

March 2000  
Expires September 2000

DHCP Schema for LDAP  
<[draft-ietf-dhc-schema-02.txt](#)>

### Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

### Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

### Abstract

This document presents an LDAP schema to represent the configuration of the DHCP protocol within a TCP/IP network. It can be used to represent the configuration(s) of an entire enterprise network, a subset of the network, or even a single server.

### 1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

In places where different sets of terminology are commonly used to represent similar DHCP concepts, this schema uses the terminology of the Internet Software Consortium's DHCP server reference implementation. For more information see [www.isc.org](http://www.isc.org).

## 2. Design Considerations

Some of the design considerations for this schema were:

- o Heterogeneous server environment - This schema is not designed to represent the configuration of a specific DHCP server implementation. The intent of this schema is to provide a basic framework for the representation of the most common elements used in the configuration of DHCP. This should allow other network services to obtain and use basic DHCP configuration information in a server-independent way. Also note that it is highly unlikely that this schema will be able to represent every feature of every implementation (and it is not intended to do so). It is expected that some implementations may need to extend the schema objects in order to fully implement all their features.
- o Use of the schema - This draft does not define any "minimal compliance criteria" for using the schema. It is recommended that you use the object classes defined in this draft if you are representing DHCP configuration information in an LDAP directory. Some implementations may choose not to support all of the objects defined here. In particular, the following two decisions are explicitly left up to the implementation:
  - it is up to the implementation to determine whether or not the lease information will be stored in the directory. Some implementations may choose not to store this information.
  - it is up to the implementation to determine if the data in the directory is considered "authoritative", or if it is simply a copy of data from an authoritative source.
- o The schema is focused on the representation of configuration information. It does not provide for the representation of statistical data, or historical lease data, only the current state of the DHCP protocol's configuration.
- o The information in this schema will be used primarily by two types

of applications: DHCP servers (for loading their configuration) and Management Interfaces (for defining/editing configurations). The schema should must be efficient for the needs of both types of applications.

- o The schema is designed to allow objects managed by DHCP (such as computers, subnets, etc) to be present anywhere in a directory hierarchy (to allow those objects to be placed in the directory for managing administrative control and access to the objects). However, the schema also provides for the possibility that any given object may have multiple sets of configuration parameters defined for different servers.
- o The schema uses a few naming conventions - all object classes and attributes are prefixed with "dhcp" and there are no object classes and attributes that have the same name. The schema also uses standard naming attributes ("cn", "ou", etc) for all objects. In some cases it is recommended that the "cn" matches another attribute value.
- o Relationship to DEN/DMTF - This document takes into consideration the object-oriented information model for representing Network information (including DHCP information) currently under development as part of the Common Information Model (CIM) activity in the Distributed Management Task Force (DMTF). It should be noted that the CIM schema is still under development and subject to change. The DMTF efforts continue and draw upon the Directory-Enabled Networks (DEN) specification. The schema described in this Internet-Draft is intended to be an LDAP implementation of the appropriate objects in the DMTF model. The DMTF schema was used as a source for defining certain terminology within this schema. For more information see [DMTF] and [DEN]. Prior versions of this draft included a mapping between the two schemas, but this has been removed since the DMTF schema is still under development. When it is complete a new draft may be published to document the mapping between the schemas.
- o Relationship to Policy Framework working group - Much of the information in this schema could be represented using the generalized schema being developed by the Policy Framework. However, there were two issues that we felt would make this a very complex and

most likely inefficient representation: (1) the complexity of the inheritance relationships between the dhcp policy objects defined in this document and (2) the Policy Framework schema represents each of the conditions and actions of a policy as separate objects. However, it is still a fairly straightforward process to map the objects from this schema into the Policy Framework Core Schema objects. For more information see [POLICY].

### [3.](#) Common Attributes

Although DHCP manages several different types of objects, the configuration of those objects is often very similar. Consequently, most

of these objects have a common set of attributes.

The dhcpConfigurableObject class is an auxiliary class which can be used to associate the basic set of configuration attributes with another object. Since some directories do not provide auxiliary classes we have also repeated these common attributes in the definition of each of the DHCP object class definitions.

An implementation of this schema is not required to provide this auxiliary object class, but it SHOULD provide it if auxiliary classes are supported. This is useful for associating DHCP configuration settings for objects that are not directly defined as part of this schema.

#### [3.1.](#) dhcpConfigurableObject Object Class

NAME	dhcpConfigurableObject
DESCRIPTION	A class that provides attributes for configuring options and server parameters for DHCP.
TYPE	Auxiliary
DERIVED FROM	Top
POSSIBLE SUPERIORS	( )
MUST CONTAIN	( )
MAY CONTAIN	( dhcpOptionSetting dhcpParameterSetting dhcpFieldSetting dhcpForcedOptions dhcpIncludeOptionSet )

#### [3.2.](#) Common Attribute Definitions

NAME dhcpOptionSetting  
DESCRIPTION Encoded option values to be sent to clients. Each value represents a single option and contains (OptionTag, Length, OptionValue) encoded in the 16-bit format used by DHCP. For more information see [DHCP OPT].  
SYNTAX OctetString MULTI-VALUE

NAME dhcpParameterSetting  
DESCRIPTION Encoded values of parameters that control server behavior. Each value represents a single parameter setting in the form (ParameterName, ParameterValue) where the parameter name is a set of ASCII characters followed by a space followed by the parameter value as a string.  
SYNTAX IA5String MULTI-VALUE

NAME dhcpFieldSetting  
DESCRIPTION Encoded settings of fields (such as siaddr, file) in the DHCP message whose values may be configurable for sending back to a client. For more information see [[RFC951](#)]. Encoded in the form (FieldName, FieldValue) where the field name is a set of ASCII characters followed by a space followed by the field value as a string.

SYNTAX IA5String MULTI-VALUE

NAME dhcpForcedOptions  
DESCRIPTION This is a list of DHCP option tags that MUST be sent to clients. If not specified, the server only sends the options back to the client which were requested.

SYNTAX Integer MULTI-VALUE

NAME dhcpIncludeOptionSet  
DESCRIPTION The distinguished name(s) of dhcpNamedOptionSet objects whose settings should be included for this object. If there are multiple option sets, the order is important so each value is preceded by it's precedence, followed by a colon as in "1:dn1", "2:dn2", etc. Settings

defined on the object take precedence over any settings found in an included option set.

SYNTAX IA5String MULTI-VALUE

#### [4.](#) Configurations and Services

The DHC working group is currently considering several proposals for failover and redundancy of DHCP servers. These may require the sharing of configuration information between servers. This schema provides a generalized mechanism for supporting any of these proposals, by separating the definition of a server from the definition of the configuration being provided by the server.

By separating these two concepts, a configuration may be provided by one or by several servers, and similarly, a server may provide one or more configurations. The schema does allow for a server to be configured as either a primary or secondary provider of a configuration.

Configurations are also defined so that one configuration can include some of the objects that are defined in another configuration (see "dhcpIncludeObjects" attribute). This allows for sharing and/or a hierarchy of related configuration items.

##### [4.1.](#) dhcpService Object Class

A "dhcpService" is a single instance of DHC server software running on a computer system that provides the DHCP service defined by a "dhcpConfiguration".

NAME	dhcpService
DESCRIPTION	This represents a single DHCP server.
TYPE	Structural
DERIVED FROM	Top
POSSIBLE SUPERIORS	( )
MUST CONTAIN	( cn )
MAY CONTAIN	( dhcpConfigurationDn dhcpImplementation )

##### [4.1.1.](#) dhcpService Attribute Definitions

NAME cn  
DESCRIPTION The "common name" of the server. This does not have any significance to the server process that provides the DHCP service - it is simply a unique name used to refer to the server. This attribute should be used as the naming attribute when constructing the dn.

NAME dhcpConfigurationDn  
DESCRIPTION The distinguished name(s) of the configurations provided by the server.  
SYNTAX DN MULTI-VALUE

NAME dhcpImplementation  
DESCRIPTION This is a string value that identifies the hardware/software platform and version which is providing the service.  
SYNTAX IA5String SINGLE-VALUE

#### [4.2.](#) dhcpConfiguration Object Class

A "dhcpConfiguration" is the collection of configuration information that represents everything a server would need to know to provide DHC service to some set of clients.

From the perspective of the schema, it is basically a collection of objects. This object class is used to capture information common to all the objects in a configuration. The algorithm used to locate all the objects in a configuration is discussed later.

NAME dhcpConfiguration  
DESCRIPTION This represents a configuration, or a collection of settings for related objects. A single service may have multiple configurations. A configuration may be provided by multiple services, but only one can be primary.  
TYPE Structural  
DERIVED FROM Top

```
POSSIBLE SUPERIORS ( )
MUST CONTAIN      ( cn )
MAY CONTAIN       ( dhcpPrimaryService dhcpSecondaryService
                  dhcpIncludeObjects dhcpOptionSetting
                  dhcpParameterSetting dhcpFieldSetting
                  dhcpForcedOptions dhcpIncludeOptionSet )
```

#### 4.2.1. dhcpConfiguration Attribute Definitions

```
NAME      cn
DESCRIPTION The "common name" of the configuration. This should be
           used as the naming attribute when constructing the dn.
```

```
NAME      dhcpPrimaryService
DESCRIPTION The "dhcpService" which is the primary for the configuration.
SYNTAX    DN SINGLE-VALUE
```

```
NAME      dhcpSecondaryService
DESCRIPTION The "dhcpService(s)" which provide backup for the configuration.
SYNTAX    DN MULTI-VALUE
```

```
NAME      dhcpIncludeObjects
DESCRIPTION This attribute defines objects that are included in a
           configuration. Each value is an LdapURL [RFC2255]
           (specifying search criteria) that is evaluated to find
           other objects that are included in this configuration.
           Note that in addition to these objects, all objects that
           are children of the configuration object in the directory
           are automatically included in the configuration.
SYNTAX    IA5String MULTI-VALUE
```

#### 5. Objects that represent Assignment Rules

Most of a DHCP configuration is the definition of rules that govern the assignment of DHCP options and addresses to clients. This schema defines a set of object classes which are common to most server implementations for defining these rules. All of these object classes are based on a higher level abstraction that represents a

This is done for several reasons: it simplifies the organization of the data and it also facilitates the mapping of the dhcp schema to other schemas being developed in other working groups (see [POLICY] for example).

This schema separates the definition of an assignment rule for an object from the object itself. This allows for the definition of multiple rules for a single object (possibly in different dhcp configurations). However, each assignment rule does maintain a link back to the definition of the object (see the "dhcpSourceObject" attribute).

The structure of a "source object" is not defined in this schema. It can be any LDAP object, and it is not required to even exist. However, if it does exist that object can use the "dhcpConfigurableObject" auxiliary class to directly associate dhcp configuration information with that object. If an object is defined in this way, this information is used on every assignment rule that references the object.

The assignment rule objects in the directory can be organized in a hierarchical fashion. If objects are organized this way, the "child" rule object inherits settings from the "parent" rule object. This can be done recursively. Furthermore, the "child" rule object inherits any conditions from the "parent" rule as well. This means that the "child" rule's settings will only be used when both sets of conditions are met.

As an example, if a "dhcpClass" is a child of a "dhcpSubnet" then the settings for that class will only be used if the client request is a member of that class AND it is also from the specified subnet.

The algorithm for resolving which option settings are applied for a policy object is defined in a later section.

### [5.1.](#) dhcpRule Object Class

The "dhcpRule" class is an abstract class that defines attributes that are common to the DHCP configuration objects that define these rules.

This class is the base class from which others are derived. Also note that it includes all the attributes from the dhcpConfigurableObject class.

---

NAME	dhcpRule
DESCRIPTION	The base class for defining rules for address and option assignment.
TYPE	Abstract
DERIVED FROM	Top
POSSIBLE SUPERIORS	( )
MUST CONTAIN	( cn dhcpRuleType )
MAY CONTAIN	( dhcpVendorCondition dhcpSourceObject dhcpOptionSetting dhcpParameterSetting dhcpFieldSetting dhcpForcedOptions dhcpIncludeOptionSet )

### [5.2.](#) dhcpRule Attribute Definitions

NAME	cn
DESCRIPTION	The "common name" of the rule. This should be used as the naming attribute when constructing the dn.
NAME	dhcpRuleType
DESCRIPTION	The type of assignment rule. This should be one of 'POOL', 'SUBNET', 'SHAREDNETWORK', 'CLIENT', 'CLASS' unless the server implementation extends this with a new type of rule.
SYNTAX	IA5String SINGLE-VALUE
NAME	dhcpVendorCondition
DESCRIPTION	If the server extends the rule types, this attribute MAY be used to specify the conditions under which the rule should be applied. The content of this attribute is defined by the vendor/server implementation.
SYNTAX	IA5String MULTI-VALUE
NAME	dhcpSourceObject
DESCRIPTION	If the rule applies to an object that is defined elsewhere in the directory, this attribute has the distinguished name of that object. The source object SHOULD be a dhcpConfigurableObject. Also note that the source object is used to determine option & parameter settings (see the algorithm discussed later in this document).
SYNTAX	DN SINGLE-VALUE

### [5.3.](#) dhcpPool Object Class

A "dhcpPool" represents a rule for a collection of addresses specified by one or more ranges of addresses. If there are multiple ranges specified, they do not need to be contiguous, and it is not

required that all the addresses be contained on the same IP subnet.

The "dhcpRuleType" attribute MUST be set to 'POOL', and the "cn" SHOULD be set to the value of the "dhcpPoolName" attribute".

NAME	dhcpPool
DESCRIPTION	This stores configuration information about one (or more) ranges of addresses.
TYPE	Structural
DERIVED FROM	dhcpRule
POSSIBLE SUPERIORS	( OrganizationalUnit dhcpRule )
MUST CONTAIN	( cn dhcpPoolName dhcpAddressRange )
MAY CONTAIN	( )

#### [5.3.1.](#) dhcpPool Attribute Definitions

NAME	dhcpPoolName
DESCRIPTION	A descriptive name of the pool.
SYNTAX	IA5String SINGLE-VALUE

NAME	dhcpAddressRange
DESCRIPTION	The starting & ending IP Addresses in the range (inclusive), separated by a hyphen; if the range only contains one address, then just the address can be specified with no hyphen. Each range is defined as a separate value.
SYNTAX	IA5String MULTI-VALUE

#### [5.4.](#) dhcpSubnet Object Class

A "dhcpSubnet" represents an assignment rule for an IP subnet.

The "dhcpRuleType" attribute MUST be set to 'SUBNET', and the "cn" SHOULD be set to the value of the "dhcpSubnetName" attribute".

NAME	dhcpSubnet
DESCRIPTION	This class defines a subnet.
TYPE	Structural
DERIVED FROM	dhcpRule
POSSIBLE SUPERIORS	( OrganizationalUnit dhcpRule )
MUST CONTAIN	( cn dhcpSubnetAddress dhcpSubnetMaskLength dhcpSubnetName )

MAY CONTAIN ( )

#### [5.4.1.](#) dhcpSubnet Attribute Definitions

NAME dhcpSubnetAddress  
DESCRIPTION The network address for the subnet.  
SYNTAX IA5String SINGLE-VALUE

Bennett, et. al.

Expires September 2000

[Page 10]

---

Internet Draft

DHCP Schema for LDAP

March 2000

NAME dhcpSubnetMaskLength  
DESCRIPTION The subnet mask length for the subnet. The mask can be easily computed from this length.  
SYNTAX Integer SINGLE-VALUE

NAME dhcpSubnetName  
DESCRIPTION A descriptive name of the subnet.  
SYNTAX IA5String SINGLE-VALUE

#### [5.5.](#) dhcpSharedNetwork Object Class

A "dhcpSharedNetwork" represents an assignment rule for multiple subnets on the same physical cabling.

The "dhcpRuleType" attribute MUST be set to 'SHAREDNETWORK', and the "cn" SHOULD be set to the value of the "dhcpSharedNetworkName" attribute.

NAME dhcpSharedNetwork  
DESCRIPTION This represents multiple subnets on the same physical cabling.  
TYPE Structural  
DERIVED FROM dhcpRule  
POSSIBLE SUPERIORS ( OrganizationalUnit dhcpRule )  
MUST CONTAIN ( cn dhcpSharedNetworkName )  
MAY CONTAIN ( )

#### [5.5.1.](#) dhcpSharedNetwork Attribute Definitions

NAME dhcpSharedNetworkName  
DESCRIPTION A descriptive name of the shared network.  
SYNTAX IA5String SINGLE-VALUE

## [5.6.](#) dhcpClient Object Class

The "dhcpClient" object class is used to store configuration information related to a specific host.

The "dhcpRuleType" attribute MUST be set to 'CLIENT'.

NAME	dhcpClient
DESCRIPTION	This represents client-specific DHCP assignments.
TYPE	Structural
DERIVED FROM	dhcpRule
POSSIBLE SUPERIORS	( OrganizationalUnit dhcpRule )
MUST CONTAIN	( cn dhcpClientIdentifier )
MAY CONTAIN	( dhcpClassMember dhcpReservedAddress )

### [5.6.1.](#) dhcpClient Attribute Definitions

NAME	dhcpClientIdentifier
DESCRIPTION	A unique identifier for the client. This is encoded as follows: the first two octets represent a type and subtype for the identifier. If the type field has a value of 0, then the subtype is a dhcp option tag, and the remainder of the octets are the value of that option to use as an id (represented as it would be sent using the DHCP protocol, including the bytes for the length). If the type field has a value of 1, then the subtype octet is the ARP hardware type (see [ <a href="#">RFC2132</a> ]) and the remainder of the bytes are the hardware address. Server implementations may choose to extend the set of types, but these two MUST be recognized. Note that a client can have more than one unique identifier specified - it is left to the server implementation to decide if one or all identifiers must be matched or which take precedence over others.
SYNTAX	OctetString MULTI-VALUE

NAME	dhcpClassMember
DESCRIPTION	This attribute indicates that the client is a member of the specified class(es).
SYNTAX	IA5String MULTI-VALUE

NAME	dhcpReservedAddress
DESCRIPTION	This attribute indicates the reserved (aka fixed) address(es) for this client (if there are any). There MAY be corresponding "dhcpAddress" objects created for tracking this reservation.
SYNTAX	IA5String MULTI-VALUE

### [5.7.](#) dhcpClass Object Class

A "dhcpClass" represents information about a collection of clients. The DHC protocol provides 2 mechanisms for managing this information (User Class and Vendor Class). The schema also provides 2 additional mechanisms for configuring groups of clients that are supported by some servers. Clients may be explicitly added to a class by setting the "dhcpClassMember" attribute in the "dhcpClient" object class. Some servers also support forms of dynamic class membership beyond the User Class and Vendor Class mechanisms - the "dhcpVendorCondition" attribute allows for the definition of dynamic classes.

The "dhcpRuleType" attribute MUST be set to 'CLASS', and the "cn" SHOULD be set to the value of the "dhcpClassName" attribute".

NAME	dhcpClass
DESCRIPTION	Represents information about a collection of related clients.
TYPE	Structural
DERIVED FROM	dhcpRule
POSSIBLE SUPERIORS	( OrganizationalUnit dhcpRule )
MUST CONTAIN	( cn dhcpClassName dhcpClassType)
MAY CONTAIN	( )

#### [5.7.1.](#) dhcpClass Attribute Definitions

NAME	dhcpClassName
DESCRIPTION	A descriptive name for the class.
SYNTAX	IA5String SINGLE-VALUE

NAME	dhcpClassType
DESCRIPTION	This attribute indicates the type of the class. It should be one of 'USERCLASS', 'VENDORCLASS', 'STATIC'

(the only members of the class are enumerated clients),  
'DYNAMIC' (membership is determined by some vendor-specific conditions).

SYNTAX IA5String SINGLE-VALUE

## 6. Other Configuration objects

Many server implementations provide other objects that simplify the configuration of the DHCP protocol. One example is the ability to assign a name to a group of option settings and then to refer to the entire group of settings by referencing the name. This is addressed by the "dhcpNamedOptionSet" object class.

It is also fairly common for server implementations to allow users to extend the default set of options with site specific option definitions. This is addressed by the "dhcpDictionary" object class. This object class is also used to define the implementation-specific parameters (and their values) that can be specified in the "dhcpParameterSetting" attribute.

### 6.1. dhcpNamedOptionSet Object Class

A "dhcpNamedOptionSet" is an object class for associating a name with a collection of option settings. The entire set of options can be associated with a DHCP object by referring to the name. This allows a common set of option settings to be re-used without repeating the option settings on each configured object. To see how an option set is referenced, see the "dhcpIncludeOptionSet" attribute.

NAME	dhcpNamedOptionSet
DESCRIPTION	This is a named collection of settings for options and/or server parameters.
TYPE	Structural
DERIVED FROM	Top
POSSIBLE SUPERIORS	( OrganizationalUnit )
MUST CONTAIN	( cn )
MAY CONTAIN	( dhcpOptionSetting dhcpParameterSetting dhcpFieldSetting dhcpForcedOptions dhcpIncludeOptionSet )

### 6.1.1. dhcpDictionary Object Class

"dhcpDictionary" objects define the options and/or parameters that can be set when configuring various DHCP entities.

NAME	dhcpDictionary
DESCRIPTION	This class defines an option or parameter that can have a value.
TYPE	Structural
DERIVED FROM	Top
POSSIBLE SUPERIORS	( OrganizationalUnit )
MUST CONTAIN	( cn dhcpTag )
MAY CONTAIN	( dhcpDisplayName dhcpDataType dhcpDefault dhcpMultiValued dhcpLegalValues dhcpTypeRestriction dhcpImplementation )

### 6.2. dhcpDictionary Attribute Definitions

NAME	cn
DESCRIPTION	The "common name" of the option or parameter. This will usually be the same as the "dhcpTag" attribute.
SYNTAX	Integer SINGLE-VALUE

NAME	dhcpTag
DESCRIPTION	A unique value that identifies an option or parameter and that is encoded in the values of the "dhcpOptionSetting" and "dhcpParameterSetting" attributes. For options this SHOULD be the numeric tag for the option (stored as a string).
SYNTAX	IA5String SINGLE-VALUE

NAME	dhcpDisplayName
DESCRIPTION	This is a string identifier for the option or parameter. This is intended for display by a management tool or GUI.
SYNTAX	IA5String SINGLE-VALUE

NAME	dhcpDataType
DESCRIPTION	The data type for values of this option. One of the following: 'INT8', 'INT16', 'INT32', 'UINT8', 'UINT16', 'UINT32', 'ADDRESS', 'ADDRESS-MASK-PAIR', 'BOOLEAN',

'STRING', 'BINARY'. Other values may be specified if the server implementation provides them.

SYNTAX IA5String SINGLE-VALUE

NAME dhcpDefault

DESCRIPTION Indicates the default value of a parameter or option definition in a dictionary object. This is encoded as it would be in the "dhcpOptionSetting" or "dhcpParameterSetting" attribute.

SYNTAX OctetString SINGLE-VALUE

NAME dhcpMultiValued

DESCRIPTION Indicates whether the parameter or option can have more than one value.

SYNTAX Boolean SINGLE-VALUE

NAME dhcpLegalValues

DESCRIPTION The list of allowed values for the option or parameter. Each "legal value" is stored as a separate value for the attribute, and is encoded based on the "dhcpDataType" attribute setting.

SYNTAX OctetString MULTI-VALUE

NAME dhcpTypeRestriction

DESCRIPTION This attribute is used to specify that the option or parameter should only be used with specific types of assignment rules. This is restricted to the same set of values as the "dhcpRuleType" attribute. If not specified it is assumed that the definition applies to all types.

SYNTAX IA5String MULTI-VALUE

NAME dhcpImplementation

DESCRIPTION This attribute is used to specify that the option or parameter should only be used with specific server implementations. If not specified it is assumed that the definition applies to all implementations.

SYNTAX IA5String MULTI-VALUE

## 7. Tracking Addresses

The behavior of a DHCP server is influenced by two factors - it's configuration and the current state of the addresses that have been assigned to clients. This schema defined a set of objects for

storing the configuration of the server, and the following object class provides the ability to record how addresses are used.

### [7.1.](#) dhcpAddress Object Class

This class represents an IP address. It may or may not be leaseable, and the object may exist even though a lease is not currently active for the associated IP address.

Note that this object class has some of the "Settings" attributes that are defined for the "dhcpConfigurableObject", but they are not used for configuration - only for tracking the settings that were assigned to the client. It is not required that the server implementation record options that were offered to the client.

NAME	dhcpAddress
DESCRIPTION	This class represents an IP Address, which may or may not have been leased.
TYPE	Structural
DERIVED FROM	Top
POSSIBLE SUPERIORS	( )
MUST CONTAIN	( cn dhcpAddressState )
MAY CONTAIN	( dhcpExpirationTime dhcpStartTimeOfState dhcpLastTransactionTime dhcpBootpFlag dhcpDomainName dhcpDnsStatus dhcpRequestedHostName dhcpAssignedHostName dhcpReservedForClient dhcpAssignedToClient dhcpRelayAgentInfo dhcpOptionSetting dhcpParameterSetting dhcpFieldSetting )

### [7.2.](#) dhcpAddress Attribute Definitions

NAME	cn
DESCRIPTION	The IP address, as a string.
SYNTAX	IA5String SINGLE-VALUE

Internet Draft

DHCP Schema for LDAP

March 2000

---

NAME dhcpAddressState  
DESCRIPTION This stores information about the current binding-status of an address. For dynamic addresses managed by DHCP, the values should be restricted to the states defined in the safe-failover draft: 'FREE', 'ACTIVE', 'EXPIRED', 'RELEASED', 'RESET', 'ABANDONED', 'BACKUP'. For more information on these states see [FAILOVR]. For other addresses, it SHOULD be one of the following: 'UNKNOWN', 'RESERVED' (an address that is managed by DHCP that is reserved for a specific client), 'RESERVED-ACTIVE' (same as reserved, but address is currently in use), 'ASSIGNED' (assigned manually or by some other mechanism), 'UNASSIGNED', 'NOTASSIGNABLE'.  
SYNTAX IA5String SINGLE-VALUE

NAME dhcpExpirationTime  
DESCRIPTION This is the time the current lease for an address expires.  
SYNTAX DateTime SINGLE-VALUE

NAME dhcpStartTimeOfState  
DESCRIPTION This is the time of the last state change for a leased address.  
SYNTAX DateTime SINGLE-VALUE

NAME dhcpLastTransactionTime  
DESCRIPTION This is the last time a valid DHCP packet was received from the client.  
SYNTAX DateTime SINGLE-VALUE

NAME dhcpBootpFlag  
DESCRIPTION This indicates whether the address was assigned via BOOTP  
SYNTAX Boolean SINGLE-VALUE

NAME dhcpDomainName  
DESCRIPTION This is the name of the domain sent to the client by the server. It is essentially the same as the value for DHCP option 15 sent to the client, and represents only the domain - not the full FQDN. To obtain the full FQDN

assigned to the client you must prepend the "dhcpAssignedHostName" to this value with a ".".  
IA5String SINGLE-VALUE

SYNTAX

Bennett, et. al.

Expires September 2000

[Page 17]

---

Internet Draft

DHCP Schema for LDAP

March 2000

NAME dhcpDnsStatus  
DESCRIPTION This indicates the status of updating DNS resource records on behalf of the client by the DHCP server for this address. The value is a 16-bit bitmask that has the same values as specified by the Failover-DDNS option (see [FAILOVR]).  
SYNTAX Integer SINGLE-VALUE

NAME dhcpRequestedHostName  
DESCRIPTION This is the hostname that was requested by the client.  
SYNTAX IA5String SINGLE-VALUE

NAME dhcpAssignedHostName  
DESCRIPTION This is the actual hostname that was assigned to a client. It may not be the name that was requested by the client. The fully qualified domain name can be determined by appending the value of "dhcpDomainName" (with a dot separator) to this name.  
SYNTAX IA5String SINGLE-VALUE

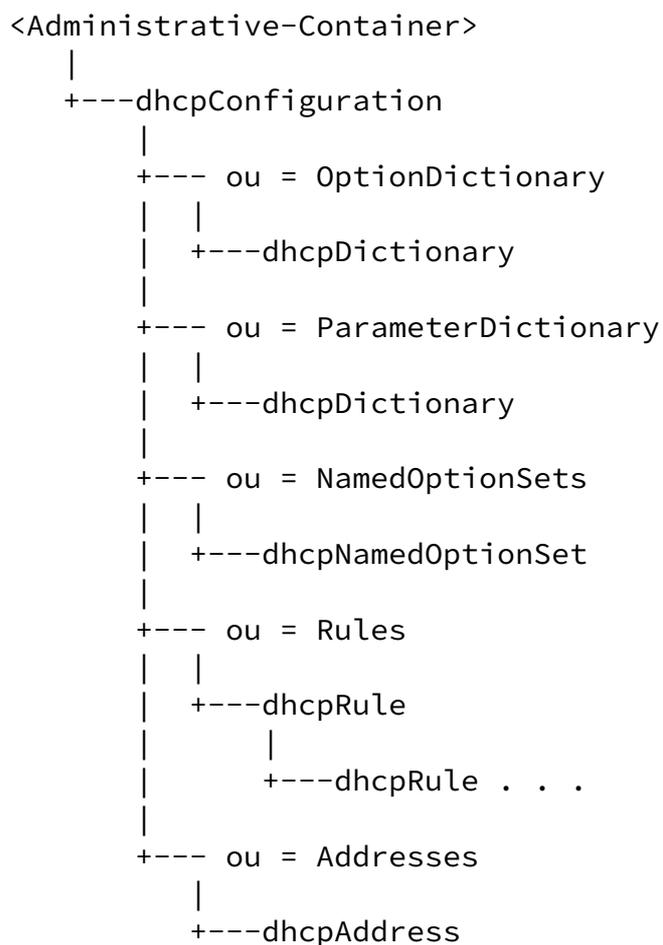
NAME dhcpReservedForClient  
DESCRIPTION The distinguished name of a "dhcpClient" that an address is reserved for. This may not be the same as the "dhcpAssignedToClient" attribute if the address is being reassigned but the current lease has not yet expired.  
SYNTAX DN SINGLE-VALUE

NAME dhcpAssignedToClient  
DESCRIPTION This is the distinguished name of a "dhcpClient" that an address is currently assigned to. This only has a value when the address is leased.  
SYNTAX DN SINGLE-VALUE

NAME dhcpRelayAgentInfo  
 DESCRIPTION If the client request was received via a relay agent, this contains information about the relay agent that was available from the DHCP request. This is a hex-encoded option value. For more information see [AGENT].  
 SYNTAX OctetString SINGLE-VALUE

## 8. Object Containment

These diagrams depict the containment hierarchy of the objects. <Administrative-Container> can be any LDAP object.



```
<Administrative-Container>
|
+---dhcpService
```

## 9. Object Class Inheritance

The following diagram shows the inheritance hierarchy of the classes:

```
Top
|
+---dhcpDictionary
|
+---dhcpService
|
+---dhcpConfiguration (aux: dhcpConfigurableObject)
|
+---dhcpNamedOptionSet (aux: dhcpConfigurableObject)
|
+---dhcpAddress (aux: dhcpConfigurableObject)
|
+---dhcpRule (aux: dhcpConfigurableObject)
    |
    +---dhcpClass
    |
    +---dhcpClient
    |
    +---dhcpPool
```

```
|
+---dhcpSharedNetwork
|
+---dhcpSubnet
```

## 10. Determining Assignment Rule settings

This section of the document defines the algorithm that should be used for determining the settings for options and/or parameters for an assignment rule. Most DHCP server implementations provide for some degree of inheritance of options between configuration objects. This algorithm is flexible enough to allow server implementations to represent their existing behavior.

The option settings directly associated with a "dhcpRule" object MUST take precedence over all other option settings. The rule also inherits its options from the following objects (in order of precedence):

- options from one or more included "dhcpNamedOptionSet" objects, as defined in the "dhcpIncludeOptionSet" attribute. If there is more than one option set, the attribute values define the order in which the option sets should be included.
- options from the "dhcpSourceObject" for the rule.
- options from "dhcpNamedOptionSet" objects associated with the "dhcpSourceObject" for the rule.
- options from the "parent" rule (only if the object's parent in the directory is also a "dhcpRule" object).
- options from "dhcpNamedOptionSet" objects associated with the "parent" rule.

- options from walking up the directory hierarchy inheriting from ancestor rules until the "dhcpConfiguration" object is reached.

## 11. References

- [[RFC2131](#)] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), March 1997.
- [[RFC2132](#)] Alexander, S., Droms, R., "DHCP Options and BOOTP Vendor Extensions", [RFC 2132](#), March 1997.
- [DMTF] Distributed Management Task Force, "Common Information Model (CIM) Specification", Version 2.0, Mar 1998.
- [DEN] Strassner, J., "Directory-Enabled Networks, Information Model and Base Schema", DEN Specification v3.0c, July 1998.

- [MSDHCP] Gu, Y., Vyaghrapuri, R., "An LDAP Schema for Dynamic Host Configuration Protocol Service", Internet Draft <[draft-gu-dhcp-ldap-schema-00.txt](#)>, August 1998.
- [NOVDHCP] Miller, T., Patel, A., Rao, P., "Lightweight Directory Access Protocol (v3): Schema for Dynamic Host Configuration Protocol (DHCP)", Internet Draft <[draft-miller-dhcp-ldap-schema-00.txt](#)>, June 1998.
- [FAILOVR] Droms, R., Rabil, G., Dooley, M., Kapur, A., Gonczi, S., Volz, B., "DHCP Failover Protocol", Internet Draft <[draft-ietf-dhc-failover-05.txt](#)>, October 1999.
- [AGENT] Patrick, M., "DHCP Relay Agent Information Option", Internet Draft <[draft-ietf-dhc-agent-options-09.txt](#)>, March 2000.
- [DHCP OPT] Carney, M., "New Option Review Guidelines and Additional Option Namespace", Internet Draft <[draft-ietf-dhc-option\\_review\\_and\\_namespace-01.txt](#)>, October 1999.
- [POLICY] Strassner, J., Elleson, E., Moore, B., "Policy Framework LDAP Core Schema", Internet Draft <[draft-ietf-policy-core-schema-06.txt](#)>, November 1999.
- [RFC2251] Wahl, M., Howes, T., Kille, S., "Lightweight Directory Access Protocol (v3)", [RFC 2251](#), December 1997.
- [RFC2252] Wahl, M., Coulbeck, A., Howes, T., Kille, S., "Lightweight Directory Access Protocol (v3) Attribute Syntax Definitions", [RFC 2252](#), December 1997.
- [RFC2255] Howes, T., Smith, M., "The LDAP URL Format", [RFC 2255](#), December 1997.
- [RFC951] Croft, B., Gilmore, J., "Bootstrap Protocol (BOOTP)", [RFC 951](#), September 1985.
- [RFC2119] Bradner, S. "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), March 1997.

## 12. Acknowledgements

This document is closely aligned with the work being done in the Distributed Management Task Force (DMTF) Networks working group.

Design ideas included in this document are primarily based on discussions during two meetings with some members of the IETF DHC Working Group and the DMTF Networks working group. The contributions of these individuals is gratefully acknowledged.

Special thanks to Andrea Westerinen, Lee Rafalow, Steve Gonczi, Steve Chirokas, Kim Kinnear, Ellen Stokes, Tom Miller, Ye Gu, Glenn Waters, Mike Carney, Ralph Droms, Greg Rabil, Ted Lemon and Steve Bazyl for their contributions.

Thanks also to Ester Burwell, Andy Sudduth, Fred Hunter, Paul Raison, Josh Littlefield, Peter Heitman, Neil Russell and Linda Scobo for their participation in these meetings.

### 13. Author information

Andy Bennett  
Bernie Volz  
Process Software Corporation  
959 Concord St.  
Framingham, MA 01701  
Phone: (508) 879-6994  
Email: bennett@process.com  
Email: volz@process.com

### 14. Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET

ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

