Renesas Synergy™ Platform

NetX™ DHCPv6 Server Module Guide

Introduction

This module guide will enable you to effectively use a module in your own design. Upon completion of this guide, you will be able to add this module to your own design, configure it correctly for the target application, and write code using the included application project code as a reference and an efficient starting point. References to more detailed API descriptions and suggestions of other application projects that illustrate more advanced uses of the module are included in this document and should be valuable resources for creating more complex designs.

In IPv6 networks, DHCPv6 replaces DHCP (which is limited to IPv4) for dynamic global IPv6 address assignment from a DHCPv6 Server, and offers most of the same features, as well as many enhancements, and explains how the NetX Duo™ DHCPv6 Server API is used for DHCPv6 Client IPv6 address requests.

This Module Guide provides an overview of the key elements related to the NetX Duo DHCPv6 Server implementation, in particular the addition and configuration of the NetX Duo DHCPv6 Server module to a Renesas Synergy™ Platform project. For operational details, consult the *NetX™ Duo Dynamic Host Protocol (DHCPv6) Server User Guide* for the Renesas Synergy™ Platform included in the *X-Ware™ Component Documents for Renesas Synergy™* zip file available at www.renesas.com/synergy/ssp when you log in.

Contents

1. NetX Duo DHCPv6 Server Module Features

- The NetX Duo DHCPv6 Server is compliant with RFC 3315, RFC 3646, and related RFCs.
- Provides high-level APIs for:
  - Creating and deleting a DHCPv6 Server instance
  - Starting and stopping a DHCPv6 Server thread task
  - Creating a pool of IPv6 addresses for lease
  - Maintaining a table of DHCPv6 leases assignable to requesting clients

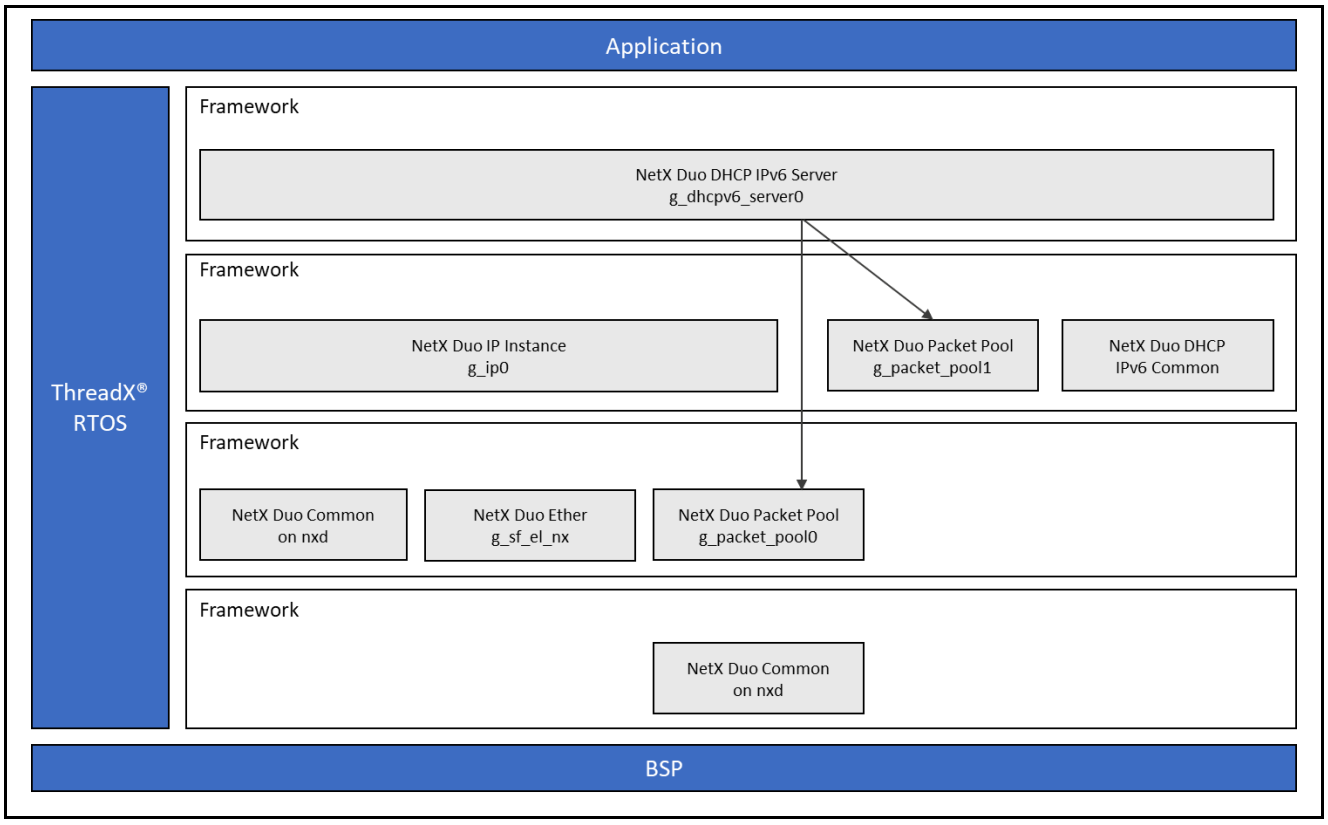

Figure 1. NetX Duo DHCPv6 Server Module Block

2. NetX Duo DHCPv6 Server Module APIs Overview

The NetX Duo DHCPv6 Server module defines APIs for creating, deleting, adding, and getting server information. The following tables summarizes the available APIs and includes an example of each, along with a brief description. A table of status return values follows the API summary table.

Table 1. NetX Duo DHCPv6 Server Module API Summary

Function Name	Example API Call and Description
nx_dhcpv6_server_create	`nx_dhcpv6_server_create(&g_dhcpv6_server0, &server_ip_address, "DHCPv6 Server", &g_packet_pool0, stack_pointer, NX_DHCPV6_SERVER_THREAD_STACK_SIZE, address_declined_handler, option_request_handler);` Create a DHCPv6 Server instance.
nx_dhcpv6_server_delete	`nx_dhcpv6_client_delete(&g_dhcpv6_server0);` Delete a DHCPv6 Server instance and release resources (unbind port, delete socket, timers and thread).

Function Name	Example API Call and Description
nx_dhcpv6_create_ip_address_lease	`nx_dhcpv6_create_ip_address_range(&g_dhcpv6_server0, &start_ipv6_address, &end_ipv6_address, &addresses_added);` Create the Server's IPv6 address lease pool.
nx_dhcpv6_reserve_ip_address_range	`nx_dhcpv6_reserve_ip_address_range(&g_dhcpv6_server0, &start_ipv6_address, &end_ipv6_address, &addresses_reserved);` Reserve the specified range of IPv6 addresses not to be leased out to a requesting Client.
nx_dhcpv6_add_ip_address_lease	`nx_dhcpv6_add_ip_address_lease(&g_dhcpv6_server0, table_index, &lease_IP_address, T1, T2, valid_lifetime, preferred_lifetime);` Copy an IPv6 lease record into the specified index into the Server table. Intended for use in non-volatile storage of IPv6 lease data.
nx_dhcpv6_add_client_record	`nx_dhcpv6_add_client_record(&g_dhcpv6_server0, table_index, message_xid, &client_address, client_state, IP_lease_time_accrued, valid_lifetime, duid_type, duid_hardware, physical_address_msw, physical_address_lsw, duid_time, duid_vendor_number, duid_vendor_private, duid_private_length);` Copy a Client record into the specified index into the Server table. Intended for use in non-volatile storage of Client IPv6 address data.
nx_dhcpv6_create_dns_address	`nx_dhcpv6_create_dns_address(&g_dhcpv6_server0, &dns_ipv6_address);` Set the DNS server address to include in network parameters sent to clients.
nx_dhcpv6_retrieve_client_record	`nx_dhcpv6_retrieve_client_record(&g_dhcpv6_server0, table_index, message_xid, &client_address, client_state, IP_lease_time_accrued, valid_lifetime, duid_type, duid_hardware, physical_address_msw, physical_address_lsw, duid_time, duid_vendor_number, duid_vendor_private, duid_private_length);` Retrieve items from the Client specified by the index into the Server table. Intended for use in non-volatile storage of Client IPv6 address data.
nx_dhcpv6_retrieve_ip_address_lease	`nx_dhcpv6_retrieve_ip_address_lease(&g_dhcpv6_server0, table_index, &lease_IP_address, T1, T2, valid_lifetime, preferred_lifetime);` Retrieve items from the IPv6 lease specified by the index into the Server table. Intended for use in non-volatile storage of IPv6 lease data.

Function Name	Example API Call and Description
nx_dhcpv6_server_interface_set	`nx_dhcpv6_server_interface_set(&g_dhcpv6_server0, 0, 1);` Set the interface the DHCPv6 Server will run on, and the global address the DHCPv6 Server will use in messages to Clients. By default, the DHCPv6 Server runs on the primary interface (index 0).
nx_dhcpv6_set_server_duid	`nx_dhcpv6_set_server_duid(&g_dhcpv6_server0, NX_DHCPV6_SERVER_DUID_TYPE, NX_DHCPV6_SERVER_HW_TYPE, physical_address_msw, physical_address_lsw, duid_time);` Create the Server DUID which is a required part of the DHCPv6 header and uniquely identifies the DHCPv6 Server.
nx_dhcpv6_server_start	`nx_dhcpv6_server_start(&g_dhcpv6_server0);` Start the DHCPv6 thread task.
nx_dhcpv6_server_suspend	`nx_dhcpv6_server_suspend(&g_dhcpv6_server0);` Suspend the DHCPv6 server task.
nx_dhcpv6_server_resume	`nx_dhcpv6_server_resume(&g_dhcpv6_server0);` Resume the DHCPv6 server task.

Note: For details on operation and definitions for the function data structures, typedefs, defines, API data, API structures, and function variables, review the *SSP User's Manual* API References for the associated module.

3. NetX Duo DHCPv6 Server Module Operational Overview

The NetX Duo DHCPv6 Server module creates a NetX Duo IP instance for the server and a UDP socket bound to the well-known DHCPv6 Server port 547 to listen for client requests. Before starting DHCPv6, the server needs a global IPv6 address by setting the **IPv6 Global Address** property in the **IP NetX Duo Instance** property box. (Note that this is a 128-bit long address, compared to the 32-bit long IPv4 address.)

The DHCPv6 Server should wait for the IPv4 address to be validated using the `nx_ip_status_check` service, even if the server does not use this IP address for DHCPv6 messages. The driver needs to be initialized with information from the IP layer and the link needs to be enabled, all of which happens with IPv4 address-registration.

Before an application can start the DHCPv6 Server, it must create a pool of assignable IPv6 addresses using the `nx_dhcpv6_create_ip_address_range` service. The application must also create a server **DUID** (DHCP Unique Identifier, usually based on the MAC address, and is required in all DHCPv6 Server messages using the `nx_dhcp_set_server_duid` service. Optionally, it can set the network DNS Server to include in the DHCPv6 option data to clients using the `nx_dhcpv6_create_dns_server` service; now the server can be started with the `nx_dhcpv6_server_start` service.

Note: All properties referenced in text are found in the **NetX Duo DHCP IPv6 Server** properties box, unless otherwise noted.

The server maintains a table of IPv6 addresses, updates their status, and indicates whether any addresses are leased out. For leased out addresses, the table indicates the client to which the address is leased. The size of this table is set by the **Maximum Size of the Server's IP lease table** property, and should be equal to or greater than the number of IPv6 addresses in the IPv6 address pool. The server maintains another table for client records; the size of this table is set by the **Size of Server's Client record table** property portion of the **DHCP Server** properties box and should be at least the size of the IPv6 lease table. If the server receives a **Client Release** or **Decline**, the server updates the IPv6 lease table and client record table, accordingly.

The DHCPv6 Server creates two timers. The first timer keeps track of the time remaining on IPv6 addresses leased to clients. The interval at which the server checks client leases is set by the **Client lease time**

expiration check interval property, which defaults to 60 seconds. If the server issues leases with extremely short lease-expirations, that value should be reduced to approximately 10 or 20 percent of the timeout value. If a lease timeout expires, the lease is returned to the pool of IPv6 addresses and the client record is deleted. The second timer is used to monitor client-session inactivity; the default timeout for session inactivity timeout is 20 seconds. The interval in which the server checks its client records for an expired session inactivity timeout is the **Interval for active session time** update property in the **NetX Duo DHCP IPv6 Common** properties box, with the interval having a 3-second default time.

3.1 NetX Duo DHCPv6 Server Module Important Operational Notes and Limitations

3.1.1 NetX Duo DHCPv6 Server Module Operational Notes

- **IPv6** and **ICMPv6** must be enabled in NetX Duo. Verify that the **NetX Duo IPv6 Support** property is **enabled** in the **NetX Duo Source** properties box; this setting automatically enables ICMPv6.

Note: If the underlying hardware supports ICMPv6 checksum computation, the following values can be left disabled.

- By default, the ICMPv6 checksum is disabled. To enable the checksum (assuming the hardware does not compute ICMPv6 checksums), locate the **Checksum computation support on transmitted ICMPv6 packets** property in the **NetX Duo Source** properties box, and set it to **enabled**. The **Checksum computation support on received ICMPv6 packets** property should also be enabled.
- Duplicate **Address Detection (DAD)** is recommended to verify the uniqueness of the server's global IPv6 address. This protocol is similar to sending gratuitous ARP probes in IPv4 to determine the uniqueness of an IPv4 address, but it is only applicable to IPv6 addresses. To enable **DAD** (which is disabled by default), set the **Duplicate Address Detection support** property to enabled in the **NetX Duo Source** properties box. The number of solicitation packets sent out for DAD is set by the **Neighbor Solicitation message count before interface address marked valid** property, which by default is 3 (and sent about a second apart). With this configuration, the application thread should wait about 4 seconds to let the DAD protocol complete.
- The following DHCPv6 parameters are supplied in DHCPv6 Server responses to the client:
  - **T1 time**: when the client should begin renewing its IPv6 address lease, is set by the **Server interval for first client IP address renewal attempt** property.
  - **Preferred time**: when the client IPv6 address is deprecated, is set by the **Time interval after which the client IP is deprecated** property. This should be double the T1 time, as per RFC 3315 recommendations.
  - **T2 time**: when the client should begin rebinding an IPv6 address if renewing efforts failed, is set by the **Server interval for the second client IP address renewal attempt** property.
  - **Valid lifetime**: when the client IPv6 address is obsolete and should no longer be used by the client. This is set by the **Time interval after which leased IP is invalid** property, and should be double the preferred time, as per RFC 3315 recommendations.
    There is no upper limit on the IPv6 lease time (valid lifetime), but the relative interval of these four time parameters must permit the logical order of renew and, if necessary, rebind state of the DHCPv6 protocol.
- The address-declined handler callback for handling a Client Decline message is not implemented on the current NetX Duo DHCPv6 Server. This callback is suggested by the RFC 3315 DHCPv6 specification for the server to notify the application of a declined address event.

3.1.2 NetX Duo DHCPv6 Server Module Limitations

The following DHCPv6 options are not supported in the NetX Duo DHCPv6 Server:

- **Rapid Commit** option: optimizes the DHCPv6 address request process to just the Solicit and Reply message-exchange
- **Reconfigure** option: allows the server to initiate changes to the client's IP address status
- **Unicast** option: all client messages must be sent to **All_DHCP_Relay_Agents_and_Servers** multicast address rather than to the DHCPv6 Server directly.
- Identity Association for the Temporary Addresses(IA_TA) option: a temporary IP address granted to a client
- Multiple IA (IPv6 addresses) option: per client request
- Relay host between DHCPv6 Client and Server: client and server must be on the same network

- NetX Duo DHCPv6 Server: directly supports only the DNS Server option request
- **Prefix Delegation** option: is not supported.
- **Option** request callback: intended for the application and determines which DHCPv6 options to support and which information to supply to the DHCPv6 Server in response to the client. However, the processing of this information into the DHCPv6 Server response is not implemented. This callback has no effect on DHCPv6 Server messages.

See the *SSP Release Notes* for any additional operational limitations for this module.

4. Including the NetX Duo DHCPv6 Server Module in an Application

Note: It is assumed you are familiar with creating a project, adding threads, adding a stack to a thread, and configuring a block within the stack. If you are unfamiliar with any of these items, refer to the first few chapters of the *SSP User's Manual* to learn how to manage each of these important steps in creating SSP-based applications.

To add the NetX Duo DHCPv6 Server module to an application, simply add it to a thread using the stacks selection sequence given in the following table. (The default name for the NetX Duo DHCPv6 Server module is `g_dhcpv6_server0`. This name can be changed in the associated **Properties** window.)

Table 2. NetX Duo DHCPv6 Server Module Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
`g_dhcpv6_server0 NetX Duo DHCPv6 Server`	Threads	New Stack > X-Ware > NetX Duo > Protocols > NetX Duo DHCP IPv6 Server

When the NetX Duo DHCPv6 Server is added to the thread stack as shown in the following figure, the configurator automatically adds the needed lower-level drivers. Any drivers that need additional configuration information are box text highlighted in Red. Modules with a Gray band are individual, standalone modules. Modules with a Blue band are shared or common and need only be added once, since they can be used by multiple stacks. Modules with a Pink band can require the selection of lower-level drivers; sometimes these modules are optional or recommended as indicated in the block text. If lower-level drivers are required, the module includes **Add** in the text. Clicking on any Pink-banded modules brings up the **New** icon and lists possible choices.

In the following figure, the DHCPv6 Server is configured with its own packet pool (`g_packet_pool1`). To adjust properties of NetX Duo relating to IPv6 and ICMPv6, which are the underlying protocols for DHCPv6, select the **Add NetX Duo Source stack element** (either box), and choose **New -> NetX Duo Source**.

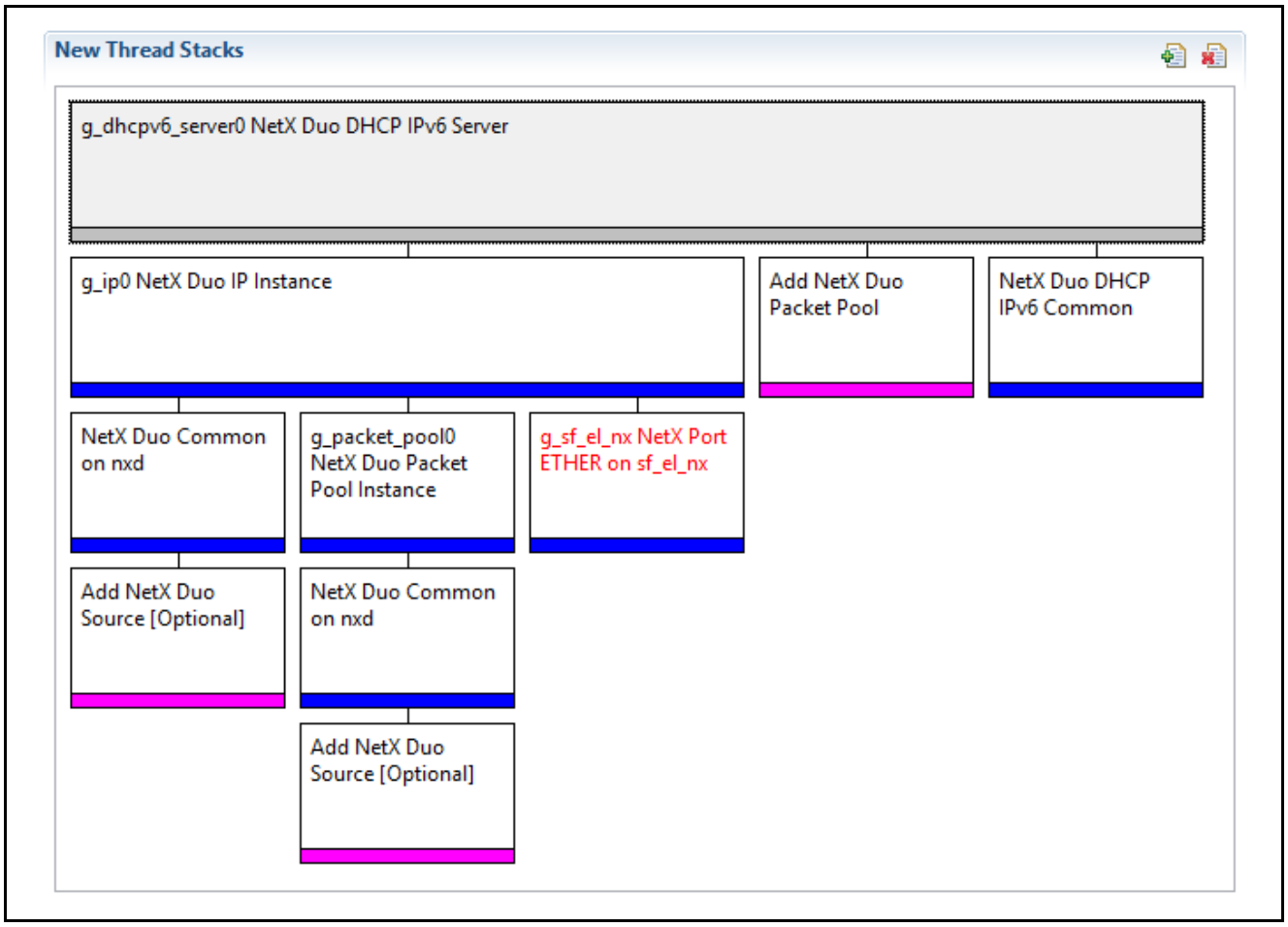

Figure 2. NetX Duo DHCPv6 Server Module Stack

5. Configuring the NetX Duo DHCPv6 Server Module

The NetX Duo DHCPv6 Server module must be configured by the user for the desired operation. The **SSP configuration** window automatically identifies (by highlighting the block in red) any required configuration selections, such as **interrupts** or **operating modes**, which must be configured for lower-level modules for successful operation. Only those properties able to be changed without causing conflicts are available for modification. Other properties are locked and not available for changes and are identified with a **lock** icon for the locked property in the **Properties** window in the ISDE. This approach simplifies the configuration process and makes it much less error-prone than previous manual approaches to configuration. The available configuration settings and defaults for all the **user-accessible** properties are given in the **Properties** tab within the SSP configurator and are shown in the following tables for easy reference.

One of the properties most often identified as requiring a change is the interrupt priority; this configuration setting is available within the **Properties** window of the associated module. Simply select the indicated module and then view the **Properties** window; the interrupt settings are often toward the bottom of the properties list, so scroll down until they become available. Note that the interrupt priorities listed in the **Properties** window in the ISDE includes an indication as to the validity of the setting based on the MCU targeted (CM4 or CM0+). This level of detail is not included in the following configuration properties tables, but is easily visible within the ISDE when configuring interrupt-priority levels.

Note: You may want to open your ISDE and create the NetX Duo DHCPv6 Client module so you can explore the property settings in parallel with the following configuration table settings; this helps to orient you and can be a useful hands-on approach to learning the ins and outs of developing applications with SSP.

Table 3. Configuration Settings for the NetX Duo DHCPv6 Server Module

ISDE Property	Value	Description
Internal thread priority	1	Internal thread priority selection
Client lease time expiration check interval (seconds)	60	Client lease time expiration check interval selection
DHCPv6 packet receive timeout (seconds)	1	DHCPv6 packet receive timeout selection
Server preference ranking for clients	0	Server preference ranking for client's selection
Maximum options to extract from a client message	6	Maximum options to extract from a client message selection
Server interval for first client IP address renewal attempt (seconds)	2000	Server interval for first client IP address renewal attempt selection
Server interval for second client IP address renewal attempt (seconds)	3000	Server interval for second client IP address renewal attempt selection
Time interval after which client IP is deprecated (seconds)	2*NX_DHCPV6_DEFAULT_T1_TIME	Time interval after which client IP is deprecated selection
Time interval after which leased IP in invalid (seconds)	2*NX_DHCPV6_DEFAULT_PREFERRED_TIME	Time interval after which leased IP in invalid selection
Maximum server status option message size (bytes)	100	Maximum server status option message size selection
Maximum Size of the Server's IP lease table (count)	100	Maximum Size of the Server's IP lease table selection
Size of the Server's Client record table (count)	120	Size of the Server's Client record table selection
Server socket fragmentation option	Don't fragment, fragment okay Default: Don't fragment	Server socket fragmentation option selection
Client request success message: granted	IA OPTION GRANTED	Client request successmessage: granted selection
Client request failure message: Failure unspecified	IA OPTION NOT GRANTED - FAILURE UNSPECIFIED	Client request failure message: Failure unspecified selection
Client request failure message: No addresses available	IA OPTION NOT GRANTED - NO ADDRESSES AVAILABLE	Client request failure message: No addresses available selection
Client request failure message: Invalid client request	IA OPTION NOT GRANTED - INVALID CLIENT REQUEST	Client request failure message: Invalid client request selection
Client request failure message: Client not on link	IA OPTION NOT GRANTED - CLIENT NOT ON LINK	Client request failure message: Client not on link selection
Client request failure message: Client must use multicast	IA OPTION NOT GRANTED - CLIENT MUST USE MULTICAST	Client request failure message: Client must use multicast selection
Name	g_dhcpv6_server0	Module name
Internal thread stack size (bytes)	4096	Internal thread stack size selection
Name of address declined handler function	dhcpv6_address_declined_handler	Name of address declined handler function selection

ISDE Property	Value	Description
Name of option request handler	dhcpv6_option_request_handler	Name of option request handler selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

In some cases, settings other than the defaults for lower-level modules can be desirable. For example, it might be useful to select different MAC or IP addresses. The configurable properties for the lower-level stack modules are given in the following sections for completeness, and as a reference.

Note: Most of the property settings for lower-level modules are intuitive and usually can be determined by inspection of the associated properties window from the SSP configurator.

5.1 Configuration Settings for the NetX Duo DHCPv6 Server Lower-Level Modules

Typically, only a small number of settings must be modified from the default for lower-level modules as indicated via the red text in the thread stack block. Notice that some of the configuration properties must be set to a certain value for proper framework operation and will be locked to prevent user modification. The following table identifies all the settings within the properties section for the module.

Table 4. Configuration Settings for the NetX Duo IP Instance

ISDE Property	Value	Description
Name	g_ip0	Module name
IPv4 Address (use commas for separation)	192,168,0,2	IPv4 Address selection
Subnet Mask (use commas for separation)	255,255,255,0	Subnet Mask selection
**IPv6 Global Address (use commas for separation)	0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0	IPv6 global address selection
**IPv6 Link Local Address (use commas for separation, all zeros mean use MAC address)	0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0	IPv6 link local address selection
IP Helper Thread Stack Size (bytes)	2048	IP Helper Thread Stack Size (bytes) selection
IP Helper Thread Priority	3	IP Helper Thread Priority selection
ARP	Enable	ARP selection
ARP Cache Size in Bytes	512	ARP Cache Size in Bytes selection
Reverse ARP	Disable	Reverse ARP selection
TCP	Enable, Disable Default: Enable	TCP selection
UDP	Enable	UDP selection
ICMP	Enable, Disable Default: Enable	ICMP selection
IGMP	Enable, Disable Default: Enable	IGMP selection
IP fragmentation	Enable, Disable Default: Disable	IP fragmentation selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

Table 5. Configuration Settings for the NetX Duo Packet Pool Instance (when added as new)

ISDE Property	Value	Description
Name	g_packet_pool1	Module name
Packet Size in Bytes	640	Packet size selection
Number of Packets in Pool	16	Number of packets in pool selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

Table 6. Configuration Settings for the NetX Duo IP Default Packet Pool Instance

ISDE Property	Value	Description
Name	g_packet_pool0	Module name
Packet Size in Bytes	640	Packet size selection
Number of Packets in Pool	16	Number of packets in pool selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

Table 7. Configuration Settings for the NetX Duo DHCPv6 Common

ISDE Property	Value	Description
Type of Service for UDP requests	Normal, Minimum delay, Maximum data, Maximum reliability, Minimum cost Default: Normal	Type of service UDP requests selection
Time to live	128	Time to live selection
Packet Queue depth	5	Packet queue depth selection
packet allocation timeout (seconds)	3	Packet allocation timeout selection
Interval for active session time update (seconds)	3	Interval for active session time update selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

Table 8. Configuration Settings for the NetX Duo Common

ISDE Property	Value	Description
No configurable properties.		

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

Table 9. Configuration Settings for the NetX Duo Port ETHER

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter checking
Channel 0 Phy Reset Pin	IOPORT_PORT_09_PIN_03	Channel 0 Phy reset pin selection
Channel 0 MAC Address High Bits	0x00002E09	Channel 0 MAC address high bits selection
Channel 0 MAC Address Low Bits	0x0A0076C7	Channel 0 MAC address low bits selection
Channel 1 Phy Reset Pin	IOPORT_PORT_07_PIN_06	Channel 1 Phy reset pin selection
Channel 1 MAC Address High Bits	0x00002E09	Channel 1 MAC address high bits selection
Channel 1 MAC Address Low Bits	0x0A0076C8	Channel 1 MAC address low bits selection
Number of Receive Buffer Descriptors	8	Number of receive buffer descriptors selection

ISDE Property	Value	Description
Number of Transmit Buffer Descriptors	32	Number of transmit buffer descriptors selection
Ethernet Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX®), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	Ethernet interrupt priority selection
Name	g_sf_el_nx	Module name
Channel	0	Channel selection
Callback	NULL	Callback selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

5.2 NetX Duo DHCPv6 Server Module Clock Configuration

The ETHERC peripheral module uses PCLKA as its clock source. The PCLKA frequency is set by using the SSP configurator clock tab prior to a build, or by using the CGC interface at run-time.

5.3 NetX Duo DHCPv6 Server Module Pin Configuration

The ETHERC peripheral module uses pins on the MCU to communicate to external devices. I/O pins must be selected and configured as required by the external device. The following table illustrates the method for selecting the pins within the SSP configuration window and the subsequent table lists an example selection for the pins.

Note: The operation mode selection indicates the peripheral signals available as well as the MCU pins required.

Table 10. Pin Selection for the ETHERC1 Module

Resource	ISDE Tab	Pin selection Sequence
ETHERC	Pins	Select Peripherals > Connectivity:ETHERC > ETHERC1.RMII

Note: The selection sequence assumes ETHERC1 is the desired hardware target for the driver.

Table 11. Pin Configuration Settings for the ETHERC1

Property	Value	Description
Operation Mode	Disabled, Custom, RMII (Default: Disabled)	Select RMII as the Operation Mode for ETHERC1
Pin Group Selection	Mixed, _A only (Default: _A only)	Pin group selection
REF50CK	P701	REF50CK Pin
TXD0	P700	TXD0 Pin
TXD1	P406	TXD1 Pin
TXD_EN	P405	TXD_EN Pin
RXD0	P702	RXD0 Pin
RXD1	P703	RXD1 Pin
RX_ER	P704	RX_ER Pin
CRS_DV	P705	CRS_DV Pin
MDC	P403	MDC Pin
MDIO	P404	MDIO Pin

Note: The example settings are for a project using the Synergy S7G2 MCU Group and the SK-S7G2 Kit. Other Synergy MCUs and kits may have pin configuration settings that differ.

6. Using the NetX Duo DHCPv6 Server Module in an Application

The following example assumes that a system is already established with a working IP, ARP, and UDP enabled, and that the link is running. The typical steps in using the NetX Duo DHCPv6 Server module in an application are:

1. Create a DUID for the DHCPv6 Server using the `nx_dhcpv6_set_server_duid` API.
2. Create a pool of assignable IPv6 addresses for the DHCPv6 Server using the `nx_dhcpv6_create_ip_address_range`.
3. Set the DNS Server option for IPv6 using the `nx_dhcpv6_create_dns_address` API [Optional].
4. Start the DHCPv6 Server with the `nx_dhcpv6_server_start` API.
5. The DHCPv6 Server can be suspended as needed using the `nx_dhcpv6_server_suspend` API.
6. The DHCPv6 Server can be resumed using `nx_dhcpv6_server_resume` API [Optional].
7. Delete the DHCPv6 Server using the `nx_dhcpv6_server_delete` API.

The following figure shows common steps in a typical operational flow.

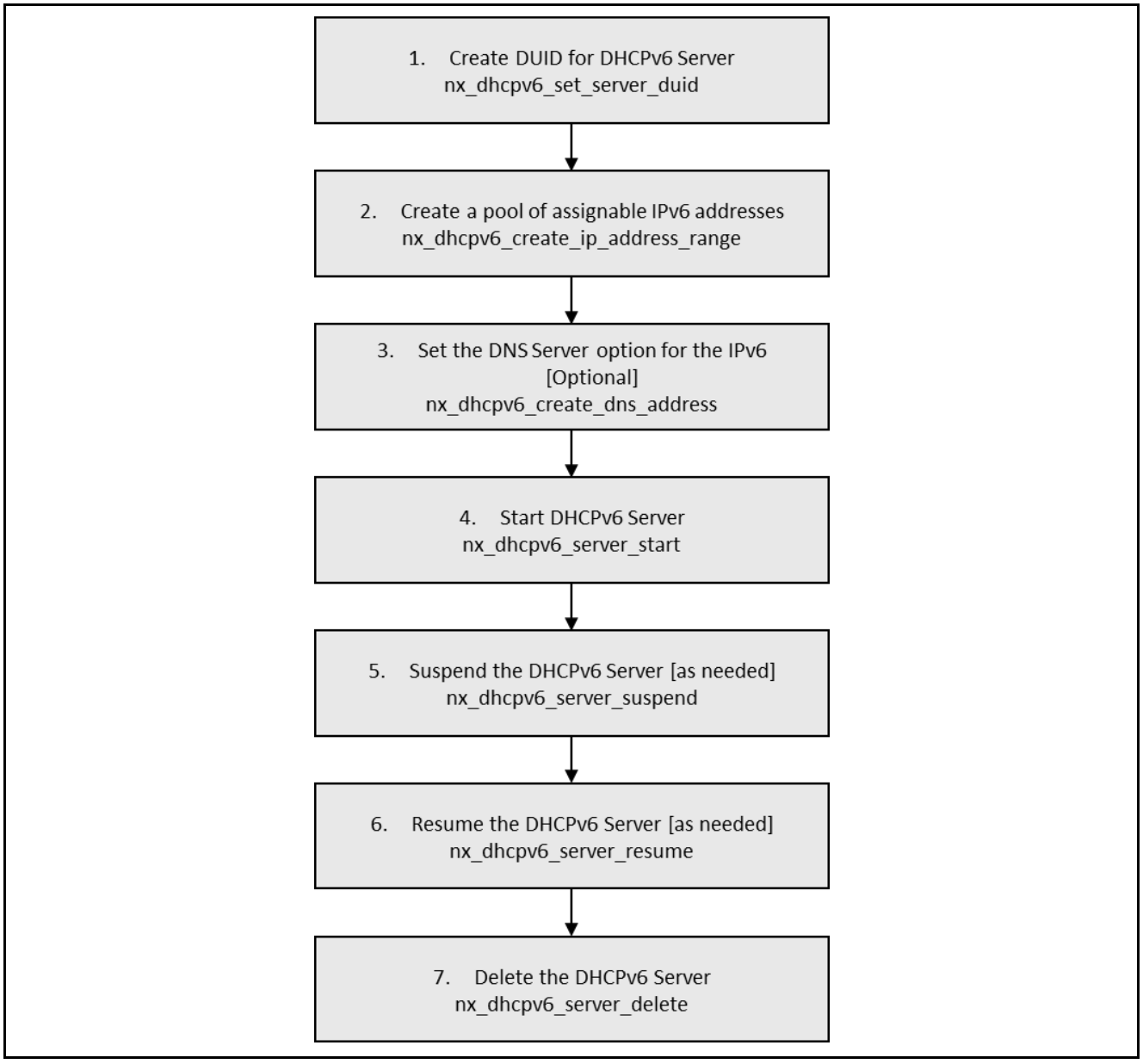

Figure 3. Flow Diagram of a Typical NetX Duo DHCPv6 Server Module Application

7. The NetX Duo DHCPv6 Server Module Application Project

The application project associated with this module guide demonstrates the steps involved in a full design. The project can be found using the link provided in the References section at the end of this document. You may want to import and open the application project within the ISDE and view the configuration settings for the NetX Duo DHCPv6 Server module. You can also read over the code (in `dhcpv6_thread_entry.c`); it's an example of the NetX Duo DHCPv6 Server module APIs in a complete design.

The application project demonstrates the typical use of the NetX Duo DHCPv6 Server module APIs. The application project main thread entry initializes the NetX Duo DHCPv6 Server module. The server is configured and then is started; an internal server thread handles all requests coming from clients using the DHCPv6 protocol. The following table identifies the target versions for the associated software and hardware used by the application project:

Table 12. Software and Hardware Resources Used by the Application

Resource	Revision	Description
e^2 studio	5.4.0.023 or later	Integrated Solution Development Environment
SSP	1.3.0 or later	Synergy Software Platform
IAR EW for Synergy	7.71.2 or later	IAR Embedded Workbench® for Renesas Synergy™
SSC	5.4.0.023 or later	Synergy Standalone Configurator
SK-S7G2	v3.0 to v3.1	Starter Kit

The following figure is a simple flow showing the application project processes.

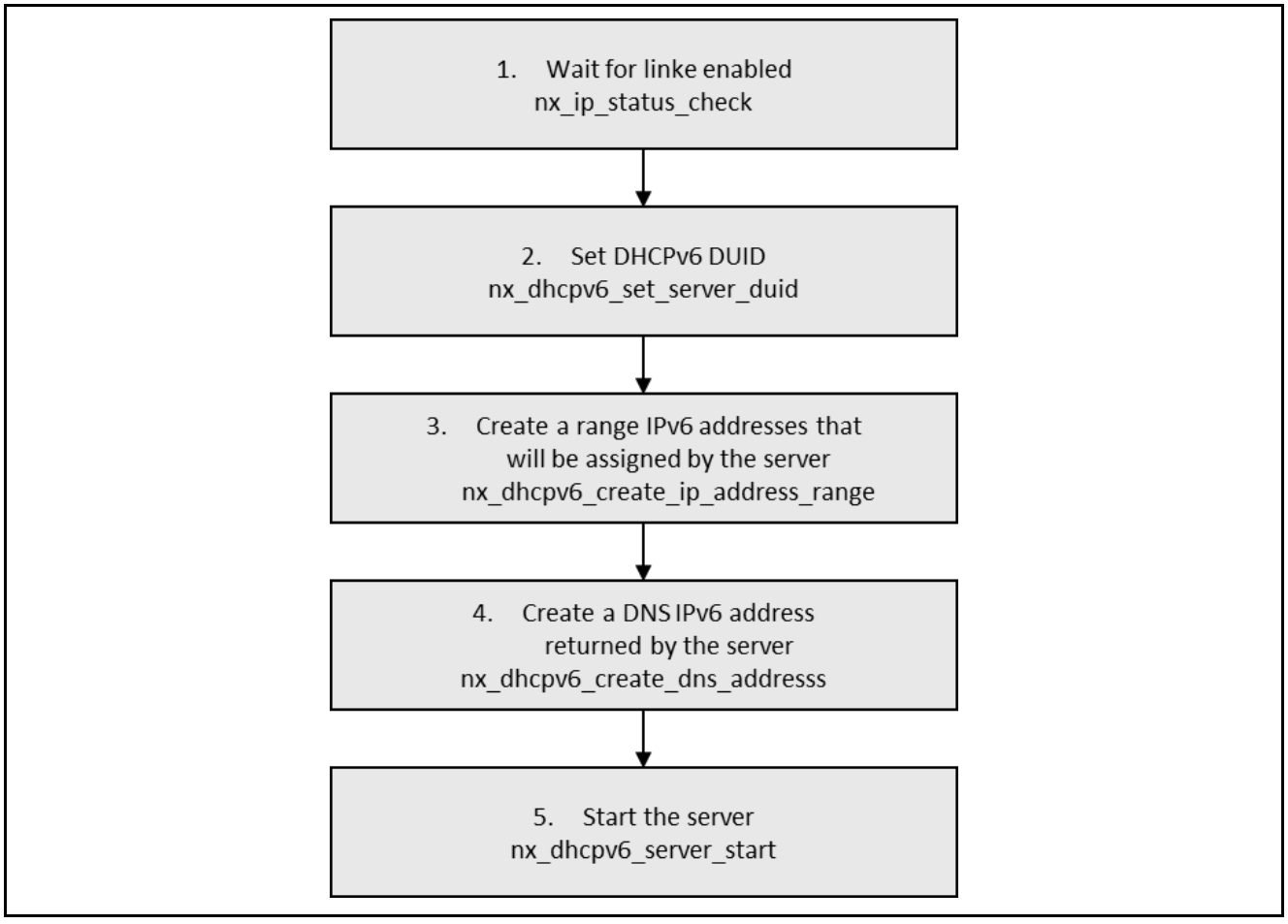

Figure 4. NetX Duo DHCPv6 Server Module Application Project Flow Diagram

The `dhcpv6_thread_entry.c` file is located in the project once it has been imported into the ISDE. You can open this file within the ISDE and follow along with the description provided to help identify key uses of APIs.

The first section of `dhcpv6_thread_entry.c` has the header file which references the DHCPv6 Server instance structure, as well as code section which contains macro constant definitions. The next section is the entry function for the main program-control section. At first, several variables are defined. The most important are three IPv6 addresses: the first and the last IPv6 addresses from the address list, and a DNS address. The address list is used by the server in reply to client requests. Then, the application waits for the link to be enabled. If it is ready, the DHCPv6 Server can be configured. Every DHCPv6 needs a DUID, so the first step is to set it. Next, an IPv6 address list and a DNS address are added to the server. After these operations, the server can be started using the `nx_dhcpv6_server_start API`. Inside the infinite while loop the thread sleep function pauses execution for a couple ThreadX-timer ticks.

Table 13. NetX Duo DHCPv6 Server Module Configuration Settings for the Application Project

ISDE Property	Value Set
Internal thread priority	1
Client lease time expiration check interval (seconds)	60
DHCPv6 packet receive timeout (seconds)	1
Server preference ranking for clients	0
Maximum options to extract from a client message	6
Server interval for first client IP address renewal attempt (seconds)	2000
Server interval for second client IP address renewal attempt (seconds)	3000
Time interval after which client IP is deprecated (seconds)	`2*NX_DHCPV6_DEFAULT_T1_TIME`
Time interval after which leased IP is invalid (seconds)	`2**NX_DHCPV6_PREFERRED_TIME`
Maximum Size of the Server's IP lease table (count)	100
Size of the Server's Client record table (count)	120
Server socket fragmentation option	Don't fragment
Client request success message: granted	IA OPTION GRANTED
Client request failure message: Failure unspecified	IA OPTION NOT GRANTED-FAILURE UNSPECIFIED
Client request failure message: No addresses available	IA OPTION NOT GRANTED-NO ADDRESSES AVAILABLE
Client request failure message: Invalid client request	IA OPTION NOT GRANTED-INVALID CLIENT REQUEST
Client request failure message: Client not on link	IA OPTION NOT GRANTED-CLIENT NOT ON LINK
Client request failure message: Client must use multicast	IA OPTION NOT GRANTED-CLIENT NOT ON LINK
Name	`g_dhcpv6_server`
Internal thread stack size (bytes)	4096
Name of address declined handler function	NULL
Name of option request handler	NULL

Table 14. NetX Duo IP Instance Configuration Settings for the Application Project

ISDE Property	Value Set
Name	`g_ip0`
IPv4 Address (use commas for separation)	192,168,0,2
Subnet Mask (use commas for separation)	255,255,255,0
IPv6 Global Address (use commas for separation)	0xFDED, 0x36B9, 0x02AB, 0x1A74, 0x0, 0x0, 0x0, 0x1
IPv6 Link Local Address (use commas for separation, All zeros means use MAC address)	0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0
IP Helper Thread Stack Size (bytes)	2048

ISDE Property	Value Set
IP Helper Thread Priority	3
ARP	Enable
ARP Cache Size in Bytes	512
Reverse ARP	Enable, Disable (Default: Disable)
TCP	Enable, Disable (Default: Disable)
UDP	Enable
ICMP	Enable, Disable (Default: Disable)
IGMP	Enable, Disable (Default: Disable)
IP fragmentation	Disable
Name of generated initialization function	Ip_init()
Auto Initialization	Enable

Table 15. NetX Duo DHCPv6 Common Configuration Settings for the Application Project

ISDE Property	Value Set
Type of Service for UDP requests	Normal, Minimum delay, Maximum data, Maximum reliability, Minimum cost (Default: Normal)
Time to live	128
Packet Queue depth	5
Packet allocation timeout (seconds)	3
Interval for active session time update (seconds)	3

Table 16. NetX Duo IP Default Packet Pool Instance Configuration Settings for the Application Project

ISDE Property	Value Set
Name	`g_packet_pool0`
Packet Size in Bytes	640
Number of Packets in Pool	16

Table 17. NetX Duo Packet Pool Instance Configuration Settings for the Application Project

ISDE Property	Value Set
Name	`g_packet_pool1`
Packet Size in Bytes	640
Number of Packets in Pool	16

Table 18. NetX Duo Port ETHER Configuration Settings for the Application Project

ISDE Property	Value
Parameter Checking	BSP, Enabled, Disabled (Default: BSP)
Channel 0 Phy Reset Pin	`IOPORT_PORT_09_PIN_03`
Channel 0 MAC Address High Bits	0x00002E09
Channel 0 MAC Address Low Bits	0x0A0076C7
Channel 1 Phy Reset Pin	`IOPORT_PORT_08_PIN_06`
Channel 1 MAC Address High Bits	0x00002E09
Channel 1 MAC Address Low Bits	0x0A0076C8
Number of Receive Buffer Descriptors	8
Number of Transmit Buffer Descriptors	32
Ethernet Interrupt Priority	Priority 9
Name	`g_sf_el_nx`
Channel	1
Callback	NULL

8. Customizing the NetX Duo DHCPv6 Server Module for a Target Application

Some configuration settings are normally be changed by the developer from those shown in the application project. For example, the user can easily change the configuration settings for the IP instance, including the IPv6 global address, which is the IPv6 address of the DHCPv6 Server. The user can also change the range of the addresses assigned by the server; the first and the last IPv6 address from the address list can be configured in the `dhcpv6_thread_entry.c` source code.

9. Running the NetX Duo DHCPv6 Server Module Application Project

To run the NetX Duo DHCPv6 Server module Application project and to see it executed on a target kit, you can simply import it into your ISDE, compile, and run debug. See *Renesas Synergy™ Project Import Guide* (included in this package) for instructions on importing the project into e^2 studio or IAR EW for Synergy, and building/running the application.

To implement the NetX Duo DHCPv6 Server module application in a new project, follow the steps to define, configure, auto-generate files, add code, compile, and debug on the target kit. Following these steps is a hands-on approach to help make the development process with SSP more practical, while just reading this guide tends to be more theoretical.

Note: The following steps are described in sufficient detail for someone experienced with the basic flow through the Synergy development process. If these steps are unfamiliar, see the *SSP User's Manual*; it includes a user tutorial in the first few chapters.

To create and run the NetX Duo DHCPv6 Server module application project, simply follow these steps:

1. Create a new Renesas Synergy project for S7G2-SK in ISDE, called **NetXDuo_DHCPv6_Server_EL_MG_AP**.
2. Select the **Threads** tab.
3. Add a new thread called:
   Symbol dhcpv6_thread
   Name DHCPv6 Thread
4. Add to the DHCPv6 Thread for the NetX Duo DHCP IPv6 Server.
5. Click on the **Generate Project Content** button.
6. Add the code from the supplied project file `dhcpv6_thread_entry.c` or copy over the generated `dhcpv6_thread_entry.c` file.
7. Connect to the host PC via a micro USB cable to J19 on the SK-S7G2 Kit.
8. Start to debug the application.
9. Connect the DHCPv6 Client via Ethernet cable to J11 on SK-S7G2 Kit.

The output can be viewed in the network parameters for the device connected to the board. For example, on a Windows machine, the network parameters received from the DHCPv6 server are in Network Connections -> interface ->status->details.

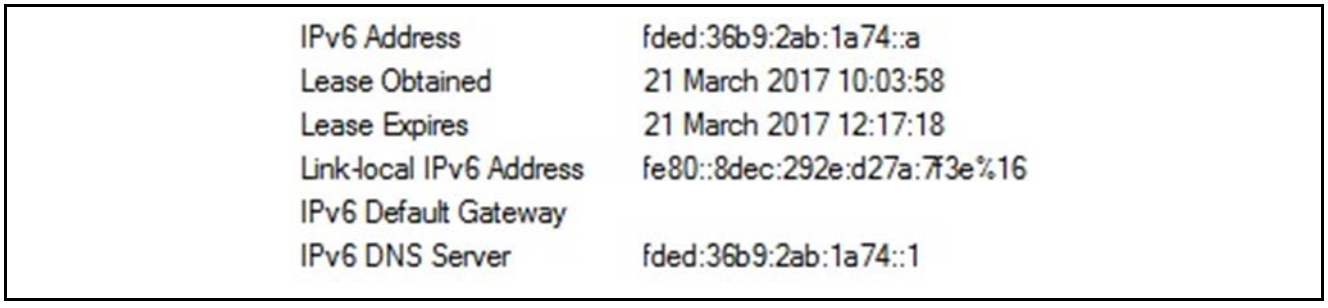

Figure 5. Example Output from NetX Duo DHCPv6 Server Module Application Project

10. NetX Duo DHCPv6 Server Module Conclusion

This module guide has provided all the background information needed to select, add, configure, and use the module in an example project. Many of these steps were time consuming and error-prone activities in previous generations of embedded systems. The Renesas Synergy Platform makes these steps much less time consuming and removes the common errors, like conflicting configuration settings or incorrect selection of lower-level drivers. The use of high-level APIs (as demonstrated in the application project) illustrates additional development time savings by allowing work to begin at a high level and avoiding the time required in older development environments to use or, in some cases, create, lower-level drivers.

11. NetX Duo DHCPv6 Server Module Next Steps

After you have mastered a simple NetX Duo DHCPv6 Server module project, you may want to review a more complex example. You may find that a different protocol should be handled in your application. NetX and NetX Duo for Synergy Platform provide many modules, including DHCP, DNS, HTTP, and more. You can find user manuals for these modules in the References section at the end of this document.

12. NetX Duo DHCPv6 Server Module Reference Information

SSP User Manual: Available in HTML format in the SSP distribution package, and as a pdf from the Synergy Gallery.

Links to all the most up-to-date NetX Duo DHCPv6 Server module reference materials and resources are available on the Synergy Knowledge Base: https://en-support.renesas.com/knowledgeBase/16977455.

Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	www.renesas.com/synergy/software
Synergy Software Package	www.renesas.com/synergy/ssp
Software add-ons	www.renesas.com/synergy/addons
Software glossary	www.renesas.com/synergy/softwareglossary
Development tools	www.renesas.com/synergy/tools
Synergy Hardware	www.renesas.com/synergy/hardware
Microcontrollers	www.renesas.com/synergy/mcus
MCU glossary	www.renesas.com/synergy/mcuglossary
Parametric search	www.renesas.com/synergy/parametric
Kits	www.renesas.com/synergy/kits
Synergy Solutions Gallery	www.renesas.com/synergy/solutionsgallery
Partner projects	www.renesas.com/synergy/partnerprojects
Application projects	www.renesas.com/synergy/applicationprojects
Self-service support resources:	
Documentation	www.renesas.com/synergy/docs
Knowledgebase	www.renesas.com/synergy/knowledgebase
Forums	www.renesas.com/synergy/forum
Training	www.renesas.com/synergy/training
Videos	www.renesas.com/synergy/videos
Chat and web ticket	www.renesas.com/synergy/resourcelibrary

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec.07.17	—	First release document
1.01	Apr.30.19	—	Updates for SSP v1.6.0

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.