

Network Working Group
Request for Comments: 4833
Updates: 2132
Category: Standards Track

E. Lear
Cisco Systems GmbH
P. Eggert
UCLA
April 2007

Timezone Options for DHCP

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

Two common ways to communicate timezone information are POSIX 1003.1 timezone strings and timezone database names. This memo specifies DHCP options for each of those methods. The DHCPv4 time offset option is deprecated.

1. Introduction

This memo specifies a means to provide hosts with more accurate timezone information than was previously available. To do this we make use of two commonly used methods to configure timezones:

- o POSIX TZ strings
- o Reference to the name of the time zone entry in the TZ Database

POSIX [1] provides a standard for how to express timezone information in a character string. Use of such a string can provide accuracy for at least one transition into and out of daylight saving time (DST), and possibly for more transitions if the transitions are regular enough (e.g., "second Sunday in March at 02:00 local time"). However, for accuracy over longer periods that involve daylight-saving rule changes or other irregular changes, a more detailed mechanism is necessary.

The TZ Database [7] that is used in many operating systems provides backwards consistency and accuracy for almost all real-world locations since 1970. The TZ database also attempts to provide a stable set of human readable timezone identifiers. In addition, many systems already make use of the TZ database, and so the names used are a de facto standard. Because the TZ database contains more information, one can heuristically derive the POSIX information from a TZ identifier (see [10] for an example), but the converse is not true.

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 [2].

1.1. Related Work

Dynamic Host Configuration Protocol (DHCP) [3] provides a means for hosts to receive configuration information relating to their current location within an IP version 4 network. [5] similarly does so for IP version 6 networks. RFC 2132 [4] specifies an option to provide client timezone information in the form of an offset in seconds from UTC. The information provided in that option is insufficient for the client to determine whether it is in daylight saving time, and when to change into and out of daylight saving time. In order for the client to properly represent local wall clock time in a consistent and accurate fashion the DHCP server would have to time lease expirations of affected clients to the beginning or end of DST, thus effecting a self stress test (to say the least) at the appointed hour.

In addition, an offset is not sufficient to determine the actual timezone in which a client resides, and thus there is no means to derive a human readable abbreviation such as "EST" or "EDT".

VTIMEZONE elements are defined in the iCalendar specification [9]. Fully specified they provide a level of accuracy similar to the TZ database. However, because there is currently no global registry of VTIMEZONE TZIDs (although one has been proposed; see [8]), complete accuracy requires that a full entry must be specified. To achieve the same information would range from 300 octets upwards with no particular bound. Furthermore, at the time of this writing the authors are aware of no operating system that natively takes advantage of VTIMEZONE entries. It might be possible to include an option for a TZURL. However, in a cold start environment, it will be bad enough that devices are stressing the DHCP server, and perhaps unwise to similarly afflict other components.

2. New Timezone Options for DHCPv4

The following two options are defined for DHCPv4:

PCode	Len	TZ-POSIX String
100	N	IEEE 1003.1 String

TCode	Len	TZ-Database String
101	N	Reference to the TZ Database

Per RFC 2939 [6], IANA allocated PCode (100) and TCode (101).

Len is the one-octet value of the length of the succeeding string for each option.

The string values that follow Len are described below. Note that they are NOT terminated by an ASCII NULL.

3. New Timezone Options for DHCPv6

The semantics and content of the DHCPv6 encoding of these options are exactly the same as the encoding described for DHCPv4, other than necessary differences between the way options are encoded in DHCPv4 and DHCPv6.

Specifically, the DHCPv6 new timezone options are described below:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| OPTION_NEW_POSIX_TIMEZONE | option-length |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               TZ POSIX String
|                               ...
+-----+-----+-----+-----+-----+-----+-----+-----+

```

option-code: OPTION_NEW_POSIX_TIMEZONE(41)

option-length: the number of octets of the TZ POSIX String Index described below.

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| OPTION_NEW_TZDB_TIMEZONE | option-length |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               TZ Name
+-----+-----+-----+-----+-----+-----+-----+-----+

```

option-code: OPTION_NEW_TZDB_TIMEZONE(42)

option-length: the number of octets of the TZ Database String Index described below.

4. The TZ POSIX String

TZ POSIX string is a string suitable for the TZ variable as specified by [1] in Section 8.3, with the exception that a string may not begin with a colon (":"). This string is NOT terminated by an ASCII NULL. Here is an example:

```
EST5EDT4,M3.2.0/02:00,M11.1.0/02:00
```

In this case, the string is interpreted as a timezone that is normally five hours behind UTC, and four hours behind UTC during DST, which runs from the second Sunday in March at 02:00 local time through the first Sunday in November at 02:00 local time. Normally the timezone is abbreviated "EST" but during DST it is abbreviated "EDT".

Clients and servers implementing other timezone options MUST support this option for basic compatibility.

5. The TZ Name

TZ Name is the name of a Zone entry in the database commonly referred to as the TZ database. Specifically, in the database's textual form, the string refers to the name field of a zone line. In order for this option to be useful, the client must already have a copy of the database. This string is NOT terminated with an ASCII NULL.

An example string is Europe/Zurich.

Clients must already have a copy of the TZ Database for this option to be useful. Configuration of the database is beyond the scope of this document. A client that supports this option SHOULD prefer this option to POSIX string if it recognizes the TZ Name that was returned. If it doesn't recognize the TZ Name, the client MUST ignore this option.

6. Use of the Timezone String(s) Returned from the Server

This specification presumes the DHCP server has some means of identifying which timezone the client is in. One obvious approach would be to associate a subnet or group of subnets with a timezone, and respond with this option accordingly.

When considering which option to implement on a client, one must choose between the TZ Name, which should be easier for users to configure and which provides accuracy over longer historical periods, and the TZ POSIX string, which does not require regular updating of a copy of the TZ Database. The TZ Name is better for most uses, in particular those cases where the timezone name might persist in a database for long periods of time, but the TZ POSIX string may be more suitable for small-footprint applications that are expertly maintained.

So that clients need not request both options, servers who implement either timezone option SHOULD implement the other one as well. This association can be established by the server's administrator. A basic server can transmit option values to the client without parsing or validating them. A more advanced server might have a copy of the TZ database and validate TZ names against this copy, or derive TZ POSIX strings heuristically from TZ names to simplify administration.

As a matter of practicality, the client will use this information at its discretion to configure the current timezone in which it resides.

It will periodically be necessary for a DHCP server to update the timezone string, based on administrative changes made by local jurisdictions (say, for instance, counties in Indiana). While the

authors do not expect this to be a lower bound on a lease time in the vast majority of cases, there may be times when anticipation of a change dictates prudence, as certain governments give little if any notification.

The effect of a changed timezone on client applications is not specified by this memo, but it may be helpful to note common problems in this area. Often, client applications consult the timezone setting only during process initialization, or inherit the setting from a parent process, so existing processes on a client may ignore a timezone change returned from the server. Sometimes it is normal and expected for processes on the same client to have different timezone settings (e.g., remote logins), and so client implementations should consider these ramifications of changing timezone settings of existing processes.

7. The New Timezone Option and Lease Times

When a lease has expired and new information is not forthcoming, the client MAY continue to use timezone information returned by the server. This follows the principle of least astonishment.

8. Deprecation of Time Offset Option

Because this option provides a superset of functionality to the previous IPv4 time offset option (tag 2), and in order to maintain consistency between IPv4 and IPv6 implementation, the older option is deprecated. Current implementations that support the time offset IPv4 option SHOULD implement this option also. Other implementations SHOULD implement this option, and SHOULD NOT implement the time offset IPv4 option. As a matter of transition, clients that already use the time offset option MAY request the time offset option and the timezone option.

9. Security Considerations

An attacker could provide erroneous information to a client. It is possible that someone might miss a meeting or otherwise show up early, or that heavy machinery or other critical functions might act at the wrong time or fail to act. If clients have job processing tools, such as cron that operate on wall clock time, it is possible that certain jobs could be triggered either earlier or later, or even repeated or skipped entirely if scheduled during a DST transition. In such cases, the client operating system might do well to confirm timezone changes with a human.

Clients using the POSIX option should beware of any time zone setting specifying unusual characters (e.g., control characters) in the standard or daylight-saving abbreviations, as this might well trigger security-relevant bugs in applications.

Clients using the POSIX option should also be suspicious of any timezone setting whose UTC offset exceeds 25 hours (the POSIX limit, if the default daylight-saving offset is used). As of this writing, the maximum UTC offset is 14 hours in practice, but governments may extend this somewhat in the future.

10. IANA Considerations

The IANA has allocated DHCPv4 and DHCPv6 option codes for this purpose and references this document.

The IANA has annotated the time offset IPv4 option (tag 2) as deprecated, with a reference to this document.

11. Acknowledgments

This document specifies a means to exchange timezone information. The hard part is actually collecting changes to the various databases from scattered sources around the world. The many volunteers on the mailing list tz@elsie.nci.nih.gov have done this nearly thankless task for many years. Thanks also go to Ralph Droms, Bernie Volz, Ted Lemon, Lisa Dusseault, John Hawkinson, Stig Venaas, and Simon Vaillancourt for their efforts to improve this work.

12. References

12.1. Normative References

- [1] "Standard for Information Technology - Portable Operating System Interface (POSIX) - Base Definitions", IEEE Std 1003.1-2004, December 2004.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [3] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [4] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.
- [5] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [6] Droms, R., "Procedures and IANA Guidelines for Definition of New DHCP Options and Message Types", BCP 43, RFC 2939, September 2000.
- [7] Eggert, P. and A. Olson, "Sources for Time Zone and Daylight Saving Time Data", <<http://www.twinsun.com/tz/tz-link.htm>>.

12.2. Informational References

- [8] Vaillancourt, S., "Calconnect.org TC Timezone Technical Committee: Timezone Registry and Service Recommendations", April 2006.
- [9] Dawson, F. and Stenerson, D., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 2445, November 1998.
- [10] Eggert, P. and E. Reingold, "cal-dst.el --- calendar functions for daylight savings rules", <http://cvs.savannah.gnu.org/viewcvs/*checkout*/emacs/lisp/calendar/cal-dst.el?root=emacs>.

Authors' Addresses

Eliot Lear
Cisco Systems GmbH
Glatt-com
Glattzentrum, ZH CH-8301
Switzerland

Phone: +41 1 878 9200
EMail: lear@cisco.com

Paul Eggert
UCLA
Computer Science Department
4532J Boelter Hall
Los Angeles, CA 90095
USA

Phone: +1 310 825 3886
EMail: eggert@cs.ucla.edu

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

