracy or completeness of such information. Dialog Semiconductor furthermore takes no responsibility whatsoever for the content in this document if provided by any information source outside of Dialog Semiconductor.

Dialog Semiconductor reserves the right to change without notice the information published in this document, including, without limitation, the specification and the design of the related semiconductor products, software and applications. Notwithstanding the foregoing, for any automotive grade version of the device, Dialog Semiconductor reserves the right to change the information published in this document, including, without limitation, the specification and the design of the related semiconductor products, software and applications, in accordance with its standard automotive change notification process.

Applications, software, and semiconductor products described in this document are for illustrative purposes only. Dialog Semiconductor makes no representation or warranty that such applications, software and semiconductor products will be suitable for the specified use without further testing or modification. Unless otherwise agreed in writing, such testing or modification is the sole responsibility of the customer and Dialog Semiconductor excludes all liability in this respect.

Nothing in this document may be construed as a license for customer to use the Dialog Semiconductor products, software and applications referred to in this document. Such license must be separately sought by customer with Dialog Semiconductor.

All use of Dialog Semiconductor products, software and applications referred to in this document is subject to Dialog Semiconductor's [Standard Terms and Conditions of Sale](http://www.dialog-semiconductor.com), available on the company website ([www.dialog-semiconductor.com](http://www.dialog-semiconductor.com)) unless otherwise stated.

Dialog, Dialog Semiconductor and the Dialog logo are trademarks of Dialog Semiconductor Plc or its subsidiaries. All other product or service names and marks are the property of their respective owners.

© 2020 Dialog Semiconductor. All rights reserved.

### RoHS Compliance

Dialog Semiconductor's suppliers certify that its products are in compliance with the requirements of Directive 2011/65/EU of the European Parliament on the restriction of the use of certain hazardous substances in electrical and electronic equipment. RoHS certificates from our suppliers are available on request.

## Contacting Dialog Semiconductor

#### United Kingdom (Headquarters)

*Dialog Semiconductor (UK) LTD*

Phone: +44 1793 757700

#### Germany

*Dialog Semiconductor GmbH*

Phone: +49 7021 805-0

#### The Netherlands

*Dialog Semiconductor B.V.*

Phone: +31 73 640 8822

#### Email:

[enquiry@diasemi.com](mailto:enquiry@diasemi.com)

#### North America

*Dialog Semiconductor Inc.*

Phone: +1 408 845 8500

#### Japan

*Dialog Semiconductor K. K.*

Phone: +81 3 5769 5100

#### Taiwan

*Dialog Semiconductor Taiwan*

Phone: +886 281 786 222

#### Web site:

[www.dialog-semiconductor.com](http://www.dialog-semiconductor.com)

#### Hong Kong

*Dialog Semiconductor Hong Kong*

Phone: +852 2607 4271

#### Korea

*Dialog Semiconductor Korea*

Phone: +82 2 3469 8200

#### China (Shenzhen)

*Dialog Semiconductor China*

Phone: +86 755 2981 3669

#### China (Shanghai)

*Dialog Semiconductor China*

Phone: +86 21 5424 9058