                        Using the Domain Name System
                     To Store Arbitrary String Attributes

Status of this Memo

Abstract

   While the Domain Name System (DNS) [2,3] is generally used to store
   predefined types of information (e.g., addresses of hosts), it is
   possible to use it to store information that has not been previously
   classified.

   This paper describes a simple means to associate arbitrary string
   information (ASCII text) with attributes that have not been defined
   by the DNS. It uses DNS TXT resource records to store the
   information. It requires no change to current DNS implementations.

1. Introduction

   The Domain Name System is designed to store information that has both
   a predefined type and structure. Examples include IP addresses of
   hosts and names of mail exchangers. It would be useful to take
   advantage of the widespread use and scaleability of the DNS to store
   information that has not been previously defined.

   This paper proposes the use of the DNS TXT resource record (defined
   in STD 13, RFC 1035) to contain new types of information. The
   principal advantage of such an approach is that it requires no change
   to most existing DNS servers. It is not intended to replace the
   process by which new resource records are defined and implemented.

2. Format of TXT record

   To store new types of information, the TXT record uses a structured
   format in its TXT-DATA field. The format consists of the attribute
   name followed by the value of the attribute. The name and value are
   separated by an equals sign (=).

For example, the following TXT records contain attributes specified
in this fashion:

```
host.widgets.com    IN    TXT    "printer=lpr5"
sam.widgets.com     IN    TXT    "favorite drink=orange juice"
```

The general syntax is:

```
<owner> <class> <ttl> TXT "<attribute name>=<attribute value>"
```

Attribute Names

Any printable ASCII character is permitted for the attribute name.
If an equals sign is embedded in the attribute name, it must be
quoted with a preceding grave accent (or backquote: "`"). A
backquote must also be quoted with an additional "`".

Attribute Name Matching Rules

The attribute name is considered case-insensitive. For example, a
lookup of the attribute "Favorite Drink" would match a TXT record
containing "favorite drink=Earl Grey tea".

During lookups, TXT records that do not contain an unquoted "=" are
ignored. TXT records that seem to contain a null attribute name,
that is, the TXT-DATA starts with the character "=", are also
ignored.

Leading and trailing whitespace (spaces and tabs) in the attribute
name are ignored unless they are quoted (with a "`"). For example,
"abc" matches " abc<tab>" but does not match "` abc".

Note that most DNS server implementations require a backslash (\) or
double quote (") in a text string to be quoted with a preceding
backslash. Accent grave ("`") was chosen as a quoting character in
this syntax to avoid confusion with "\" (and remove the need for
confusing strings that include sequences like "\\\\").

Attribute Values

All printable ASCII characters are permitted in the attribute value.
No characters need to be quoted with a "`". In other words, the
first unquoted equals sign in the TXT record is the name/value
delimiter. All subsequent characters are part of the value.

Once again, note that in most implementations the backslash character
is an active quoting character (and must, itself, be quoted).

All whitespace in the attribute value is returned to the requestor
(it is up to the application to decide if it is significant.)

Examples

<sp> indicates a space character.

```
Attribute      Attribute       Internal Form          External Form
Name           Value           (server to resolver)   (TXT record)

color          blue            color=blue             "color=blue"
equation       a=4             equation=a=4           "equation=a=4"
a=a            true            a`=a=true              "a`=a=true"
a\=a false             a\`=a=false            "a\\`=a=false"
=              \=              `==\=                  "`==\\="

string         "Cat"           string="Cat"           "string=\"Cat\""
string2        `abc`           string2=``abc``        "string2=``abc``"
novalue                        novalue=               "novalue="
a b            c d             a b=c d                "a b=c d"
abc<sp>        123<sp>         abc` =123<sp>          "abc` =123 "
```

3. Application Usage

The attributes can be accessed by the standard resolver library, but
it is recommended that a library routine designed specially for this
attribute format be used. Such a routine might provide an analogue
to gethostbyname:

```
        getattributebyname(objectname,         name of object
                           attributename,      name of attribute
                           attributevalue,     pointer to buffer
                           attributevaluelen)  length of buffer
```

This routine would remove all quoting characters before returning the
information to the caller. A more complex routine could return
attributes with multiple values, or several different attributes.

4. Attribute Name Registration

To permit ease of interoperability and to reduce the chance of naming
conflicts, a registration process for well known attribute names
might be established. This could be a periodically updated list of
names and/or adherence to other name registration mechanisms such as
published object identifiers.

This paper does not address attribute name registration.

5. Restrictions

   Some DNS server implementations place limits on the size or number of
   TXT records associated with a particular owner. Certain
   implementations may not support TXT records at all.

6. REFERENCES and BIBLIOGRAPHY

   [1] Stahl, M., "Domain Administrators Guide", RFC 1032, Network
       Information Center, SRI International, November 1987.

   [2] Mockapetris, P., "Domain Names - Concepts and Facilities", STD
       13, RFC 1034, USC/Information Sciences Institute, November 1987.

   [3] Mockapetris, P., "Domain Names - Implementation and
       Specification", STD 13, RFC 1035, USC/Information Sciences
       Institute, November 1987.

   [4] Mockapetris, P., "DNS Encoding of Network Names and Other Types",
       RFC 1101, USC/Information Sciences Institute, April 1989.

7. Security Considerations

   Security issues are not discussed in this memo.

8. Author's Address

   Rich Rosenbaum
   Digital Equipment Corporation
   550 King Street, LKG2-2/Z7
   Littleton, MA 01460-1289

   Phone: 508-486-5922
   Email: rosenbaum@lkg.dec.com