

IPWorks DHCP Parameter Description

PARAMETER DESCRIPTION

Copyright

© Ericsson AB 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in IPWorks Trademark Information.



Contents

1	Introduction	1
1.1	Prerequisites	1
1.1.1	Documents	1
1.2	Related Information	1
2	Basic Concepts	3
3	Managed Object Format	5
3.1	Data Types	5
3.2	Enumerations	7
4	Managed Objects	9
4.1	DHCP Policy Objects	10
4.1.1	Subnet	11
4.1.2	Pool	15
4.1.3	Link	20
4.1.4	Client	23
4.1.5	ClientClass	26
4.1.6	ClientSubclass	30
4.2	Dhcpv4AuthKey	33
4.3	Option	35
4.3.1	ISC DHCPv4 Options	37
4.3.2	IPWorks Specific DHCPv4 Options	43
4.3.3	DhcpV4Option	51
4.4	Option82	54
4.4.1	Dhcpv4Option82Format	54
4.4.2	Dhcpv4Option82IPRange	58
4.5	Dhcpv4Server	61
4.6	Lease	67
5	DHCP Expressions	73
5.1	Boolean Expressions	73
5.2	Data Expressions	74
5.3	Numeric Expressions	76
	Reference List	77





1 Introduction

This document describes the objects and fields for DHCP in IPWorks.

Scope

This document covers the following topics:

- Managed Object Format
- Managed Object Class Descriptions
- Managed Objects (MO)
- Managed Object Attributes

Target Groups

This document is intended for personnel configuring and fine tuning the IPWorks. All the objects in this document are configured by using IPWorks CLI. It is assumed that readers of this document are familiar with basic concepts and operations of IPWorks CLI. For details, refer to *Command Line Interface User Guide for IPWorks SS*, Reference [1].

1.1 Prerequisites

Not applicable.

1.1.1 Documents

Not applicable.

1.2 Related Information

Trademark information, typographic conventions, definition, and explanation of acronyms and terminology can be found in the following documents:

- *Trademark Information*, Reference [2]
- *Glossary of Terms and Acronyms*, Reference [3]
- *Typographic Conventions*, Reference [4]





2 Basic Concepts

This section describes the following:

- The MOM concept
- The MOC concept
- Data types
- Object attributes
- Conventions

The Managed Object Model (MOM) presents a view of manageable resources in the **IPWorks**, and attributes and actions associated with the resources.

A Managed Object (MO) is an entity presented to the user for the purpose of controlling the aspects of a function. The object carries attributes that reflect the behavior of the function.

The MOs are identified by means of a naming attribute, also called the Relative Distinguished Name (RDN). The ID part of this attribute is defined when the MO is created, and cannot be changed afterwards. A Local Distinguished Name (LDN) is a sequence of RDNs, which forms a unique name within the node.

All the objects in this document are configured by using IPWorks CLI. It is assumed that readers of this document are familiar with basic concepts and operations of IPWorks CLI. For details, refer to *Command Line Interface User Guide for IPWorks SS*, Reference [1].





3 Managed Object Format

This section describes the format used to display information about objects.

Key The key is an identifier of an object. The combination of the key field values must be unique for an object.

Required The required field indicates that the field must be configured, otherwise the CLI generates an error.

- **<Field>**: Field name.
 - **Aliases**: The alternative names of a field.
 - **Type**: Shows whether this field can contain multiple values by putting them into a separated table in the database.
 - **Data Type**: The predefined data type of a field. See Section 3.1 on page 5.
 - **Read-only**: If the `readonly` attribute is “yes”, the value of this field cannot be set from the CLI.
 - **Enumeration**: If a field has an enumeration attribute, the value of this field should be selected from a limited value set. See Section 3.2 on page 6.
 - **Description**: The brief description of a field.
 - **Examples**: The examples of valid values in a field.

Note: If the field `Partition` is required, unless the user wants to work within a different partition, the partition being worked on does not need to be specified. IPWorks assumes the *active* partition at all times. For more information, see the *Partition* section of *IPWorks Configuration ManagementReference* [5].

3.1 Data Types

Blob A large text string. The theoretic maximum size of this data type is 2GB.

BooleanTrueFalse A Boolean value that is either true or false. You can enter yes/no or 1/0 and they will be converted to true/false as appropriate.

DnsName A domain name with maximum length not exceeding 128 characters.



Dhcpv4Option	A DHCPv4 configuration option, where each option setting is in the form "option value [, value...]". The option and subsequent value(s) are separated by a single space. The "option" is specified by the option tag. The format of the option value(s) are determined by the option definition.
Int32	A 32 bit signed integer value.
IPv4Address	An IPv4 address in dotted decimal notation.
IPv4AddressRange	<p>An IPv4 address range, in dotted decimal notation using the hyphen as a separator between the start and end addresses.</p> <p>Note: Note that you can omit the first three octets in the end address if they are the same as the first three octets in the first address, for example, 192.168.1.1-255.</p>
IPv6Address	An IPv6 address, which can be specified using the standard abbreviated forms, but stored in the fully expanded form.
MACAddress	Short for Media Access Control address, a hardware address that uniquely identifies each node of a network.
Name	The name of the configured object.
SortKey	A signed 32 bit integer that is computed from an IPv4 address that can be used as an alternate sorting key that returns addresses in sorted order. For example, 0.0.0.0 is equivalent to the smallest 32 bit integer using this scheme, and 255.255.255.255 is the largest.
SubnetMask	The subnet mask is the network address plus the bits reserved for identifying the subnet work. (By convention, the bits for the network address are all set to 1, though it would also work if the bits were set exactly as in the network address.) It is called a mask because it can be used to identify the subnet to which an IP address belongs by performing a bitwise AND operation on the mask and the IP address. The result is the subnet work address.
UInt8	An 8-bit unsigned integer value, between 0 and 255.
UInt16	A 16-bit unsigned integer value, between 0 and 65535.



3.2 Enumerations

DhcpV4ImplementationDHCPv4

DhcpV4OptionDatatype

Boolean, Custom, DnsName, Encapsulated, GmtDate, Hba, Int8, Int16, Int32, IPAddress, IPAddressMask, IPAddressMaskPair, IPAddressMaskIPAddressTriplet, IPAddressPair, LocalTimeDate, MTU, NameString, QuotedString, String, Text, Time, UInt8, UInt16, UInt32, UIntPer, IPv4Address

DhcpV4Scope

Client, ClientClass, ClientSubclass, Link, Subnet, Pool, DhcpV4Server

PtrStrategy

Manual, Auto, Prompt, Dynamic, Generated, Delegated





4 Managed Objects

This section describes some of the concepts that are required for managing DHCP for IPv4, the Managed Object Classes, and attributes.

The Dynamic Host Configuration Protocol (DHCP) is an extension of the Bootstrap Protocol (BOOTP) that handles dynamic IP address assignments for remote clients. It allows a remote client to plug into a network and broadcast a request, to one or more DHCP servers managing a particular subnet, for network configuration information. One or more DHCP servers reply to the remote client and one server ultimately negotiates a lease of terms including an IP address, a specified period of usage for the address, and other client configuration parameters.

If a client finds a favorable lease offer with the required configuration parameters, the client can accept the offer. After successful negotiation, the server instructs the client to set a lease-renewal timer and a lease-expiry timer, and directs the client where to find system resources (such as the system printer, mail server and terminal fonts).

When the DHCP server receives a client request packet, the process that determines which IP address to assign to the client and which options to send to the client is as follows:

Note: This may not be the exact algorithm used by the server, but it is accurate enough to understand how the configuration defined impacts the way the server responds to clients.

1. When the packet is first received, the server checks the source subnet of the packet to determine if it came from a subnet for which the server is responsible. The source subnet is determined in the following ways:

If the packet was received from a relay agent, the relay agent includes its IP address in the `giaddr` field and that is used to determine the source subnet.

Otherwise, the IP address of the interface on which the packet was received is used to determine the source subnet.

2. Based on the packet's source hardware address and the DHCP client identifier (if present), the server finds any host declarations for this client. If there are any matching host declarations with a fixed address for the source subnet (or related shared network), that will be the address assigned to the client. If there are any host declarations that deny addresses to the client, the request is dropped and logged. The presence of host declarations also determines whether the client is considered a known client. If there are any global option settings that would deny an address to the client, the request is dropped.



3. Next, the server determines which classes the client belongs to by evaluating the match expression of each class against the contents of the client packet. This also determines the subclasses for those classes that have them.
4. Based on the class membership and whether the client is known, the server examines each of the lease pools for the source subnet (or related shared network) to see if the client is allowed to lease addresses in that pool. The first pool that allows the client to lease an address is used.
5. If no address is available for the client the request is dropped.
6. Once the address has been determined, the server determines which options will be sent to the client by merging the option settings from all the declarations that apply to the client. If there is a conflict between two declarations, the most specific declaration is used as follows (with highest precedence first):

The relevant host client declaration

Subclasses

Classes

The pool declaration (if it is not a fixed address)

The subnet declaration

The shared network link declaration (if there is one)

The global default option settings

7. Once all the option settings have been determined, the server sends a response back to the client.

4.1 DHCP Policy Objects

In IPWorks, DHCP is configured for both IPv4 and IPv6 by creating policy objects. Different types of policy objects define how a DHCP server behaves under different situations. The policies are specific to IPv4, IPv6, or both.

If a server is not specified for a policy, most policies will be associated with all servers for that area. There are a few exceptions to this, but they are noted in the descriptions of the specific types of policies.

Common Parameters

Some parameters are common to all policy objects. These parameters are used to associate the policy with the servers that implement the policy. The parameters are listed as follows:

- **Server:** The name of the server(s) that use this policy. This field is set to the names of the servers that include the policy in their configuration. The



referenced servers must be defined before the policy is created. This field can also be set to two special values:

- **all** indicates that the policy is to be included in all servers associated with that area.
- **none** indicates that the policy is not included in any of the servers (this provides a simple method for temporarily disabling a policy without deleting it).

4.1.1 Subnet

A subnet is a DHCP policy object that represents the configuration for a contiguous set of addresses within an IPv4 subnet. Subnet objects can be used to specify information about the address space, but most of the time, they are created to assist with the configuration of the DHCP protocol.

Key Partition, Area, Name

Required Partition, Area, Name, Address, Mask, MaskLength

- **Name**

- **Type:** Single
- **Data Type:** Name
- **Description:** A unique identifier for the subnet — if one is not specified, IPWorks will generate a name based on the address and masklength.

- **Address**

- **Type:** Single
- **Data Type:** IPv4Address
- **Description:** The address of the subnet.

- **Mask**

- **Type:** Single
- **Data Type:** SubnetMask
- **Description:** The subnet mask for this subnet.

- **MaskLength**

- **Type:** Single
- **Data Type:** UInt8



- **Description:** The subnet mask length for this subnet. The value is between 0 and 32 indicating the number of bits that are set in the mask.

Note: Only one of the two fields (Mask, MaskLength) need to be specified and the other will be computed.

- **FirstSortKey**

- **Type:** Single
- **Data Type:** SortKey
- **Read-only:** Yes
- **Description:** The sort key of the first address in the subnet.

- **LastSortKey**

- **Type:** Single
- **Data Type:** SortKey
- **Read-only:** Yes
- **Description:** The sort key of the last address in the subnet.

- **BroadcastAddress**

- **Type:** Single
- **Data Type:** IPv4Address
- **Description:** The broadcast address for this subnet.

- **PtrStrategy**

- **Type:** Single
- **Enumeration:** PtrStrategy
- **Description:** The strategy used for managing reverse lookup (PTR) records for addresses in this subnet.

- **PtrDomain**

- **Type:** Single
- **Data Type:** DnsName
- **Description:** The domain that will contain the reverse lookup PTR resource records for dnsnames assigned to addresses in this subnet.



Note: PtrStrategy and PtrDomain allow the user to define a subnet-specific strategy for managing PTRRecords for addresses in this subnet. For more information about PTRRecord, see the section **PTRRecord** in *IPWorks DNS*, *ASDNS*, *ENUM Parameter Description*, Reference [6].

IPWorks allows the representation of a hierarchy of subnets, but the DHCP server cannot handle subnet definitions that overlap. Therefore, whether hierarchical subnets are implemented, the **server** field on the higher level subnets must be set to *none* so that they are not included in the configuration files for any DHCP servers with the lower level subnets.

- **Link**
 - **Type:** Single
 - **Description:** The name of the link that contains this subnet. If the subnet shares a physical network segment with other subnets the link policy will need to be defined, so that the DHCP server can handle requests from these subnets correctly. In such a situation this field is set to the name of the Link policy (which must already exist).
- **IsDistinct**
 - **Type:** Single
 - **Data Type:** BooleanTrueFalse
 - **Read-only:** Yes
 - **Description:** Indicates whether this subnet is the only configured subnet on the LAN where it is running.
- **V4Option**
 - **Aliases:** Option, ConfigOption, Opt, V4ConfigOption, V4Opt
 - **Data Type:** DhcpV4Option
 - **Description:** DHCPv4 Configuration setting(s) specific to this policy.
- **V4Config**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** The compiled configuration setting(s) specific to this policy that will be used for the DHCPv4 protocol.
- **Server**



- **Description:** The name of the server(s) that use this policy. For details, see Common Parameters. The default value is *none*.
- **Area**
 - **Type:** Single
 - **Description:** The name of the area that contains this object. The area represents a single address space and the policies within that area define how the DHCP protocol is managed within that address space. The area must exist before the policy is created.
- **Partition**
 - **Type:** Single
 - **Description:** The name of the partition that contains this object.
- **Creationtime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **Creator**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes



- **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **LastModifiedTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **InternalData**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.
- **Format**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.

4.1.2

Pool

A pool, also called a lease pool, is a DHCP policy object that represents a contiguous set of addresses that are available for dynamic address assignment. All the addresses in a pool must be contained within a declared subnet and the subnet must be defined prior to defining the pool.

Key Partition, Name

Required Partition, Name, Subnet, AddressRange, Server, Area

- **Name**



- **Type:** Single
- **Data Type:** Name
- **Description:** The name of this lease pool policy (must be unique).
- **Subnet**
 - **Type:** Single
 - **Description:** The name of the subnet that contains this lease pool. This should be set to the name of the subnet that contains the pool. When this field blank, IPWorks examines all the existing subnets to determine which one contains the pool.
- **AddressRange**
 - **Aliases:** Range
 - **Data Type:** IPv4AddressRange
 - **Description:** The range of addresses in the lease pool. This field can have multiple values. Each value can specify a single address (such as 1.2.3.4), a range of addresses (such as 1.2.3.1-100), or an entire subnet (such as 1.2.3.128/25).
- **FirstSortkey**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The sort key of the first address in the range (or subnet).
- **LastSortkey**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The sort key of the last address in the range (or subnet).
- **AllowKnownClients**
 - **Type:** Single
 - **Data Type:** BooleanTrueFalse
 - **Description:** Indicates whether this pool should either allow or prevent allocation of addresses to known clients. If no value is specified then they are neither explicitly allowed or denied, and other criteria is used to determine if an address should be assigned.
- **AllowUnknownClients**



- **Type:** Single
- **Data Type:** BooleanTrueFalse
- **Description:** Indicates whether this pool should either allow or prevent allocation of addresses to unknown clients. If no value is specified then they are neither explicitly allowed or denied, and other criteria is used to determine if an address should be assigned.
- **AllowBootpClients**
 - **Type:** Single
 - **Data Type:** BooleanTrueFalse
 - **Description:** Indicates whether this pool should either allow or prevent allocation of addresses to BOOTP clients. If no value is specified then they are neither explicitly allowed or denied, and other criteria is used to determine if an address should be assigned.
- **AllowClientClass**
 - **Description:** The names of any client classes that are explicitly allowed in this pool. If no value is specified, then the pool is open to any clients (except any that are explicitly denied)
- **DenyClientClass**
 - **Description:** The names of any client classes that are explicitly denied from using addresses in this pool.
- **AllowedClient**
 - **Aliases:** AllowedClass, Allow, AllowClient, AllowClass
 - **Description:** Enter values to describe clients that should be explicitly allowed to obtain addresses in this pool. If no value is specified for either of the two fields (AllowedClient, DeniedClient), the pool is available to any client that requests an address.

Make the value of the field the name of a *client class* or one of the following predefined values. Multiple client classes are allowed to be specified.

 - **all:** all clients
 - **bootp:** clients that use the Dynamic BOOTP protocol
 - **known:** clients that have an explicit Client declaration
 - **unknown:** clients that do not have an explicit Client declaration

Note: When specifying client class names, the names must exactly match the case of the **Name** of the client class.



- **DeniedClient**
 - **Aliases:** DeniedClass, Deny, DenyClient, DenyClass
 - **Description:** Enter values to describe clients that should be explicitly denied addresses in this pool. For more information, see AllowedClient
 - For detailed information, see the description for AllowedClient.
- **V4Option**
 - **Aliases:** Option, ConfigOption, Opt, V4ConfigOption, V4Opt
 - **Data Type:** DhcpV4Option
 - **Description:** DHCPv4 Configuration setting(s) specific to this policy.
- **V4Config**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** The compiled configuration setting(s) specific to this policy that will be used for the DHCPv4 protocol.
- **Server**
 - **Description:** The name of the server(s) that use this policy. For details, see Common Parameters. The default value is *none*.
- **Area**
 - **Type:** Single
 - **Description:** The name of the area that contains this object. The area represents a single address space and the policies within that area define how the DHCP protocol is managed within that address space. The area must exist before the policy is created.
- **Partition**
 - **Type:** Single
 - **Description:** The name of the partition that contains this object.
- **Creationtime**
 - **Type:** Single
 - **Read-only:** Yes



- **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **Creator**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **LastModifiedTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **InternalData**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes



- **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.
- **Format**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.

4.1.3

Link

A Link Policy stores information about how a DHCP server is configured for a shared network (a set of one or more logical subnets that all share the same physical wire). A link, also known as a shared network, is a DHCP policy object that represents a physical network segment that has subnets operating on it. If there are multiple subnets on the same network segment, the DHCP server must be declared to correctly process DHCP requests on those subnets. During the DHCP server request processing, the server tries to find an address in other subnets of the same link before dropping the request if no address is available in any subnets.

Note: A link must include at least one subnet. If a link includes no subnets, the DHCP server fails to start.

Key Partition, Area, Name

Required Partition, Area, Name

- **Name**
 - **Type:** Single
 - **Data Type:** Name
 - **Description:** The name of this shared network (must be unique).
- **V4Option**
 - **Aliases:** V4ConfigOption, V4Opt
 - **Data Type:** DhcpV4Option
 - **Description:** DHCPv4 Configuration setting(s) specific to this policy.
- **V4Config**



- **Type:** Single
- **Data Type:** Blob
- **Read-only:** Yes
- **Description:** The compiled configuration setting(s) specific to this policy that will be used for the DHCPv4 protocol. These are determined from the Option field (and other fields) and cannot be modified directly.
- **Server**
 - **Description:** The name of the server(s) that use this policy. For details, see Common Parameters. The default value is *all*.
- **Area**
 - **Type:** Single
 - **Description:** The name of the area that contains this object. The area represents a single address space and the policies within that area define how the DHCP protocol is managed within that address space. The area must exist before the policy is created.
- **Partition**
 - **Type:** Single
 - **Description:** The name of the partition that contains this object.
- **Creationtime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **Creator**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single



- **Data Type:** Blob
- **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **LastModifiedTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **InternalData**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.
- **Format**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.



4.1.4 Client

A client, also known as a host policy, is a DHCP policy object that represents the configuration for a specific DHCP client (usually a specific computer system) on the network.

Key Partition, Area, Name

Required Partition, Area, Name

- **Name**
 - **Type:** Single
 - **Data Type:** Name
 - **Description:** The name of the policy (usually a recognizable name for the client itself). If there are two different client policies that refer to the same client, then each policy must have a unique name.
- **MACAddress**
 - **Type:** Single
 - **Data Type:** MACAddress
 - **Description:** The MAC address, in hexadecimal format. This can be entered using several different methods, but is always stored using the colon character to separate the byte values in the address.
 - Note:** When the client identifiers are assigned to the host systems, certain characters that are considered legal are to be avoided. These characters are problematic when configuring the server, for example quotation marks.
 - **Example:** 01:02:03:04:AA:BB
- **ClientIdentifier**
 - **Type:** Single
 - **Description:** The DHCPv4 client identifier for this client, as specified in RFC2131.
 - Note:** When the client identifier is assigned to the host systems, certain characters that are considered legal are to be avoided. These characters are problematic when configuring the server, for example quotation marks.
- **FixedAddress**
 - **Data Type:** IPv4Address



- **Description:** The reserved (fixed) IPv4 address(es) for the client. Multiple addresses can be specified for the client, as long as each address is on a different subnet. When a client has a fixed address on a subnet, it is ineligible for dynamic addresses.
- **FixedV6Address**
 - **Data Type:** IPv6Address
 - **Description:** The reserved (fixed) IPv6 address(es) for the client.
- **DenyBooting**
 - **Type:** Single
 - **Data Type:** BooleanTrueFalse
 - **Description:** Indicates whether this host should be denied from communicating with DHCP server. If this field is set to true, the client will not be allowed to obtain an address from the DHCP server. This is a useful feature for handling cumbersome clients on the network
- **DenyAddress**
 - **Type:** Single
 - **Data Type:** BooleanTrueFalse
 - **Description:** Indicates whether this host should be denied addresses by the DHCP server.
- **V4Option**
 - **Aliases:** V4ConfigOption, V4Opt
 - **Data Type:** DhcpV4Option
 - **Description:** DHCPv4 Configuration setting(s) specific to this policy.
- **V4Config**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** The compiled configuration setting(s) specific to this policy that will be used for the DHCPv4 protocol. These are determined from the Option field (and other fields) and cannot be modified directly.
- **Server**
 - **Description:** The name of the server(s) that use this policy. For details, see Common Parameters. The default value is *all*.



- **Area**
 - **Type:** Single
 - **Description:** The name of the area that contains this object. The area represents a single address space and the policies within that area define how the DHCP protocol is managed within that address space. The area must exist before the policy is created.
- **Partition**
 - **Type:** Single
 - **Description:** The name of the partition that contains this object.
- **Creationtime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **Creator**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.



- **LastModifiedTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **InternalData**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.
- **Format**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.

4.1.5

ClientClass

A ClientClass is a DHCP policy object that represents configuration for a group of clients. When the DHCP server receives a request from a client, the contents of the DHCP packet are examined to determine which client classes should be associated with that request. Each request can be associated with multiple classes.

Key Partition, Area, Name

Required Partition, Area, Name

- **Name**
 - **Type:** Single



- **Data Type:** Name
- **Description:** The name of the client class, which must be unique.
- **Match**
 - **Type:** Single
 - **Description:** This specifies the criteria that the request packet must satisfy in order for the client to be considered as a member of the client class. The value of this field must be one of the following:
 - the value of **MACAddress**
 - the value of **ClientIdentifier**
 - the value of **UserClass**
 - *<DHCP data expression>* (discussed in **Data Expressions**)
 - *if <DHCP boolean expression>* (discussed in **Boolean Expressions**)

For the first four values, the specified value is determined for the request packet by extracting the specified piece of information from the packet or by evaluating the specified data expression. This value is then compared to the existing subclasses for this client class and if a subclass exists with the specified value, the client is considered to be a member of the class and a member of the subclass as well. For the last value, if the specified boolean expression evaluates to *true*, the client is considered to be a member of the client class.

- **Spawn**
 - **Type:** Single
 - **Description:** This is to define a spawn class which indicates that all clients will be considered to be members of this spawn class. The spawning class is a class that automatically produces subclasses based on the relay agent information sending from client. The value of the spawn field is one of the following:
 - `agent.remote-id`
 - `agent.circuit-id`
 - `agent.agent-id`

There are two ways that subclasses are created for a client class: they can be explicitly declared by creating a client subclass object or they can be automatically created if the **Spawn** value is defined. In the second case, this implies that every request will be considered to be a member of the class since it will either match an existing subclass, or one will be spawned.



- **LeaseLimit**
 - **Type:** Single
 - **Data Type:** Int32
 - **Description:** If this is set, then this client class will limit the maximum number of members of this class that can have addresses to the specified value. This limit applies to all addresses the server allocates, not just addresses on a particular network segment. If a client is a member of more than one class with lease limits, the server will assign the client an address based on either class. If a client is a member of one or more classes with limits and one or more classes without limits, the classes without limits are not considered.
- **Preference**
 - **Type:** Single
 - **Data Type:** UInt16
 - **Description:** A numeric value indicating the relative preference of this client class in relation to other client classes in the server's configuration. This is used to determine the order in which classes are used by the server when resolving conflicting option settings. If an option is set for both classes, the value in the class with the lowest numeric preference will be used.
- **V4Option**
 - **Aliases:** V4ConfigOption, V4Opt
 - **Data Type:** DhcpV4Option
 - **Description:** DHCPv4 Configuration setting(s) specific to this policy.
- **V4Config**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** The compiled configuration setting(s) specific to this policy that will be used for the DHCPv4 protocol. These are determined from the Option field (and other fields) and cannot be modified directly.
- **Server**
 - **Description:** The name of the server(s) that use this policy. For details, see Common Parameters. The default value is *all*.
- **Area**



- **Type:** Single
- **Description:** The name of the area that contains this object. The area represents a single address space and the policies within that area define how the DHCP protocol is managed within that address space. The area must exist before the policy is created.
- **Partition**
 - **Type:** Single
 - **Description:** The name of the partition that contains this object.
- **Creationtime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **Creator**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **LastModifiedTime**



- **Type:** Single
- **Read-only:** Yes
- **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **InternalData**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.
- **Format**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.

4.1.6 ClientSubclass

A ClientSubclass is a DHCP policy object that represents the configuration for a subset of the members of a client class.

Key Partition, Area, ClientClass, Name

Required Partition, Area, ClientClass, Name

- **Name**
 - **Type:** Single
 - **Description:** This is a unique identifier for the subclass and must exactly match the client's match criteria in incoming packets.
- **ClientClass**



- **Type:** Single
- **Description:** The name of the client class to which the subclass applies. It must be created before any subclasses that refer to it.
- **V4Option**
 - **Aliases:** Option, ConfigOption, Opt, V4ConfigOption, V4Opt
 - **Data Type:** DhcpV4Option
 - **Description:** DHCPv4 Configuration setting(s) specific to this policy.
- **V4Config**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** The compiled configuration setting(s) specific to this policy that will be used for the DHCPv4 protocol. These are determined from the Option field (and other fields) and cannot be modified directly.
- **Server**
 - **Description:** The name of the server(s) that use this policy. For details, see Common Parameters. The default value is *all*.
- **Area**
 - **Type:** Single
 - **Description:** The name of the area that contains this object. The area represents a single address space and the policies within that area define how the DHCP protocol is managed within that address space. The area must exist before the policy is created.
- **Partition**
 - **Type:** Single
 - **Description:** The name of the partition that contains this object.
- **Creationtime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone



must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.

- **Creator**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **LastModifiedTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **InternalData**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.



- **Format**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.

4.2 Dhcpv4AuthKey

An Authentication Key is used for DHCPv4 Server Authentication.

Key Partition, AuthKeyID

Required Partition, AuthKeyID, AuthKey, ClientID, Server

- **AuthKeyID**
 - **Type:** Single
 - **Data Type:** Int32
 - **Description:** The ID used to refer to the key.
- **AuthKey**
 - **Type:** Single
 - **Description:** A string of variable length.
- **ClientID**
 - **Type:** Single
 - **Description:** The MAC address, in hexadecimal format or a string.
 - **Examples:** 01:02:03:04:AA:AB
- **Server**
 - **Description:** The name of the dhcp server(s) where this key is used.
- **Partition**
 - **Type:** Single
 - **Description:** The name of the partition that contains this object.



- **Creationtime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **Creator**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **LastModifiedTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **InternalData**



- **Type:** Single
- **Data Type:** Blob
- **Read-only:** Yes
- **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.
- **Format**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.

4.3 Option

All of the DHCP management concepts described thus far (DHCPV4 servers and all the policy objects) have the ability to be further customized by specifying configuration options. This is done by adding values on a special field on the objects.

Each of these objects have a field called `V4Option`. This field can have multiple values. Each value is used to specify a configuration option. The syntax of the value is as follows:

```
<option-name> <value 1> <value 2> ...
```

There are many configuration options that can be specified. Some options can only be applied to certain types of objects. Many of these options may have somewhat complicated values when they are specified, so it is important to understand the meanings of both the option and the syntax. IPWorks CLI provides a command to display information about a configuration option.

```
IPWorks> show dhcpv4 option routers -verbose on
routers
Tag: 3
Alias(es): router, gateway, gateways, default-gateway
Multi-Valued: yes
Description: This specifies a list of router IP addresses on the
              subnet. List routers in order of preference. This
              serve as default gateway to the client.
Datatype: IPAddress
```



Description: A valid IPv4 address. The server also allows you to a valid dns name and it will be translated (via DNS) obtain an IPv4 address.

Example(s): routers rtr1.domain.com, rtr2.domain.com
router 10.1.0.1

Server Config: option routers <values>;

IPWorks>

If unsure how to define a configuration option for an object, use this command to review the option before setting it. IPWorks checks the syntax of any values set for configuration options.

In the following example, the options for both a pool and subnet that were previously created are set and the new configuration with these options are examined.

```
IPWorks> modify pool pool1 -add option="default-lease-time
86400"
Working on 1 object(s).
1 object(s) created.
IPWorks> select subnet 10.1.1.0/24
Selected 1 Object(s)
IPWorks> modify -add option="routers rtr.example.com" -add
option="dns-server 10.1.0.1"
Working on 1 object(s).
1 object(s) created.
IPWorks> list -format=conf
# Subnet 10.1.1.0/24
subnet 10.1.1.0 netmask 255.255.255.0 {
    option routers rtr.example.com;
    option domain-name-servers 10.1.0.1;

# Pool pool1
pool {
    pool-index 739763669;
    range 10.1.1.1 10.1.1.100;
    allow known clients;
    default-lease-time 86400;
}
```

Note: An IPWorks component must be created before using the `select` command. If the selected object does not exist, the following message will be displayed:

```
IPWorks> No matching object(s) found
```




4.3.1

ISC DHCPv4 Options

The following table shows the Internet Systems Consortium, Inc. (ISC) standard DHCPv4 options supported by the IPWorks DHCP server:

Table 1 Standard DHCPv4 Options

Option	Description
all-subnets-local	This specifies whether the client assumes that all subnets of the IP network, which the client is connected, use the same Maximum Transmission Unit (MTU) as the subnet of that network, to which the client is directly connected. A value of <i>true</i> means that all subnets share the same MTU.
arp-cache-timeout	This specifies the timeout, in seconds, for Address Resolution Protocol (ARP) cache entries.
boot-size	This specifies the length, in 512-octet blocks, of the default boot image for the client.
bootfile-name	This identifies a bootfile when the file field or the DHCP header has been used for DHCP options.
broadcast-address	This specifies the broadcast address in use on the client's subnet.
cookie-servers	This specifies a list of cookie servers available to the client. See RFC 865 for the list of cookie servers. List servers in order of preference.
classless-static-routes	The classless-static-routes option (121) supported by the DHCPv4 server used to specify the subnet mask with the subnet number and gateway IP address. This subnet mask is implicit in case of the static-route option (33).
default-ip-ttl	This specifies the default TTL the client uses on outgoing datagrams.
default-tcp-ttl	This specifies the default TTL the client uses when sending TCP segments. The value is represented as an 8-bit unsigned integer, with the minimum value of 1.
domain-name	This specifies the domain name the client uses when resolving host names through the DNS.
domain-name-servers	This specifies a list of DNS name servers available to the client. See STD 13, RFC 1035 for the list of name servers. List servers in order of preference.



Option	Description
extensions-path	This specifies a file, retrievable using TFTP, which contains information that can be interpreted the same way as the 64-octet vendor-extension field within the BOOTP response. There are exceptions: the length of the file is unconstrained and all references to Tag 18 (instances of the BOOTP Extensions Path field) within the file are ignored.
finger-server	This specifies a list of Finger servers available to the client. List servers in order of preference.
font-servers	This specifies a list of Window System font servers available to the client. List servers in order of preference.
host-name	This specifies the name of the client, which may be qualified with the local domain name. See RFC 2132, Section 3.17 for the preferred way to retrieve the domain name and RFC 1035 for character set restrictions.
ieee802-3-encapsulation	This specifies whether the client uses Ethernet Version 2 (see RFC 894) or IEEE 802.3 (see RFC 1042) encapsulation, if the interface is ethernet. A value of <i>false</i> indicates the client needs to use RFC encapsulation; a value of <i>true</i> means the client needs to use RFC 1042 encapsulation. If the option is not specified, the client uses its default.
ien116-name-servers	This specifies a list of IEN 116 name servers available to the client. List servers in order of preference.
impress-servers	This specifies a list of Imagen Impress servers available to the client. List servers in order of preference.
interface-mtu	This specifies the Maximum Transmission Unit (MTU) to use on the network interface.
ip-forwarding	This specifies whether the client configures its IP layer for packet forwarding. A value of <i>false</i> means to disable IP forwarding; a value of <i>true</i> means to enable IP forwarding. If the option is not specified, the client uses its default.
irc-server	This specifies a list of Internet Relay Chat (IRC) servers available to the client. List servers in order of preference.



Option	Description
log-servers	This specifies a list of MIT-LCS UDP log servers available to the client. List servers in order of preference.
lpr-servers	This specifies a list of line printer servers available to the client. See RFC 1179 for the list of line printer servers. List servers in order of preference.
mask-supplier	This specifies whether the client responds to subnet mask requests using ICMP. A value of <i>false</i> indicates the client should not respond; a value of <i>true</i> means the client should respond. If the option is not specified, the client uses its default.
max-dgram-reassembly	This specifies the maximum size datagram that the client should be prepared to reassemble.
merit-dump	This specifies the pathname of a file where the client's core image is to be dumped, if the client crashes.
mobile-ip-home-agent	This specifies a list of IP addresses indicating mobile IP home agents available to the client. List agents in order of preference.
nds-context	This specifies the initial Network Directory Service (NDS) context for the client.
nds-servers	This specifies a list of NDS servers available to the client. List servers in order of preference.
nds-tree-name	This specifies the NDS tree name for the client to use.
netbios-dd-server	This specifies a list of RFC 1001/1002 NBDD servers. List the servers in order of preference.
netbios-name-servers	This specifies a list of RFC 1001/1002 NBNS name servers. List the servers in order of preference.
netbios-node-type	This allows NetBIOS over TCP/IP clients to be configured as described in RFC 1001/1002. The value is specified as a single octet, which identifies the client type.
netbios-scope	This specifies the NetBIOS over TCP/IP <i>scope</i> parameter for the client as specified in RFC 1001/1002.



Option	Description
nis-domain	This specifies the name of the client's Network Information Service (NIS) domain. The domain is formatted as a string of characters from the NVT ASCII character set.
nis-servers	This specifies a list of IP addresses indicating NIS servers available to the client. List servers in order of preference.
nisplus-domain	This specifies the name of the client's NIS+ domain. The domain is formatted as a string of characters from the NVT ASCII character set.
nisplus-servers	This specifies a list of IP addresses indicating NIS+ servers available to the client. List servers in order of preference.
nnntp-server	This specifies a list of Network News Transport Protocol (NNTP) servers available to the client. List servers in order of preference.
non-local-source-routing	This specifies whether the client configures its IP layer to allow forwarding of datagrams with non-local source routes. A value of <i>false</i> prevents forwarding of such datagrams; a value of <i>true</i> allows forwarding. If the option is not specified, the client uses the default value.
ntp-servers	This specifies a list of IP addresses indicating NTP (Network Time Protocol) servers available to the client. List servers in order of preference.
path-mtu-aging-timeout	This specifies the timeout period, in seconds, for Path MTU values discovered by the mechanism defined in RFC 1191.
path-mtu-plateau-table	This specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit, unsigned integers in ascending numerical order. The minimum MTU value is 68.
perform-mask-discovery	This specifies whether the client performs subnet mask discovery using ICMP. A value of <i>false</i> indicates the client should not perform mask discovery; a value of <i>true</i> means the client should perform subnet mask discovery. If the option is not specified, the client uses the default value.



Option	Description
policy-filter	This specifies policy filters for non-local source routing. The filters are a list of IP addresses and masks that specify destination or mask pairs to filter incoming source routes. Any source routed datagram whose next hop address does not match one of the filters should be discarded by the client.
pop-server	This specifies a list of Post Office Protocol (POP3) servers available to the client. List servers in order of preference.
resource-location-servers	This specifies a list of Resource Location servers available to the client. See RFC 887 for the list of Resource Location servers. List servers in order of preference.
root-path	This specifies the pathname that contains the client's root disk. The path is formatted as a string of characters from the NVT ASCII character set.
router-discovery	This specifies whether the client solicits routers using the Router Discovery mechanism, as defined in RFC1256. A value of <i>false</i> indicates the client should not perform router discovery; a value of <i>true</i> means the client should perform router discovery. If the option is not specified, the client uses its default.
router-solicitation-addresses	This specifies the address to which the client transmits router solicitation requests.
routers or default-gateway	This specifies a list of router IP addresses on the client's subnet. List routers in order of preference.
sip-servers	This specifies a list of Session Initiation Protocol (SIP) servers to be used by the SIP client for all outbound SIP requests, also known as outbound proxy servers. The list can be IP addresses or domain names, but it should not contain both. The order of the addresses or domain names dictates which SIP server the client tries first.
smtp-server	This specifies a list of SMTP servers available to the client. List servers in order of preference.



Option	Description
static-routes	<p>This specifies a list of static routes the client installs in its routing cache.</p> <p>If multiple routes to the same destination are specified, the routes are listed in descending order of priority. The routes are a list of IP address pairs: the first address is the destination and the second address is the router for the destination.</p>
streettalk-directory-assistance-server	This specifies a list of STDA (StreetTalk Directory Assistance) servers available to the client. List servers in order of preference.
streettalk-server	This specifies a list of StreetTalk servers available to the client. List servers in order of preference.
subnet-mask	This specifies the client's subnet mask. By default, the server will provide this option to the client using the subnet mask for the lease pool. See RFC 950 for further information.
swap-server	This specifies the IP address of the client's swap server.
tcp-keepalive-garbage	This specifies whether the client sends TCP keep-alive messages with an octet of garbage for compatibility with older implementations. A value of <i>false</i> indicates a garbage octet should not be sent; a value of <i>true</i> indicates a garbage octet should be sent. If the option is not specified, the client uses its default.
tcp-keepalive-interval	This specifies the interval, in seconds, the client TCP waits before sending a keep alive message on a TCP connection. The time is specified as a 32-bit unsigned integer. A value of 0 indicates the client should not generate keep-alive messages on connections, unless specifically requested by an application.
tftp-server-name	This identifies a TFTP server when the <code>sname</code> field in the DHCP header has been used for DHCP options.
time-offset	This specifies the time offset of the client's subnet, in seconds, from Coordinated Universal Time (UTC). The offset is expressed as a complement 32-bit integer for 2 and is positive (east) or negative (west).



Option	Description
time-servers	This specifies a list of time servers available to the client. See RFC 868 for the time servers list. List servers in order of preference.
trailer-encapsulation	This specifies whether the client negotiates the use of trailers (see RFC 893) when using the ARP protocol. A value of <i>false</i> indicates the client should not attempt to use trailers; a value of <i>true</i> means the client should attempt to use trailers. If the option is not specified, the client uses the default value.
vendor-encapsulated-options	This is used by clients and servers to exchange vendor-specific information. The information is an object of <i>n</i> octets and the definition of the information is vendor-specific.
www-server	This specifies a list of World Wide Web (WWW) servers available to the client. List servers in order of preference.
x-display-managers	This specifies a list of X Window System display managers available to the client. List managers in order of preference.

4.3.2 IPWorks Specific DHCPv4 Options

The following table shows the IPWorks-specific DHCPv4 options that are available.

Table 2 IPWorks Specific DHCPv4 Options

Option	Description
allow-all-clients	This indicates whether a pool is available (or unavailable) for all clients. It is either <i>true</i> (any client can use addresses in the pool) or <i>false</i> (no clients can use addresses in the pool) Setting the value to <i>false</i> is useful when the user is staging a pool that is not ready to be activated. If no value is specified, then access to the pool is determined by the other <i>allow</i> options.
allow-bootp	This indicates whether the server responds to BOOTP queries. It is either <i>true</i> (queries are processed) or <i>false</i> (queries are ignored). The default value is <i>false</i> if no value is specified.



Option	Description
allow-dhcp-inform	This option is used to enable or disable DHCPINFORM command processing. By default, the DHCP server will process DHCPINFORM commands.
allow-known-clients	This indicates whether addresses need to be leased to known clients. Known clients are those which have an explicit client policy (aka Host) defined. It is either <i>true</i> (known clients are allowed) or <i>false</i> (known clients are denied). The default value is <i>true</i> .
allow-ras-servers	This option is used to allow or deny RAS Servers from obtaining addresses. By default, RAS Servers are allowed to obtain addresses.
allow-unknown-clients	This indicates whether addresses should be leased to unknown clients. Unknown clients are those which do not have an explicit Client (aka Host) policy defined. It is either <i>true</i> (unknown clients are allowed) or <i>false</i> (unknown clients are denied). The default value is <i>true</i> .
always-reply-rfc1048	Some BOOTP clients expect RFC 1048-style responses, if the user wants to send RFC1048 options to BOOTP clients set this option to <i>true</i> . The default value is <i>false</i> .
authoritative	The DHCP server normally assumes it does not have the complete configuration information for all networks. Thus, if a client requests an invalid IP address for a network segment, the server ignores the client and does not send it a DHCPNAK message to force the client to obtain a new (and valid) address. If the DHCP server has complete configuration information, add this option to allow the server to send DHCPNAK messages. The default value is <i>false</i> .
clf-address	<p>Specifies the IPv4 address of the CLF node which the Network Access Configuration Function (NACF) or IPWorks DHCPv4 Server interacts with.</p> <p>It is required to ensure that the NACF interacts with the CLF.</p>



Option	Description
clf-address-zone	Specifies the collective address space (Topology Zone) that one pool is designated to. It is a pool level option, and can contain either a string or DNS name. It is used when the NACF interacts with the CLF, and the NACF will only send message to CLF if the IP address belongs to a pool having this option.
custom	This allows the user to specify custom options that are not supported by the server.
ddns-ttl	This specifies the time (in seconds) to be used as the TTL value in the A and PTR resource records when the DHCP server performs a DDNS update. The default value is 600 seconds.
default-lease-time	This specifies the time (in seconds) to be used as the length of the lease if the client does not ask for a specific expiration time. The servers default lease time is 12 hours (43200 seconds). The <code>maximum-lease-time</code> must also be specified if the user wants leases longer than its default of 1 day.
deny-booting	This indicates whether the server is to respond to queries. Define this option only for specific clients. It is true (ignore queries from this client) or false (respond to queries). The default value is <i>false</i> .
dynamic-bootp-lease-cut off	This specifies a fixed date when leases assigned dynamically to BOOTP clients will end. BOOTP clients do not have a way of renewing leases and do not know leases can expire, therefore, the server normally assigns unlimited leases to all BOOTP clients. It is recommended in certain situations to set a cutoff date for all BOOTP leases — for example, the end of a school term, or the time at night when a facility is closed and all machines are required to be powered off.
dynamic-bootp-lease-length	This specifies the length (in seconds) of leases assigned to BOOTP clients. At some sites, it may be possible to assume that a lease is no longer in use if the holder has not used BOOTP or DHCP to get its address within a certain time period. If a client restarts using BOOTP during the time-out period, the lease duration is reset, so a BOOTP client that boots frequently enough never loses its lease.



Option	Description
dynamic-update	This specifies whether Dynamic DNS needs to be enabled. It is either <i>true</i> (Dynamic DNS updates will be performed) or <i>false</i> (Dynamic DDNS updates are not performed). The default value is <i>false</i> .
filename	This specifies the name of the initial bootfile to be loaded by a client. It should be the name of a file that is recognizable to whatever file transfer protocol the client is expected to use to load the file. Some clients may prefer to receive this information in the DHCP <code>bootfile-name</code> option.
get-lease-hostnames	This indicates whether the DHCP server should look up the host name for a client in DNS (based on the address being assigned). It is either <i>true</i> (get the name from DNS) or <i>false</i> (do not get the name). The default value is <i>false</i> .
hba	This specifies the load balancing hash bucket entries the server will respond to. When a client request is received the server hashes the client's unique identifier using an algorithm that hashes to a value between 0 and 255. If the <code>hba</code> is set, then the server will only respond to clients when the hash value corresponds to a bit that is set to 1 in the <code>hba</code> . This allows the user to divide the clients between any number of servers and to divide the load between the servers any way.
lease-limit	Limits the number of members of a class that can be assigned a lease at any given time.
leasethreshold	This specifies the threshold for active lease on server or pool level. Once the percentage of the active-state occupied leases in configured leases is crossing the value, the corresponding level alarm will be generated and sent to agent. Its range is [10,100] with integer and the fraction is forbidden.
max-delayed-acks	To improve performance under heavy loads, the DHCP server delays sending DHCPACK messages by up to 2 seconds. All DHCPACKs accumulated in that time are batched. This parameter specifies the maximum number of DHCPACKs in a batch. The default value is 8. To disable the delaying of DHCPACKs, specify a value of 1.



Option	Description
max-lease-time	This specifies the maximum length of a lease, in seconds. If not specified, the default value is 1 day (86400 seconds).
min-lease-time	This specifies the minimum length of a lease, in seconds. If not specified, the default value is 300 seconds.
name-by-client	This specifies whether clients are allowed to specify their own DNS names. It is either <i>true</i> (clients can specify their own names) or <i>false</i> (clients cannot specify their own names). The default value is <i>false</i> .
network-type	This indicates the type of access aggregation network where the messages are coming from. It can be set as ATM Aggregation network or Gigabit Ethernet Aggregation network. If the network-type is set, the clf-address must be set as well.
next-server	This specifies the address of the server where the boot file for the client is located.
one-lease-per-client	If <i>true</i> , it allows only one lease per client. The default value is <i>false</i> .
ping	This specifies whether the server needs to ping an address before offering it to a client. This is done to make sure that the address is not already in use by another network host. Set the option to either <i>true</i> (ping) or <i>false</i> (do not ping). The default value is <i>false</i> .
ping-retries	If the <code>ping</code> option is enabled, this specifies the number of times an address will be pinged unsuccessfully before it is considered to be unused (and therefore available to be leased).
ping-timeout	If the <code>ping</code> option is enabled, this specifies the maximum number of seconds to wait for a response from the address.
pool-index	It is a unique number generated automatically when a pool is created. It is used to differentiate the pools. This option cannot be either modified or deleted.
requested-options-only	Some DHCP clients cannot handle receiving any options other than the ones they request or a specific list of options. In order to send the options requested by the client, set this option to <i>true</i> . The default value is <i>false</i> .



Option	Description
secure-ddns	If <i>true</i> , the DHCP server will sign all DNS transactions relating to DDNS with a TSIGKey. The default value is <i>false</i> .
server-identifier	Specifies the IP Address sent to clients for the DHCP server. Normally, it is not specified as the server determines it. However, either the server has multiple interfaces or an interface has multiple addresses or the user wants to use an IP alias address for the server, it needs to be specified using this option.
server-name	This specifies the server name sent to the client when it is booting.
stash-agent-options	This specifies whether the DHCP server is to keep a copy of the relay agent information option it receives when a DHCP client acquires its lease, for use when the lease is renewed. If the flag is <i>true</i> , the relay agent information options are saved.
update-a-record	This is only used if the dynamic-update option is <i>true</i> . This specifies whether the DHCP server should make Dynamic DNS updates to ARecords. It is either <i>true</i> (update ARecords) or <i>false</i> (do not update ARecords). The default value is <i>false</i> .
use-host-decl-names	This indicates that the DHCP server needs to use the name of the host declaration as the client's host name if no other name is specified. If the option is not specified, the host declaration name is not used as the client's host name.
use-lease-addr-for-default-route	If <i>true</i> , the IP address of the lease being assigned is used as the router address and is sent to the client instead of the value specified in the router option. The default value is <i>false</i> .
option-100	A POSIX TZ string. Use of such a string can provide accuracy for at least one transition into and out of daylight saving time (DST), and possibly for more transitions if the transitions are regular enough. Forms(spaces inserted for clarity): std offset dst offset, rule



Table 3 Configure Failover Options

Option	Description
failover-load-balance-max-seconds	This specifies the maximum number of seconds, based on the <code>secs</code> field of DHCP requests, before the peer will respond to a client's request regardless of the load balancing hash bucket assignment for the server.
failover-split	This specifies the percentage of the address space in dynamic pools that will be allocated to the primary server for failover purposes. The remainder of the address space will be assigned to the secondary server. This option is limited by the <code>hba</code> option. If there are n "1" bits in <code>hba</code> , the option should have values between 0 and n . If no value is specified, this defaults to $n/2$.
failover-max-response-delay	If a message is not received from the failover peer within this time (in seconds), the peer is assumed to be down.
failover-max-unacked-updates	This specifies the maximum number of outstanding binding updates allowed at any time.
failover-mclt	This is the length (in seconds) of a first-time lease given to a client by the server for the failover protocol. The lead time is used by the server to communicate the lease activity to the secondary server and should be less than the normal lease time for the client.



Option	Description
max-lease-misbalance	<p>It specifies the DHCP server what percentage of total free leases (as defined as the total number of leases in either the free or backup states) a peer is allowed to own before a rebalance check is made. Configuring higher values causes the server to rebalance less frequently, but permits a larger imbalance between the free and backup lease pools. Configuring a lower value causes the server to rebalance more frequently, but keeps the pools more balanced. ISC DHCP servers no longer send POOLREQ messages unless the imbalance is at least twice this percentage in favor of the peer . Valid values are between 0 and 100. The default value is 15.</p>
max-lease-ownership	<p>It specifies the DHCP server what percentage of total free leases either it or its peer are normally allowed to own in excess of balance for the purpose of MAC address affinity. When a server undergoes a lease rebalancing operation, it first tries to move as many leases as it can to the peer whose previous client was load-balanced to that peer (as governed by the Load Balance Algorithm, see the split configuration value). The max-lease-ownership value determines the maximum percentage of leases either server holds before giving its peer the oldest leases (regardless of the previous client's place in the Load Balance Algorithm). Valid values are between 0 and 100, and should be less than the max-lease-misbalance value. Larger values will allow servers to retain leases to reallocate to returning clients, smaller values promote pool balance. The default value is 10.</p>



Option	Description
max-balance	It specifies the maximum length of time that the DHCP server schedules pool rebalance events. The default value is 3600 (seconds).
min-balance	It specifies the minimum length of time that the DHCP server schedules pool rebalance events. The default value is 60 (seconds).

4.3.3 DhcpV4Option

A DHCP Option defines information about both configurable parameters in the DHCP Server and DHCP options that are exchanged between the client and server in the DHCP protocol. The server determines how to respond to a client by examining the values for these options in the policies within the server's configuration. The collection of all the DHCP Option objects serves as a dictionary of the all the configurable parameters for the server.

Key Name

Required Name, Datatype

- **Implementation**

- **Enumeration:** DhcpV4Implementation
- **Description:** The server software implementation(s) that this option applies to. If not set, this option will be apply to all server implementations.

- **Datatype**

- **Type:** Single
- **Enumeration:** DhcpV4OptionDatatype
- **Description:** The expected datatype for value(s) of this option.

- **Scope**

- **Enumeration:** DhcpV4Scope
- **Description:** The types of policies on which this option can be set. If no value is specified, then the option can be used for all types of policies.

- **Name**

- **Type:** Single
- **Description:** The name of the option as a text string.



- **Tag**
 - **Type:** Single
 - **Description:** A unique tag used to identify the option. This tag is encoded in the option value when the options are assigned in configuration objects as the first part of the value. It must NOT contain any spaces or the option parsing/validation will fail.
- **Alias**
 - **Description:** Alternate name(s) that can be used to assign values for the option. These names are never used to store the values or to configure the server. They are only present to assist the user in specifying values.
- **DefaultValue**
 - **Type:** Single
 - **Description:** The default value for this option.
- **LegalValue**
 - **Description:** The legal value(s) for this option.
- **Webtype**
 - **Type:** Single
 - **Description:** Determines the web editor used for value(s) of this option.
- **Webdata**
 - **Type:** Single
 - **Description:** Determines the webdata used with the web editor for value(s) of this option.
- **Format**
 - **Description:** The value(s) for this field define the format in which this option's value(s) will be written to the configuration file for the servers where it is set.
- **Example**
 - **Aliases:** Examples
 - **Description:** Contains examples of values for this option. This is used primarily for the help text describing the option.
- **Flags**
 - **Type:** Single



- **Description:** This is a string that consists of a series of keywords to specify the flags that apply to this option.
- **Creationtime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **Creator**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **LastModifiedTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone



must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.

- **InternalData**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.
- **Format**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.

4.4 Option82

NACF offers a flexible mechanism to extract the input data stream from Option82 based on the predefined input formats of virtual circuits or physical links, and convert the CLID or RemoteId to the new one based on the predefined output formats. Then the reformatted CLID or RemoteId are sent to CLF. To make the mechanism operational, `dhcipv4option82format` and `dhcipv4option82iprange` are introduced.

4.4.1 Dhcpv4Option82Format

This Dhcpv4Option82Format is used to store input format parsing rule of Option82 and format of the output message which will be sent to CLF

Key	Partition, Name
Required	Partition, Name, Category, Suboptionid, Informat, Outformat

- **Name:**
 - **Type:** Single



- **Data Type:** Name
- **Description:** The name of the dhcpv4Option82Format, a unique identifier for the object.
- **Category:**
 - **Type:** Single
 - **Description:** The input data stream's category, ASCII or binary: 0 for ASCII, 1 for binary. For different categories, a different parsing method will be used.
- **Suboptionid:**
 - **Type:** Single
 - **Description:** The sub-option id where the data is stored.
The value for this field can be any integer except 5.
- **Informat:**
 - **Type:** Single
 - **Description:** The rule to parse the inputting data format.
 - For ASCII category, the parsing rule is based on the standard regular expression that is used to match and fetch the specified data.
 - For binary category, the parsing rule is based on the private mechanism. The format, $\$ [num] (n.m,N.M)$, is described by the position specification, which provides the values for NACF.

Where:

 - $[num]$ represents the id of the variable.
 - n represents the start byte position.
 - m represents the start bit position of n .
 - N represents the end byte position.
 - M represents the end bit position of N .
- **Outformat:**
 - **Type:** Single
 - **Description:** This defines the rule to describe the output data format that is compatible with the protocol of the CLF. The output data format consists of variable $\$ [n]$.



where:

- For ASCII category, *n* represents the *n* subexpression in the parsing result.
- For binary category, *n* represents the [num] that is the id of the variable in the **informat**.

The output format is in accordance with the private protocol between NACF and CLF.

- For CLID, four characters# are used as tokens to separate different parts of the output data.
- For Remoteld, one character ! is used as a token to separate different parts. The output format is defined as \$1!\$2.

Examples:

- For ASCII category,
 - Input data: "ERX N+1:ATM11/0.715101:15.101"
 - **informat**: "^(.*) :ATM([0-9]{1,2})/([0-9]{1,2}) . ?[0-9]* : ([0-9]{1,3}) . ([0-9]{1,5}) \$"
 - **outformat**: "\$1#\$2#\$3#\$4#\$5"
 - Output data: "ERX N+1#11#0#15#101"
- For binary category,
 - Input data: "\x01\x00\x00\x00\x00\x00\x00\x00\x20\x25\x00\xaa"
 - **informat**: "\$1(5):\$2(6):\$3(7):\$4(8):\$5(9.1,9.4):\$6(9.6,9.8):\$7(10):\$8(11,12) "
 - **outformat**: "\$1.\$2.\$3.\$4#\$5#\$6#\$7#\$8"
 - Output data: "0.0.0.0#2#0#37#170"

- **Partition**

- **Type**: Single
- **Description**: The name of the partition that contains this object.

- **Creationtime**

- **Type**: Single
- **Read-only**: Yes



- **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **Creator**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **LastModifiedTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **InternalData**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes



- **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.
- **Format**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.

4.4.2 Dhcpv4Option82IPRange

The Dhcpv4Option82IPRange will be used to specify the rules used for each Relay Agent IP range.

Sometimes, two or more format rules can be used to parse the specific input data. However, only one of those format rules is the correct match for the specific input data. Therefore, the object `dhcpv4option82iprange` is introduced to determine which format rule applies to the input data from a certain IP range.

Key	Partition, Name
Required	Partition, Name, Address, Mask, MaskLength, Option82Format

- **Name:**
 - **Type:** Single
 - **Data Type:** Name
 - **Description:** The name and unique identifier of the `dhcpv4Option82IPRange`.
- **Address:**
 - **Type:** Single
 - **Description:** The address of the `dhcpv4Option82IPRange`.
- **Mask:**
 - **Type:** Single
 - **Description:** The subnet mask for this `dhcpv4Option82IPRange`.



- **MaskLength:**
 - **Type:** Single
 - **Data Type:** UInt8
 - **Description:** The subnet mask length for this dhcpv4Option82IPRange. This is a value between 0 and 32 indicating the number of bits that are set in the mask.
- **Server:**
 - **Description:** The name of the server(s) that use the dhcpv4Option82Format(s) in this IP range. This does not include the name of any secondary servers, only the masters.
- **Option82Format:**
 - **Description:** The Option82Format name list which will be used in this IP range.

When two format rules apply to the two IP ranges where one is contained in another, the format rule application follows a certain sequence. The IP range with the smaller address space has priority. When the two IP ranges have the same address space size, the one with the larger first octet has priority. If the first octets of the two IP ranges are the same, the one with the larger second octet has priority, and so forth. The format rule associated with the prioritized IP range is applied first.

An example is shown below, with the IP ranges being ordered by priority. Highest priority is first, followed by the second highest priority, and so on.

10.20.0.0/24 < 10.10.0.0/24 < 10.10.0.0/10 < 10.0.0.0/8 < 0.0.0.0/0

- **Partition**
 - **Type:** Single
 - **Description:** The name of the partition that contains this object.
- **Creationtime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **Creator**



- **Type:** Single
- **Read-only:** Yes
- **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **LastModifiedTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **InternalData**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.
- **Format**
 - **Type:** Single



- **Data Type:** Blob
- **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.

4.5 Dhcpv4Server

A Dhcpv4Server object represents an IPv4 DHCP server running on the network.

Key Partition, Name

Required Partition, Name

- **Primary**
 - **Aliases:** Master, MasterServer, PrimaryServer
 - **Type:** Single
 - **Description:** The name of the server that is the primary server for this server. If the server is part of a failover pair, this field is set to the name of the primary server on the secondary server only. If the server is not part of a failover pair, then this field should not be set. If the server is the primary server in a failover pair, this field should not be set.
- **FailoverAddress**
 - **Type:** Single
 - **Data Type:** IPv4Address
 - **Description:** The address that is used to listen for a connection from this server's failover peer. This should only be set if this server is part of a failover pair. If you do not explicitly set this, then it will be defaulted to the primary address.
 - **Examples:**

```
192.168.1.1
10.0.0.1
```
- **FailoverPort**
 - **Type:** Single
 - **Data Type:** UInt16



- **Description:** The port that is used to listen for a connection from this server's failover peer. This should only be set if this server is part of a failover pair. If you do not explicitly set this, it defaults to 647 for the secondary server and 847 for the primary server.
- **V4Option**
 - **Aliases:** Option, ConfigOption, Opt, V4ConfigOption, V4Opt.
 - **Data Type:** DhcpV4Option.
 - **Description:** DHCPv4 Configuration setting(s) specific to this server.
- **V4Config**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** The compiled configuration setting(s) specific to this server that will be used for the DHCPv4 protocol. These are determined from the Option field (and other fields) and cannot be modified directly.
- **FailoverConfig**
 - **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** The compiled failover setting(s) specific to this server if it's in a failover pair. This is determined from the Option field (and other failover related fields) and cannot be modified directly.
- **Filename**
 - **Type:** Single
 - **Description:** The name of the configuration file.
- **Area**
 - **Type:** Single
 - **Description:** This is the name of the area that will be used as the source for the policies to configure the server. The area represents an address-space and contains DHCP policies that represent how that address space should be managed by the DHCP protocol. This area must exist before the server is created.



Note: When dnsnames and addresses are assigned to the server, IPWorks does not automatically create any A and PTRRecords to bind the name and address together.

This should not be set if this is a secondary server (there is no value for the MasterServer field).

- **Option82Filename**
 - **Type:** Single
 - **Description:** The name of the Option82 format parsing rule configuration file.
- **EnableAuthentication**
 - **Type:** Single
 - **Data Type:** BooleanTrueFalse
 - **Description:** This flag is set to indicate if the authentication on the Server is enabled. If enabled, then the authentication keys are loaded on the server when "update" is issued.
 - **Examples:** yes/no or 1/0
- **KeysFilename**
 - **Type:** Single
 - **Description:** The name of the authentication keys file.
- **Name**
 - **Type:** Single
 - **Data Type:** Name
 - **Description:** This is a unique identifier for this server. It is not required to be the same as the primary dns name, but that value can be used. If DHCPv4 Servers are installed on the same system, the name must be unique for each server.
- **Address**
 - **Type:** Ordered
 - **Data Type:** Address
 - **Description:** The Server Manager use the addresses to establish a connection with the central IPWorks database to identify the corresponding server. Leaving the field unassigned causes the server to be unable to establish a connection correctly. One or more addresses can be specified for a server. If multiple addresses are



specified, the first address in the list is consider as the *primary* address for the server. When server manager is running, the address cannot be deleted but can be modified.

- **Examples:**

192.168.0.1

FE80::1.2.3.4

FE80::0102:0304

FE80:0000:0000:0000:0000:0102:0304

- **DnsName**

- **Type:** Ordered

- **Data Type:** DnsName

- **Description:** The dns name(s) for the server. During the configuration of this server, if more than one dnsname is specified, then the first dnsname in the list is used as the *primary* dnsname — this is the preferred name to use when contacting the server.

- **PrimaryDnsName**

- **Type:** Single

- **Data Type:** DnsName

- **Read-only:** Yes

- **Description:** The primary dns name for the server. This is simply the first name in the list of all dns names and is used any time the server's dns name is needed. It cannot be set directly, to change it simply change the DnsName field.

- **PrimaryAddress**

- **Aliases:** The alternative names of a field.

- **Type:** Single

- **Data Type:** Address

- **Read-only:** Yes

- **Description:** The primary address for the server. This is simply the first address in the list of all addresses and is used any time the server's address is needed. It cannot be set directly, to change it simply change the Address field.

- **Implementation**



- **Type:** Single
- **Description:** The server software implementation.
- **Status**
 - **Type:** Single
 - **Description:** The last known operational status of the server.
- **PostUpdateScript**
 - **Type:** Single
 - **Description:** The name and arguments for a shell script to be run during every update operation performed on the server. The script is run after new configuration files have been exported from the Storage Server and before the server has been reloaded. If an argument is a token enclosed in braces, {token}, symbol substitution will be performed on the token before the argument is passed to the script. Possible tokens are: ServerType, ServerIdentifier, ServerDirectory, StorageServerAddress, StorageServerPort, StorageServerUserName, StorageServerPassword, StorageServerPartition. Other arguments are passed directly to the script which must reside in `/var/ipworks/scripts` or another configured scripts directory. See the Sm.ScriptsDirectory configuration property for the Server Manager.
- **Type**
 - **Type:** Single.
 - **Read-only:** Yes
 - **Description:** The type of server that is initialized when the object is created based on the storage class of the object. This is provided primarily for querying purposes.
- **ExportNeeded**
 - **Type:** Single
 - **Data Type:** BooleanTrueFalse
 - **Description:** This flag is set to indicate if the data for this server needs to be exported. If set to 'true' it means that the central database contains data that has not yet been exported. If it has no value (or is set to 'false') then the data in the central database is the same data that the server is providing on the live network.
- **Partition**
 - **Type:** Single
 - **Description:** The name of the partition that contains this object.



- **Creationtime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **Creator**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **LastModifiedTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **InternalData**



- **Type:** Single
- **Data Type:** Blob
- **Read-only:** Yes
- **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.
- **Format**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.

4.6 Lease

A lease object represents an actively leased address in the DHCP server. Lease objects are read-only objects: they cannot be created or modified. They are not used for configuration and are only created by the normal processing of the DHCP server. In other words, they are managed by and stored in the DHCP server, not the central IPWorks database.

Key Partition, Server, Address

Required Partition, Server, Address

- **Partition**
 - **Type:** Single
 - **Description:** The name of the partition that contains this object.
- **Server**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The name of the server that issued this lease.
- **Address**
 - **Type:** Single



- **Read-only:** Yes
 - **Description:** The address, which is a DHCPv4 address, depending on which server is managing the lease.
- **State**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The state of the address. This is a constant value that is assigned by the server to represent how the address is being used.
- **ClientIdentifier**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The DHCPv4 Client Identifier for the client that leased this address.
- **HardwareAddress**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The Hardware Address for the client that leased this address. This is, usually, a 7 byte value, where the first byte designates the hardware type, and the remaining 6 bytes specify the MACAddress for the client. The hardware type codes are 01 for Ethernet, 06 for Token Ring, and 08 for FDDI.
- **AssignedHostName**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The host name that was assigned to the client. This may be different from the name that was requested by the client, depending on how the server is configured.
- **RequestedHostName**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The host name that was requested by the client.
- **RequestedDomain**



- **Type:** Single
- **Read-only:** Yes
- **Description:** The domain in which the client requests a name. This is used to determine the FQDN that was associated with the client.
- **FQDN**
 - **Aliases:** DnsName
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The "Fully Qualified Domain Name" that was assigned to the client. This only has a value if DDNS was enabled for the client when an address was assigned.
- **StartTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time at which the lease was granted to the client.
- **EndTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time at which the lease will expire.
- **IAID:**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The identity association identifier.
- **IAType:**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The identity association type.
- **Lifetime:**
 - **Type:** Single



- **Read-only:** Yes
- **Description:** The lifetime of the binding between the address and the DUID.
- **PreferredLifetime:**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The preferred lifetime of the binding between the address and the DUID.
- **MaximumLifetime:**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The maximum lifetime of the binding between the address and the DUID.
- **LinkLocalAddress:**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The link local address of the machine that has this lease. This is a non-routeable IPv6 address.
- **Prefix:**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The address prefix for the IPv6 address for this lease.
- **PoolIndex:**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The Pool Index for this lease.
- **PoolName:**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The pool lease to which this lease belongs.



- **Creationtime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time this object was created, based on the LDAPv3 "createTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified, and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **Creator**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The creator of this object, based on the LDAPv3 "creatorsName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **Description**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This attribute contains a human-readable description of the object, based on the X.500 "description" attribute (RFC2256).
- **LastModifiedBy**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The last modifier of this object, based on the LDAPv3 "modifiersName" attribute (RFC2252). The value is a distinguished name. This field can only be set by the server.
- **LastModifiedTime**
 - **Type:** Single
 - **Read-only:** Yes
 - **Description:** The time object was last modified, based on the LDAPv3 "modifyTimestamp" attribute (RFC2252). Values are encoded as printable strings, represented as specified in X.208. The time zone must be specified and it is strongly recommended that GMT time be used. This field can only be set by the server.
- **InternalData**



- **Type:** Single
 - **Data Type:** Blob
 - **Read-only:** Yes
 - **Description:** This is used for keeping track of internal data that is specific to this object. The value for this field is managed as a single string that has the same format as a Java properties file.
- **Format**
 - **Type:** Single
 - **Data Type:** Blob
 - **Description:** This can be used to define the object-specific CONF format. The CONF format is the format that is used to configure this object in the server(s). The value of this field is a template of the text that should be inserted into the configuration file(s) for this object, where the xml tags embedded in the text will be replaced based on format definitions for those tags.



5 DHCP Expressions

Three types of DHCP expressions are shown as follows:

- Boolean Expressions
- Data Expressions (that evaluate to a string value)
- Numeric Expressions.

5.1 Boolean Expressions

The following is the current supported list of boolean expressions. All boolean expressions depend on the results of evaluating data expressions.

- `data-expression-1 = data-expression-2`

The `=` operator compares the values of two data expressions, returning true if they are the same, false if they are not. If either the left-hand side or the right-hand side are null, the result is also null.

- `boolean-expression-1 and boolean-expression-2`

The `and` operator evaluates to true if the boolean expression on the left-hand side and the boolean expression on the right-hand side both evaluate to true. Otherwise, it evaluates to false. If either the expression on the left-hand side or the expression on the right-hand side are null, the result is null.

- `boolean-expression-1 or boolean-expression-2`

The `or` operator evaluates to true if either the boolean expression on the left-hand side or the boolean expression on the right-hand side evaluate to true. Otherwise, it evaluates to false. If either the expression on the left-hand side or the expression on the right-hand side are null, the result is null.

- `not boolean-expression`

The `not` operator evaluates to true if `boolean-expression` evaluates to false and returns false if `boolean-expression` evaluates to true. If `boolean-expression` evaluates to null, the result is also null.

- `! boolean-expression`

The `!` operator behaves exactly the same as the `not` operator, except for the handling of the null value. If the `boolean-expression` evaluates to null, the result is false.



- `exists option-name`

The `exists` expression returns true if the specified option exists in the incoming DHCP packet being processed.

- `known`

The `known` expression returns true if the client whose request is currently being processed is known - that is, if there is a host declaration for it.

5.2 Data Expressions

A list of data expressions is provided below. The boolean expressions listed above depend on the results of evaluating data expressions.

- `Substring (data-expr, offset, length)`

The `substring` operator evaluates the data expression and returns the substring of the result of that evaluation that starts offset bytes from the beginning, continuing for length bytes. Offset and length are both numeric expressions. If data-expr, offset or length evaluate to null, then the result is also null. If offset is greater than or equal to the length of the evaluated data, then a zero-length data string is returned. If length is greater than the remaining length of the evaluated data after offset, then a data string containing all data from offset to the end of the evaluated data is returned.

- `Suffix (data-expr, length)`

The `suffix` operator evaluates data-expr and returns the last length bytes of the result of that evaluation. Length is a numeric expression. If data-expr or length evaluate to null, then the result is also null. If suffix evaluates to a number greater than the length of the evaluated data, then the evaluated data is returned.

- `Option option-name`

The `option` operator returns the contents of the specified option in the packet to which the server is responding.

- `Hardware`

The `hardware` operator returns a data string whose first element is the type of network interface indicated in packet being considered and whose subsequent elements are client's link-layer address. If there is no packet, or if the RFC2131 hlen field is invalid, then the result is null. Hardware types include ethernet (1), token-ring (6) and fddi (8). Hardware types are specified by the IETF and details on how the type numbers are defined can be found in RFC2131.

- `Packet (offset, length)`



The `packet` operator returns the specified portion of the packet being considered, or null in contexts where no packet is being considered. Offset and length are applied to the contents packet as in the substring operator.

- String

A string, enclosed in quotes, may be specified as a data expression and returns the text between the quotes, encoded in ASCII. The backslash `\` character is treated specially, as in C programming: `\t` means TAB, `\r` means carriage return, `\n` means newline and `\b` means bell. Any octal value can be specified with `"\nnn"`, where `nnn` is any positive octal number less than 0400. Any hexadecimal value can be specified with `'\xnn'`, where `nn` is any positive hexadecimal number less than 0xff.

- Colon-separated hexadecimal list

A list of hexadecimal octet values, separated by colons, may be specified as a data expression.

- Concat (data-expr1, ..., data-exprN)

The expressions are evaluated and the results of each evaluation are concatenated in the sequence that the sub expressions are listed. If any sub-expression evaluates to null, the result of the concatenation is null.

- Reverse (numeric-expr1, data-expr2)

The two expressions are evaluated and then the result of evaluating the data expression is reversed in place, using portions of the size specified in the numeric expression. For example, if the numeric expression evaluates to four and the data expression evaluates to twelve bytes of data. The `reverse` expression will evaluate to twelve bytes of data, consisting of the last four bytes of the input data, followed by the middle four bytes, followed by the first four bytes.

- Binary-to-ascii (numeric-expr1, numeric-expr2, data-expr1, data-expr2)

Converts the result of evaluating data-expr2 into a text string, which contains one number representing each element of the result. Each number is separated from the other by the data-expr1 result. The result of evaluating numeric-expr1 specifies the base (2 through 16) into which the numbers are to be converted. The result of evaluating numeric-expr2 specifies the width in bits of each number, which may be either 8, 16 or 32.

- Encode-int (numeric-expr, width)

Numeric-expr is evaluated and encoded as a data string of the specified width, in network byte order (most significant byte first). If the numeric expression evaluates to the null value, the result is also null.

- Pick-first-value (data-expr1 [... exprn])



The `pick-first-value` function takes any number of data expressions as its arguments. Each expression is evaluated, starting with the first in the list, until an expression is found that does not evaluate to a null value. That expression is returned and none of the subsequent expressions are evaluated. If all expressions evaluate to a null value, the null value is returned.

- `Host-decl-name`

The `host-decl-name` function returns the name of the host declaration that matched the client whose request is currently being processed, if any. If no host declaration matched, the result is the null value.

5.3 Numeric Expressions

Numeric expressions are expressions that evaluate to an integer. In general, the maximum size of such an integer should not be assumed to be represented in less than 32 bits, but the precision of such integers may be more than 32 bits. This can be elaborated as in the following:

- `Extract-int (data-expr, width)`

The `extract-int` operator extracts an integer value in network byte order from the result of evaluating the specified data expression. Width is the width in bits of the integer to extract. Currently, the only supported widths are 8, 16 and 32. If the evaluation of the data expression does not provide sufficient bits to extract an integer of the specified size, the null value is returned.

- `Lease-time`

The duration of the current lease—that is, the difference between the current time and the time that the lease expires.

- `Number`

Any number between zero and the maximum that may be the represented size specified as a numeric expression.



Reference List

IPWorks Library Documents

- [1] *Command Line Interface User Guide for IPWorks SS*
- [2] *Trademark Information*
- [3] *Glossary of Terms and Acronyms*
- [4] *Typographic Conventions*
- [5] *IPWorks Configuration Management*
- [6] *IPWorks DNS, ASDNS, ENUM Parameter Description*