M. Smith
T. Howes
University of Michigan
21 November 1995

Definition of the leaf and nonLeaf Object Classes
Filename: draft-ietf-asid-leafnonleaf-00.txt

Status of this Memo

Abstract

Applications of X.500, LDAP, and similar directory services need to be able to efficiently and unambiguously determine if an entry is a leaf entry (no entries exist beneath the entry) or a non-leaf entry (entries do exist beneath the entry). While some implementations use proprietary object classes to allow directory clients to make the distinction, there is no standard defined. This document defines two object classes that may be used by all implementations to allow directory clients to distinguish leaf entries from non-leaf entries.

Background and Intended Usage

This document defines two object classes (leaf and nonLeaf) that may

be used by X.500 [1], LDAP [2], and other directory implementations to allow applications to distinguish leaf entries from non-leaf entries. A leaf entry is one that does not have any other entries beneath it (that is, it is not a container). A non-leaf entry is one that does have other entries beneath it (that is, it is a container for other entries). Each entry in the directory is one of these two types.

Directory servers that comply with this specification should automatically include the leaf object class value in all entries that do not have other entries beneath them. Similarly, the nonLeaf object class value should be included in all other entries. If one of these two object class values is present in an entry, directory clients can unambiguously know whether an entry may have other entries beneath it. This capability is particularly useful when browsing the directory in a hierarchical fashion. If neither value is present, the client must use other means (such as a one-level or subtree search based at the entry in question to see if any "child" entries are present).

It is intended that the schema elements defined in this document will be progressed according to the process defined by the Internet X.500 Schema Working Group [3].

Definition of the leaf Object Class

```
Name:             leaf
Description:      object that does not contain other entries
OID:              umichObjectClass.19 (1.3.6.1.4.1.250.3.19)
SubclassOf:       top
MustContain:
MayContain:
```

Definition of the nonLeaf Object Class

```
Name:             nonLeaf
Description:      object that contains other entries
OID:              umichObjectClass.20 (1.3.6.1.4.1.250.3.20)
SubclassOf:       top
MustContain:
MayContain:       numberOfChildren, numberOfDescendants
```

Definition of the numberOfChildren Attribute

```
   Name:              numberOfChildren
   ShortName:
   OID:               umichAttributeType.62 (1.3.6.1.4.1.250.1.62)
   Syntax:            Integer
   SizeRestriction:   none
   SingleValued:      TRUE
   MatchesFor:
```

Definition of the numberOfDescendants Attribute

```
   Name:              numberOfDescendants
   ShortName:
   OID:               umichAttributeType.63 (1.3.6.1.4.1.250.1.63)
   Syntax:            Integer
   SizeRestriction:   none
   SingleValued:      TRUE
   MatchesFor:
```

Discussion of the Object Class and Attribute Schema

Entries that do not have other entries beneath them belong to the leaf object class. Entries that have other entries beneath them belong to the nonLeaf object class.

The numberOfChildren attribute, if present, contains a count of the entries that are listed directly below the non-leaf entry (that is, the number of entries that are in the single-level beneath the entry). The numberOfDescendants attribute, if present, contains a count of the total number of entries that are listed anywhere beneath the non-leaf entry (that is, the total number of entries that are contained in the directory subtree beneath the entry).

References

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

[2] Yeong, W., Howes, T., and S. Kille, "Lightweight Directory Access Protocol", RFC 1777, Performance Systems International, University of Michigan, ISODE Consortium, March 1995, <URL:ftp://ds.internic.net/rfc/rfc1777.txt>

[3] Howes, T., Rossen, K., Sataluri, S., and Wright, R., "Procedures for Formalizing, Evolving, and Maintaining the Internet X.500

Directory Schema", Internet Draft (Work In Progress) of the Schema Working Group, <URL:ftp://ds.internic.net/internet-drafts/draft-howes-x500-schema-03.txt>

Security Considerations

Security considerations are not discussed in this memo.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. NCR-9416667.

Authors' Addresses

Mark Smith
University of Michigan
Information Technology Division
535 W. William St.
Ann Arbor, MI 48103-4943, USA
Phone: +1 313 764-2277
EMail: mcs@umich.edu

Tim Howes
University of Michigan
Information Technology Division
535 W. William St.
Ann Arbor, MI 48103-4943, USA
Phone: +1 313 747-4454
EMail: tim@umich.edu

This Internet Draft expires on 21 May 1995.