Network Working Group
INTERNET-DRAFT

Andy Bennett
Bernie Volz
Process Software

Andrea Westerinen
Microsoft

October 1999
Expires April 2000

DHCP Schema for LDAP
<draft-ietf-dhc-schema-01.txt>

Status of this Memo

Copyright Notice

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.

2. Open Issues

The following issues still need to be resolved:

- The Policy Framework Mapping section is superficial and needs to be updated.

- The CIM Mapping section is based on the object class definitions from the previous revision of this draft and needs to be updated.

3. Design Considerations

Some of the design considerations for this schema were:

- 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.

- 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 DHC 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, though in some cases the "cn" may be the same as another attribute already defined on the object.

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].

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].

4. 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.

4.1. 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)

4.2. 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 [DHCPOPT].
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

5. 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.

5.1. 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)

5.1.1. 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 hard-ware/software platform and version which is providing the service.
SYNTAX	IA5String SINGLE-VALUE

5.2. 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 ser-
                    vice may have multiple configurations.  A config-
                    uration 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 )
```

5.2.1. 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 configu-
             ration.
SYNTAX       DN SINGLE-VALUE

NAME         dhcpSecondaryService
DESCRIPTION  The "dhcpService(s)" which provide backup for the con-
             figuration.
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 direc-
             tory are automatically included in the configuration.
SYNTAX       IA5String MULTI-VALUE
```

6. Policy Objects

Most of a DHCP configuration is the definition of policies 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 policies. All of these object classes are based on a higher level abstraction that represents a dhcp policy.

This is done for several reasons: it simplifies the organization of the data and it also facilitates a mapping of the dhcp schema into the work being developed in the Policy Framework working group. This is discussed in more detail in a later section.

This schema separates the definition of a policy for an object from the object itself. This allows for the definition of multiple policies for an object (possibly in multiple dhcp configurations). However, the policy object does maintain a link back to the original 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 policy that references the object.

Furthermore, the policy objects in the directory can be organized in a hierarchical fashion. If objects are organized this way, the "child" policy object inherits settings from the "parent" policy object. This can be done recursively. Furthermore, the "child" policy object inherits any conditions from the "parent" policy as well. This means that the "child" policy'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.

6.1. dhcpPolicy Object Class

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

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

```
NAME                dhcpPolicy
DESCRIPTION         The base class for defining policies for address
                    and option assignment.
TYPE                Abstract
DERIVED FROM        Top
POSSIBLE SUPERIORS  ( )
MUST CONTAIN        ( cn dhcpPolicyType )
MAY CONTAIN         ( dhcpPoolName dhcpSubnetName
                      dhcpSharedNetworkName dhcpClientIdentifier
                      dhcpClassName dhcpVendorCondition
                      dhcpSourceObject dhcpOptionSetting
                      dhcpParameterSetting dhcpFieldSetting
                      dhcpForcedOptions dhcpIncludeOptionSet )
```

6.2. dhcpPolicy Attribute Definitions

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

NAME         dhcpPolicyType
DESCRIPTION  The type of the policy.  This should be one of 'POOL',
             'SUBNET', 'SHAREDNETWORK', 'CLIENT', 'CLASS' unless the
             server implementation extends this with a new policy
             type.
SYNTAX       IA5String SINGLE-VALUE

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

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

NAME         dhcpClientIdentifier
DESCRIPTION  The DHCP client identifier for the client.  This is
             encoded in the binary format used for dhcp option 61 -
             the first octet is the ARP hardware type, followed by
             the link layer (or MAC) address.  If the value is an
             arbitrary identifier (instead of a MAC address) then the
             first octet is 0.  For more information see [RFC2132].
SYNTAX       OctetString SINGLE-VALUE

NAME         dhcpClassName
DESCRIPTION  The name of the class for the policy.
SYNTAX       IA5String SINGLE-VALUE
```

```
NAME         dhcpVendorCondition
DESCRIPTION  If the server extends the policy types, this attribute
             MAY be used to specify the conditions under which the
             policy should be applied.  The content of this attribute
             is defined by the vendor/server implementation.
SYNTAX       IA5String MULTI-VALUE

NAME         dhcpSourceObject
DESCRIPTION  If the policy applies to an object that is defined else-
             where in the directory, this attribute has the distin-
             guished name of that object.
SYNTAX       DN SINGLE-VALUE
```

6.3. dhcpPool Object Class

A "dhcpPool" represents a policy 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 "dhcpPolicyType" 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        dhcpPolicy
POSSIBLE SUPERIORS  ( OrganizationalUnit dhcpPolicy )
MUST CONTAIN        ( cn dhcpPoolName dhcpAddressRange )
MAY CONTAIN         ( )
```

6.3.1. dhcpPool Attribute Definitions

```
NAME         dhcpAddressRange
DESCRIPTION  The starting & ending IP Addresses in the range (inclu-
             sive), separated by a hyphen.  Each range is defined as
             a separate value.
SYNTAX       IA5String MULTI-VALUE
```

6.4. dhcpSubnet Object Class

A "dhcpSubnet" represents a policy for an IP subnet.

The "dhcpPolicyType" 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        dhcpPolicy
POSSIBLE SUPERIORS  ( OrganizationalUnit dhcpPolicy )
MUST CONTAIN        ( cn dhcpSubnetAddress dhcpSubnetMaskLength
                      dhcpSubnetName )
MAY CONTAIN         ( )
```

6.4.1. dhcpSubnet Attribute Definitions

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

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

6.5. dhcpSharedNetwork Object Class

A "dhcpSharedNetwork" represents a policy for multiple subnets on the same physical cabling.

The "dhcpPolicyType" 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        dhcpPolicy
POSSIBLE SUPERIORS  ( OrganizationalUnit dhcpPolicy )
MUST CONTAIN        ( cn dhcpSharedNetworkName )
MAY CONTAIN         ( )
```

6.6. dhcpClient Object Class

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

The "dhcpPolicyType" attribute MUST be set to 'CLIENT', and the "cn" SHOULD be set to the value of the "dhcpClientIdentifier" attribute".

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

6.6.1. dhcpClient Attribute Definitions

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

6.7. 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 - setting the "dhcpVendor-Condition" attribute allows for the definition of dynamic classes.

The "dhcpPolicyType" 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	dhcpPolicy
POSSIBLE SUPERIORS	(OrganizationalUnit dhcpPolicy)
MUST CONTAIN	(cn dhcpClassName)
MAY CONTAIN	(dhcpClassType)

6.7.1. dhcpClass Attribute Definitions

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

7. 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.

7.1. 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)

7.1.1. 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 )
MAY CONTAIN         ( dhcpKeyword dhcpTag dhcpDataType dhcpDefault
                      dhcpMultiValued dhcpDescription dhcpLegalValues
                      dhcpTypeRestriction dhcpImplementation )
```

7.2. dhcpDictionary Attribute Definitions

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

NAME         dhcpKeyword
DESCRIPTION  This is a string tag (or keyword) that is used to iden-
             tify the option or parameter.
SYNTAX       IA5String SINGLE-VALUE

NAME         dhcpTag
DESCRIPTION  The numeric tag that identifies an option.  This MUST be
             defined for options, but is not required for parameters.
SYNTAX       Integer SINGLE-VALUE

NAME         dhcpDataType
DESCRIPTION  The data type for values of this option.  The set of
             valid data types are defined by the DHCP protocol.  For
             more information see [RFC2131].
SYNTAX       Integer 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 "dhcpParame-
             terSetting" 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         dhcpDescription
DESCRIPTION  A textual description of an object.
SYNTAX       String 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 each one is encoded as it would be in the
             "dhcpOptionSetting" or "dhcpParameterSetting" attribute.
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
             policies.  This is restricted to the same set of values
             as the "dhcpPolicyType" attribute.
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.
SYNTAX       IA5String MULTI-VALUE
```

8. 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.

8.1. 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 dhcpAddressType dhcpAddressState )
MAY CONTAIN         ( dhcpExpirationTime dhcpStartTimeOfState
                      dhcpLastTransactionTime dhcpBootpFlag
                      dhcpDomainName dhcpDnsStatus
                      dhcpRequestedHostName dhcpAssignedHostName
                      dhcpReservedForClient dhcpAssignedToClient
                      dhcpRelayAgentInfo dhcpOptionSetting
                      dhcpParameterSetting dhcpFieldSetting )
```

8.2. dhcpAddress Attribute Definitions

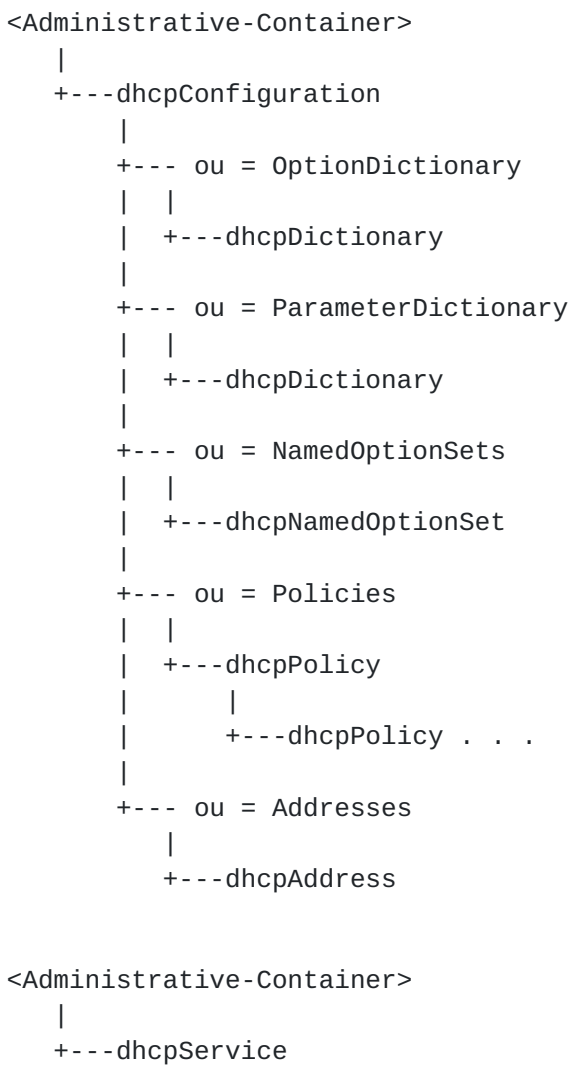
```
NAME         cn
DESCRIPTION  The IP address, as a string.
SYNTAX       IA5String SINGLE-VALUE

NAME         dhcpAddressType
DESCRIPTION  This describes how the address is to be assigned to a
             client.  One of 'UNKNOWN', 'DYNAMIC', 'FIXED', 'UNAS-
             SIGNED', 'NOTASSIGNABLE'.
SYNTAX       IA5String SINGLE-VALUE

NAME         dhcpAddressState
DESCRIPTION  This stores information about the current binding-status
             of an address, using the states defined in the safe-
             failover draft.  For more information see [FAILOVR].
SYNTAX       Integer 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 is assigned via BOOTP
   SYNTAX       Boolean SINGLE-VALUE

   NAME         dhcpDomainName
   DESCRIPTION  This is the name of the domain assigned to the client by
                the server.
   SYNTAX       IA5String SINGLE-VALUE

   NAME         dhcpDnsStatus
   DESCRIPTION  This indicates which resource records were added to the
                domain on behalf of the client by the DHCP server.  The
                allowed values are: "No DNS Activity", "Update A
                Records", "Update PTR Records", and "Update Both A and
                PTR Records"
   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 deter-
                mined 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 "dhc-
                pAssignedToClient" 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 [RELAY].
SYNTAX       OctetString SINGLE-VALUE
```

9. Object Containment

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



10. 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)
   |
   +---dhcpPolicy (aux: dhcpConfigurableObject)
          |
          +---dhcpClass
          |
          +---dhcpClient
          |
          +---dhcpPool
          |
          +---dhcpSharedNetwork
          |
          +---dhcpSubnet
```

11. Determining Policy settings

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

The option settings directly associated with a "dhcpPolicy" object MUST take precedence over all other option settings. The policy also inherits 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 policy.
- options from "dhcpNamedOptionSet" objects associated with the "dhcpSourceObject" for the policy.
- options from the "parent" policy (only if the object's parent in the directory is also a "dhcpPolicy" object.
- options from "dhcpNamedOptionSet" objects associated with the "parent" policy.

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

12. Policy Framework Schema Mapping

This section of the document provides a mapping to the Policy Framework Core schema being developed by the Policy Framework working group.

The Policy Framework Core schema provides a general representation of policies. Mapping the "dhcpPolicy" objects in this schema to that general model is a fairly straightforward exercise. For the most part each "dhcpPolicy" maps directly into a "policyRule" object from the Core schema.

When a dhcpPolicy has a value for any of the following attributes: { "dhcpPoolName", "dhcpSubnetName", "dhcpSharedNetworkName", "dhcpClientIdentifier", "dhcpClassName", "dhcpVendorCondition" } this would be mapped into a "policyCondition" object. Values for the { "dhcpOptionSetting", "dhcpParameterSetting", "dhcpFieldSetting", "dhcpForcedOptions", "dhcpIncludeOptionSet" } attributes would be mapped into "policyAction" objects.

13. CIM Mapping

This section of the document provides a mapping to the CIM (Common Information Model) DHCP Schema defined by the DMTF.

13.1. dhcpConfigurableObject

The DMTF Schema maps the setting and parameter data in the dhcpConfigurableObject class using instances of CIM_Setting classes and associations.

The individual dhcpOption, Parameter and Field Settings are instances of the CIM_DHCPSetting class (derived from CIM_Setting). The CIM_DHCPSetting class defines key properties and an octet string to hold the OptionData. The CIM_DHCPSetting instances are associated with the various subclasses of dhcpConfigurableObject using the CIM_ElementSetting, CIM_SettingContext or CIM_CollectionSetting relationships. (Which association is used depends on the derivation of the dhcpConfigurableObject in the CIM class hierarchy.)

The individual dhcpConfigurations and dhcpIncludeOptionSets are instances of the CIM_DHCPConfiguration class or of its subclass, CIM_DHCPOptionConfiguration (both classes were discussed earlier in this document). The CIM_DHCPConfiguration instances are associated

with the various subclasses of dhcpConfigurableObject using the CIM_ElementConfiguration, CIM_ConfigurationComponent or CIM_CollectionConfiguration relationships. (As above, the association that to be used depends on the derivation of the dhcpConfigurableObject in the CIM class hierarchy.)

dhcpForcedOptions is mapped to the CIM_DHCPForcedOptions class (derived from CIM_Setting). CIM_DHCPForcedOptions defines key properties and an integer array to hold the option tags being "forced". It is associated with the various subclasses of dhcpConfigurableObject using the CIM_ElementSetting, CIM_SettingContext or CIM_CollectionSetting relationships. (As above, the association that is used depends on the derivation of the dhcpConfigurableObject in the CIM class hierarchy.)

Since the CIM_DHCPSetting and CIM_DHCPForcedOptions classes were not discussed previously, a brief overview of their properties is needed. Both classes contain only key properties for identification, plus one additional property in each object. The CIM_DHCPSetting class has an octet string to hold the option, parameter or field data. The CIM_DHCPForcedOptions class has an integer array holding option tags.

The key properties of CIM_DHCPSetting and CIM_DHCPForcedOptions are:

- ConfiguredObject, a string identifying the entity to which the Setting or ForcedOptions applies
- SettingID, a string identifier for the instance
- For CIM_DHCPSetting only, a SettingType enumerated integer defining whether an option, parameter or field setting is specified.

These string properties taken together form the objects' key and conceptually are a DN.

13.2. Configurations and Services

13.2.1. dhcpService

The DMTF Schema specifies a CIM_DHCPService object (derived from CIM_Service) that corresponds to dhcpService. In CIM, Services are defined and named relative to the System that hosts them. Therefore, the keys of CIM_DHCPService are defined as:

- The hosting System's CreationClassName (for example, CIM_UnitaryComputerSystem) and Name.
- The Service's CreationClassName (= "CIM_DHCPService") and Name.

These string properties taken together form the object's key and conceptually are a DN.

That CIM_DHCPService is named and scoped by a CIM_System is consistent with the definition of a <Computer> as a "possible superior" of dhcpService. In CIM, this relationship is described using the

CIM_HostedService association.

CIM_DHCPService also contains the following properties:
- Description and Caption strings (inherited from CIM_ManagedSystemElement)
- ImplementationMatchingStrings, an array of strings
- Numerous other properties defining runtime characteristics of the DHCP Service. For example, whether or not the Service provides ProxyDNSSupport (a boolean) or the BootPLeaseLength (a uint32) are properties of CIM_DHCPService.

The dhcpServiceName string can be mapped into the CIM_DHCPService class' Name property, if the uniqueness characteristics of CIM naming are satisfied. Otherwise, it should be mapped into the Caption string. And, the dhcpImplementation multi-valued string can be mapped into the ImplementationMatchingStrings property.

The dhcpConfiguration property is mapped to an association, CIM_ServiceForDCHPConfiguration, in the DMTF Schema. This association was explained above.

13.2.2. dhcpConfiguration

The DMTF Schema defines a CIM_DHCPConfiguration object (derived from CIM_Configuration) that corresponds to dhcpConfiguration. The CIM class has the following properties:
- A Name string that is the object's key, and conceptually is a DN
- Description and Caption strings

The dhcpConfigurationName string is mapped into the CIM_DHCPConfiguration class' Name property, if it satisfies the uniqueness characteristics of CIM naming. Otherwise, it should be mapped into the Caption property.

The remainder of the dhcpConfiguration properties are mapped to associations in the DMTF Schema. The correspondence is as follows:
- dhcpPrimaryService and dhcpSecondaryService map to the CIM_ServiceForDHCPConfiguration association. This association ties together a CIM_DHCPConfiguration and a CIM_DHCPService, and includes a boolean property to identify the PrimaryService.
- dhcpParentConfiguration (subsetted by the dhcpIncludeFromParent property) is defined by instantiating various CIM_DHCPConfiguration objects and then grouping them using the CIM_ConfigurationComponent relationship. The aggregation in CIM defines the parts or components of the parent, as opposed to the LDAP DHCP Schema which defines the parent and then subsets it.

The dhcpLocator property is unique to the LDAP Schema and has no CIM correspondence.

13.3. Policy Objects

13.3.1. dhcpPool

Different than the LDAP schema, the DMTF Schema defines a CIM_IPAddressRange entity to uniquely identify and manipulate ranges. The key properties of the class are its start and end addresses. CIM_IPAddressRanges can be aggregated into CIM_RangeGroups using the CIM_CollectionOfRanges association.

Regarding dhcpSubnet being a "possible superior" of dhcpPool, this is mapped in the DMTF Schema using the CIM_RangeGroupInSubnet aggregation association.

13.3.2. dhcpSubnet

The DMTF Schema specifies a CIM_IPSubnet object (derived from CIM_LogicalNetwork) that corresponds to the dhcpSubnet class. The property correspondence is as follows:

- dhcpIpAddress = CIM_IPSubnet.SubnetNumber
- dhcpSubnetMaskLength = (mapping is not 1 to 1 in that CIM_IPSubnet specifies the complete mask, as the SubnetMask property)
- dhcpSubnetName = CIM_IPSubnet.Name, inherited from CIM_LogicalNetwork and part of the key structure for the class

The dhcpSubnetName string can be mapped into the CIM_DHCPService class' Name property, only if the uniqueness characteristics of CIM naming are satisfied. Otherwise, it should be mapped into the Caption property.

Participation in a SharedNetwork (indicated by the dhcpSharedNetwork attribute in the LDAP Schema) is indicated by the CIM_SubnetInSharedNetwork aggregation. This association would relate an instance of CIM_IPSubnet and an instance of CIM_SharedNetwork.

Regarding dhcpSubnet being a "possible superior" of another Subnet, this is mapped in the DMTF Schema using the CIM_SubnetInSubnet aggregation association.

13.3.3. dhcpSharedNetwork

CIM_SharedNetwork (subclassed from CIM_CollectionOfMSEs) is defined in the DMTF Schema to correspond to the dhcpSharedNetwork class. A property of the CIM_SharedNetwork class, Name (its key), maps the dhcpSharedNetworkName information. This mapping is allowed if the dhcpSharedNetworkName satisfies the uniqueness characteristics of CIM naming. Otherwise, the Name data should be mapped into the CIM_SharedNetwork.Caption property, inherited from

CIM_CollectionOfMSEs.

13.3.4. dhcpClient

The DMTF Schema defines a CIM_DHCPClientEndpoint object (derived from CIM_ProtocolEndpoint) that corresponds to dhcpClient. The CIM class maps the LDAP Schema properties as follows:

- dhcpClientIdentifier = DHCPClientID string
- dhcpUserClass = UserClass string array
- dhcpVendorClass = VendorClass string
- dhcpCharacteristics = Characteristics string array
- dhcpCreatedByServer = CreatedForUnlistedClient boolean

The remainder of the dhcpClient properties are mapped to associations in the DMTF Schema. The correspondence is as follows:

- Information in dhcpAssignedAddress is mapped to the CIM_DHCPAddressAssignment association, relating CIM_DHCPClientEndpoint and CIM_DHCPAllocatedEndpoint (as discussed in Section 3.2.1)
- Information in dhcpMemberOfGroup is mapped to the CIM_DHCPEndpointCollection association, relating CIM_DHCPClientEndpoint and CIM_DHCPClientCollection (as discussed in Section 3.3

To identify instances of CIM_DHCPClientEndpoint, key (naming) properties are needed. Names are defined relative to the System that hosts the Endpoints. So, the keys of any CIM_ProtocolEndpoint are specified as:

- The hosting System's CreationClassName (for example, CIM_UnitaryComputerSystem) and Name,
- The ProtocolEndpoint's CreationClassName (= "CIM_DHCPClientEndpoint") and Name.

These string properties taken together form the object's key and conceptually are a DN.

13.3.5. dhcpClass

The DMTF Schema defines CIM_DHCPClientCollection (derived from CIM_CollectionOfMSEs) to correspond to dhcpClass. In the CIM_DHCPClientCollection object, UserClass and VendorClass information is individually called out (addressing the DHC protocol mechanisms for determining a collection of clients). Also, it is permissible to define other membership Characteristics, as well as to specify explicit membership. In CIM, properties can be left undefined/unspecified. Therefore, if the UserClass, VendorClass or other Characteristics are not applicable, they can be left NULL (undefined).

The mapping of properties from dhcpClass to CIM_DHCPClientCollection is straightforward.

- dhcpClassName corresponds to the CIM_DHCPClientCollection's Name property (if CIM naming requirements for uniqueness are met). Otherwise, dhcpClassName should be mapped into the Caption property, inherited from CIM_CollectionOfMSEs.
- dhcpClassType is mapped to the CollectionType enumeration.
- dhcpClassCharacteristics corresponds to the Characteristics string array property.
- dhcpUserClass and dhcpVendorClass map to the UserClass and VendorClass string arrays, respectively.

Explicit membership by a client in a particular instance of CIM_DHCPClientCollection is specified using the CIM_DHCPEndpointCollection aggregation association. (In the LDAP Schema, this is defined by placing the DN of the dhcpClass in the client's dhcpMemberOfGroup attribute.) CIM_DHCPEndpointCollection is also useful when enumerating the client members of a Collection, after evaluation of the UserClass, VendorClass and Characteristics requirements.

Regarding dhcpConfiguration as a "possible superior" for dhcpClass, this is mapped in the DMTF Schema as an instance of the CIM_CollectionConfiguration association. The association would be instantiated to reference the CIM_DHCPClientCollection and the appropriate CIM_DHCPConfiguration.

13.4. Other Configuration Objects

13.4.1. dhcpNamedOptionSet

In the DMTF Schema, the CIM_DHCPOptionConfiguration object corresponds to the dhcpNamedOptionSet class. dhcpOptionSetName is mapped into the Name property (a key property) of the CIM class. This is acceptable if dhcpOptionSetName meets CIM's requirements for uniqueness. Otherwise, its data should be placed into the Caption property, inherited from CIM_Configuration.

If the instance of CIM_DHCPOptionConfiguration is part of a higher level CIM_DHCPConfiguration, this is indicated by instantiating the CIM_ConfigurationComponent association and referencing both Configurations.

13.4.2. dhcpDictionary

The DMTF Schema defines a CIM_DHCPOptionDictionary object (derived from CIM_Setting) that corresponds to dhcpOptionDictionary. Instances of CIM_DHCPOptionDictionary may be grouped by CIM_DHCPConfigurations using the CIM_SettingContext association. This association aggregates one or more Settings into a Configuration, and corresponds to the definition of dhcpConfiguration as a "possible

superior" of dhcpOptionDictionary. It also is the mapping for the dhcpConfigurationName string attribute.

The property values of the CIM_DHCPOptionDictionary class map one-to-one to those of dhcpOptionDictionary. The explicit mapping is:

- dhcpOptionName = OptionName string
- dhcpOptionTag = OptionTag uint32
- dhcpDataType = DataType enumeration
- dhcpMultiValued = MultiValued boolean
- dhcpDefault = DefaultValue octet string
- dhcpVendorClass = VendorClass string
- dhcpDescription = Description string, inherited from CIM_Setting
- dhcpLegalValues = LegalValues octet string
- dhcpTypeRestriction = RestrictedTo enumerated integer array

To identify instances of CIM_DHCPOptionDictionary, one or more key (naming) properties are needed. The key of CIM_DHCPOptionDictionary is specified as its SettingID (a string). This property conceptually is a DN.

13.5. Tracking Addresses

13.5.1. dhcpAddress

The DMTF Schema defines a CIM_DHCPAllocatedEndpoint (a subclass of CIM_ProtocolEndpoint) that corresponds to the dhcpAddress class. However, in CIM, to obtain all the data for an IP address, one must examine several classes. These are:

- CIM_DHCPAllocatedEndpoint containing lease specific information
- CIM_IPProtocolEndpoint containing address specific data, such as address type and IP version support
- CIM_EndpointIdentity, an association instantiated between the CIM_DHCPAllocatedEndpoint and CIM_IPProtocolEndpoint, to tie together these different aspects of the same address
- CIM_DHCPAddressAssignment, an association relating a CIM_DHCPAllocatedEndpoint with a DHCP client

Mapping the dhcpAddress properties into the DMTF Schema results in the following list:

- dhcpIPAddress = CIM_DHCPAllocatedEndpoint.Address string
- dhcpAddressType = CIM_DHCPAllocatedEndpoint.DHCPAddressType enumeration
- dhcpAddressState = CIM_DHCPAllocatedEndpoint.AddressState enumeration
- dhcpDomainName = CIM_DHCPAllocatedEndpoint.DomainName string
- dhcpLastTransactionTime = CIM_DHCPAllocatedEndpoint.LastTransactionTime datetime

- dhcpExpirationTime = CIM_DHCPAllocatedEndpoint.Expiration date-time
- dhcpStartTimeOfState = CIM_DHCPAllocatedEndpoint.TimeOfLastState-Change datetime
- dhcpRequestedHostName = CIM_DHCPAllocatedEnd-point.RequestedDNSName string
- dhcpBootpFlag = CIM_DHCPAllocatedEndpoint.BootPAddress boolean
- dhcpAssignedHostName = CIM_DHCPAllocatedEndpoint.AssignedDNSName string
- dhcpDnsStatus = CIM_DHCPAllocatedEndpoint.DNSStatus enumeration
- dhcpReservedForClient = An instance of the CIM_DHCPAddressAssign-ment association with the Reserved boolean set to TRUE
- dhcpAssignedToClient = An instance of the CIM_DHCPAddressAssign-ment association indicating the current lease assignment
- dhcpRelayAgentInfo = CIM_DHCPAllocatedEndpoint.RelayAgent octet string

To identify CIM_IPProtocolEndpoint or CIM_DHCPAllocatedEndpoint instances, key (naming) properties are needed. Names are defined relative to the System that hosts the Endpoints. So, the keys of any CIM_ProtocolEndpoint are specified as:

- The hosting System's CreationClassName (for example, CIM_Uni-taryComputerSystem) and Name.
- The ProtocolEndpoint's CreationClassName (= "CIM_IPProtocolEnd-point" or "CIM_DHCPAllocatedEndpoint") and Name.

These string properties taken together form the object's key and conceptually are a DN.

Any instances of CIM_ProtocolEndpoints (or instances of its sub-classes) can be members of CIM_LogicalNetworks. This is specified using the CIM_InLogicalNetwork association. Using this association, one can relate CIM_DHCPAllocatedEndpoints and CIM_IPProtocolEnd-points to a CIM_IPSubnet. The InLogicalNetwork association becomes the mapping for the dhcpSubnet "possible superior" relationship to dhcpAddress.

14. 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-03.txt>, November 1998.

[RELAY] Patrick, M., "DHCP Relay Agent Information Option", Internet Draft <draft-ietf-dhc-agent-options-05.txt>, November 1998.

[DHCPOPT] Carney, M., "New Option Review Guidelines and Additional Option Namespace", Internet Draft <draft-ietf-dhc-option_review_and_namespace-00.txt>, June 1999.

[POLICY] Strassner, J., Elleson, E., Moore, B., "Policy Framework LDAP Core Schema", Internet Draft <draft-ietf-policy-core-schema-05.txt>, October 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.

15. 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 Lee Rafalow, Steve Gonczi, Steve Chirokas, Kim Kinnear, Ellen Stokes, Tom Miller, Ye Gu, Glenn Waters, Mike Carney, Ralph Droms, Greg Rabil 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.

16. 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

Andrea Westerinen
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
Phone: +1 425-705-2553
Email: andreawe@microsoft.com

17. Full Copyright Statement