



DHCP Client for IPv6 User Guide

Version 2.10

For use with Dynamic Host Control Protocol (DHCP)
Client for IPv6 module versions 1.12 and above

Table of Contents

1. System Overview	3
1.1. Introduction	4
1.2. Feature Check	5
1.3. Packages and Documents	6
1.4. Change History	7
2. Source File List	8
3. Configuration Options	9
4. Application Programming Interface	10
4.1. dhcp_v6_add_custom_option_parser	11
4.2. dhcp_v6_del_custom_option_parser	12
4.3. t_dhcp_v6_custom_option_parser_ntf_fn	13
5. Integration	14
5.1. OS Abstraction Layer	14
5.2. Utilities	14
5.3. PSP Porting	15
6. Version	16

1. System Overview

This chapter contains the fundamental information for this module.

The component sections are as follows:

- [Introduction](#) - describes the main elements of the module. This section includes a diagram showing the position of this module within HCC's TCP/IP stack.
- [Feature Check](#) - summarizes the main features of the module as bullet points.
- [Packages and Documents](#) - the *Packages* section lists the packages that you need in order to use this module. The *Documents* section lists the relevant user guides.
- [Change History](#) - lists the earlier versions of this manual, giving the software version that each manual describes.

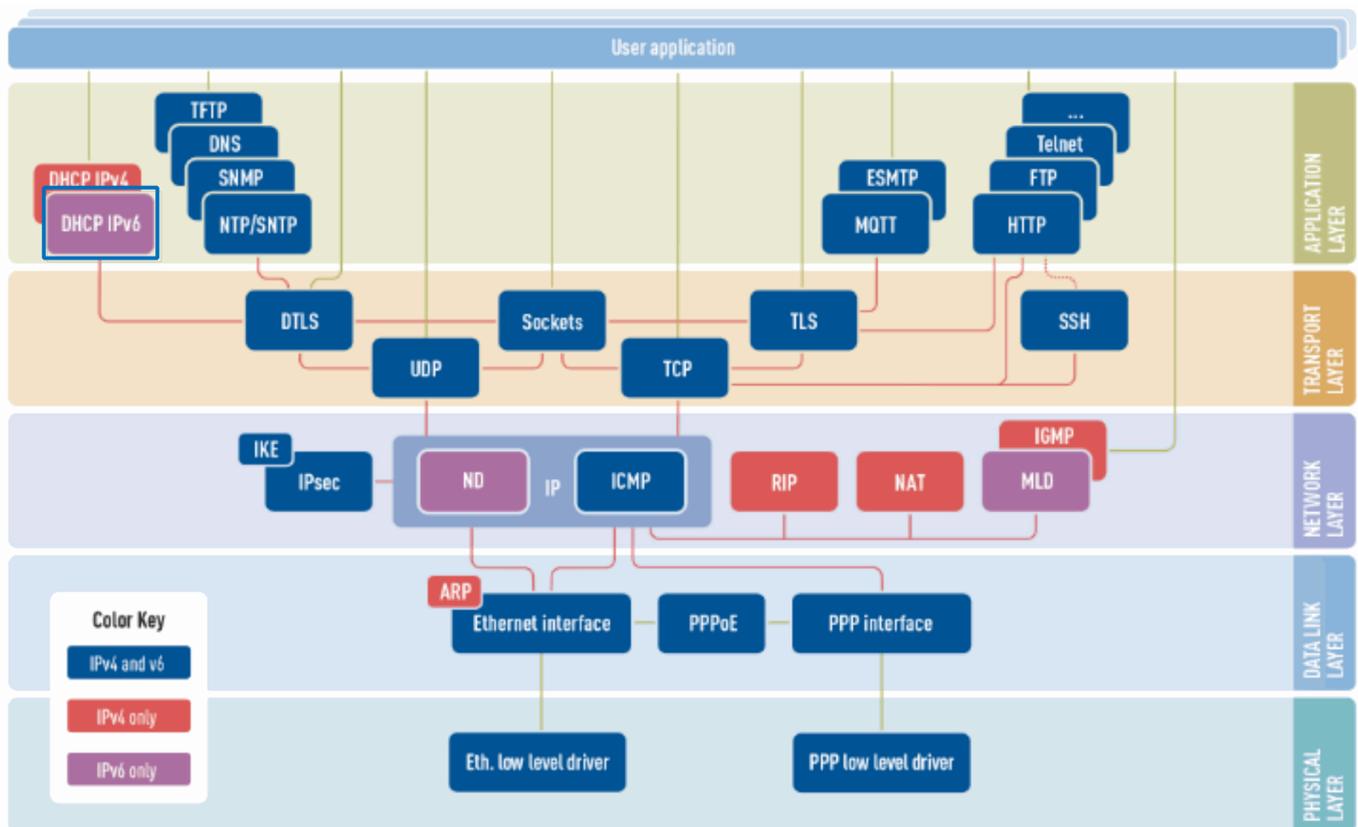
All rights reserved. This document and the associated software are the sole property of HCC Embedded. Reproduction or duplication by any means of any portion of this document without the prior written consent of HCC Embedded is expressly forbidden.

HCC Embedded reserves the right to make changes to this document and to the related software at any time and without notice. The information in this document has been carefully checked for its accuracy; however, HCC Embedded makes no warranty relating to the correctness of this document.

1.1. Introduction

This guide is for those who want to implement a Dynamic Host Control Protocol (DHCP) client module as part of their TCP/IP stack. The DHCP Client for IPv6 module is used by a client (a computer or other device) to get an IP address automatically from a remote DHCP server. This client module supports IPv6 addresses. It can be used in the HCC dual TCP/IP stack alongside the DHCP client for IPv4.

The DHCP Client for IPv6 module is part of the HCC MISRA-compliant TCP/IP stack, as shown below, and is designed specifically for use with it. (In this diagram green lines show interfaces available to users of the stack, red lines show interfaces internal to the TCP/IP system.)



When a DHCP-configured client connects to a network, it sends a broadcast query to a DHCP server, requesting necessary information. If the request is valid, the server assigns the device an IP address, a lease (the length of time the allocation is valid), and other IP configuration parameters, such as the subnet mask and the default gateway.

The query is typically sent straight after booting and must complete before the client can initiate IP-based communication with other hosts. Upon disconnection, the IP address is returned to the pool for use by another computer. In this way many computers may use the same IP address in a short time.

1.2. Feature Check

The main features of the module are the following:

- Conforms to the HCC Advanced Embedded Framework.
- Compliant with the HCC MISRA-compliant TCP/IP stack.
- Supports IPv6 addresses.
- Designed for integration with both RTOS and non-RTOS based systems.
- Compliant with [RFC 3315](#).

1.3. Packages and Documents

Packages

The table below lists the packages that you need in order to use this module:

Package	Description
<code>hcc_base_doc</code>	This contains the two guides that will help you get started.
<code>ip_app_dhcp_v6</code>	The DHCP client for IPv6 package described in this manual.
<code>ip_base_v6</code>	The TCP/IP IPv6 base package.
<code>mip_udp</code>	The UDP package.

Documents

For an overview of the HCC TCP/IP stack software, see [Product Information](#) on the main HCC website.

Readers should note the points in the [HCC Documentation Guidelines](#) on the HCC documentation website.

HCC Firmware Quick Start Guide

This document describes how to install packages provided by HCC in the target development environment. Also follow the *Quick Start Guide* when HCC provides package updates.

HCC Source Tree Guide

This document describes the HCC source tree. It gives an overview of the system to make clear the logic behind its organization.

HCC TCP/IP Dual Stack System User Guide

This is the core document that describes the complete TCP/IP stack. It covers both IPv4 and IPv6 systems.

HCC DHCP Client for IPv6 User Guide

This is this document.

HCC DHCP Client for IPv4 User Guide

This document describes the similar DHCP client that supports IPv4 addresses.

1.4. Change History

This section describes past changes to this manual.

- To download this manual or a PDF describing an [earlier software version, see TCP/IP PDFs](#).
- For the history of changes made to the package code itself, see [History: ip_app_dhcp_v6](#).

The current version of this manual is 2.10. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
2.10	2020-05-28	1.12	Added DHCPV6_CUSTOM_OPTION_PARSER_COUNT configuration option. Added API section with three functions.
2.00	2020-02-20	1.10	New document template.
1.40	2018-02-06	1.09	psp_getrand() replaced psp_tick() in <i>PSP Porting</i> .
1.30	2017-06-20	1.06	New <i>Change History</i> format.
1.20	2017-03-28	1.06	Updated network diagram.
1.10	2017-01-18	1.06	Updated network diagram.
1.00	2016-08-05	1.04	First online version.

2. Source File List

The following sections describe all the source code files included in the system. These files follow the HCC Embedded standard source tree system, described in the [HCC Source Tree Guide](#). All references to file pathnames refer to locations within this standard source tree, not within the package you initially receive.

Note: Do not modify any files except the configuration file.

API Header File

The file `src/api/api_ip_app_dhcp_v6.h` is the only file that should be included by an application using this module. For details of the API functions and callback, see [Application Programming Interface](#).

Configuration File

The file `src/config/config_ip_app_dhcp_v6.h` contains all the configurable parameters of the system. Configure these as required. For details of these options, see [Configuration Options](#).

DHCP Client System

These files are in the directory `src/ip/apps/dhcp_v6`. **These files should only be modified by HCC.**

File	Description
<code>dhcp_v6.c</code>	DHCP client for IPv6 source code.
<code>dhcp_v6.h</code>	Header file.

Version File

The file `src/version/ver_ip_app_dhcp_v6.h` contains the version number of this module. This version number is checked by all modules that use this module to ensure system consistency over upgrades.

3. Configuration Options

Set the system configuration options in the file **src/config/config_ip_app_dhcp_v6.h**. This section lists the available configuration options and their default values.

DHCPV6_TASK_STACK_SIZE

The DHCP for IPv6 task stack size. The default is 1024.

DHCPV6_TIMER_PERIOD

The timer period in ms. The default is 1000.

DHCPV6_NUM_SERVERS

The maximum number of DHCP IPv6 servers that are accepted on an single interface. The default is 1.

DHCPV6_NUM_ADDRS

The maximum number of IPv6 addresses that can be assigned to the target. This must be not less than DHCPV6_NUM_SERVERS. The default is 1.

DHCPV6_CUSTOM_OPTION_PARSER_COUNT

The maximum number of custom option parsers that can be added to the DHCPv6 client. The default value is 1. Set this to 0 to disable this element.

4. Application Programming Interface

This section describes the Application Programming Interface (API) functions and the callback function.

4.1. dhcp_v6_add_custom_option_parser

Use this function to add a custom option code parser callback function. The option code is used during DHCPv6 negotiation.

The callback is invoked when it is received in the OFFER or ACK message from the DHCP server.

Note: This is only available if DHCPV6_CUSTOM_OPTION_PARSER_COUNT is set.

Format

```
t_ip_ret dhcp_v6_add_custom_option_parser (
    const uint16_t          option_code,
    const t_dhcp_custom_option_parser_ntf_fn  dhcp_custom_option_parser_ntf_fn )
```

Arguments

Arguments	Description	Type
option_code	The option code to search for.	uint16_t
dhcp_custom_option_parser_ntf_fn	The function to call after the custom option code is received.	t_dhcp_custom_option_parser_ntf_fn

Return Values

Return value	Description
IP_SUCCESS	Successful execution.
IP_ERR_INVALID_PARAM	The option code is already registered.
IP_ERR_NO_MORE_ENTRY	There is no free entry available to register the code.

4.2. dhcp_v6_del_custom_option_parser

Use this function to delete a custom option code parser callback function.

Note: This is only available if DHCPV6_CUSTOM_OPTION_PARSER_COUNT is set.

Format

```
t_ip_ret dhcp_v6_del_custom_option_parser ( const uint16_t option_code )
```

Arguments

Arguments	Description	Type
option_code	The option code to search for.	uint16_t

Return Values

Return value	Description
IP_SUCCESS	Successful execution.
IP_ERR_INVALID_PARAM	The option code parser given has not been registered.

4.3. t_dhcp_v6_custom_option_parser_ntf_fn

This is the function type used for the callback function invoked after a custom option code is received from the DHCPv6 server in the OFFER or ACK message.

Format

```
typedef void ( * t_dhcp_v6_custom_option_parser_ntf_fn )(
    const t_ip_ifc_hdl    ifc_hdl,
    const uint16_t        option_code,
    const uint16_t        data_length,
    const uint8_t         data[] )
```

Arguments

Arguments	Description	Type
ifc_hdl	The handle of the IP interface on which the option was received.	t_ip_ifc_hdl
option_code	The option code of the received option.	uint16_t
data_length	The length of the received option data.	uint16_t
data[]	The received option data.	uint8_t

5. Integration

This section describes all aspects of the DHCP client for IPv6 module that require integration with your target project. This includes porting and configuration of external resources.

5.1. OS Abstraction Layer

All HCC modules use the OS Abstraction Layer (OAL) that allows the module to run seamlessly with a wide variety of RTOSes, or without an RTOS.

This module uses the following OAL components:

OAL Resource	Number Required
Tasks	1
Mutexes	1
Events	0

The DHCP task is started automatically by the IP stack if DHCP is enabled (that is, if the `IP_DHCP_ENABLE` configuration option in the IP base package is set).

The DHCP task function is named **dhcp_task()**.

5.2. Utilities

The DHCP code creates and uses a single timer in the **hcc_timer** module.

The **hcc_timer** module is included in your system when you install the base TCP/IP modules.

5.3. PSP Porting

The Platform Support Package (PSP) is designed to hold all platform-specific functionality, either because it relies on specific features of a target system, or because this provides the most efficient or flexible solution for the developer. For full details of its functions and macros, see the *HCC Base Platform Support Package User Guide*.

The module makes use of the following standard PSP functions:

Function	Package	Element	Description
psp_memcmp()	psp_base	psp_string	Compares two blocks of memory.
psp_memcpy()	psp_base	psp_string	Copies a block of memory. The result is a binary copy of the data.
psp_memset()	psp_base	psp_string	Sets the specified area of memory to the defined value.
psp_getrand()	psp_base	psp_rand	Gets random identifier.

The module makes use of the following standard PSP macros:

Macro	Package	Element	Description
PSP_RD_BE16	psp_base	psp_endianness	Reads a 16 bit value stored as big-endian from a memory location.
PSP_RD_BE24	psp_base	psp_endianness	Reads a 24 bit value stored as big-endian from a memory location.
PSP_RD_BE32	psp_base	psp_endianness	Reads a 32 bit value stored as big-endian from a memory location.
PSP_WR_BE16	psp_base	psp_endianness	Writes a 16 bit value to be stored as big-endian to a memory location.
PSP_WR_BE24	psp_base	psp_endianness	Writes a 24 bit value to be stored as big-endian to a memory location.
PSP_WR_BE32	psp_base	psp_endianness	Writes a 32 bit value to be stored as big-endian to a memory location.

6. Version

Version 2.10

For use with Dynamic Host Control Protocol (DHCP) Client for IPv6 module versions 1.12 and above