

# Kea Messages Manual

---

Copyright © 2011-2014 Internet Systems Consortium, Inc.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Kea Log Messages</b>	<b>2</b>

## **Abstract**

This is the messages manual for Kea version 0.9.1. The most up-to-date version of this document, along with other documents for Kea, can be found at <http://kea.isc.org/docs>.

# Chapter 1

## Introduction

This document lists each message that can be logged by the programs in the Kea package. Each entry in this manual is of the form:

```
IDENTIFICATION message-text
```

... where "IDENTIFICATION" is the message identification included in each message logged and "message-text" is the accompanying message text. The "message-text" may include placeholders of the form "%1", "%2" etc.; these parameters are replaced by relevant values when the message is logged.

Each entry is also accompanied by a description giving more information about the circumstances that result in the message being logged.

For information on configuring and using Kea logging, refer to the [Kea Guide](#).

## Chapter 2

# Kea Log Messages

**ASIODNS\_FD\_ADD\_TCP adding a new TCP server by opened fd %1**

A debug message informing about installing a file descriptor as a server. The file descriptor number is noted.

**ASIODNS\_FD\_ADD\_UDP adding a new UDP server by opened fd %1**

A debug message informing about installing a file descriptor as a server. The file descriptor number is noted.

**ASIODNS\_FETCH\_COMPLETED upstream fetch to %1(%2) has now completed**

A debug message, this records that the upstream fetch (a query made by the resolver on behalf of its client) to the specified address has completed.

**ASIODNS\_FETCH\_STOPPED upstream fetch to %1(%2) has been stopped**

An external component has requested the halting of an upstream fetch. This is an allowed operation, and the message should only appear if debug is enabled.

**ASIODNS\_OPEN\_SOCKET error %1 opening %2 socket to %3(%4)**

The asynchronous I/O code encountered an error when trying to open a socket of the specified protocol in order to send a message to the target address. The number of the system error that caused the problem is given in the message.

**ASIODNS\_READ\_DATA error %1 reading %2 data from %3(%4)**

The asynchronous I/O code encountered an error when trying to read data from the specified address on the given protocol. The number of the system error that caused the problem is given in the message.

**ASIODNS\_READ\_TIMEOUT receive timeout while waiting for data from %1(%2)**

An upstream fetch from the specified address timed out. This may happen for any number of reasons and is most probably a problem at the remote server or a problem on the network. The message will only appear if debug is enabled.

**ASIODNS\_SEND\_DATA error %1 sending data using %2 to %3(%4)**

The asynchronous I/O code encountered an error when trying to send data to the specified address on the given protocol. The number of the system error that caused the problem is given in the message.

**ASIODNS\_SYNC\_UDP\_CLOSE\_FAIL failed to close a DNS/UDP socket: %1**

This is the same to ASIODNS\_UDP\_CLOSE\_FAIL but happens on the "synchronous UDP server", mainly used for the authoritative DNS server daemon.

**ASIODNS\_TCP\_ACCEPT\_FAIL failed to accept TCP DNS connection: %1**

Accepting a TCP connection from a DNS client failed due to an error that could happen but should be rare. The reason for the error is included in the log message. The server still keeps accepting new connections, so unless it happens often it's probably okay to ignore this error. If the shown error indicates something like "too many open files", it's probably because the run time environment is too restrictive on this limitation, so consider adjusting the limit using a tool such as ulimit. If you see other types of errors too often, there may be something overlooked; please file a bug report in that case.

**ASIODNS\_TCP\_CLEANUP\_CLOSE\_FAIL failed to close a DNS/TCP socket on port cleanup: %1**

A TCP DNS server tried to close a TCP socket (one created on accepting a new connection or is already unused) as a step of cleaning up the corresponding listening port, but it failed to do that. This is generally an unexpected event and so is logged as an error. See also the description of `ASIODNS_TCP_CLOSE_ACCEPTOR_FAIL`.

**ASIODNS\_TCP\_CLOSE\_ACCEPTOR\_FAIL failed to close listening TCP socket: %1**

A TCP DNS server tried to close a listening TCP socket (for accepting new connections) as a step of cleaning up the corresponding listening port (e.g., on server shutdown or updating port configuration), but it failed to do that. This is generally an unexpected event and so is logged as an error. See `ASIODNS_TCP_CLOSE_FAIL` on the implication of related system resources.

**ASIODNS\_TCP\_CLOSE\_FAIL failed to close DNS/TCP socket with a client: %1**

A TCP DNS server tried to close a TCP socket used to communicate with a client, but it failed to do that. While closing a socket should normally be an error-free operation, there have been known cases where this happened with a "connection reset by peer" error. This might be because of some odd client behavior, such as sending a TCP RST after establishing the connection and before the server closes the socket, but how exactly this could happen seems to be system dependent (i.e, it's not part of the standard socket API), so it's difficult to provide a general explanation. In any case, it is believed that an error on closing a socket doesn't mean leaking system resources (the kernel should clean up any internal resource related to the socket, just reporting an error detected in the close call), but, again, it seems to be system dependent. This message is logged at a debug level as it's known to happen and could be triggered by a remote node and it would be better to not be too verbose, but you might want to increase the log level and make sure there's no resource leak or other system level troubles when it's logged.

**ASIODNS\_TCP\_CLOSE\_NORESP\_FAIL failed to close DNS/TCP socket with a client: %1**

A TCP DNS server tried to close a TCP socket used to communicate with a client without returning an answer (which normally happens for zone transfer requests), but it failed to do that. See `ASIODNS_TCP_CLOSE_FAIL` for more details.

**ASIODNS\_TCP\_GETREMOTE\_FAIL failed to get remote address of a DNS TCP connection: %1**

A TCP DNS server tried to get the address and port of a remote client on a connected socket but failed. It's expected to be rare but can still happen. See also `ASIODNS_TCP_READLEN_FAIL`.

**ASIODNS\_TCP\_READDATA\_FAIL failed to get DNS data on a TCP socket: %1**

A TCP DNS server tried to read a DNS message (that follows a 2-byte length field) but failed. It's expected to be rare but can still happen. See also `ASIODNS_TCP_READLEN_FAIL`.

**ASIODNS\_TCP\_READLEN\_FAIL failed to get DNS data length on a TCP socket: %1**

A TCP DNS server tried to get the length field of a DNS message (the first 2 bytes of a new chunk of data) but failed. This is generally expected to be rare but can still happen, e.g, due to an unexpected reset of the connection. A specific reason for the failure is included in the log message.

**ASIODNS\_TCP\_WRITE\_FAIL failed to send DNS message over a TCP socket: %1**

A TCP DNS server tried to send a DNS message to a remote client but failed. It's expected to be rare but can still happen. See also `ASIODNS_TCP_READLEN_FAIL`.

**ASIODNS\_UDP\_ASYNC\_SEND\_FAIL Error sending UDP packet to %1: %2**

The low-level ASIO library reported an error when trying to send a UDP packet in asynchronous UDP mode. This can be any error reported by `send_to()`, and can indicate problems such as too high a load on the network, or a problem in the underlying library or system. This packet is dropped and will not be sent, but service should resume normally. If you see a single occurrence of this message, it probably does not indicate any significant problem, but if it is logged often, it is probably a good idea to inspect your network traffic.

**ASIODNS\_UDP\_CLOSE\_FAIL failed to close a DNS/UDP socket: %1**

A UDP DNS server tried to close its UDP socket, but failed to do that. This is generally an unexpected event and so is logged as an error.

**ASIODNS\_UDP\_RECEIVE\_FAIL failed to receive UDP DNS packet: %1**

Receiving a UDP packet from a DNS client failed due to an error that could happen but should be very rare. The server still keeps receiving UDP packets on this socket. The reason for the error is included in the log message. This log message is basically not expected to appear at all in practice; if it does, there may be some system level failure and other system logs may have to be checked.

**ASIODNS\_UDP\_SYNC\_RECEIVE\_FAIL failed to receive UDP DNS packet: %1**

This is the same to ASIODNS\_UDP\_RECEIVE\_FAIL but happens on the "synchronous UDP server", mainly used for the authoritative DNS server daemon.

**ASIODNS\_UDP\_SYNC\_SEND\_FAIL Error sending UDP packet to %1: %2**

The low-level ASIO library reported an error when trying to send a UDP packet in synchronous UDP mode. See ASIODNS\_UDP\_ASYNC\_SEND\_FAIL for more information.

**ASIODNS\_UNKNOWN\_ORIGIN unknown origin for ASIO error code %1 (protocol: %2, address %3)**

An internal consistency check on the origin of a message from the asynchronous I/O module failed. This may indicate an internal error; please submit a bug report.

**ASIODNS\_UNKNOWN\_RESULT unknown result (%1) when IOFetch::stop() was executed for I/O to %2(%3)**

An internal error indicating that the termination method of the resolver's upstream fetch class was called with an unknown result code (which is given in the message). Please submit a bug report.

**CC\_ASYNC\_READ\_FAILED asynchronous read failed (error code = %1)**

This marks a low level error, we tried to read data from the message queue daemon asynchronously, but the ASIO library returned an error.

**CC\_CONN\_ERROR error connecting to message queue (%1)**

It is impossible to reach the message queue daemon for the reason given. It is unlikely there'll be reason for whatever program this currently is to continue running, as the communication with the rest of Kea is vital for the components.

**CC\_DISCONNECT disconnecting from message queue daemon**

The library is disconnecting from the message queue daemon. This debug message indicates that the program is trying to shut down gracefully.

**CC\_ESTABLISH trying to establish connection with message queue daemon at %1**

This debug message indicates that the command channel library is about to connect to the message queue daemon, which should be listening on the UNIX-domain socket listed in the output.

**CC\_ESTABLISHED successfully connected to message queue daemon**

This debug message indicates that the connection was successfully made, this should follow CC\_ESTABLISH.

**CC\_GROUP\_RECEIVE trying to receive a message with seq %1**

Debug message, noting that a message is expected to come over the command channel.

**CC\_GROUP\_RECEIVED message arrived ('%1', '%2')**

Debug message, noting that we successfully received a message (its envelope and payload listed). This follows CC\_GROUP\_RECEIVE but might happen some time later, depending if we waited for it or just polled.

**CC\_GROUP\_SEND sending message '%1' to group '%2'**

Debug message, we're about to send a message over the command channel.

**CC\_INVALID\_LENGTHS invalid length parameters (%1, %2)**

This happens when garbage comes over the command channel or some kind of confusion happens in the program. The data received from the socket make no sense if we interpret it as lengths of message. The first one is total length of the message; the second is the length of the header. The header and its length (2 bytes) is counted in the total length.

**CC\_LENGTH\_NOT\_READY length not ready**

There should be data representing the length of message on the socket, but it is not there.

**CC\_LNAME\_RECEIVED received local name: %1**

Debug message: the local module received its unique identifier (name) from msgq on completion of establishing the session with msgq.

**CC\_NO\_MESSAGE no message ready to be received yet**

The program polled for incoming messages, but there was no message waiting. This is a debug message which may happen only after CC\_GROUP\_RECEIVE.



**CC\_NO\_MSGQ unable to connect to message queue (%1)**

It isn't possible to connect to the message queue daemon, for reason listed. It is unlikely any program will be able continue without the communication.

**CC\_READ\_ERROR error reading data from command channel (%1)**

A low level error happened when the library tried to read data from the command channel socket. The reason is listed.

**CC\_READ\_EXCEPTION error reading data from command channel (%1)**

We received an exception while trying to read data from the command channel socket. The reason is listed.

**CC\_REPLY replying to message from '%1' with '%2'**

Debug message, noting we're sending a response to the original message with the given envelope.

**CC\_SET\_TIMEOUT setting timeout to %1ms**

Debug message. A timeout for which the program is willing to wait for a reply is being set.

**CC\_START\_READ starting asynchronous read**

Debug message. From now on, when a message (or command) comes, it'll wake the program and the library will automatically pass it over to correct place.

**CC\_SUBSCRIBE subscribing to communication group %1**

Debug message. The program wants to receive messages addressed to this group.

**CC\_TIMEOUT timeout reading data from command channel**

The program waited too long for data from the command channel (usually when it sent a query to different program and it didn't answer for whatever reason).

**CC\_UNSUBSCRIBE unsubscribing from communication group %1**

Debug message. The program no longer wants to receive messages addressed to this group.

**CC\_WRITE\_ERROR error writing data to command channel (%1)**

A low level error happened when the library tried to write data to the command channel socket.

**CC\_ZERO\_LENGTH invalid message length (0)**

The library received a message length being zero, which makes no sense, since all messages must contain at least the envelope.

**CONFIG\_CCSESSION\_MSG error in CC session message: %1**

There was a problem with an incoming message on the command and control channel. The message does not appear to be a valid command, and is missing a required element or contains an unknown data format. This most likely means that another Kea module is sending a bad message. The message itself is ignored by this module.

**CONFIG\_CCSESSION\_MSG\_INTERNAL error handling CC session message: %1**

There was an internal problem handling an incoming message on the command and control channel. An unexpected exception was thrown, details of which are appended to the message. The module will continue to run, but will not send back an answer.

The most likely cause of this error is a programming error. Please raise a bug report.

**CONFIG\_CCSESSION\_STOPPING error sending stopping message: %1**

There was a problem when sending a message signaling that the module using this CCSession is stopping. This message is sent so that the rest of the system is aware that the module is no longer running. Apart from logging this message, the error itself is ignored, and the ModuleCCSession is still stopped. The specific exception message is printed.

**CONFIG\_CCSESSION\_STOPPING\_UNKNOWN unknown error sending stopping message**

Similar to CONFIG\_CCSESSION\_STOPPING, but in this case the exception that is seen is not a standard exception, and further information is unknown. This is a bug.

**CONFIG\_GET\_FAIL error getting configuration from cfgmgr: %1**

The configuration manager returned an error when this module requested the configuration. The full error message answer from the configuration manager is appended to the log error. The most likely cause is that the module is of a different (command specification) version than the running configuration manager.

---

**CONFIG\_JSON\_PARSE JSON parse error in %1: %2**

There was an error parsing the JSON file. The given file does not appear to be in valid JSON format. Please verify that the filename is correct and that the contents are valid JSON.

**CONFIG\_LOG\_EXPLICIT will use logging configuration for explicitly-named logger %1**

This is a debug message. When processing the "loggers" part of the configuration file, the configuration library found an entry for the named logger that matches the logger specification for the program. The logging configuration for the program will be updated with the information.

**CONFIG\_LOG\_IGNORE\_EXPLICIT ignoring logging configuration for explicitly-named logger %1**

This is a debug message. When processing the "loggers" part of the configuration file, the configuration library found an entry for the named logger. As this does not match the logger specification for the program, it has been ignored.

**CONFIG\_LOG\_IGNORE\_WILD ignoring logging configuration for wildcard logger %1**

This is a debug message. When processing the "loggers" part of the configuration file, the configuration library found the named wildcard entry (one containing the "\*" character) that matched a logger already matched by an explicitly named entry. The configuration is ignored.

**CONFIG\_LOG\_WILD\_MATCH will use logging configuration for wildcard logger %1**

This is a debug message. When processing the "loggers" part of the configuration file, the configuration library found the named wildcard entry (one containing the "\*" character) that matches a logger specification in the program. The logging configuration for the program will be updated with the information.

**CONFIG\_MOD\_SPEC\_FORMAT module specification error in %1: %2**

The given file does not appear to be a valid specification file: details are included in the message. Please verify that the filename is correct and that its contents are a valid Kea module specification.

**CONFIG\_MOD\_SPEC\_REJECT module specification rejected by cfgmgr: %1**

The specification file for this module was rejected by the configuration manager. The full error message answer from the configuration manager is appended to the log error. The most likely cause is that the module is of a different (specification file) version than the running configuration manager.

**CONFIG\_OPEN\_FAIL error opening %1: %2**

There was an error opening the given file. The reason for the failure is included in the message.

**CONFIG\_RPC\_SEQ RPC call %1 to %2 with seq %3**

Debug message, saying there's a RPC call of given command to given module. It has internal sequence number as listed in the message.

**DCTL\_CCSESSION\_ENDING %1 ending control channel session**

This debug message is issued just before the controller attempts to disconnect from its session with the Kea control channel.

**DCTL\_CCSESSION\_STARTING %1 starting control channel session, specfile: %2**

This debug message is issued just before the controller attempts to establish a session with the Kea control channel.

**DCTL\_COMMAND\_RECEIVED %1 received command: %2, arguments: %3**

A debug message listing the command (and possible arguments) received from the Kea control system by the controller.

**DCTL\_CONFIG\_COMPLETE server has completed configuration: %1**

This is an informational message announcing the successful processing of a new configuration. It is output during server startup, and when an updated configuration is committed by the administrator. Additional information may be provided.

**DCTL\_CONFIG\_FILE\_LOAD\_FAIL %1 reason: %2**

This fatal error message indicates that the application attempted to load its initial configuration from file and has failed. The service will exit.

**DCTL\_CONFIG\_LOAD\_FAIL %1 configuration failed to load: %2**

This critical error message indicates that the initial application configuration has failed. The service will start, but will not process requests until the configuration has been corrected.

**DCTL\_CONFIG\_START parsing new configuration: %1**

A debug message indicating that the application process has received an updated configuration and has passed it to its configuration manager for parsing.

**DCTL\_CONFIG\_STUB %1 configuration stub handler called**

This debug message is issued when the dummy handler for configuration events is called. This only happens during initial startup.

**DCTL\_CONFIG\_UPDATE %1 updated configuration received: %2**

A debug message indicating that the controller has received an updated configuration from the Kea configuration system.

**DCTL\_DISCONNECT\_FAIL %1 controller failed to end session with Bundy: %2**

This message indicates that while shutting down, the DHCP-DDNS controller encountered an error terminating communication with the Bundy configuration system. The service will still exit. While theoretically possible, this situation is rather unlikely.

**DCTL\_INIT\_PROCESS %1 initializing the application**

This debug message is issued just before the controller attempts to create and initialize its application instance.

**DCTL\_INIT\_PROCESS\_FAIL %1 application initialization failed: %2**

This error message is issued if the controller could not initialize the application and will exit.

**DCTL\_NOT\_RUNNING %1 application instance is not running**

A warning message is issued when an attempt is made to shut down the application when it is not running.

**DCTL\_PARSER\_FAIL : %1**

On receipt of a new configuration, the server failed to create a parser to decode the contents of the named configuration element, or the creation succeeded but the parsing actions and committal of changes failed. The reason for the failure is given in the message.

**DCTL\_PROCESS\_FAILED %1 application execution failed: %2**

The controller has encountered a fatal error while running the application and is terminating. The reason for the failure is included in the message.

**DCTL\_RUN\_PROCESS %1 starting application event loop**

This debug message is issued just before the controller invokes the application run method.

**DCTL\_SESSION\_FAIL %1 controller failed to establish Kea session: %1**

The controller has failed to establish communication with the rest of Kea and will exit.

**DCTL\_STANDALONE %1 skipping message queue, running standalone**

This is a debug message indicating that the controller is running in the application in standalone mode. This means it will not be connected to the Kea message queue. Standalone mode is only useful during program development, and should not be used in a production environment.

**DHCP4\_ACTIVATE\_INTERFACE activating interface %1**

This message is printed when DHCPv4 server enabled an interface to be used to receive DHCPv4 traffic. IPv4 socket on this interface will be opened once Interface Manager starts up procedure of opening sockets.

**DHCP4\_CCSESSION\_STARTED control channel session started on socket %1**

A debug message issued during startup after the DHCPv4 server has successfully established a session with the Kea control channel.

**DHCP4\_CCSESSION\_STARTING starting control channel session, specfile: %1**

This debug message is issued just before the DHCPv4 server attempts to establish a session with the Kea control channel.

**DHCP4\_CLASS\_ASSIGNED client packet has been assigned to the following class(es): %1**

This debug message informs that incoming packet has been assigned to specified class or classes. This is a normal behavior and indicates successful operation.

**DHCP4\_CLIENT\_NAME\_PROC\_FAIL failed to process the fqdn or hostname sent by a client: %1**

This debug message is issued when the DHCP server was unable to process the FQDN or Hostname option sent by a client. This is likely because the client's name was malformed or due to internal server error.

**DHCP4\_COMMAND\_RECEIVED received command %1, arguments: %2**

A debug message listing the command (and possible arguments) received from the Kea control system by the DHCPv4 server.

**DHCP4\_CONFIG\_COMPLETE DHCPv4 server has completed configuration: %1**

This is an informational message announcing the successful processing of a new configuration. It is output during server startup, and when an updated configuration is committed by the administrator. Additional information may be provided.

**DHCP4\_CONFIG\_LOAD\_FAIL configuration error using file: %1, reason: %2**

This error message indicates that the DHCPv4 configuration has failed. If this is an initial configuration (during server's startup) the server will fail to start. If this is a dynamic reconfiguration attempt the server will continue to use an old configuration.

**DHCP4\_CONFIG\_NEW\_SUBNET a new subnet has been added to configuration: %1**

This is an informational message reporting that the configuration has been extended to include the specified IPv4 subnet.

**DHCP4\_CONFIG\_OPTION\_DUPLICATE multiple options with the code %1 added to the subnet %2**

This warning message is issued on an attempt to configure multiple options with the same option code for a particular subnet. Adding multiple options is uncommon for DHCPv4, but is not prohibited.

**DHCP4\_CONFIG\_RECEIVED received configuration %1**

A debug message listing the configuration received by the DHCPv4 server. The source of that configuration depends on used configuration backend.

**DHCP4\_CONFIG\_START DHCPv4 server is processing the following configuration: %1**

This is a debug message that is issued every time the server receives a configuration. That happens at start up and also when a server configuration change is committed by the administrator.

**DHCP4\_CONFIG\_UPDATE updated configuration received: %1**

A debug message indicating that the DHCPv4 server has received an updated configuration from the Kea configuration system.

**DHCP4\_DDNS\_REQUEST\_SEND\_FAILED failed sending a request to kea-dhcp-ddns, error: %1, ncr: %2**

This error message indicates that DHCPv4 server attempted to send a DDNS update request to the DHCP-DDNS server. This is most likely a configuration or networking error.

**DHCP4\_DEACTIVATE\_INTERFACE deactivate interface %1**

This message is printed when DHCPv4 server disables an interface from being used to receive DHCPv4 traffic. Sockets on this interface will not be opened by the Interface Manager until interface is enabled.

**DHCP4\_DHCID\_COMPUTE\_ERROR failed to compute the DHCID for lease: %1, reason: %2**

This error message is logged when the attempt to compute DHCID for a specified lease has failed. The lease details and reason for failure is logged in the message.

**DHCP4\_DISCOVER\_CLASS\_PROCESSING\_FAILED client class specific processing failed for DHCPDISCOVER**

This debug message means that the server processing that is unique for each client class has reported a failure. The response packet will not be sent.

**DHCP4\_DYNAMIC\_RECONFIGURATION initiate server reconfiguration using file: %1, after receiving SIGHUP signal**

This is the info message logged when the DHCPv4 server starts reconfiguration as a result of receiving SIGHUP signal.

**DHCP4\_DYNAMIC\_RECONFIGURATION\_FAIL dynamic server reconfiguration failed with file: %1**

This is an error message logged when the dynamic reconfiguration of the DHCP server failed.

**DHCP4\_EMPTY\_HOSTNAME received empty hostname from the client, skipping processing of this option**

This debug message is issued when the server received an empty Hostname option from a client. Server does not process empty Hostname options and therefore option is skipped.

**DHCP4\_HOOKS\_LIBS\_RELOAD\_FAIL reload of hooks libraries failed**

A "libreload" command was issued to reload the hooks libraries but for some reason the reload failed. Other error messages issued from the hooks framework will indicate the nature of the problem.

**DHCP4\_HOOK\_BUFFER\_RCVD\_SKIP received DHCPv4 buffer was dropped because a callout set the skip flag.**

This debug message is printed when a callout installed on buffer4\_receive hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to drop the packet.

**DHCP4\_HOOK\_BUFFER\_SEND\_SKIP prepared DHCPv4 response was dropped because a callout set the skip flag.**

This debug message is printed when a callout installed on buffer4\_send hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to drop the packet. Server completed all the processing (e.g. may have assigned, updated or released leases), but the response will not be send to the client.

**DHCP4\_HOOK\_LEASE4\_RELEASE\_SKIP DHCPv4 lease was not released because a callout set the skip flag.**

This debug message is printed when a callout installed on lease4\_release hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to not release a lease.

**DHCP4\_HOOK\_PACKET\_RCVD\_SKIP received DHCPv4 packet was dropped, because a callout set the skip flag.**

This debug message is printed when a callout installed on the pkt4\_receive hook point sets the skip flag. For this particular hook point, the setting of the flag instructs the server to drop the packet.

**DHCP4\_HOOK\_PACKET\_SEND\_SKIP prepared DHCPv6 response was not sent, because a callout set skip flag.**

This debug message is printed when a callout installed on the pkt4\_send hook point sets the skip flag. For this particular hook point, the setting of the flag instructs the server to drop the packet. This means that the client will not get any response, even though the server processed client's request and acted on it (e.g. possibly allocated a lease).

**DHCP4\_HOOK\_SUBNET4\_SELECT\_SKIP no subnet was selected, because a callout set skip flag.**

This debug message is printed when a callout installed on the subnet4\_select hook point sets the skip flag. For this particular hook point, the setting of the flag instructs the server not to choose a subnet, an action that severely limits further processing; the server will be only able to offer global options - no addresses will be assigned.

**DHCP4\_INFORM\_CLASS\_PROCESSING\_FAILED client class specific processing failed for DHCPINFORM**

This debug message means that the server processing that is unique for each client class has reported a failure. The response packet will not be sent.

**DHCP4\_INIT\_FAIL failed to initialize Kea server: %1**

The server has failed to initialize. This may be because the configuration was not successful, or it encountered any other critical error on startup. Attached error message provides more details about the issue.

**DHCP4\_INVALID\_ADDRESS\_INIT\_REBOOT invalid address %1 requested by INIT-REBOOT client (id: %2, hwaddr: %3)**

This debug message is issued when the client being in the INIT-REBOOT state requested an address which is not assigned to him. The server will respond to this client with DHCPNAK.

**DHCP4\_LEASE\_ADVERT lease %1 advertised (client client-id %2, hwaddr %3)**

This debug message indicates that the server successfully advertised a lease. It is up to the client to choose one server out of other advertised and continue allocation with that server. This is a normal behavior and indicates successful operation.

**DHCP4\_LEASE\_ADVERT\_FAIL failed to advertise a lease for client-id %1, hwaddr %2, client sent ciaddr %3, requested-ip-a**

This message indicates that the server has failed to offer a lease to the specified client after receiving a DISCOVER message from it. There are many possible reasons for such a failure.

**DHCP4\_LEASE\_ALLOC lease %1 has been allocated for client-id %2, hwaddr %3**

This debug message indicates that the server successfully granted a lease in response to client's REQUEST message. This is a normal behavior and indicates successful operation.

**DHCP4\_LEASE\_ALLOC\_FAIL failed to grant a lease for client-id %1, hwaddr %2, client sent ciaddr %3, requested-ip-address**

This message indicates that the server failed to grant a lease to the specified client after receiving a REQUEST message from it. There are many possible reasons for such a failure. Additional messages will indicate the reason.

**DHCP4\_LEASE\_DATABASE\_TIMERS\_EXEC\_FAIL failed to execute timer-based functions for lease database: %1**

A warning message executed when a server process is unable to execute the periodic actions for the lease database. An example of the periodic action is a Lease File Cleanup. One of the reasons for the failure is a misconfiguration of the lease database, whereby the lease database hasn't been selected.

**DHCP4\_NAME\_GEN\_UPDATE\_FAIL failed to update the lease after generating name for a client: %1**

This message indicates the failure when trying to update the lease and/or options in the server's response with the hostname generated by the server from the acquired address. The message argument indicates the reason for the failure.

**DHCP4\_NCR\_CREATION\_FAILED failed to generate name change requests for DNS: %1**

This message indicates that server was unable to generate NameChangeRequests which should be sent to the kea-dhcp\_ddns module to create new DNS records for the lease being acquired or to update existing records for the renewed lease. The reason for the failure is printed in the logged message.

**DHCP4\_NOT\_RUNNING DHCPv4 server is not running**

A warning message is issued when an attempt is made to shut down the DHCPv4 server but it is not running.

**DHCP4\_NO\_LEASE\_INIT\_REBOOT no lease for address %1 requested by INIT-REBOOT client (id: %2, hwaddr: %3)**

This debug message is issued when the client being in the INIT-REBOOT state requested an address but this client is unknown. The server will not respond.

**DHCP4\_NO\_SOCKETS\_OPEN no interface configured to listen to DHCP traffic**

This warning message is issued when current server configuration specifies no interfaces that server should listen on, or specified interfaces are not configured to receive the traffic.

**DHCP4\_NO\_SUBNET\_FOR\_DIRECT\_CLIENT no suitable subnet configured for a direct client sending packet with transaction id %1**

This info message is logged when received a message from a directly connected client but there is no suitable subnet configured for the interface on which this message has been received. The IPv4 address assigned on this interface must belong to one of the configured subnets. Otherwise received message is dropped.

**DHCP4\_OPEN\_SOCKET opening sockets on port %1**

A debug message issued during startup, this indicates that the DHCPv4 server is about to open sockets on the specified port.

**DHCP4\_OPEN\_SOCKET\_FAIL failed to open socket: %1**

A warning message issued when IfaceMgr fails to open and bind a socket. The reason for the failure is appended as an argument of the log message.

**DHCP4\_PACKET\_DROP\_NO\_TYPE packet received on interface %1 dropped, because of missing msg-type option**

This is a debug message informing that incoming DHCPv4 packet did not have mandatory DHCP message type option and thus was dropped.

**DHCP4\_PACKET\_NOT\_FOR\_US received DHCPv4 message (transid=%1, iface=%2) dropped because it contains foreign server identifier %3**

This debug message is issued when received DHCPv4 message is dropped because it is addressed to a different server, i.e. a server identifier held by this message doesn't match the identifier used by our server. The arguments of this message hold the name of the transaction id and interface on which the message has been received.

**DHCP4\_PACKET\_PARSE\_FAIL failed to parse incoming packet: %1**

The DHCPv4 server has received a packet that it is unable to interpret. The reason why the packet is invalid is included in the message.

**DHCP4\_PACKET\_PROCESS\_FAIL failed to process packet received from %1: %2**

This is a general catch-all message indicating that the processing of a received packet failed. The reason is given in the message. The server will not send a response but will instead ignore the packet.

**DHCP4\_PACKET\_RECEIVED %1 (type %2) packet received on interface %3**

A debug message noting that the server has received the specified type of packet on the specified interface. Note that a packet marked as UNKNOWN may well be a valid DHCP packet, just a type not expected by the server (e.g. it will report a received OFFER packet as UNKNOWN).

**DHCP4\_PACKET\_RECEIVE\_FAIL error on attempt to receive packet: %1**

The DHCPv4 server tried to receive a packet but an error occurred during this attempt. The reason for the error is included in the message.

**DHCP4\_PACKET\_SEND\_FAIL failed to send DHCPv4 packet: %1**

This error is output if the DHCPv4 server fails to send an assembled DHCP message to a client. The reason for the error is included in the message.

---

**DHCP4\_PARSER\_COMMIT\_EXCEPTION parser failed to commit changes**

On receipt of message containing details to a change of the DHCPv4 server configuration, a set of parsers were successfully created, but one of them failed to commit its changes due to a low-level system exception being raised. Additional messages may be output indicating the reason.

**DHCP4\_PARSER\_COMMIT\_FAIL parser failed to commit changes: %1**

On receipt of message containing details to a change of the DHCPv4 server configuration, a set of parsers were successfully created, but one of them failed to commit its changes. The reason for the failure is given in the message.

**DHCP4\_PARSER\_CREATED created parser for configuration element %1**

A debug message output during a configuration update of the DHCPv4 server, notifying that the parser for the specified configuration element has been successfully created.

**DHCP4\_PARSER\_EXCEPTION failed to create or run parser for configuration element %1**

On receipt of message containing details to a change of its configuration, the DHCPv4 server failed to create a parser to decode the contents of the named configuration element, or the creation succeeded but the parsing actions and committal of changes failed. The message has been output in response to a non-Kea exception being raised. Additional messages may give further information.

**DHCP4\_PARSER\_FAIL failed to create or run parser for configuration element %1: %2**

On receipt of message containing details to a change of its configuration, the DHCPv4 server failed to create a parser to decode the contents of the named configuration element, or the creation succeeded but the parsing actions and committal of changes failed. The reason for the failure is given in the message.

**DHCP4\_QUERY\_DATA received packet type %1, data is <%2>**

A debug message listing the data received from the client.

**DHCP4\_QUEUE\_NCR name change request to %1 DNS entry queued: %2**

A debug message which is logged when the NameChangeRequest to add or remove a DNS entries for a particular lease has been queued. The first parameter of this log message indicates whether the DNS entry is to be added or removed. The second parameter carries the details of the NameChangeRequest.

**DHCP4\_RELEASE address %1 belonging to client-id %2, hwaddr %3 was released properly.**

This debug message indicates that an address was released properly. It is a normal operation during client shutdown.

**DHCP4\_RELEASE\_EXCEPTION exception %1 while trying to release address %2**

This message is output when an error was encountered during an attempt to process a RELEASE message. The error will not affect the client, which does not expect any response from the server for RELEASE messages. Depending on the nature of problem, it may affect future server operation.

**DHCP4\_RELEASE\_FAIL failed to remove lease for address %1 for duid %2, hwaddr %3**

This error message indicates that the software failed to remove a lease from the lease database. It is probably due to an error during a database operation: resolution will most likely require administrator intervention (e.g. check if DHCP process has sufficient privileges to update the database). It may also be triggered if a lease was manually removed from the database during RELEASE message processing.

**DHCP4\_RELEASE\_FAIL\_NO\_LEASE client (client-id %1, hwaddr %2) tried to release address, %3, but there is no such lease**

This warning message is printed when client attempts to release a lease, but no such lease is known to the server.

**DHCP4\_RELEASE\_FAIL\_WRONG\_CLIENT\_ID client (client-id %2) tried to release address %1, but it belongs to client (cli**

This warning message indicates that client tried to release an address that belongs to a different client. This should not happen in normal circumstances and may indicate a misconfiguration of the client. However, since the client releasing the address will stop using it anyway, there is a good chance that the situation will correct itself.

**DHCP4\_RELEASE\_FAIL\_WRONG\_HWADDR client (client-id %2) tried to release address %1, but sent from a wrong hardw**

This warning message indicates that client tried to release an address that does belong to it, but the lease information was associated with a different hardware address. One possible reason for using different hardware address is that a cloned virtual machine was not updated and both clones use the same client-id.

**DHCP4\_REQUEST\_CLASS\_PROCESSING\_FAILED client class specific processing failed for DHCPREQUEST**

This debug message means that the server processing that is unique for each client class has reported a failure. The response packet will not be sent.

**DHCP4\_RESPONSE\_DATA responding with packet type %1, data is <%2>**

A debug message listing the data returned to the client.

**DHCP4\_SERVER\_FAILED server failed: %1**

The DHCPv4 server has encountered a fatal error and is terminating. The reason for the failure is included in the message.

**DHCP4\_SHUTDOWN server shutdown**

The DHCPv4 server has terminated normally.

**DHCP4\_SHUTDOWN\_REQUEST shutdown of server requested**

This debug message indicates that a shutdown of the DHCPv4 server has been requested via a call to the 'shutdown' method of the core Dhcpv4Srv object.

**DHCP4\_SRV\_CONSTRUCT\_ERROR error creating Dhcpv4Srv object, reason: %1**

This error message indicates that during startup, the construction of a core component within the DHCPv4 server (the Dhcpv4 server object) has failed. As a result, the server will exit. The reason for the failure is given within the message.

**DHCP4\_STARTED Kea DHCPv4 server version %1 started**

This informational message indicates that the DHCPv4 server has processed all configuration information and is ready to process DHCPv4 packets. The version is also printed.

**DHCP4\_STARTING Kea DHCPv4 server version %1 starting**

This informational message indicates that the DHCPv4 server has processed any command-line switches and is starting. The version is also printed.

**DHCP4\_START\_INFO pid: %1, port: %2, verbose: %3**

This is a debug message issued during the DHCPv4 server startup. It lists some information about the parameters with which the server is running.

**DHCP4\_SUBNET\_SELECTED the %1 subnet was selected for client assignment**

This is a debug message noting the selection of a subnet to be used for address and option assignment. Subnet selection is one of the early steps in the processing of incoming client message.

**DHCP4\_SUBNET\_SELECTION\_FAILED failed to select a subnet for incoming packet, src: %1, type: %2**

This warning message is output when a packet was received from a subnet for which the DHCPv4 server has not been configured. The most probable cause is a misconfiguration of the server.

**DHCP4\_UNRECOGNIZED\_RCVD\_PACKET\_TYPE received message (transaction id %1) has unrecognized type %2 in option**

This debug message indicates that the message type carried in DHCPv4 option 53 is unrecognized by the server. The valid message types are listed on the IANA website: <http://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml#message-type-53>. The message will not be processed by the server.

**DHCP4\_UNSUPPORTED\_RCVD\_PACKET\_TYPE received message (transaction id %1), having type %2 is not supported**

This debug message indicates that the message type carried in DHCPv4 option 53 is valid but the message will not be processed by the server. This includes messages being normally sent by the server to the client, such as Offer, ACK, NAK etc.

**DHCP6\_ACTIVATE\_INTERFACE activating interface %1**

This message is printed when DHCPv6 server enabled an interface to be used to receive DHCPv6 traffic. IPv6 socket on this interface will be opened once Interface Manager starts up procedure of opening sockets.

**DHCP6\_CCSESSION\_STARTED control channel session started on socket %1**

A debug message issued during startup after the IPv6 DHCP server has successfully established a session with the Kea control channel.

**DHCP6\_CCSESSION\_STARTING starting control channel session, specfile: %1**

This debug message is issued just before the IPv6 DHCP server attempts to establish a session with the Kea control channel.



**DHCP6\_CLASS\_ASSIGNED client packet has been assigned to the following class(es): %1**

This debug message informs that incoming packet has been assigned to specified class or classes. This is a normal message.

**DHCP6\_CLIENTID\_MISSING mandatory client-id option is missing, message from %1 dropped**

This error message indicates that the received message is being dropped because it does not include the mandatory client-id option necessary for address assignment. The most likely cause is a problem with the client.

**DHCP6\_COMMAND\_RECEIVED received command %1, arguments: %2**

A debug message listing the command (and possible arguments) received from the Kea control system by the IPv6 DHCP server.

**DHCP6\_CONFIG\_COMPLETE DHCPv6 server has completed configuration: %1**

This is an informational message announcing the successful processing of a new configuration. It is output during server startup, and when an updated configuration is committed by the administrator. Additional information may be provided.

**DHCP6\_CONFIG\_LOAD\_FAIL configuration error using file: %1, reason: %2**

This error message indicates that the DHCPv6 configuration has failed. If this is an initial configuration (during server's startup) the server will fail to start. If this is a dynamic reconfiguration attempt the server will continue to use an old configuration.

**DHCP6\_CONFIG\_NEW\_SUBNET a new subnet has been added to configuration: %1**

This is an informational message reporting that the configuration has been extended to include the specified subnet.

**DHCP6\_CONFIG\_OPTION\_DUPLICATE multiple options with the code: %1 added to the subnet: %2**

This warning message is issued on an attempt to configure multiple options with the same option code for the particular subnet. Adding multiple options is uncommon for DHCPv6, but it is not prohibited.

**DHCP6\_CONFIG\_RECEIVED received configuration: %1**

A debug message listing the configuration received by the DHCPv6 server. The source of that configuration depends on used configuration backend.

**DHCP6\_CONFIG\_START DHCPv6 server is processing the following configuration: %1**

This is a debug message that is issued every time the server receives a configuration. That happens at start up and also when a server configuration change is committed by the administrator.

**DHCP6\_CONFIG\_UPDATE updated configuration received: %1**

A debug message indicating that the IPv6 DHCP server has received an updated configuration from the Kea configuration system.

**DHCP6\_DB\_BACKEND\_STARTED lease database started (type: %1, name: %2)**

This informational message is printed every time the IPv6 DHCP server is started. It indicates what database backend type is being used to store lease and other information.

**DHCP6\_DDNS\_CREATE\_ADD\_NAME\_CHANGE\_REQUEST created name change request: %1**

This debug message is logged when the new Name Change Request has been created to perform the DNS Update, which adds new RRs.

**DHCP6\_DDNS\_CREATE\_REMOVE\_NAME\_CHANGE\_REQUEST created name change request: %1**

This debug message is logged when the new Name Change Request has been created to perform the DNS Update, which removes RRs from the DNS.

**DHCP6\_DDNS\_LEASE\_ASSIGN\_FQDN\_CHANGE FQDN for the allocated lease: %1 has changed. New values: hostname = %2**

This debug message is logged when FQDN mapping for a particular lease has been changed by the recent Request message. This mapping will be changed in DNS.

**DHCP6\_DDNS\_LEASE\_RENEW\_FQDN\_CHANGE FQDN for the renewed lease: %1 has changed. New values: hostname = %2**

This debug message is logged when FQDN mapping for a particular lease has been changed by the recent Renew message. This mapping will be changed in DNS.

**DHCP6\_DDNS\_RECEIVE\_FQDN received DHCPv6 Client FQDN Option: %1**

This debug message is logged when server has found the DHCPv6 Client FQDN Option sent by a client and started processing it.

**DHCP6\_DDNS\_REMOVE\_INVALID\_HOSTNAME invalid FQDN: %1 for the lease: %2 when removing DNS bindings**

This error message is issued when a lease being deleted contains an indication that the DNS Update has been performed for it, but the FQDN held in the lease database has invalid format and can't be transformed to the canonical on-wire format.

**DHCP6\_DDNS\_REQUEST\_SEND\_FAILED failed sending a request to kea-dhcp-ddns, error: %1, ncr: %2**

This error message indicates that IPv6 DHCP server failed to send a DDNS update request to the DHCP-DDNS server. This is most likely a configuration or networking error.

**DHCP6\_DDNS\_SEND\_FQDN sending DHCPv6 Client FQDN Option to the client: %1**

This debug message is logged when server includes an DHCPv6 Client FQDN Option in its response to the client.

**DHCP6\_DEACTIVATE\_INTERFACE deactivate interface %1**

This message is printed when DHCPv6 server disables an interface from being used to receive DHCPv6 traffic. Sockets on this interface will not be opened by the Interface Manager until interface is enabled.

**DHCP6\_DYNAMIC\_RECONFIGURATION initiate server reconfiguration using file: %1, after receiving SIGHUP signal**

This is the info message logged when the DHCPv6 server starts reconfiguration as a result of receiving SIGHUP signal.

**DHCP6\_DYNAMIC\_RECONFIGURATION\_FAIL dynamic server reconfiguration failed with file: %1**

This is an error message logged when the dynamic reconfiguration of the DHCP server failed.

**DHCP6\_EXTEND\_LEASE\_SUBNET\_SELECTED the %1 subnet was selected for client extending its lease**

This is a debug message informing that a given subnet was selected. It will be used for extending lifetime of the lease. This is one of the early steps in the processing of incoming client message.

**DHCP6\_EXTEND\_LEASE\_SUBNET\_SELECT\_FAILED failed to select a subnet for received %1: src=%2 type=%3**

This warning message is output when a Renew or Rebind was received from a subnet for which the DHCPv6 server has not been configured. The cause is most likely due to a misconfiguration of the server. The packet processing will continue, but the response will only contain generic configuration parameters and no addresses or prefixes.

**DHCP6\_EXTEND\_NA\_UNKNOWN received unknown IA\_NA from client (duid=%1, iaid=%2) in subnet %3**

This warning message is printed when client attempts to extend the lease for the address (in the IA\_NA option) but no such lease is known by the server. It typically means that client has attempted to use its lease past its lifetime: causes of this include a adjustment of the client's date/time setting or poor support on the client for sleep/recovery. A properly implemented client will recover from such a situation by restarting the lease allocation process after receiving a negative reply from the server.

An alternative cause could be that the server has lost its database recently and does not recognize its well-behaving clients. This is more probable if you see many such messages. Clients will recover from this, but they will most likely get a different IP addresses and experience a brief service interruption.

**DHCP6\_EXTEND\_NA\_UNKNOWN\_SUBNET %1 message received from client on unknown subnet (duid=%2, iaid=%3)**

A warning message indicating that a client is attempting to extend lease lifetime for the address, but the server does not have any information about the subnet this client belongs to. This may mean that faulty the mobile client changed its location and is trying to renew its old address (client is supposed to send confirm, not renew in such cases, according to RFC3315) or the server configuration has changed and information about existing subnet was removed. Note that in a sense this is worse case of DHCP6\_EXTEND\_NA\_UNKNOWN, as not only the lease is unknown, but also the subnet is. Depending on the reasons of this condition, it may or may not correct on its own.

**DHCP6\_EXTEND\_PD\_NO\_BINDING client sent a %1 message to extend lifetimes of prefixes for an unknown binding: duid=**

This warning message is logged when a client attempts to extend the lifetime of the prefix it is using, but the server was unable to find the binding to which this lease belongs. This error condition has different implications for the Renew

and Rebind messages. For the former, the server will respond with NoBinding status code. For the latter, the server will discard the message.

**DHCP6\_EXTEND\_PD\_UNKNOWN\_SUBNET %1 message received from client on unknown subnet (duid=%2, iaid=%3)**

A warning message indicating that a client is attempting to extend lease lifetime for the prefix, but the server doesn't have any information about the subnet this client belongs to.

**DHCP6\_HOOKS\_LIBS\_RELOAD\_FAIL reload of hooks libraries failed**

A "libreload" command was issued to reload the hooks libraries but for some reason the reload failed. Other error messages issued from the hooks framework will indicate the nature of the problem.

**DHCP6\_HOOK\_BUFFER\_RCVD\_SKIP received DHCPv6 buffer was dropped because a callout set the skip flag**

This debug message is printed when a callout installed on buffer6\_receive hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to drop the packet.

**DHCP6\_HOOK\_BUFFER\_SEND\_SKIP prepared DHCPv6 response was dropped because a callout set the skip flag**

This debug message is printed when a callout installed on buffer6\_send hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to drop the packet. Server completed all the processing (e.g. may have assigned, updated or released leases), but the response will not be send to the client.

**DHCP6\_HOOK\_LEASE6\_EXTEND\_SKIP DHCPv6 lease lifetime was not extended because a callout set the skip flag for mess**

or lease6\_rebind hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to not extend the lifetime for a lease. If client requested renewal of multiples leases (by sending multiple IA options), the server will skip the renewal of the one in question and will proceed with other renewals as usual.

**DHCP6\_HOOK\_LEASE6\_RELEASE\_NA\_SKIP DHCPv6 address lease was not released because a callout set the skip flag**

This debug message is printed when a callout installed on the lease6\_release hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to not release a lease. If a client requested the release of multiples leases (by sending multiple IA options), the server will retain this particular lease and proceed with other releases as usual.

**DHCP6\_HOOK\_LEASE6\_RELEASE\_PD\_SKIP DHCPv6 prefix lease was not released because a callout set the skip flag**

This debug message is printed when a callout installed on lease6\_release hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to not release a lease. If client requested release of multiples leases (by sending multiple IA options), the server will retains this particular lease and will proceed with other renewals as usual.

**DHCP6\_HOOK\_LEASE6\_RENEW\_SKIP DHCPv6 lease was not renewed because a callout set the skip flag**

This debug message is printed when a callout installed on lease6\_renew hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to not renew a lease. If client requested renewal of multiples leases (by sending multiple IA options), the server will skip the renewal of the one in question and will proceed with other renewals as usual.

**DHCP6\_HOOK\_PACKET\_RCVD\_SKIP received DHCPv6 packet was dropped because a callout set the skip flag**

This debug message is printed when a callout installed on the pkt6\_receive hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to drop the packet.

**DHCP6\_HOOK\_PACKET\_SEND\_SKIP prepared DHCPv6 response was not sent because a callout set the skip flag**

This debug message is printed when a callout installed on the pkt6\_send hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to drop the packet. This effectively means that the client will not get any response, even though the server processed client's request and acted on it (e.g. possibly allocated a lease).

**DHCP6\_HOOK\_SUBNET6\_SELECT\_SKIP no subnet was selected because a callout set the skip flag**

This debug message is printed when a callout installed on the subnet6\_select hook point set the skip flag. For this particular hook point, the setting of the flag instructs the server not to choose a subnet, an action that severely limits further processing; the server will be only able to offer global options - no addresses or prefixes will be assigned.

---

**DHCP6\_INIT\_FAIL failed to initialize Kea server: %1**

The server has failed to establish communication with the rest of Kea, failed to read JSON configuration file or encountered any other critical issue that prevents it from starting up properly. Attached error message provides more details about the issue.

**DHCP6\_LEASE\_ADVERT address lease %1 advertised (client duid=%2, iaid=%3)**

This debug message indicates that the server successfully advertised an address lease. It is up to the client to choose one server out of the advertised servers and continue allocation with that server. This is a normal behavior and indicates successful operation.

**DHCP6\_LEASE\_ADVERT\_FAIL failed to advertise an address lease for client duid=%1, iaid=%2**

This message indicates that in response to a received SOLICIT, the server failed to advertise a non-temporary lease for a given client. There may be many reasons for such failure. Each failure is logged in a separate log entry.

**DHCP6\_LEASE\_ALLOC address lease %1 has been allocated (client duid=%2, iaid=%3)**

This debug message indicates that in response to a client's REQUEST message, the server successfully granted a non-temporary address lease. This is a normal behavior and indicates successful operation.

**DHCP6\_LEASE\_ALLOC\_FAIL failed to grant an address lease for client duid=%1, iaid=%2**

This message indicates that in response to a received REQUEST, the server failed to grant a non-temporary address lease for the client. There may be many reasons for such failure. Each failure is logged in a separate log entry.

**DHCP6\_LEASE\_DATABASE\_TIMERS\_EXEC\_FAIL failed to execute timer-based functions for lease database: %1**

A warning message executed when a server process is unable to execute the periodic actions for the lease database. An example of the periodic action is a Lease File Cleanup. One of the reasons for the failure is a misconfiguration of the lease database, whereby the lease database hasn't been selected.

**DHCP6\_LEASE\_NA\_WITHOUT\_DUID address lease for address %1 does not have a DUID**

This error message indicates a database consistency problem. The lease database has an entry indicating that the given address is in use, but the lease does not contain any client identification. This is most likely due to a software error: please raise a bug report. As a temporary workaround, manually remove the lease entry from the database.

**DHCP6\_LEASE\_PD\_WITHOUT\_DUID prefix lease for address %1 does not have a DUID**

This error message indicates a database consistency failure. The lease database has an entry indicating that the given prefix is in use, but the lease does not contain any client identification. This is most likely due to a software error: please raise a bug report. As a temporary workaround, manually remove the lease entry from the database.

**DHCP6\_NAME\_GEN\_UPDATE\_FAIL failed to update the lease using address %1, after generating FQDN for a client, reason %2**

This message indicates the failure when trying to update the lease and/or options in the server's response with the hostname generated by the server from the acquired address.

**DHCP6\_NOT\_RUNNING IPv6 DHCP server is not running**

A warning message is issued when an attempt is made to shut down the IPv6 DHCP server but it is not running.

**DHCP6\_NO\_INTERFACES failed to detect any network interfaces**

During startup the IPv6 DHCP server failed to detect any network interfaces and is therefore shutting down.

**DHCP6\_NO\_SOCKETS\_OPEN no interface configured to listen to DHCP traffic**

This warning message is issued when current server configuration specifies no interfaces that server should listen on, or specified interfaces are not configured to receive the traffic.

**DHCP6\_OPEN\_SOCKET opening sockets on port %1**

A debug message issued during startup, this indicates that the IPv6 DHCP server is about to open sockets on the specified port.

**DHCP6\_OPEN\_SOCKET\_FAIL failed to open socket: %1**

A warning message issued when IfaceMgr fails to open and bind a socket. The reason for the failure is appended as an argument of the log message.

**DHCP6\_PACKET\_MISMATCH\_SERVERID\_DROP dropping packet %1 (transid=%2, interface=%3) having mismatched server identifier**

A debug message noting that server has received message with server identifier option that not matching server identifier that server is using.

**DHCP6\_PACKET\_PARSE\_FAIL failed to parse incoming packet: %1**

The DHCPv6 server has received a packet that it is unable to interpret. There may be many causes: truncated header, truncated or malformed option, trailing padding that contains garbage etc. More information is specified as a parameter.

**DHCP6\_PACKET\_PROCESS\_FAIL processing of %1 message received from %2 failed: %3**

This is a general catch-all message indicating that the processing of the specified packet type from the indicated address failed. The reason is given in the message. The server will not send a response but will instead ignore the packet.

**DHCP6\_PACKET\_RECEIVED %1 packet received**

A debug message noting that the server has received the specified type of packet. Note that a packet marked as UNKNOWN may well be a valid DHCP packet, just a type not expected by the server (e.g. it will report a received OFFER packet as UNKNOWN).

**DHCP6\_PACKET\_RECEIVE\_FAIL error on attempt to receive packet: %1**

The IPv6 DHCP server tried to receive a packet but an error occurred during this attempt. The reason for the error is included in the message.

**DHCP6\_PACKET\_SEND\_FAIL failed to send DHCPv6 packet: %1**

This error is output if the IPv6 DHCP server fails to send an assembled DHCP message to a client. The reason for the error is included in the message.

**DHCP6\_PACK\_FAIL failed to assemble response correctly**

This error is output if the server failed to assemble the data to be returned to the client into a valid packet. The reason is most likely to be to a programming error: please raise a bug report.

**DHCP6\_PARSER\_COMMIT\_EXCEPTION parser failed to commit changes**

On receipt of message containing details to a change of the IPv6 DHCP server configuration, a set of parsers were successfully created, but one of them failed to commit its changes due to a low-level system exception being raised. Additional messages may be output indicating the reason.

**DHCP6\_PARSER\_COMMIT\_FAIL parser failed to commit changes: %1**

On receipt of message containing details to a change of the IPv6 DHCP server configuration, a set of parsers were successfully created, but one of them failed to commit its changes. The reason for the failure is given in the message.

**DHCP6\_PARSER\_CREATED created parser for configuration element %1**

A debug message output during a configuration update of the IPv6 DHCP server, notifying that the parser for the specified configuration element has been successfully created.

**DHCP6\_PARSER\_EXCEPTION failed to create or run parser for configuration element %1**

On receipt of message containing details to a change of its configuration, the IPv6 DHCP server failed to create a parser to decode the contents of the named configuration element, or the creation succeeded but the parsing actions and committal of changes failed. The message has been output in response to a non-Kea exception being raised. Additional messages may give further information.

The most likely cause of this is that the specification file for the server (which details the allowable contents of the configuration) is not correct for this version of Kea. This may be the result of an interrupted installation of an update to Kea.

**DHCP6\_PARSER\_FAIL failed to create or run parser for configuration element %1: %2**

On receipt of message containing details to a change of its configuration, the IPv6 DHCP server failed to create a parser to decode the contents of the named configuration element, or the creation succeeded but the parsing actions and committal of changes failed. The reason for the failure is given in the message.

**DHCP6\_PD\_LEASE\_ADVERT prefix lease %1/%2 advertised (client duid=%3, iaid=%4)**

This debug message indicates that the server successfully advertised a prefix lease. It is up to the client to choose one server out of the advertised servers and continue allocation with that server. This is a normal behavior and indicates successful operation.

**DHCP6\_PD\_LEASE\_ADVERT\_FAIL failed to advertise a prefix lease for client duid=%1, iaid=%2**

This message indicates that in response to a received SOLICIT, the server failed to advertise a prefix lease for the client. There may be many reasons for such failure. Each failure is logged in a separate log entry.

**DHCP6\_PD\_LEASE\_ALLOC prefix lease %1/%2 has been allocated (client duid=%3, iaid=%4)**

This debug message indicates that in response to a client's REQUEST message, the server successfully granted a prefix delegation lease. This is a normal behavior and indicates successful operation.

**DHCP6\_PD\_LEASE\_ALLOC\_FAIL failed to grant a prefix lease for client duid=%1, iaid=%2**

This message indicates that the server failed to grant (in response to received REQUEST) a prefix lease for a given client. There may be many reasons for such failure. Each failure is logged in a separate log entry.

**DHCP6\_PROCESS\_IA\_NA\_REQUEST server is processing IA\_NA option (duid=%1, iaid=%2, hint=%3)**

This is a debug message that indicates the processing of a received IA\_NA option. It may optionally contain an address that may be used by the server as a hint for possible requested address.

**DHCP6\_PROCESS\_IA\_PD\_REQUEST server is processing IA\_PD option (duid=%1, iaid=%2, hint=%3)**

This is a debug message that indicates a processing of received IA\_PD option. It may optionally contain a prefix that may be used by the server as a hint for possible requested prefix.

**DHCP6\_QUERY\_DATA received packet length %1, data length %2, data is %3**

A debug message listing the data received from the client or relay.

**DHCP6\_RELEASE\_MISSING\_CLIENTID client (address=%1) sent RELEASE message without mandatory client-id**

This warning message indicates that client sent RELEASE message without mandatory client-id option. This is most likely caused by a buggy client (or a relay that malformed forwarded message). This request will not be processed and a response with error status code will be sent back.

**DHCP6\_RELEASE\_NA address %1 belonging to client duid=%2, iaid=%3 was released properly**

This debug message indicates that an address was released properly. It is a normal operation during client shutdown.

**DHCP6\_RELEASE\_NA\_FAIL failed to remove address lease for address %1 for duid=%2, iaid=%3**

This error message indicates that the software failed to remove an address lease from the lease database. It probably due to an error during a database operation: resolution will most likely require administrator intervention (e.g. check if DHCP process has sufficient privileges to update the database). It may also be triggered if a lease was manually removed from the database during RELEASE message processing.

**DHCP6\_RELEASE\_NA\_FAIL\_WRONG\_DUID client (duid=%1) tried to release address %2, but it belongs to another client (duid=%3)**

This warning message indicates that a client tried to release an address that belongs to a different client. This should not happen in normal circumstances and may indicate a misconfiguration of the client. However, since the client releasing the address will stop using it anyway, there is a good chance that the situation will correct itself.

**DHCP6\_RELEASE\_NA\_FAIL\_WRONG\_IAID client (duid=%1) tried to release address %2, but it used wrong IAID (expected %3)**

This warning message indicates that client tried to release an address that does belong to it, but the address was expected to be in a different IA (identity association) container. This probably means that the client's support for multiple addresses is flawed.

**DHCP6\_RELEASE\_PD prefix %1 belonging to client duid=%2, iaid=%3 was released properly**

This debug message indicates that a prefix was released properly. It is a normal operation during client shutdown.

**DHCP6\_RELEASE\_PD\_FAIL failed to remove prefix lease for address %1 for duid=%2, iaid=%3**

This error message indicates that the software failed to remove a prefix lease from the lease database. It probably due to an error during a database operation: resolution will most likely require administrator intervention (e.g. check if DHCP process has sufficient privileges to update the database). It may also be triggered if a lease was manually removed from the database during RELEASE message processing.

**DHCP6\_RELEASE\_PD\_FAIL\_WRONG\_DUID client (duid=%1) tried to release prefix %2, but it belongs to another client (duid=%3)**

This warning message indicates that client tried to release a prefix that belongs to a different client. This should not happen in normal circumstances and may indicate a misconfiguration of the client. However, since the client releasing the prefix will stop using it anyway, there is a good chance that the situation will correct itself.

**DHCP6\_RELEASE\_PD\_FAIL\_WRONG\_IAID client (duid=%1) tried to release prefix %2, but it used wrong IAID (expected**

This warning message indicates that client tried to release a prefix that does belong to it, but the address was expected to be in a different IA (identity association) container. This probably means that the client's support for multiple prefixes is flawed.

**DHCP6\_REQUIRED\_OPTIONS\_CHECK\_FAIL %1 message received from %2 failed the following check: %3**

This message indicates that received DHCPv6 packet is invalid. This may be due to a number of reasons, e.g. the mandatory client-id option is missing, the server-id forbidden in that particular type of message is present, there is more than one instance of client-id or server-id present, etc. The exact reason for rejecting the packet is included in the message.

**DHCP6\_RESPONSE\_DATA responding with packet type %1 data is %2**

A debug message listing the data returned to the client.

**DHCP6\_SERVERID\_GENERATED server-id %1 has been generated and will be stored in %2**

This informational messages indicates that the server was not able to read its server identifier (DUID) and has generated a new one. This server-id will be stored in a file and will be read and used during next restart. It is normal behavior when the server is started for the first time. If this message is printed every start, please check that the server have sufficient permission to write its server-id file and that the file is not corrupt.

Changing the server identifier in a production environment is not recommended as existing clients will not recognize the server and may go through a rebind phase. However, they should be able to recover without losing their leases.

**DHCP6\_SERVERID\_LOADED server-id %1 has been loaded from file %2**

This debug message indicates that the server loaded its server identifier. That value is sent in all server responses and clients use it to discriminate between servers. This is a part of normal startup or reconfiguration procedure.

**DHCP6\_SERVERID\_WRITE\_FAIL server was not able to write its ID to file %1**

This warning message indicates that server was not able to write its server identifier (DUID) to a file. This likely indicates lack of write permission to a given file or directory. This is not critical and the server will continue to operate, but server will generate different DUID during every start and clients will need to go through a rebind phase to recover.

**DHCP6\_SERVER\_FAILED server failed: %1**

The IPv6 DHCP server has encountered a fatal error and is terminating. The reason for the failure is included in the message.

**DHCP6\_SHUTDOWN server shutdown**

The IPv6 DHCP server has terminated normally.

**DHCP6\_SHUTDOWN\_REQUEST shutdown of server requested**

This debug message indicates that a shutdown of the IPv6 server has been requested via a call to the 'shutdown' method of the core Dhcpv6Srv object.

**DHCP6\_SOCKET\_UNICAST server is about to open socket on address %1 on interface %2**

This is a debug message that inform that a unicast socket will be opened.

**DHCP6\_SRV\_CONSTRUCT\_ERROR error creating Dhcpv6Srv object, reason: %1**

This error message indicates that during startup, the construction of a core component within the IPv6 DHCP server (the Dhcpv6 server object) has failed. As a result, the server will exit. The reason for the failure is given within the message.

**DHCP6\_STANDALONE skipping message queue, running standalone**

This is a debug message indicating that the IPv6 server is running in standalone mode, not connected to the message queue. Standalone mode is only useful during program development, and should not be used in a production environment.

**DHCP6\_STARTED Kea DHCPv6 server version %1 started**

This informational message indicates that the IPv6 DHCP server has processed all configuration information and is ready to process DHCPv6 packets. The version is also printed.

**DHCP6\_STARTING Kea DHCPv6 server version %1 starting**

This informational message indicates that the IPv6 DHCP server has processed any command-line switches and is starting. The version is also printed.

**DHCP6\_START\_INFO pid: %1, port: %2, verbose: %3**

This is a debug message issued during the IPv6 DHCP server startup. It lists some information about the parameters with which the server is running.

**DHCP6\_SUBNET\_SELECTED the %1 subnet was selected for client assignment**

This is a debug message informing that a given subnet was selected. It will be used for address and option assignment. This is one of the early steps in the processing of incoming client message.

**DHCP6\_SUBNET\_SELECTION\_FAILED failed to select a subnet for incoming packet, src=%1 type=%2**

This warning message is output when a packet was received from a subnet for which the DHCPv6 server has not been configured. The cause is most likely due to a misconfiguration of the server. The packet processing will continue, but the response will only contain generic configuration parameters and no addresses or prefixes.

**DHCP6\_UNKNOWN\_MSG\_RECEIVED received unknown message (type %d) on interface %2**

This debug message is printed when server receives a message of unknown type. That could either mean missing functionality or invalid or broken relay or client. The list of formally defined message types is available here: <http://www.iana.org/assignment> parameters.

**DHCP6\_UNKNOWN\_RELEASE\_NA received RELEASE from unknown client (IA\_NA, duid=%1, iaid=%2)**

This warning message is printed when client attempts to release an address lease, but no such lease is known by the server. See the explanation of the status code DHCP6\_UNKNOWN\_RENEW\_NA for possible reasons for such behavior.

**DHCP6\_UNKNOWN\_RELEASE\_PD received RELEASE from unknown client (IA\_PD, duid=%1, iaid=%2)**

This warning message is printed when client attempts to release a prefix lease, but no such lease is known by the server. See the explanation of the status code DHCP6\_UNKNOWN\_RENEW\_PD for possible reasons for such behavior.

**DHCPSRV\_ADDRESS4\_ALLOC\_ERROR error during attempt to allocate an IPv4 address: %1**

An error occurred during an attempt to allocate an IPv4 address, the reason for the failure being contained in the message. The server will return a message to the client refusing a lease.

**DHCPSRV\_ADDRESS4\_ALLOC\_FAIL failed to allocate an IPv4 address after %1 attempt(s)**

The DHCP allocation engine gave up trying to allocate an IPv4 address after the specified number of attempts. This probably means that the address pool from which the allocation is being attempted is either empty, or very nearly empty. As a result, the client will have been refused a lease.

This message may indicate that your address pool is too small for the number of clients you are trying to service and should be expanded. Alternatively, if you know that the number of concurrently active clients is less than the addresses you have available, you may want to consider reducing the lease lifetime. In this way, addresses allocated to clients that are no longer active on the network will become available sooner.

**DHCPSRV\_ADDRESS6\_ALLOC\_ERROR error during attempt to allocate an IPv6 address: %1**

An error occurred during an attempt to allocate an IPv6 address, the reason for the failure being contained in the message. The server will return a message to the client refusing a lease.

**DHCPSRV\_ADDRESS6\_ALLOC\_FAIL failed to allocate an IPv6 address after %1 attempt(s)**

The DHCP allocation engine gave up trying to allocate an IPv6 address after the specified number of attempts. This probably means that the address pool from which the allocation is being attempted is either empty, or very nearly empty. As a result, the client will have been refused a lease.

This message may indicate that your address pool is too small for the number of clients you are trying to service and should be expanded. Alternatively, if you know that the number of concurrently active clients is less than the addresses you have available, you may want to consider reducing the lease lifetime. In this way, addresses allocated to clients that are no longer active on the network will become available sooner.

**DHCPSRV\_CFGMGR\_ADD\_IFACE listening on interface %1**

An info message issued when a new interface is being added to the collection of interfaces on which the server listens to DHCP messages.

**DHCPSRV\_CFGMGR\_ADD\_SUBNET4 adding subnet %1**

A debug message reported when the DHCP configuration manager is adding the specified IPv4 subnet to its database.

**DHCPSRV\_CFGMGR\_ADD\_SUBNET6 adding subnet %1**

A debug message reported when the DHCP configuration manager is adding the specified IPv6 subnet to its database.



**DHCPSRV\_CFGMGR\_ALL\_IFACES\_ACTIVE enabling listening on all interfaces**

A debug message issued when the server is being configured to listen on all interfaces.

**DHCPSRV\_CFGMGR\_CFG\_DHCP\_DDNS Setting DHCP-DDNS configuration to: %1**

A debug message issued when the server's DHCP-DDNS settings are changed.

**DHCPSRV\_CFGMGR\_CLEAR\_ACTIVE\_IFACES stop listening on all interfaces**

A debug message issued when configuration manager clears the internal list of active interfaces. This doesn't prevent the server from listening to the DHCP traffic through open sockets, but will rather be used by Interface Manager to select active interfaces when sockets are re-opened.

**DHCPSRV\_CFGMGR\_NO\_SUBNET4 no suitable subnet is defined for address hint %1**

This debug message is output when the DHCP configuration manager has received a request for an IPv4 subnet for the specified address, but no such subnet exists.

**DHCPSRV\_CFGMGR\_NO\_SUBNET6 no suitable subnet is defined for address hint %1**

This debug message is output when the DHCP configuration manager has received a request for an IPv6 subnet for the specified address, but no such subnet exists.

**DHCPSRV\_CFGMGR\_ONLY\_SUBNET4 retrieved subnet %1 for address hint %2**

This is a debug message reporting that the DHCP configuration manager has returned the specified IPv4 subnet when given the address hint specified because it is the only subnet defined.

**DHCPSRV\_CFGMGR\_ONLY\_SUBNET6 retrieved subnet %1 for address hint %2**

This is a debug message reporting that the DHCP configuration manager has returned the specified IPv6 subnet when given the address hint specified because it is the only subnet defined.

**DHCPSRV\_CFGMGR\_SOCKET\_RAW\_UNSUPPORTED use of raw sockets is unsupported on this OS, UDP sockets will be used**

This warning message is logged when the user specified that the DHCPv4 server should use the raw sockets to receive the DHCP messages and respond to the clients, but the use of raw sockets is not supported on the particular environment. The raw sockets are useful when the server must respond to the directly connected clients which don't have an address yet. If the raw sockets are not supported by Kea on the particular platform, Kea will fall back to use of the IP/UDP sockets. The responses to the directly connected clients will be broadcast. The responses to relayed clients will be unicast as usual.

**DHCPSRV\_CFGMGR\_SOCKET\_TYPE\_DEFAULT "dhcp-socket-type" not specified , using default socket type %1**

This informational message is logged when the administrator hasn't specified the "dhcp-socket-type" parameter in configuration for interfaces. In such case, the default socket type will be used.

**DHCPSRV\_CFGMGR\_SOCKET\_TYPE\_SELECT using socket type %1**

This informational message is logged when the DHCPv4 server selects the socket type to be used for all sockets that will be opened on the interfaces. Typically, the socket type is specified by the server administrator. If the socket type hasn't been specified, the raw socket will be selected. If the raw socket has been selected but Kea doesn't support the use of raw sockets on the particular OS, it will use an UDP socket instead.

**DHCPSRV\_CFGMGR\_SUBNET4 retrieved subnet %1 for address hint %2**

This is a debug message reporting that the DHCP configuration manager has returned the specified IPv4 subnet when given the address hint specified as the address is within the subnet.

**DHCPSRV\_CFGMGR\_SUBNET4\_RELAY selected subnet %1, because of matching relay addr %2**

This is a debug message reporting that the DHCP configuration manager has returned the specified IPv4 subnet, because detected relay agent address matches value specified for this subnet.

**DHCPSRV\_CFGMGR\_SUBNET6 retrieved subnet %1 for address hint %2**

This is a debug message reporting that the DHCP configuration manager has returned the specified IPv6 subnet when given the address hint specified as the address is within the subnet.

**DHCPSRV\_CFGMGR\_SUBNET6\_IFACE selected subnet %1 for packet received over interface %2**

This is a debug message reporting that the DHCP configuration manager has returned the specified IPv6 subnet for a packet received over given interface. This particular subnet was selected, because it was specified as being directly reachable over given interface. (see 'interface' parameter in the subnet6 definition).

**DHCPSRV\_CFGMGR\_SUBNET6\_IFACE\_ID selected subnet %1 (interface-id match) for incoming packet**

This is a debug message reporting that the DHCP configuration manager has returned the specified IPv6 subnet for a received packet. This particular subnet was selected, because value of interface-id option matched what was configured in the server's interface-id option for that selected subnet6. (see 'interface-id' parameter in the subnet6 definition).

**DHCPSRV\_CFGMGR\_SUBNET6\_RELAY selected subnet %1, because of matching relay addr %2**

This is a debug message reporting that the DHCP configuration manager has returned the specified IPv6 subnet, because detected relay agent address matches value specified for this subnet.

**DHCPSRV\_CFGMGR\_UNICAST\_LINK\_LOCAL specified link local address %1 for unicast traffic on interface %2**

This warning message is logged when user specified a link-local address to receive unicast traffic. The warning message is issued because it is an uncommon use.

**DHCPSRV\_CFGMGR\_USE\_ADDRESS listening on address %1, on interface %2**

A message issued when the server is configured to listen on the explicitly specified IP address on the given interface.

**DHCPSRV\_CFGMGR\_USE\_UNICAST listening on unicast address %1, on interface %2**

An info message issued when configuring the DHCP server to listen on the unicast address on the specific interface.

**DHCPSRV\_CLOSE\_DB closing currently open %1 database**

This is a debug message, issued when the DHCP server closes the currently open lease database. It is issued at program shutdown and whenever the database access parameters are changed: in the latter case, the server closes the currently open database, and opens a database using the new parameters.

**DHCPSRV\_DHCP\_DDNS\_ERROR\_EXCEPTION error handler for DHCP\_DDNS IO generated an expected exception: %1**

This is an error message that occurs when an attempt to send a request to kea-dhcp-ddns fails there registered error handler threw an uncaught exception. This is a programmatic error which should not occur. By convention, the error handler should not propagate exceptions. Please report this error.

**DHCPSRV\_DHCP\_DDNS\_HANDLER\_NULL error handler for DHCP\_DDNS IO is not set.**

This is an error message that occurs when an attempt to send a request to kea-dhcp-ddns fails and there is no registered error handler. This is a programmatic error which should never occur and should be reported.

**DHCPSRV\_DHCP\_DDNS\_NCR\_REJECTED NameChangeRequest rejected by the sender: %1, ncr: %2**

This is an error message indicating that NameChangeSender used to deliver DDNS update requests to kea-dhcp-ddns rejected the request. This most likely cause is the sender's queue has reached maximum capacity. This would imply that requests are being generated faster than they can be delivered.

**DHCPSRV\_DHCP\_DDNS\_NCR\_SENT NameChangeRequest sent to kea-dhcp-ddns: %1**

A debug message issued when a NameChangeRequest has been successfully sent to kea-dhcp-ddns.

**DHCPSRV\_DHCP\_DDNS\_SENDER\_STARTED NameChangeRequest sender has been started: %1**

A informational message issued when a communications with kea-dhcp-ddns has been successfully started.

**DHCPSRV\_DHCP\_DDNS\_SENDER\_STOPPED NameChangeRequest sender has been stopped.**

A informational message issued when a communications with kea-dhcp-ddns has been stopped. This normally occurs during reconfiguration and as part of normal shutdown. It may occur if kea-dhcp-ddns communications breakdown.

**DHCPSRV\_DHCP\_DDNS\_SUSPEND\_UPDATES DHCP\_DDNS updates are being suspended.**

This is a warning message indicating the DHCP\_DDNS updates have been turned off. This should only occur if IO errors communicating with kea-dhcp-ddns have been experienced. Any such errors should have preceding entries in the log with details. No further attempts to communicate with kea-dhcp-ddns will be made without intervention.

**DHCPSRV\_DISCOVER\_ADDRESS\_CONFLICT conflicting reservation for address %1 with existing lease %2**

This warning message is issued when the DHCP server finds that the address reserved for the client can't be offered because this address is currently allocated to another client. The server will try to allocate a different address to the client to use until the conflict is resolved.

**DHCPSRV\_HOOK\_LEASE4\_RENEW\_SKIP DHCPv4 lease was not renewed because a callout set the skip flag.**

This debug message is printed when a callout installed on lease4\_renew hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to not renew a lease. The server will use existing lease as it is, without extending its lifetime.

**DHCPSRV\_HOOK\_LEASE4\_SELECT\_SKIP Lease4 creation was skipped, because of callout skip flag.**

This debug message is printed when a callout installed on lease4\_select hook point sets the skip flag. It means that the server was told that no lease4 should be assigned. The server will not put that lease in its database and the client will get a NAK packet.

**DHCPSRV\_HOOK\_LEASE6\_EXTEND\_SKIP DHCPv6 lease lifetime was not extended because a callout set the skip flag for m**

This debug message is printed when a callout installed on lease6\_renew or the lease6\_rebind hook point set the skip flag. For this particular hook point, the setting of the flag by a callout instructs the server to not extend the lifetime for a lease. If the client requested renewal of multiple leases (by sending multiple IA options), the server will skip the renewal of the one in question and will proceed with other renewals as usual.

**DHCPSRV\_HOOK\_LEASE6\_SELECT\_SKIP Lease6 (non-temporary) creation was skipped, because of callout skip flag.**

This debug message is printed when a callout installed on lease6\_select hook point sets the skip flag. It means that the server was told that no lease6 should be assigned. The server will not put that lease in its database and the client will get a NoAddrAvail for that IA\_NA option.

**DHCPSRV\_HR\_RESERVED\_ADDR\_GRANTED reserved address %1 was assigned to client (duid=%2)**

This informational message signals that the specified client was assigned the address reserved for it.

**DHCPSRV\_HR\_RESERVED\_PREFIX\_GRANTED reserved prefix %1/%2 was assigned to client (duid=%3)**

This informational message signals that the specified client was assigned the prefix reserved for it.

**DHCPSRV\_HR\_REVOKED\_ADDR6\_LEASE address %1 was revoked from client %2 as it is reserved for client %3**

This informational message is an indication that the specified IPv6 address was used by client A but it is now reserved for client B. Client A has been told to stop using it so that it can be leased to client B. This is a normal occurrence during conflict resolution, which can occur in cases such as the system administrator adding a reservation for an address that is currently in use by another client. The server will fully recover from this situation, but clients will change their addresses.

**DHCPSRV\_HR\_REVOKED\_PREFIX6\_LEASE Prefix %1/%2 was revoked from client %3 as it is reserved for client %4**

This informational message is an indication that the specified IPv6 prefix was used by client A but it is now reserved for client B. Client A has been told to stop using it so that it can be leased to client B. This is a normal occurrence during conflict resolution, which can occur in cases such as the system administrator adding a reservation for an address that is currently in use by another client. The server will fully recover from this situation, but clients will change their prefixes.

**DHCPSRV\_INVALID\_ACCESS invalid database access string: %1**

This is logged when an attempt has been made to parse a database access string and the attempt ended in error. The access string in question - which should be of the form 'keyword=value keyword=value...' is included in the message.

**DHCPSRV\_MEMFILE\_ADD\_ADDR4 adding IPv4 lease with address %1**

A debug message issued when the server is about to add an IPv4 lease with the specified address to the memory file backend database.

**DHCPSRV\_MEMFILE\_ADD\_ADDR6 adding IPv6 lease with address %1**

A debug message issued when the server is about to add an IPv6 lease with the specified address to the memory file backend database.

**DHCPSRV\_MEMFILE\_COMMIT committing to memory file database**

The code has issued a commit call. For the memory file database, this is a no-op.

**DHCPSRV\_MEMFILE\_DB opening memory file lease database: %1**

This informational message is logged when a DHCP server (either V4 or V6) is about to open a memory file lease database. The parameters of the connection including database name and username needed to access it (but not the password if any) are logged.

**DHCPSRV\_MEMFILE\_DELETE\_ADDR deleting lease for address %1**

A debug message issued when the server is attempting to delete a lease for the specified address from the memory file database for the specified address.

**DHCPSRV\_MEMFILE\_GET\_ADDR4 obtaining IPv4 lease for address %1**

A debug message issued when the server is attempting to obtain an IPv4 lease from the memory file database for the specified address.

**DHCPSRV\_MEMFILE\_GET\_ADDR6 obtaining IPv6 lease for address %1 and lease type %2**

A debug message issued when the server is attempting to obtain an IPv6 lease from the memory file database for the specified address.

**DHCPSRV\_MEMFILE\_GET\_CLIENTID obtaining IPv4 leases for client ID %1**

A debug message issued when the server is attempting to obtain a set of IPv4 leases from the memory file database for a client with the specified client identification.

**DHCPSRV\_MEMFILE\_GET\_CLIENTID\_HWADDR\_SUBID obtaining IPv4 lease for client ID %1, hardware address %2 and subnet ID %3**

A debug message issued when the server is attempting to obtain an IPv4 lease from the memory file database for a client with the specified client ID, hardware address and subnet ID.

**DHCPSRV\_MEMFILE\_GET\_HWADDR obtaining IPv4 leases for hardware address %1**

A debug message issued when the server is attempting to obtain a set of IPv4 leases from the memory file database for a client with the specified hardware address.

**DHCPSRV\_MEMFILE\_GET\_IAID\_DUID obtaining IPv6 leases for IAID %1 and DUID %2 and lease type %3**

A debug message issued when the server is attempting to obtain a set of IPv6 lease from the memory file database for a client with the specified IAID (Identity Association ID) and DUID (DHCP Unique Identifier).

**DHCPSRV\_MEMFILE\_GET\_IAID\_SUBID\_DUID obtaining IPv6 leases for IAID %1, Subnet ID %2, DUID %3 and lease type %4**

A debug message issued when the server is attempting to obtain an IPv6 lease from the memory file database for a client with the specified IAID (Identity Association ID), Subnet ID and DUID (DHCP Unique Identifier).

**DHCPSRV\_MEMFILE\_GET\_SUBID\_CLIENTID obtaining IPv4 lease for subnet ID %1 and client ID %2**

A debug message issued when the server is attempting to obtain an IPv4 lease from the memory file database for a client with the specified subnet ID and client ID.

**DHCPSRV\_MEMFILE\_GET\_SUBID\_HWADDR obtaining IPv4 lease for subnet ID %1 and hardware address %2**

A debug message issued when the server is attempting to obtain an IPv4 lease from the memory file database for a client with the specified subnet ID and hardware address.

**DHCPSRV\_MEMFILE\_GET\_VERSION obtaining schema version information**

A debug message issued when the server is about to obtain schema version information from the memory file database.

**DHCPSRV\_MEMFILE\_LEASE\_FILE\_LOAD loading leases from file %1**

An info message issued when the server is about to start reading DHCP leases from the lease file. All leases currently held in the memory will be replaced by those read from the file.

**DHCPSRV\_MEMFILE\_LEASE\_LOAD loading lease %1**

A debug message issued when DHCP lease is being loaded from the file to memory.

**DHCPSRV\_MEMFILE\_LFC\_EXECUTE executing Lease File Cleanup using: %1**

An informational message issued when the Memfile lease database backend starts a new process to perform Lease File Cleanup.

**DHCPSRV\_MEMFILE\_LFC\_LEASE\_FILE\_RENAME\_FAIL failed to rename the current lease file %1 to %2, reason: %3**

An error message logged when the Memfile lease database backend fails to move the current lease file to a new file on which the cleanup should be performed. This effectively means that the lease file cleanup will not take place.

**DHCPSRV\_MEMFILE\_LFC\_LEASE\_FILE\_REOPEN\_FAIL failed to reopen lease file %1 after preparing input file for lease cleanup**

An error message logged when the Memfile lease database backend failed to re-open or re-create the lease file after renaming the lease file for lease file cleanup. The server will continue to operate but leases will not be persisted to disk.

**DHCPSRV\_MEMFILE\_LFC\_SETUP setting up the Lease File Cleanup interval to %1 sec**

An informational message logged when the Memfile lease database backend configures the LFC to be executed periodically. The argument holds the interval in seconds in which the LFC will be executed.

**DHCPSRV\_MEMFILE\_LFC\_SPAWN\_FAIL lease file cleanup failed to run because kea-lfc process couldn't be spawned**

This error message is logged when the the Kea server fails to run kea-lfc, the program that cleans up the lease file. The server will try again the next time a lease file cleanup is scheduled. Although this message should not appear and the reason why it did investigated, the occasional failure to start the lease file cleanup will not impact operations. Should the failure persist however, the size of the lease file will increase without bound.

**DHCPSRV\_MEMFILE\_LFC\_START starting Lease File Cleanup**

An informational message issued when the Memfile lease database backend starts the periodic Lease File Cleanup.

**DHCPSRV\_MEMFILE\_NO\_STORAGE running in non-persistent mode, leases will be lost after restart**

A warning message issued when writes of leases to disk have been disabled in the configuration. This mode is useful for some kinds of performance testing but should not be enabled in normal circumstances. Non-persistence mode is enabled when 'persist4=no persist6=no' parameters are specified in the database access string.

**DHCPSRV\_MEMFILE\_READ\_HWADDR\_FAIL failed to read hardware address from lease file: %1**

A warning message issued when read attempt of the hardware address stored in a disk file failed. The parameter should provide the exact nature of the failure. The database read will continue, but that particular lease will no longer have hardware address associated with it.

**DHCPSRV\_MEMFILE\_ROLLBACK rolling back memory file database**

The code has issued a rollback call. For the memory file database, this is a no-op.

**DHCPSRV\_MEMFILE\_UPDATE\_ADDR4 updating IPv4 lease for address %1**

A debug message issued when the server is attempting to update IPv4 lease from the memory file database for the specified address.

**DHCPSRV\_MEMFILE\_UPDATE\_ADDR6 updating IPv6 lease for address %1**

A debug message issued when the server is attempting to update IPv6 lease from the memory file database for the specified address.

**DHCPSRV\_MULTIPLE\_RAW\_SOCKETS\_PER\_IFACE current configuration will result in opening multiple broadcast capable**

A warning message issued when the current configuration indicates that multiple sockets, capable of receiving broadcast traffic, will be opened on some of the interfaces. It must be noted that this may lead to receiving and processing the same DHCP message multiple times, as it will be received by each socket individually.

**DHCPSRV\_MYSQL\_ADD\_ADDR4 adding IPv4 lease with address %1**

A debug message issued when the server is about to add an IPv4 lease with the specified address to the MySQL backend database.

**DHCPSRV\_MYSQL\_ADD\_ADDR6 adding IPv6 lease with address %1, lease type %2**

A debug message issued when the server is about to add an IPv6 lease with the specified address to the MySQL backend database.

**DHCPSRV\_MYSQL\_COMMIT committing to MySQL database**

The code has issued a commit call. All outstanding transactions will be committed to the database. Note that depending on the MySQL settings, the committal may not include a write to disk.

**DHCPSRV\_MYSQL\_DB opening MySQL lease database: %1**

This informational message is logged when a DHCP server (either V4 or V6) is about to open a MySQL lease database. The parameters of the connection including database name and username needed to access it (but not the password if any) are logged.

**DHCPSRV\_MYSQL\_DELETE\_ADDR deleting lease for address %1**

A debug message issued when the server is attempting to delete a lease for the specified address from the MySQL database for the specified address.

**DHCPSRV\_MYSQL\_GET\_ADDR4 obtaining IPv4 lease for address %1**

A debug message issued when the server is attempting to obtain an IPv4 lease from the MySQL database for the specified address.

**DHCPSRV\_MYSQL\_GET\_ADDR6 obtaining IPv6 lease for address %1, lease type %2**

A debug message issued when the server is attempting to obtain an IPv6 lease from the MySQL database for the specified address.

**DHCPSRV\_MYSQL\_GET\_CLIENTID obtaining IPv4 leases for client ID %1**

A debug message issued when the server is attempting to obtain a set of IPv4 leases from the MySQL database for a client with the specified client identification.

**DHCPSRV\_MYSQL\_GET\_HWADDR obtaining IPv4 leases for hardware address %1**

A debug message issued when the server is attempting to obtain a set of IPv4 leases from the MySQL database for a client with the specified hardware address.

**DHCPSRV\_MYSQL\_GET\_IAID\_DUID obtaining IPv6 leases for IAID %1, DUID %2, lease type %3**

A debug message issued when the server is attempting to obtain a set of IPv6 lease from the MySQL database for a client with the specified IAID (Identity Association ID) and DUID (DHCP Unique Identifier).

**DHCPSRV\_MYSQL\_GET\_IAID\_SUBID\_DUID obtaining IPv6 leases for IAID %1, Subnet ID %2, DUID %3, lease type %4**

A debug message issued when the server is attempting to obtain an IPv6 lease from the MySQL database for a client with the specified IAID (Identity Association ID), Subnet ID and DUID (DHCP Unique Identifier).

**DHCPSRV\_MYSQL\_GET\_SUBID\_CLIENTID obtaining IPv4 lease for subnet ID %1 and client ID %2**

A debug message issued when the server is attempting to obtain an IPv4 lease from the MySQL database for a client with the specified subnet ID and client ID.

**DHCPSRV\_MYSQL\_GET\_SUBID\_HWADDR obtaining IPv4 lease for subnet ID %1 and hardware address %2**

A debug message issued when the server is attempting to obtain an IPv4 lease from the MySQL database for a client with the specified subnet ID and hardware address.

**DHCPSRV\_MYSQL\_GET\_VERSION obtaining schema version information**

A debug message issued when the server is about to obtain schema version information from the MySQL database.

**DHCPSRV\_MYSQL\_ROLLBACK rolling back MySQL database**

The code has issued a rollback call. All outstanding transaction will be rolled back and not committed to the database.

**DHCPSRV\_MYSQL\_UPDATE\_ADDR4 updating IPv4 lease for address %1**

A debug message issued when the server is attempting to update IPv4 lease from the MySQL database for the specified address.

**DHCPSRV\_MYSQL\_UPDATE\_ADDR6 updating IPv6 lease for address %1, lease type %2**

A debug message issued when the server is attempting to update IPv6 lease from the MySQL database for the specified address.

**DHCPSRV\_NOTYPE\_DB no 'type' keyword to determine database backend: %1**

This is an error message, logged when an attempt has been made to access a database backend, but where no 'type' keyword has been included in the access string. The access string (less any passwords) is included in the message.

**DHCPSRV\_NO\_SOCKETS\_OPEN no interface configured to listen to DHCP traffic**

This warning message is issued when the current server configuration specifies no interfaces that the server should listen on, or when the specified interfaces are not configured to receive the traffic.

**DHCPSRV\_OPEN\_SOCKET\_FAIL failed to open socket: %1**

A warning message issued when IfaceMgr fails to open and bind a socket. The reason for the failure is appended as an argument of the log message.

**DHCPSRV\_PGSQL\_ADD\_ADDR4 adding IPv4 lease with address %1**

A debug message issued when the server is about to add an IPv4 lease with the specified address to the PostgreSQL backend database.

**DHCPSRV\_PGSQL\_ADD\_ADDR6 adding IPv6 lease with address %1**

A debug message issued when the server is about to add an IPv6 lease with the specified address to the PostgreSQL backend database.

**DHCPSRV\_PGSQL\_COMMIT committing to MySQL database**

The code has issued a commit call. All outstanding transactions will be committed to the database. Note that depending on the PostgreSQL settings, the committal may not include a write to disk.

**DHCPSRV\_PGSQL\_DB opening PostgreSQL lease database: %1**

This informational message is logged when a DHCP server (either V4 or V6) is about to open a PostgreSQL lease database. The parameters of the connection including database name and username needed to access it (but not the password if any) are logged.

**DHCPSRV\_PGSQL\_DEALLOC\_ERROR An error occurred deallocating SQL statements while closing the PostgreSQL lease**

This is an error message issued when a DHCP server (either V4 or V6) experienced an error freeing database SQL resources as part of closing its connection to the PostgreSQL database. The connection is closed as part of normal server shutdown. This error is most likely a programmatic issue that is highly unlikely to occur or negatively impact server operation.

**DHCPSRV\_PGSQL\_DELETE\_ADDR deleting lease for address %1**

A debug message issued when the server is attempting to delete a lease for the specified address from the PostgreSQL database for the specified address.

**DHCPSRV\_PGSQL\_GET\_ADDR4 obtaining IPv4 lease for address %1**

A debug message issued when the server is attempting to obtain an IPv4 lease from the PostgreSQL database for the specified address.

**DHCPSRV\_PGSQL\_GET\_ADDR6 obtaining IPv6 lease for address %1 (lease type %2)**

A debug message issued when the server is attempting to obtain an IPv6 lease from the PostgreSQL database for the specified address.

**DHCPSRV\_PGSQL\_GET\_CLIENTID obtaining IPv4 leases for client ID %1**

A debug message issued when the server is attempting to obtain a set of IPv4 leases from the PostgreSQL database for a client with the specified client identification.

**DHCPSRV\_PGSQL\_GET\_HWADDR obtaining IPv4 leases for hardware address %1**

A debug message issued when the server is attempting to obtain a set of IPv4 leases from the PostgreSQL database for a client with the specified hardware address.

**DHCPSRV\_PGSQL\_GET\_IAID\_DUID obtaining IPv4 leases for IAID %1 and DUID %2, lease type %3**

A debug message issued when the server is attempting to obtain a set of IPv6 lease from the PostgreSQL database for a client with the specified IAID (Identity Association ID) and DUID (DHCP Unique Identifier).

**DHCPSRV\_PGSQL\_GET\_IAID\_SUBID\_DUID obtaining IPv4 leases for IAID %1, Subnet ID %2, DUID %3, and lease type %4**

A debug message issued when the server is attempting to obtain an IPv6 lease from the PostgreSQL database for a client with the specified IAID (Identity Association ID), Subnet ID and DUID (DHCP Unique Identifier).

**DHCPSRV\_PGSQL\_GET\_SUBID\_CLIENTID obtaining IPv4 lease for subnet ID %1 and client ID %2**

A debug message issued when the server is attempting to obtain an IPv4 lease from the PostgreSQL database for a client with the specified subnet ID and client ID.

**DHCPSRV\_PGSQL\_GET\_SUBID\_HWADDR obtaining IPv4 lease for subnet ID %1 and hardware address %2**

A debug message issued when the server is attempting to obtain an IPv4 lease from the PostgreSQL database for a client with the specified subnet ID and hardware address.

**DHCPSRV\_PGSQL\_GET\_VERSION obtaining schema version information**

A debug message issued when the server is about to obtain schema version information from the PostgreSQL database.

**DHCPSRV\_PGSQL\_ROLLBACK rolling back PostgreSQL database**

The code has issued a rollback call. All outstanding transaction will be rolled back and not committed to the database.

**DHCPSRV\_PGSQL\_UPDATE\_ADDR4 updating IPv4 lease for address %1**

A debug message issued when the server is attempting to update IPv4 lease from the PostgreSQL database for the specified address.

**DHCPSRV\_PGSQL\_UPDATE\_ADDR6 updating IPv6 lease for address %1**

A debug message issued when the server is attempting to update IPv6 lease from the PostgreSQL database for the specified address.

**DHCPSRV\_RENEW6\_ERROR allocation engine experienced error with attempting to renew a lease: %1**

This error message indicates that an error was experienced during Renew or Rebind processing. Additional explanation is provided with this message. Depending on its nature, manual intervention may be required to continue processing messages from this particular client; other clients will be unaffected.

**DHCPSRV\_UNEXPECTED\_NAME database access parameters passed through '%1', expected 'lease-database'**

The parameters for access the lease database were passed to the server through the named configuration parameter, but the code was expecting them to be passed via the parameter named "lease-database". If the database opens successfully, there is no impact on server operation. However, as this does indicate an error in the source code, please submit a bug report.

**DHCPSRV\_UNKNOWN\_DB unknown database type: %1**

The database access string specified a database type (given in the message) that is unknown to the software. This is a configuration error.

**DHCP\_DDNS\_ADD\_FAILED DHCP\_DDNS Transaction outcome: %1**

This is an error message issued after DHCP\_DDNS attempts to submit DNS mapping entry additions have failed. The precise reason for the failure should be documented in preceding log entries.

**DHCP\_DDNS\_ADD\_SUCCEEDED DHCP\_DDNS successfully added the DNS mapping addition for this request: %1**

This is an informational message issued after DHCP\_DDNS has submitted DNS mapping additions which were received and accepted by an appropriate DNS server.

**DHCP\_DDNS\_AT\_MAX\_TRANSACTIONS application has %1 queued requests but has reached maximum number of %2 concurrent**

This is a debug message that indicates that the application has DHCP\_DDNS requests in the queue but is working as many concurrent requests as allowed.

**DHCP\_DDNS\_CFG\_FILE\_RELOAD\_ERROR configuration reload failed: %1, reverting to current configuration.**

This is an error message indicating that the application attempted to reload its configuration from file and encountered an error. This is likely due to invalid content in the configuration file. The application should continue to operate under its current configuration.

**DHCP\_DDNS\_CFG\_FILE\_RELOAD\_SIGNAL\_RECVD OS signal %1 received, reloading configuration from file: %2**

This is an informational message indicating the application has received a signal instructing it to reload its configuration from file.

**DHCP\_DDNS\_CLEARED\_FOR\_SHUTDOWN application has met shutdown criteria for shutdown type: %1**

This is a debug message issued when the application has been instructed to shutdown and has met the required criteria to exit.

**DHCP\_DDNS\_COMMAND command directive received, command: %1 - args: %2**

This is a debug message issued when the DHCP-DDNS application command method has been invoked.

**DHCP\_DDNS\_CONFIGURE configuration update received: %1**

This is a debug message issued when the DHCP-DDNS application configure method has been invoked.

**DHCP\_DDNS\_FAILED application experienced a fatal error: %1**

This is a debug message issued when the DHCP-DDNS application encounters an unrecoverable error from within the event loop.

**DHCP\_DDNS\_FORWARD\_ADD\_BAD\_DNSCLIENT\_STATUS DHCP\_DDNS received an unknown DNSClient status: %1, w**

This is an error message issued when DNSClient returns an unrecognized status while DHCP\_DDNS was adding a forward address mapping. The request will be aborted. This is most likely a programmatic issue and should be reported.



**DHCP\_DDNS\_FORWARD\_ADD\_BUILD\_FAILURE DNS update message to add a forward DNS entry could not be constructed**

This is an error message issued when an error occurs attempting to construct the server bound packet requesting a forward address addition. This is due to invalid data contained in the NameChangeRequest. The request will be aborted. This is most likely a configuration issue.

**DHCP\_DDNS\_FORWARD\_ADD\_IO\_ERROR DHCP\_DDNS encountered an IO error sending a forward mapping add for FQDN**

This is an error message issued when a communication error occurs while DHCP\_DDNS is carrying out a forward address update. The application will retry against the same server or others as appropriate.

**DHCP\_DDNS\_FORWARD\_ADD\_REJECTED DNS Server, %1, rejected a DNS update request to add the address mapping for FQDN**

This is an error message issued when an update was rejected by the DNS server it was sent to for the reason given by the RCODE. The rcode values are defined in RFC 2136.

**DHCP\_DDNS\_FORWARD\_ADD\_RESP\_CORRUPT DHCP\_DDNS received a corrupt response from the DNS server, %1, while adding forward address mapping**

This is an error message issued when the response received by DHCP\_DDNS, to a update request to add a forward address mapping, is mangled or malformed. The application will retry against the same server or others as appropriate.

**DHCP\_DDNS\_FORWARD\_REMOVE\_ADDRS\_BAD\_DNSCLIENT\_STATUS DHCP\_DDNS received an unknown DNSClient status**

This is an error message issued when DNSClient returns an unrecognized status while DHCP\_DDNS was removing a forward address mapping. The request will be aborted. This is most likely a programmatic issue and should be reported.

**DHCP\_DDNS\_FORWARD\_REMOVE\_ADDRS\_BUILD\_FAILURE DNS update message to remove a forward DNS Address entry could not be constructed**

This is an error message issued when an error occurs attempting to construct the server bound packet requesting a forward address (A or AAAA) removal. This is due to invalid data contained in the NameChangeRequest. The request will be aborted. This is most likely a configuration issue.

**DHCP\_DDNS\_FORWARD\_REMOVE\_ADDRS\_IO\_ERROR DHCP\_DDNS encountered an IO error sending a forward mapping remove**

This is an error message issued when a communication error occurs while DHCP\_DDNS is carrying out a forward address remove. The application will retry against the same server or others as appropriate.

**DHCP\_DDNS\_FORWARD\_REMOVE\_ADDRS\_REJECTED DNS Server, %1, rejected a DNS update request to remove the forward address mapping**

This is an error message issued when an update was rejected by the DNS server it was sent to for the reason given by the RCODE. The rcode values are defined in RFC 2136.

**DHCP\_DDNS\_FORWARD\_REMOVE\_ADDRS\_RESP\_CORRUPT DHCP\_DDNS received a corrupt response from the DNS server, %1, while removing forward address mapping**

This is an error message issued when the response received by DHCP\_DDNS, to a update request to remove a forward address mapping, is mangled or malformed. The application will retry against the same server or others as appropriate.

**DHCP\_DDNS\_FORWARD\_REMOVE\_RRS\_BAD\_DNSCLIENT\_STATUS DHCP\_DDNS received an unknown DNSClient status**

This is an error message issued when DNSClient returns an unrecognized status while DHCP\_DDNS was removing forward RRs. The request will be aborted. This is most likely a programmatic issue and should be reported.

**DHCP\_DDNS\_FORWARD\_REMOVE\_RRS\_BUILD\_FAILURE DNS update message to remove forward DNS RR entries could not be constructed**

This is an error message issued when an error occurs attempting to construct the server bound packet requesting forward RR (DHCID RR) removal. This is due to invalid data contained in the NameChangeRequest. The request will be aborted. This is most likely a configuration issue.

**DHCP\_DDNS\_FORWARD\_REMOVE\_RRS\_IO\_ERROR DHCP\_DDNS encountered an IO error sending a forward RR remove**

This is an error message issued when a communication error occurs while DHCP\_DDNS is carrying out a forward RR remove. The application will retry against the same server.

**DHCP\_DDNS\_FORWARD\_REMOVE\_RRS\_REJECTED DNS Server, %1, rejected a DNS update request to remove forward**

This is an error message issued when an update was rejected by the DNS server it was sent to for the reason given by the RCODE. The rcode values are defined in RFC 2136.

**DHCP\_DDNS\_FORWARD\_REMOVE\_RRS\_RESP\_CORRUPT DHCP\_DDNS received a corrupt response from the DNS server**

This is an error message issued when the response received by DHCP\_DDNS, to a update request to remove forward RRs mapping, is mangled or malformed. The application will retry against the same server or others as appropriate.

**DHCP\_DDNS\_FORWARD\_REPLACE\_BAD\_DNSCLIENT\_STATUS DHCP\_DDNS received an unknown DNSClient status: %1**

This is an error message issued when DNSClient returns an unrecognized status while DHCP\_DDNS was replacing a forward address mapping. The request will be aborted. This is most likely a programmatic issue and should be reported.

**DHCP\_DDNS\_FORWARD\_REPLACE\_BUILD\_FAILURE DNS update message to replace a forward DNS entry could not be built**

This is an error message issued when an error occurs attempting to construct the server bound packet requesting a forward address replacement. This is due to invalid data contained in the NameChangeRequest. The request will be aborted. This is most likely a configuration issue.

**DHCP\_DDNS\_FORWARD\_REPLACE\_IO\_ERROR DHCP\_DDNS encountered an IO error sending a forward mapping replacement**

This is an error message issued when a communication error occurs while DHCP\_DDNS is carrying out a forward address update. The application will retry against the same server or others as appropriate.

**DHCP\_DDNS\_FORWARD\_REPLACE\_REJECTED DNS Server, %1, rejected a DNS update request to replace the address mapping**

This is an error message issued when an update was rejected by the DNS server it was sent to for the reason given by the RCODE. The rcode values are defined in RFC 2136.

**DHCP\_DDNS\_FORWARD\_REPLACE\_RESP\_CORRUPT DHCP\_DDNS received a corrupt response from the DNS server, %1**

This is an error message issued when the response received by DHCP\_DDNS, to a update request to replace a forward address mapping, is mangled or malformed. The application will retry against the same server or others as appropriate.

**DHCP\_DDNS\_FWD\_REQUEST\_IGNORED Forward updates are disabled, the forward portion of request will be ignored: %1**

This is a debug message issued when forward DNS updates are disabled and DHCP\_DDNS receives an update request containing a forward DNS update. The forward update will not be performed.

**DHCP\_DDNS\_INVALID\_NCR application received an invalid DNS update request: %1**

This is an error message that indicates that an invalid request to update a DNS entry was received by the application. Either the format or the content of the request is incorrect. The request will be ignored.

**DHCP\_DDNS\_INVALID\_RESPONSE received response to DNS Update message is malformed: %1**

This is a debug message issued when the DHCP-DDNS application encountered an error while decoding a response to a DNS Update message. Typically, this error will be encountered when a response message is malformed.

**DHCP\_DDNS\_NCR\_FLUSH\_IO\_ERROR DHCP-DDNS Last send before stopping did not complete successfully: %1**

This is an error message that indicates the DHCP-DDNS client was unable to complete the last send prior to exiting send mode. This is a programmatic error, highly unlikely to occur, and should not impair the application's ability to process requests.

**DHCP\_DDNS\_NCR\_LISTEN\_CLOSE\_ERROR application encountered an error while closing the listener used to receive NameChangeRequests**

This is an error message that indicates the application was unable to close the listener connection used to receive NameChangeRequests. Closure may occur during the course of error recovery or during normal shutdown procedure. In either case the error is unlikely to impair the application's ability to process requests but it should be reported for analysis.

**DHCP\_DDNS\_NCR\_RECV\_NEXT\_ERROR application could not initiate the next read following a request receive.**

This is a error message indicating that NameChangeRequest listener could not start another read after receiving a request. While possible, this is highly unlikely and is probably a programmatic error. The application should recover on its own.

**DHCP\_DDNS\_NCR\_SEND\_CLOSE\_ERROR DHCP-DDNS client encountered an error while closing the sender connection**

This is an error message that indicates the DHCP-DDNS client was unable to close the connection used to send NameChangeRequests. Closure may occur during the course of error recovery or during normal shutdown procedure. In either case the error is unlikely to impair the client's ability to send requests but it should be reported for analysis.

**DHCP\_DDNS\_NCR\_SEND\_NEXT\_ERROR DHCP-DDNS client could not initiate the next request send following send complete**

This is a error message indicating that NameChangeRequest sender could not start another send after completing the send of the previous request. While possible, this is highly unlikely and is probably a programmatic error. The application should recover on its own.

**DHCP\_DDNS\_NCR\_UDP\_CLEAR\_READY\_ERROR NCR UDP watch socket failed to clear: %1**

This is an error message that indicates the application was unable to reset the UDP NCR sender ready status after completing a send. This is programmatic error that should be reported. The application may or may not continue to operate correctly.

**DHCP\_DDNS\_NCR\_UDP\_RECV\_CANCELED UDP socket receive was canceled while listening for DNS Update requests**

This is a debug message indicating that the listening on a UDP socket for DNS update requests has been canceled. This is a normal part of suspending listening operations.

**DHCP\_DDNS\_NCR\_UDP\_RECV\_ERROR UDP socket receive error while listening for DNS Update requests: %1**

This is an error message indicating that an I/O error occurred while listening over a UDP socket for DNS update requests. This could indicate a network connectivity or system resource issue.

**DHCP\_DDNS\_NCR\_UDP\_SEND\_CANCELED UDP socket send was canceled while sending a DNS Update request to DHCP**

This is an informational message indicating that sending requests via UDP socket to DHCP\_DDNS has been interrupted. This is a normal part of suspending send operations.

**DHCP\_DDNS\_NCR\_UDP\_SEND\_ERROR UDP socket send error while sending a DNS Update request: %1**

This is an error message indicating that an IO error occurred while sending a DNS update request to DHCP\_DDNS over a UDP socket. This could indicate a network connectivity or system resource issue.

**DHCP\_DDNS\_NOT\_ON\_LOOPBACK the DHCP-DDNS server has been configured to listen on %1 which is not the local loopback**

This is a warning message issued when the DHCP-DDNS server is configured to listen at an address other than the loopback address (127.0.0.1 or ::1). It is possible for a malicious attacker to send bogus NameChangeRequests to it and change entries in the DNS. For this reason, addresses other than the IPv4 or IPv6 loopback addresses should only be used for testing purposes. A future version of Kea will implement authentication to guard against such attacks.

**DHCP\_DDNS\_NO\_ELIGIBLE\_JOBS although there are queued requests, there are pending transactions for each Queue count**

This is a debug message issued when all of the queued requests represent clients for which there is a an update already in progress. This may occur under normal operations but should be temporary situation.

**DHCP\_DDNS\_NO\_FWD\_MATCH\_ERROR the configured list of forward DDNS domains does not contain a match for: %1 T**

This is an error message that indicates that DHCP\_DDNS received a request to update a the forward DNS information for the given FQDN but for which there are no configured DDNS domains in the DHCP\_DDNS configuration. Either the DHCP\_DDNS configuration needs to be updated or the source of the FQDN itself should be investigated.

**DHCP\_DDNS\_NO\_MATCH No DNS servers match FQDN %1**

This is warning message issued when there are no domains in the configuration which match the cited fully qualified domain name (FQDN). The DNS Update request for the FQDN cannot be processed.

**DHCP\_DDNS\_NO\_REV\_MATCH\_ERROR** the configured list of reverse DDNS domains does not contain a match for: %1 Th

This is an error message that indicates that DHCP\_DDNS received a request to update a the reverse DNS information for the given FQDN but for which there are no configured DDNS domains in the DHCP\_DDNS configuration. Either the DHCP\_DDNS configuration needs to be updated or the source of the FQDN itself should be investigated.

**DHCP\_DDNS\_PROCESS\_INIT** application init invoked

This is a debug message issued when the DHCP-DDNS application enters its initialization method.

**DHCP\_DDNS\_QUEUE\_MGR\_QUEUE\_FULL** application request queue has reached maximum number of entries %1

This an error message indicating that DHCP-DDNS is receiving DNS update requests faster than they can be processed. This may mean the maximum queue needs to be increased, the DHCP-DDNS clients are simply generating too many requests too quickly, or perhaps upstream DNS servers are experiencing load issues.

**DHCP\_DDNS\_QUEUE\_MGR\_RECONFIGURING** application is reconfiguring the queue manager

This is an informational message indicating that DHCP\_DDNS is reconfiguring the queue manager as part of normal startup or in response to a new configuration.

**DHCP\_DDNS\_QUEUE\_MGR\_RECOVERING** application is attempting to recover from a queue manager IO error

This is an informational message indicating that DHCP\_DDNS is attempting to restart the queue manager after it suffered an IO error while receiving requests.

**DHCP\_DDNS\_QUEUE\_MGR\_RECV\_ERROR** application's queue manager was notified of a request receive error by its listener

This is an error message indicating that the NameChangeRequest listener used by DHCP-DDNS to receive requests encountered a IO error. There should be corresponding log messages from the listener layer with more details. This may indicate a network connectivity or system resource issue.

**DHCP\_DDNS\_QUEUE\_MGR\_RESUME\_ERROR** application could not restart the queue manager, reason: %1

This is an error message indicating that DHCP\_DDNS's Queue Manager could not be restarted after stopping due to a full receive queue. This means that the application cannot receive requests. This is most likely due to DHCP\_DDNS configuration parameters referring to resources such as an IP address or port, that is no longer unavailable. DHCP\_DDNS will attempt to restart the queue manager if given a new configuration.

**DHCP\_DDNS\_QUEUE\_MGR\_RESUMING** application is resuming listening for requests now that the request queue size has

This is an informational message indicating that DHCP\_DDNS, which had stopped accepting new requests, has processed enough entries from the receive queue to resume accepting requests.

**DHCP\_DDNS\_QUEUE\_MGR\_STARTED** application's queue manager has begun listening for requests.

This is a debug message indicating that DHCP\_DDNS's Queue Manager has successfully started and is now listening for NameChangeRequests.

**DHCP\_DDNS\_QUEUE\_MGR\_START\_ERROR** application could not start the queue manager, reason: %1

This is an error message indicating that DHCP\_DDNS's Queue Manager could not be started. This means that the application cannot receive requests. This is most likely due to DHCP\_DDNS configuration parameters referring to resources such as an IP address or port, that are unavailable. DHCP\_DDNS will attempt to restart the queue manager if given a new configuration.

**DHCP\_DDNS\_QUEUE\_MGR\_STOPPED** application's queue manager has stopped listening for requests.

This is a debug message indicating that DHCP\_DDNS's Queue Manager has stopped listening for NameChangeRequests. This may be because of normal event such as reconfiguration or as a result of an error. There should be log messages preceding this one to indicate why it has stopped.

**DHCP\_DDNS\_QUEUE\_MGR\_STOPPING** application is stopping the queue manager for %1

This is an informational message indicating that DHCP\_DDNS is stopping the queue manager either to reconfigure it or as part of application shutdown.

**DHCP\_DDNS\_QUEUE\_MGR\_STOP\_ERROR** application encountered an error stopping the queue manager: %1

This is an error message indicating that DHCP\_DDNS encountered an error while trying to stop the queue manager. This error is unlikely to occur or to impair the application's ability to function but it should be reported for analysis.

**DHCP\_DDNS\_QUEUE\_MGR\_UNEXPECTED\_HANDLER\_ERROR application's queue manager request receive handler error**

This is an error message indicating that an unexpected error occurred within the DHCP\_DDNS's Queue Manager request receive completion handler. This is most likely a programmatic issue that should be reported. The application may recover on its own.

**DHCP\_DDNS\_QUEUE\_MGR\_UNEXPECTED\_STOP application's queue manager receive was**

aborted unexpectedly while queue manager state is: %1 This is an error message indicating that DHCP\_DDNS's Queue Manager request receive was unexpectedly interrupted. Normally, the read is receive is only interrupted as a normal part of stopping the queue manager. This is most likely a programmatic issue that should be reported.

**DHCP\_DDNS\_REMOVE\_FAILED DHCP\_DDNS Transaction outcome: %1**

This is an error message issued after DHCP\_DDNS attempts to submit DNS mapping entry removals have failed. The precise reason for the failure should be documented in preceding log entries.

**DHCP\_DDNS\_REMOVE\_SUCCEEDED DHCP\_DDNS successfully removed the DNS mapping addition for this request: %1**

This is an informational message issued after DHCP\_DDNS has submitted DNS mapping removals which were received and accepted by an appropriate DNS server.

**DHCP\_DDNS\_REQUEST\_DROPPED Request contains no enabled update requests and will be dropped: %1**

This is a debug message issued when DHCP\_DDNS receives a request which does not contain updates in a direction that is enabled. In other words, if only forward updates are enabled and request is received that asks only for reverse updates then the request is dropped.

**DHCP\_DDNS\_REVERSE\_REMOVE\_BAD\_DNSCLIENT\_STATUS DHCP\_DDNS received an unknown DNSClient status: %1**

This is an error message issued when DNSClient returns an unrecognized status while DHCP\_DDNS was removing a reverse address mapping. The request will be aborted. This is most likely a programmatic issue and should be reported.

**DHCP\_DDNS\_REVERSE\_REMOVE\_BUILD\_FAILURE DNS update message to remove a reverse DNS entry could not be constructed**

This is an error message issued when an error occurs attempting to construct the server bound packet requesting a reverse PTR removal. This is due to invalid data contained in the NameChangeRequest. The request will be aborted. This is most likely a configuration issue.

**DHCP\_DDNS\_REVERSE\_REMOVE\_IO\_ERROR DHCP\_DDNS encountered an IO error sending a reverse mapping removal**

This is an error message issued when a communication error occurs while DHCP\_DDNS is carrying out a reverse address update. The application will retry against the same server or others as appropriate.

**DHCP\_DDNS\_REVERSE\_REMOVE\_REJECTED DNS Server, %1, rejected a DNS update request to remove the reverse mapping**

This is an error message issued when an update was rejected by the DNS server it was sent to for the reason given by the RCODE. The rcode values are defined in RFC 2136.

**DHCP\_DDNS\_REVERSE\_REMOVE\_RESP\_CORRUPT DHCP\_DDNS received a corrupt response from the DNS server, %1**

This is an error message issued when the response received by DHCP\_DDNS, to a update request to remove a reverse address, is mangled or malformed. The application will retry against the same server or others as appropriate.

**DHCP\_DDNS\_REVERSE\_REPLACE\_BAD\_DNSCLIENT\_STATUS DHCP\_DDNS received an unknown DNSClient status: %1**

This is an error message issued when DNSClient returns an unrecognized status while DHCP\_DDNS was replacing a reverse address mapping. The request will be aborted. This is most likely a programmatic issue and should be reported.

**DHCP\_DDNS\_REVERSE\_REPLACE\_BUILD\_FAILURE DNS update message to replace a reverse DNS entry could not be constructed**

This is an error message issued when an error occurs attempting to construct the server bound packet requesting a reverse PTR replacement. This is due to invalid data contained in the NameChangeRequest. The request will be aborted. This is most likely a configuration issue.

---

**DHCP\_DDNS\_REVERSE\_REPLACE\_IO\_ERROR DHCP\_DDNS encountered an IO error sending a reverse mapping replace**

This is an error message issued when a communication error occurs while DHCP\_DDNS is carrying out a reverse address update. The application will retry against the same server or others as appropriate.

**DHCP\_DDNS\_REVERSE\_REPLACE\_REJECTED DNS Server, %1, rejected a DNS update request to replace the reverse ma**

This is an error message issued when an update was rejected by the DNS server it was sent to for the reason given by the RCODE. The rcode values are defined in RFC 2136.

**DHCP\_DDNS\_REVERSE\_REPLACE\_RESP\_CORRUPT DHCP\_DDNS received a corrupt response from the DNS server, %1**

This is an error message issued when the response received by DHCP\_DDNS, to a update request to replace a reverse address, is mangled or malformed. The application will retry against the same server or others as appropriate.

**DHCP\_DDNS\_REV\_REQUEST\_IGNORED Reverse updates are disabled, the reverse portion of request will be ignored: %1**

This is a debug message issued when reverse DNS updates are disabled and DHCP\_DDNS receives an update request containing a reverse DNS update. The reverse update will not be performed.

**DHCP\_DDNS\_RUN\_EXIT application is exiting the event loop**

This is a debug message issued when the DHCP-DDNS server exits its event loop.

**DHCP\_DDNS\_SHUTDOWN DHCP-DDNS has shut down**

This is an informational message indicating that the DHCP-DDNS service has shut down.

**DHCP\_DDNS\_SHUTDOWN\_COMMAND application received shutdown command with args: %1**

This is a debug message issued when the application has been instructed to shut down by the controller.

**DHCP\_DDNS\_SHUTDOWN\_SIGNAL\_RECVD OS signal %1 received, starting shutdown**

This is a debug message indicating the application has received a signal instructing it to shutdown.

**DHCP\_DDNS\_SIGNAL\_ERROR signal handler for signal %1, threw an unexpected exception: %2**

This is an error message indicating that the application encountered an unexpected error after receiving a signal. This is a programmatic error and should be reported. While the application will likely continue to operating, it may be unable to respond correctly to signals.

**DHCP\_DDNS\_STARTED Kea DHCP-DDNS server version %1 started**

This informational message indicates that the DHCP-DDNS server has processed all configuration information and is ready to begin processing. The version is also printed.

**DHCP\_DDNS\_STARTING DHCP-DDNS starting, pid: %1, version: %2**

This is an informational message issued when controller for the service first starts. Version is also reported.

**DHCP\_DDNS\_STARTING\_TRANSACTION Transaction Key: %1**

This is a debug message issued when DHCP-DDNS has begun a transaction for a given request.

**DHCP\_DDNS\_STATE\_MODEL\_UNEXPECTED\_ERROR application encountered an unexpected error while carrying out a**

This is error message issued when the application fails to process a NameChangeRequest correctly. Some or all of the DNS updates requested as part of this update did not succeed. This is a programmatic error and should be reported.

**DHCP\_DDNS\_TRANS\_SEND\_ERROR application encountered an unexpected error while attempting to send a DNS update:**

This is error message issued when the application is able to construct an update message but the attempt to send it suffered an unexpected error. This is most likely a programmatic error, rather than a communications issue. Some or all of the DNS updates requested as part of this request did not succeed.

**DHCP\_DDNS\_UNCAUGHT\_NCR\_RECV\_HANDLER\_ERROR unexpected exception thrown from the application receive co**

This is an error message that indicates that an exception was thrown but not caught in the application's request receive completion handler. This is a programmatic error that needs to be reported. Dependent upon the nature of the error the application may or may not continue operating normally.

**DHCP\_DDNS\_UNCAUGHT\_NCR\_SEND\_HANDLER\_ERROR unexpected exception thrown from the DHCP-DDNS client send handler**

This is an error message that indicates that an exception was thrown but not caught in the application's send completion handler. This is a programmatic error that needs to be reported. Dependent upon the nature of the error the client may or may not continue operating normally.

**DHCP\_DDNS\_UNSUPPORTED\_SIGNAL ignoring reception of unsupported signal: %1**

This is a debug message indicating that the application received an unsupported signal. This is a programming error indicating that the application has registered to receive the signal but no associated processing logic has been added.

**DHCP\_DDNS\_UPDATE\_REQUEST\_SENT %1 for transaction key: %2 to server: %3**

This is a debug message issued when DHCP\_DDNS sends a DNS request to a DNS server.

**DHCP\_DDNS\_UPDATE\_RESPONSE\_RECEIVED for transaction key: %1 to server: %2 status: %3**

This is a debug message issued when DHCP\_DDNS receives sends a DNS update response from a DNS server.

**DHCP\_DDNS\_WATCH\_SINK\_CLOSE\_ERROR Sink-side watch socket failed to close: %1**

This is an error message that indicates the application was unable to close the inbound side of a NCR sender's watch socket. While technically possible this error is highly unlikely to occur and should not impair the application's ability to process requests.

**DHCP\_DDNS\_WATCH\_SOURCE\_CLOSE\_ERROR Source-side watch socket failed to close: %1**

This is an error message that indicates the application was unable to close the outbound side of a NCR sender's watch socket. While technically possible this error is highly unlikely to occur and should not impair the application's ability to process requests.

**HOOKS\_ALL\_CALLOUTS\_DEREGISTERED hook library at index %1 removed all callouts on hook %2**

A debug message issued when all callouts on the specified hook registered by the library with the given index were removed. This is similar to the HOOKS\_CALLOUTS\_REMOVED message (and the two are likely to be seen together), but is issued at a lower-level in the hook framework.

**HOOKS\_CALLOUTS\_REMOVED callouts removed from hook %1 for library %2**

This is a debug message issued during library unloading. It notes that one or more callouts registered by that library have been removed from the specified hook. This is similar to the HOOKS\_DEREGISTER\_ALL\_CALLOUTS message (and the two are likely to be seen together), but is issued at a higher-level in the hook framework.

**HOOKS\_CALLOUT\_CALLED hooks library with index %1 has called a callout on hook %2 that has address %3**

Only output at a high debugging level, this message indicates that a callout on the named hook registered by the library with the given index (in the list of loaded libraries) has been called and returned a success state. The address of the callout is given in the message

**HOOKS\_CALLOUT\_DEREGISTERED hook library at index %1 deregistered a callout on hook %2**

A debug message issued when all instances of a particular callouts on the hook identified in the message that were registered by the library with the given index have been removed.

**HOOKS\_CALLOUT\_ERROR error returned by callout on hook %1 registered by library with index %2 (callout address %3)**

If a callout returns an error status when called, this error message is issued. It identifies the hook to which the callout is attached, the index of the library (in the list of loaded libraries) that registered it and the address of the callout. The error is otherwise ignored.

**HOOKS\_CALLOUT\_EXCEPTION exception thrown by callout on hook %1 registered by library with index %2 (callout address %3)**

If a callout throws an exception when called, this error message is issued. It identifies the hook to which the callout is attached, the index of the library (in the list of loaded libraries) that registered it and the address of the callout. The error is otherwise ignored.

**HOOKS\_CALLOUT\_REGISTRATION hooks library with index %1 registering callout for hook %2**

This is a debug message, output when a library (whose index in the list of libraries (being) loaded is given) registers a callout.

**HOOKS\_CLOSE\_ERROR failed to close hook library %1: %2**

Kea has failed to close the named hook library for the stated reason. Although this is an error, this should not affect the running system other than as a loss of resources. If this error persists, you should restart Kea.

**HOOKS\_HOOK\_LIST\_RESET the list of hooks has been reset**

This is a message indicating that the list of hooks has been reset. While this is usual when running the Kea test suite, it should not be seen when running Kea in a production environment. If this appears, please report a bug through the usual channels.

**HOOKS\_INCORRECT\_VERSION hook library %1 is at version %2, require version %3**

Kea has detected that the named hook library has been built against a version of Kea that is incompatible with the version of Kea running on your system. It has not loaded the library.

This is most likely due to the installation of a new version of Kea without rebuilding the hook library. A rebuild and re-install of the library should fix the problem in most cases.

**HOOKS\_LIBRARY\_LOADED hooks library %1 successfully loaded**

This information message is issued when a user-supplied hooks library has been successfully loaded.

**HOOKS\_LIBRARY\_LOADING loading hooks library %1**

This is a debug message output just before the specified library is loaded. If the action is successful, it will be followed by the HOOKS\_LIBRARY\_LOADED informational message.

**HOOKS\_LIBRARY\_UNLOADED hooks library %1 successfully unloaded**

This information message is issued when a user-supplied hooks library has been successfully unloaded.

**HOOKS\_LIBRARY\_UNLOADING unloading library %1**

This is a debug message called when the specified library is being unloaded. If all is successful, it will be followed by the HOOKS\_LIBRARY\_UNLOADED informational message.

**HOOKS\_LIBRARY\_VERSION hooks library %1 reports its version as %2**

A debug message issued when the version check on the hooks library has succeeded.

**HOOKS\_LOAD\_ERROR 'load' function in hook library %1 returned error %2**

A "load" function was found in the library named in the message and was called. The function returned a non-zero status (also given in the message) which was interpreted as an error. The library has been unloaded and no callouts from it will be installed.

**HOOKS\_LOAD\_EXCEPTION 'load' function in hook library %1 threw an exception**

A "load" function was found in the library named in the message and was called. The function threw an exception (an error indication) during execution, which is an error condition. The library has been unloaded and no callouts from it will be installed.

**HOOKS\_LOAD\_FRAMEWORK\_EXCEPTION 'load' function in hook library %1 threw an exception: reason %2**

A "load" function was found in the library named in the message and was called. Either the hooks framework or the function threw an exception (an error indication) during execution, which is an error condition; the cause of the exception is recorded in the message. The library has been unloaded and no callouts from it will be installed.

**HOOKS\_LOAD\_SUCCESS 'load' function in hook library %1 returned success**

This is a debug message issued when the "load" function has been found in a hook library and has been successfully called.

**HOOKS\_NO\_LOAD no 'load' function found in hook library %1**

This is a debug message saying that the specified library was loaded but no function called "load" was found in it. Providing the library contained some "standard" functions (i.e. functions with the names of the hooks for the given server), this is not an issue.

**HOOKS\_NO\_UNLOAD no 'unload' function found in hook library %1**

This is a debug message issued when the library is being unloaded. It merely states that the library did not contain an "unload" function.

**HOOKS\_NO\_VERSION no 'version' function found in hook library %1**

The shared library named in the message was found and successfully loaded, but Kea did not find a function named "version" in it. This function is required and should return the version of Kea against which the library was built. The value is used to check that the library was built against a compatible version of Kea. The library has not been loaded.



**HOOKS\_OPEN\_ERROR failed to open hook library %1: %2**

Kea failed to open the specified hook library for the stated reason. The library has not been loaded. Kea will continue to function, but without the services offered by the library.

**HOOKS\_STD\_CALLOUT\_REGISTERED hooks library %1 registered standard callout for hook %2 at address %3**

This is a debug message, output when the library loading function has located a standard callout (a callout with the same name as a hook point) and registered it. The address of the callout is indicated.

**HOOKS\_UNLOAD\_ERROR 'unload' function in hook library %1 returned error %2**

During the unloading of a library, an "unload" function was found. It was called, but returned an error (non-zero) status, resulting in the issuing of this message. The unload process continued after this message and the library has been unloaded.

**HOOKS\_UNLOAD\_EXCEPTION 'unload' function in hook library %1 threw an exception**

During the unloading of a library, an "unload" function was found. It was called, but in the process generated an exception (an error indication). The unload process continued after this message and the library has been unloaded.

**HOOKS\_UNLOAD\_FRAMEWORK\_EXCEPTION 'unload' function in hook library %1 threw an exception, reason %2**

During the unloading of a library, an "unload" function was found. It was called, but in the process either it or the hooks framework generated an exception (an error indication); the cause of the error is recorded in the message. The unload process continued after this message and the library has been unloaded.

**HOOKS\_UNLOAD\_SUCCESS 'unload' function in hook library %1 returned success**

This is a debug message issued when an "unload" function has been found in a hook library during the unload process, called, and returned success.

**HOOKS\_VERSION\_EXCEPTION 'version' function in hook library %1 threw an exception**

This error message is issued if the version() function in the specified hooks library was called and generated an exception. The library is considered unusable and will not be loaded.

**LFC\_FAIL\_PID\_CREATE : %1**

This message is issued if LFC detected a failure when trying to create the PID file. It includes a more specific error string.

**LFC\_FAIL\_PID\_DEL : %1**

This message is issued if LFC detected a failure when trying to delete the PID file. It includes a more specific error string.

**LFC\_FAIL\_PROCESS : %1**

This message is issued if LFC detected a failure when trying to process the files. It includes a more specific error string.

**LFC\_FAIL\_ROTATE : %1**

This message is issued if LFC detected a failure when trying to rotate the files. It includes a more specific error string.

**LFC\_PROCESSING Previous file: %1, copy file: %2**

This message is issued just before LFC starts processing the lease files.

**LFC\_READ\_STATS Leases: %1, attempts: %2, errors: %3.**

This message prints out the number of leases that were read, the number of attempts to read leases and the number of errors encountered while reading.

**LFC\_ROTATING LFC rotating files**

This message is issued just before LFC starts rotating the lease files - removing the old and replacing them with the new.

**LFC\_RUNNING LFC instance already running**

This message is issued if LFC detects that a previous copy of LFC may still be running via the PID check.

**LFC\_START Starting lease file cleanup**

This message is issued as the LFC process starts.

**LFC\_TERMINATE LFC finished processing**

This message is issued when the LFC process completes. It does not indicate that the process was successful only that it has finished.

---

**LFC\_WRITE\_STATS Leases: %1, attempts: %2, errors: %3.**

This message prints out the number of leases that were written, the number of attempts to write leases and the number of errors encountered while writing.

**LOGIMPL\_ABOVE\_MAX\_DEBUG debug level of %1 is too high and will be set to the maximum of %2**

A message from the interface to the underlying logger implementation reporting that the debug level (as set by an internally-created string DEBUGn, where n is an integer, e.g. DEBUG22) is above the maximum allowed value and has been reduced to that value. The appearance of this message may indicate a programming error - please submit a bug report.

**LOGIMPL\_BAD\_DEBUG\_STRING debug string '%1' has invalid format**

A message from the interface to the underlying logger implementation reporting that an internally-created string used to set the debug level is not of the correct format (it should be of the form DEBUGn, where n is an integer, e.g. DEBUG22). The appearance of this message indicates a programming error - please submit a bug report.

**LOGIMPL\_BELOW\_MIN\_DEBUG debug level of %1 is too low and will be set to the minimum of %2**

A message from the interface to the underlying logger implementation reporting that the debug level (as set by an internally-created string DEBUGn, where n is an integer, e.g. DEBUG22) is below the minimum allowed value and has been increased to that value. The appearance of this message may indicate a programming error - please submit a bug report.

**LOG\_BAD\_DESTINATION unrecognized log destination: %1**

A logger destination value was given that was not recognized. The destination should be one of "console", "file", or "syslog".

**LOG\_BAD\_SEVERITY unrecognized log severity: %1**

A logger severity value was given that was not recognized. The severity should be one of "DEBUG", "INFO", "WARN", "ERROR", "FATAL" or "NONE".

**LOG\_BAD\_STREAM bad log console output stream: %1**

Logging has been configured so that output is written to the terminal (console) but the stream on which it is to be written is not recognised. Allowed values are "stdout" and "stderr".

**LOG\_DUPLICATE\_MESSAGE\_ID duplicate message ID (%1) in compiled code**

During start-up, Kea detected that the given message identification had been defined multiple times in the Kea code. This indicates a programming error; please submit a bug report.

**LOG\_DUPLICATE\_NAMESPACE line %1: duplicate \$NAMESPACE directive found**

When reading a message file, more than one \$NAMESPACE directive was found. (This directive is used to set a C++ namespace when generating header files during software development.) Such a condition is regarded as an error and the read will be abandoned.

**LOG\_INPUT\_OPEN\_FAIL unable to open message file %1 for input: %2**

The program was not able to open the specified input message file for the reason given.

**LOG\_INVALID\_MESSAGE\_ID line %1: invalid message identification '%2'**

An invalid message identification (ID) has been found during the read of a message file. Message IDs should comprise only alphanumeric characters and the underscore, and should not start with a digit.

**LOG\_LOCK\_TEST\_MESSAGE this is a test message.**

This is a log message used in testing.

**LOG\_NAMESPACE\_EXTRA\_ARGS line %1: \$NAMESPACE directive has too many arguments**

The \$NAMESPACE directive in a message file takes a single argument, a namespace in which all the generated symbol names are placed. This error is generated when the compiler finds a \$NAMESPACE directive with more than one argument.

**LOG\_NAMESPACE\_INVALID\_ARG line %1: \$NAMESPACE directive has an invalid argument ('%2')**

The \$NAMESPACE argument in a message file should be a valid C++ namespace. This message is output if the simple check on the syntax of the string carried out by the reader fails.

**LOG\_NAMESPACE\_NO\_ARGS line %1: no arguments were given to the \$NAMESPACE directive**

The \$NAMESPACE directive in a message file takes a single argument, a C++ namespace in which all the generated symbol names are placed. This error is generated when the compiler finds a \$NAMESPACE directive with no arguments.

**LOG\_NO\_MESSAGE\_ID line %1: message definition line found without a message ID**

Within a message file, message are defined by lines starting with a "%". The rest of the line should comprise the message ID and text describing the message. This error indicates the message compiler found a line in the message file comprising just the "%" and nothing else.

**LOG\_NO\_MESSAGE\_TEXT line %1: line found containing a message ID ('%2') and no text**

Within a message file, message are defined by lines starting with a "%". The rest of the line should comprise the message ID and text describing the message. This error indicates the message compiler found a line in the message file comprising just the "%" and message identification, but no text.

**LOG\_NO\_SUCH\_MESSAGE could not replace message text for '%1': no such message**

During start-up a local message file was read. A line with the listed message identification was found in the file, but the identification is not one contained in the compiled-in message dictionary. This message may appear a number of times in the file, once for every such unknown message identification.

There may be several reasons why this message may appear:

- The message ID has been mis-spelled in the local message file.
- The program outputting the message may not use that particular message (e.g. it originates in a module not used by the program).
- The local file was written for an earlier version of the Kea software and the later version no longer generates that message.

Whatever the reason, there is no impact on the operation of Kea.

**LOG\_OPEN\_OUTPUT\_FAIL unable to open %1 for output: %2**

Originating within the logging code, the program was not able to open the specified output file for the reason given.

**LOG\_PREFIX\_EXTRA\_ARGS line %1: \$PREFIX directive has too many arguments**

Within a message file, the \$PREFIX directive takes a single argument, a prefix to be added to the symbol names when a C++ file is created. This error is generated when the compiler finds a \$PREFIX directive with more than one argument.

Note: the \$PREFIX directive is deprecated and will be removed in a future version of Kea.

**LOG\_PREFIX\_INVALID\_ARG line %1: \$PREFIX directive has an invalid argument ('%2')**

Within a message file, the \$PREFIX directive takes a single argument, a prefix to be added to the symbol names when a C++ file is created. As such, it must adhere to restrictions on C++ symbol names (e.g. may only contain alphanumeric characters or underscores, and may not start with a digit). A \$PREFIX directive was found with an argument (given in the message) that violates those restrictions.

Note: the \$PREFIX directive is deprecated and will be removed in a future version of Kea.

**LOG\_READING\_LOCAL\_FILE reading local message file %1**

This is an informational message output by Kea when it starts to read a local message file. (A local message file may replace the text of one or more messages; the ID of the message will not be changed though.)

**LOG\_READ\_ERROR error reading from message file %1: %2**

The specified error was encountered reading from the named message file.

**LOG\_UNRECOGNISED\_DIRECTIVE line %1: unrecognised directive '%2'**

Within a message file, a line starting with a dollar symbol was found (indicating the presence of a directive) but the first word on the line (shown in the message) was not recognised.

**LOG\_WRITE\_ERROR error writing to %1: %2**

The specified error was encountered by the message compiler when writing to the named output file.

**USER\_CHK\_HOOK\_LOAD\_ERROR DHCP UserCheckHook could not be loaded: %1**

This is an error message issued when the DHCP UserCheckHook could not be loaded. The exact cause should be explained in the log message. User subnet selection will revert to default processing.

**USER\_CHK\_HOOK\_UNLOAD\_ERROR DHCP UserCheckHook an error occurred unloading the library: %1**

This is an error message issued when an error occurs while unloading the UserCheckHook library. This is unlikely to occur and normal operations of the library will likely resume when it is next loaded.

**USER\_CHK\_SUBNET4\_SELECT\_ERROR DHCP UserCheckHook an unexpected error occurred in subnet4\_select callout: %**

This is an error message issued when the DHCP UserCheckHook subnet4\_select hook encounters an unexpected error. The message should contain a more detailed explanation.

**USER\_CHK\_SUBNET4\_SELECT\_REGISTRY\_NULL DHCP UserCheckHook UserRegistry has not been created.**

This is an error message issued when the DHCP UserCheckHook subnet4\_select hook has been invoked but the UserRegistry has not been created. This is a programmatic error and should not occur.

**USER\_CHK\_SUBNET6\_SELECT\_ERROR DHCP UserCheckHook an unexpected error occurred in subnet6\_select callout: %**

This is an error message issued when the DHCP UserCheckHook subnet6\_select hook encounters an unexpected error. The message should contain a more detailed explanation.

**USER\_CHK\_SUBNET6\_SELECT\_REGISTRY\_NULL DHCP UserCheckHook UserRegistry has not been created.**

This is an error message issued when the DHCP UserCheckHook subnet6\_select hook has been invoked but the UserRegistry has not been created. This is a programmatic error and should not occur.

---