Application Working Group
INTERNET-DRAFT
Expires in six months
Intended Category: Standard

Tim Howes
Netscape Communications Corp.
Luke Howard
Independent Consultant
October 1997

A Simple Caching Scheme for LDAP and X.500 Directories
<draft-ietf-asid-ldap-cache-01.txt>

1. Status of this Memo

This document is an Internet-Draft. 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.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet- Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

2. Abstract

This memo defines an object class and attribute type in support of a simple caching mechanism for use in LDAP and X.500 directories. The object class allows a simple 'time-to-live' attribute to be included in entries, enabling clients retrieving the attribute to tell when the other information they have retrieved from the entry should be thrown away. The use of this scheme does not preclude the subsequent or additional use of other more complicated schemes, for example, allowing different TTLs on individual attributes.

3. Need for Caching and Overview

LDAP [ldapv3-1, ldapv3-2] and X.500 [x500] define a distributed database. To achieve greater performance and availability, it is desirable to replicate information close to the entities accessing it. Formal replication schemes have been devised for this purpose. Caching is an informal method of replication designed to make repeated use of

information by the same or co-located clients more efficient. Systems relying on fast performance that can tolerate temporarily out-of-date data, such as the Domain Name System [rfc1034], often make heavy use of caching to achieve the desired level of performance. LDAP and X.500 comprise another system that could similarly benefit from caching.

Until now, there has been no agreed scheme for providing a consistent caching mechanism for LDAP and X.500. Caching occurs, but it is up to the caching agent to determine the appropriate length of time a piece of information can safely be cached. There is support in X.500 for ignoring all cached or replicated information copies in favor of the master copy, at the client's discretion (the dontUseCopy service control). There is no guidance on the length of time that information (master or not) can safely be cached.

This draft defines a simple caching model similar to that used by the DNS. A new operational attribute, ttl, is defined to specify the maximum time for which a cached copy of any attributes in the entry should be considered valid. The ttl attribute SHOULD be allowed in every entry that may be cached.

A new object class, cacheObject is defined, which allows an entry to have the new ttl attribute, even if the server implementation does not support operational attributes (e.g., an LDAPv2 server).

Note that use of this scheme means that all attributes in an entry are subject to the same TTL. This is satisfactory in many cases, but more complicated situations where different attributes (or even values of the same attribute) may have different TTL requirements can easily arise. The scheme described here is not intended to address these situations, nor is it intended to hinder the development of other schemes that do.

4. The ttl Attribute

The definition of the ttl attribute is as follows:

```
( 1.3.6.1.4.1.250.1.60 NAME 'ttl' EQUALITY integerMatch
  SYNTAX '1.3.6.1.4.1.1466.115.121.1.27' SINGLE-VALUE )
```

The ttl attribute contains the time, in seconds, that any information from the entry should be kept by a client before it is considered "stale" and a new copy fetched. A value of 0 implies that the object must not be cached.

The behaviour of caching clients with respect to entries lacking the ttl is not prescribed. Caching agents may use any appropriate method for determining whether an entry without a ttl attribute should be refetched. For example, clients may compare the modifyTimestamp

attribute of the entry with the current one and refetch the entry only if the entry has been updated since it was cached. A number of factors, such as network latency, may render this policy inefficient. As such, clients may assume entries lacking the ttl attribute never expire, or that they expire in some client-defined time period, or that they should never be cached.

5. The cacheObject Object Class

The cacheObject object class is defined as follows:

```
( 1.3.6.1.4.1.250.3.18 NAME 'cacheObject' AUXILIARY SUP top
  DESC 'Auxiliary object class to hold TTL caching information'
  MAY ttl )
```

6. Coexistence with entryTtl and DNS-related attributes

The entryTtl attribute, defined in [v3ext], is an operational attribute maintained by the directory server which appears to bear superficial resemblance to the ttl attribute. The entryTtl attribute is only present in entries of the dynamicObject object class, and may not be modified by the user. A value of 0 indicates that the entry may disappear from the directory without warning.

By contrast, the ttl attribute as defined here refers not to dynamic entries, but to those defined by the user which are accorded a specific time to live.

Clients caching entries of class dynamicObject should use the entryTtl attribute instead of the ttl to determine an object's TTL. The same behaviour applies: if the value is 0, the entry should not be cached.

The dNSDomain object class [rfc1279] contains attributes, such as dNSRecord, which may include embedded TTLs. If the caching agent has specific cognizance of these attributes, it may wish to honour them in preference to the entryTtl or ttl attributes. This is not required.

7. Security Considerations

A caching scheme has implications on the accuracy and security of data. Both applications and data providers should be aware of how important information consistency is for the data they are using or providing.

When accessing anything but publicly available information, care must be taken by the caching entity to ensure that the intended access control policy of the directory is not violated. This may be accomplished by not caching non-public information at all, or by having an understanding of the source site's access control policies. Note that understanding a

site's access control policy may be difficult, given the existence of proprietary schemes, and the fact that there may be mechanisms in place not visible or detectable by the caching entity. These mechanisms may even make the determination of what information is publicly accessible difficult or impossible.

8. Bibliography

[ldapv3-1] Wahl, M., Howes, T., Kille, S., "Lightweight Directory Access Protocol (v3)", INTERNET-DRAFT <draft-ietf-asid-ldapv3-protocol-07.txt>, August 1997.

[rfc1034] Mockapetris, P., "Domain Names - Concepts and Facilities", Request for Comments (RFC) 1034, November 1987.

[ldapv3-2] Wahl, M., Coulbeck, A., Howes, T., Kille, S., "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", INTERNET-DRAFT <draft-ietf-asid-ldapv3-attributes-07.txt>, August 1997.

[x500] "Information Processing Systems - Open Systems Interconnection - The Directory: Overview of Concepts, Models and Services", ISO/IEC JTC 1/SC21, International Standard 9594-1, 1988.

[v3ext] Yaacovi, Y., Wahl, M., Genovese, T., "Lightweight Directory Access Protocol (v3): Extensions for Dynamic Directory Services", INTERNET-DRAFT <draft-ietf-asid-ldapv3-dynamic-06.txt>, September 1997.

[rfc1279] Kille, S., "X.500 and Domains", Request for Comments (RFC) 1279, November 1991.

9. Authors' Addresses

Tim Howes
Netscape Communications Corp.
501 E. Middlefield Rd.
Mountain View, CA 94043
USA
+1 415 937-3419
howes@netscape.com

Luke Howard
PO Box 59
Central Park Vic 3145
Australia
lukeh@xedoc.com