Network Working Group   M. Douglass
Internet-Draft   RPI
Intended status: Standards Track   C. Daboo
Expires: April 30, 2012   Apple
                                                   October 28, 2011

                      Timezone Service Protocol
                  draft-douglass-timezone-service-03

Abstract

   This document defines a timezone service protocol that allows
   reliable, secure and fast delivery of timezone information to client
   systems such as calendaring and scheduling applications or operating
   systems.

Table of Contents

1. Introduction

   Timezone information, in general, combines a Universal Coordinated
   Time (UTC) offset with daylight saving time (DST) rules. Timezones
   are typically tied to specific geographic and geopolitical regions.
   Whilst the UTC offset for particular regions changes infrequently,
   DST rules can change frequently and sometimes with very little notice
   (sometimes hours before a change comes into effect).

   Calendaring and scheduling systems, such as those that use iCalendar
   [RFC5545], as well as operating systems, critically rely on timezone
   information to determine the correct local time. As such they need
   to be kept up to date with changes to timezone information. To date
   there has been no fast and easy way to do that. Timezone data is
   often supplied in the form of a set of data files that have to be
   "compiled" into a suitable database format for use by the client
   application or operating system. In the case of operating systems,
   those changes often only get propagated out to client machines when
   there is an operating system update and those may not be frequent
   enough to ensure accurate timezone data is always in place.

   This specification defines a timezone service protocol that allows
   for fast, reliable and accurate delivery of timezone information to
   client systems. This protocol is based on HTTP [RFC2616] using a
   REST style API.

   This specification does not specify the source of the timezone
   information. It is assumed that a reliable and accurate source is
   available. One such source is the Olson database - see
   [I-D.lear-iana-timezone-database] for a proposal to host the data in
   IANA.

   This specification does not address the need for global timezone
   identifiers for timezone data.

   Discussion of this document should take place on the calsify mailing
   list calsify@ietf.org

1.1. Conventions

   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 [RFC2119].

   This document uses XML DTD fragments ([W3C.REC-xml-20081126], Section
   3.2) as a purely notational convention.

   The namespace "urn:ietf:params:xml:ns:timezone-service" is reserved

for the XML elements defined in this specification, its revisions,
and related specifications. XML elements defined by individual
implementations MUST NOT use the
"urn:ietf:params:xml:ns:timezone-service" namespace, and instead
should use a namespace that they control.

When XML element types in the
"urn:ietf:params:xml:ns:timezone-service" namespace are referenced in
this document outside of the context of an XML fragment, the string
"TZ:" will be prefixed to the element types.

1.2. Glossary of terms

The following terms with the given meanings are used throughout this
document.

Timezone Server: A server implementing the Timezone Service Protocol
   defined by this specification;

Timezone Identifier: A globally unique name which identifies
   timezone information.

2. Architectural Overview

The overall process for the delivery of timezone information can be
visualized via the diagram shown below.

```
                    =====================  =====================
      (a)           |    Contributors   |  |    Contributors   |
                    =====================  =====================
                             |                      |
                    =====================  =====================
      (b)           |    Publisher A    |  |    Publisher B    |
                    =====================  =====================
                                       |
                             =====================
      (c)                    |      Provider     |
                             =====================
                            /           |          \
                           /            |           \
                    =====================  |  =====================
      (d)           |      Provider     |  |  |      Provider     |
                    =====================  |  =====================
                       |         |         |              |
                       |         |         |              |
                    =========  =========  =========     =========
      (e)           | Client |  | Client |  | Client |     | Client |
                    =========  =========  =========     =========
```
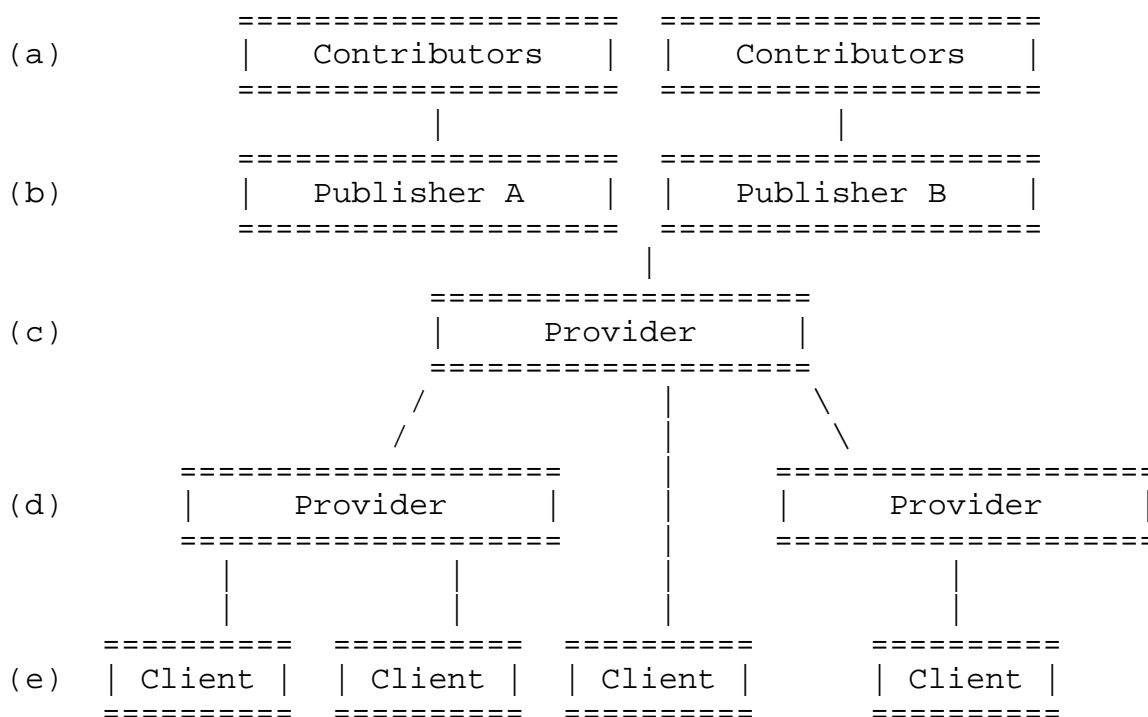
                  Figure 1: Timezone Service Architecture

   The overall service is made up of several layers:

   (a) Contributors: Individuals, governments or organizations which
       provide information about timezone definitions to the publishing
       process. There can be many contributors.

   (b) Publishers: Publishers aggregate information from contributors,
       determine the reliability of the information and, based on that,
       generate timezone definitions. There can be many publishers, each
       getting data from many different contributors. In some cases a
       publisher may choose to "re-publish" data from another publisher.

    (c) Root Providers: Servers which obtain and then provide the
        timezone data from publishers and make that available to other
        servers or clients. There can be many root providers. Root
        providers can choose to supply timezone data from one or more (or
        all) publishers.

   (d) Local Providers: Servers which handle the bulk of the requests
       and reduce the load on root servers. These will typically be
       simple caches of the root server, located closer to clients. For
       example a large Internet Service Provider (ISP) may choose to
       setup their own local provider to allow clients within their
       network to make requests of that server rather than making

requests of servers outside their network. Local servers will
cache and periodically refresh data from the root servers.

(e) Clients: Applications, operating systems etc., that make use of
    timezone data and retrieve that from either root or local
    providers.

Some of those layers may be coalesced by implementors. For example,
a vendor may choose to implement the entire service as a single
monolithic virtual server with the address embedded in distributed
systems. Others may choose to provide a service consisting of
multiple layers of providers, many local servers and a small number
of root servers.

This specification is only concerned with the protocol used to
exchange data between providers and from provider to client. This
specification does not specify how contributors pass their
information to publishers, nor how those publishers vet that
information to obtain the level of trust required of the final
result.

3. Timezone Naming

3.1. Timezone Identitifiers

Timezone identitiers are the canonical names for identifiers. There
MUST be one and only one identifier per specification. iCalendar
Section 3.8.3.1 [RFC5545] has this text on the identifier:

This is the label by which a time zone calendar component is
referenced by any iCalendar properties whose value type is
either DATE-TIME or TIME and not intended to specify a UTC
or a "floating" time. The presence of the SOLIDUS character
as a prefix, indicates that this "TZID" represents an unique
ID in a globally defined time zone registry (when such
registry is defined).

This specification does not define what identifiers should be used.
While the above specifies that names must start with "/" to be
globally unique, it is assumed that a set of timezone identifiers
will be considered the default set with an implied "/" preceding the
identifier.

3.2. Timezone Identitifier Aliases

Timezone identifier aliases map one timezone identifier onto another.
Aliases allow the timezone service to map old timezone specifications

onto a new specification.

A timezone alias can be provided in the timezone data and should be
mapped on to the target timezone specification. This can be done by
the client or by the server.

The client can request that aliases be returned along with summary
information when listing timezones provided by the timezone service.
Clients can then map received timezone identifiers on to the target
perhaps avoiding a fetch of the target specification.

Alternatively, the client can just fetch the timezone specification
by supplying the timezone identifier alias. The service will return
the target timezone specification with the tzid optionally set to
that requested.

Aliases are identifiers and as such are NOT localized names. They
are also inexact matches to canonical names and may vary over time.
For example US/Eastern is usually mapped on to America/New_York even
though it covers many more local timezones.

3.3. Timezone Localized Names

Localized names are names for timezones which can be presented to a
user in their own language. Each timezone specification may have one
or more localized names associated with it. Each name SHOULD be
unique in its locale as it may be used in a list presented to the
user.

The timezone service can be requested to return localized names for
timezones by the addition of request parameters. Names in multiple
lanaguages can be returned in a single request.

3.4. Timezone Name Searches

When requesting information from the timezone service identifiers,
aliases or names may be used depending on context. This will be more
explicitly defined below for each action. In general however, if a
"tzid" request parameter is used then the value may be an identifier
or an alias. When the "name" parameter is used it may be an
identifier, an alias or a localized name.

4. Timezones Service Protocol

4.1. General Considerations

4.1.1. Timezone Formats

   The default format for returning timezone definitions is the
   iCalendar [RFC5545] data format. In addition, the iCalendar-in-XML
   [RFC6321] representation is also available.

4.1.2. Inactive Timezones

   Over time zone timezone definitions are replaced by others, but are
   maintained for historical purposes. Often times clients are only
   concerned with timezone whose definitions are valid for current and
   future dates and times. When listing timezones provided by a
   timezone service, the server will by default only provide the list of
   "active" timezones. However, clients can use a request parameter to
   have the server also return details for "inactive" timezones.

4.1.3. Conditional Timezone Requests

   Timezone information is generally slow moving. However, changes need
   to be distributed in a timely manner. The list of timezones that
   change from even year-to-year will typically be relatively small.

   When listing timezones, a global timestamp is returned by the server,
   and that can be used later by clients to determine if any
   "substantive" change has occurred in the timezone data. Clients can
   use a conditional list request, supplying a previous global timestamp
   value, to limit the results to timezones which have changed in a
   "substantive" manner since that previous global timestamp. This
   allows clients to cache the last global timestamp and to periodically
   poll the server for possible changes. Clients SHOULD limit automatic
   polling to no more than once a day.

   A "substantive" change is one which affects the calculated onsets for
   a timezone or a change to the region it covers. Changes to
   properties such as description are not treated as a "substantive"
   change.

4.1.4. Expanded Timezone Data

   Determining timezone offsets at a particular point in time is often a
   complicated process as the rules for daylight saving time can be
   complex. To help with this, the timezone service provides an
   operation that allows clients to request the server to expand a
   timezone definition into a set of "observances" over a fixed period
   of time. Each of these observances describes a local onset time and
   UTC offsets for the prior time and the observance time. Together,

these provide a quick way for "thin" clients to determine an
appropriate UTF offset for an arbitrary date without having to do
full timezone expansion themselves.

4.1.5. Server Requirements

All servers MUST deliver timezone information for all timezones.
This means that any client API implementation can go to a single
server to get all timezone information. In turn, any server can
refresh any of the data from any other server - though the root
servers may provide the most up-to-date copy of the data.

All service providers MUST deliver functionally equivalent data for
all timezones. This ensures that all parties in a contract agree on
the UTC time. Service providers (or publishers) may choose to map a
particular region on to a different timezone identifier to correct a
deficiency in the original timezone specification.

With opaque timezone identifiers this remapping may be short lived
and the mapping can revert to the original identifier once the
deficiency has been addressed by the publisher.

4.1.6. Extensions

This protocol is designed to be extensible through a standards based
registration mechanism (see Section 10). It is anticipated, that
other useful timezone operations will be added in the future (e.g.,
mapping a geographical location to timezone identifiers, getting
change history for timezones). To that end, servers MUST return a
description of their capabilities. This will allow clients to
determine if new features have been installed and, if not, fall back
on earlier features or disable some client capabilities.

4.2. Server Protocol

The interactions are simple HTTP [RFC2616] requests. Most security
considerations are already handled adequately by HTTP. However,
given the nature of the data being transferred and the requirement it
be correct all interactions between client and server SHOULD use an
HTTP connection protected with TLS [RFC5246] as defined in [RFC2818].

The HTTP GET request method is used, with information passed in
request parameters. The "action" request parameter specifies which
operation is to take place, other request parameters act as arguments
to that operation.

4.2.1. Error Responses

   The following are examples of response codes one would expect to be
   used by the server. Note, however, that unless explicitly prohibited
   any 2/3/4/5xx series response code may be used in a response.

      200 (OK) - The command succeeded.

      400 (Bad Request) - The Sender has provided an invalid request
      parameter.

      404 (Not Found) - The timezone was not found.

   When an error status is set the server SHOULD respond with some
   descriptive text in an error element Section 7.29

4.3. Client Guidelines

4.3.1. Discovery

   Client implementations need to either know where the timezone service
   is located or discover it through some mechanism. To use a timezone
   service, a client needs an FQDN, port and HTTP request-URI path.

4.3.1.1. Timezone Service SRV Service Labels

   [RFC2782] defines a DNS-based service discovery protocol that has
   been widely adopted as a means of locating particular services within
   a local area network and beyond, using SRV RR records. This can be
   used to discover a service's FQDN and port.

   This specification adds two service types for use with SRV records:

   timezone: Identifies a Timezone server that uses HTTP without
      transport layer security ([RFC2818]).

   timezones: Identifies a Timezone server that uses HTTP with
      transport layer security ([RFC2818]).

   Clients MUST honor "TTL", "Priority" and "Weight" values in the SRV
   records, as described by [RFC2782].

   Example: service record for server without transport layer security

         _timezone._tcp   SRV 0 1 80 tz.example.com.

Example: service record for server with transport layer security

        _timezones._tcp   SRV 0 1 443 tz.example.com.

4.3.1.2. Timezone Service TXT records

When SRV RRs are used to advertise a timezone service, it is also
convenient to be able to specify a "context path" in the DNS to be
retrieved at the same time. To enable that, this specification uses
a TXT RR that follows the syntax defined in Section 6 of
[I-D.cheshire-dnsext-dns-sd] and defines a "path" key for use in that
record. The value of the key MUST be the actual "context path" to
the corresponding service on the server.

A site might provide TXT records in addition to SRV records for each
service. When present, clients MUST use the "path" value as the
"context path" for the service in HTTP requests. When not present,
clients use the ".well-known" URI approach described next.

Example: text record for service with transport layer security

          _timezones._tcp   TXT path=/timezones

4.3.1.3. Timezone Service Well-Known URI

A "well-known" URI [RFC5785] is registered by this specification for
the Timezone service, "timezone" (see Section 10). This URI points
to a resource that the client can use as the initial "context path"
for the service they are trying to connect to. The server MUST
redirect HTTP requests for that resource to the actual "context path"
using one of the available mechanisms provided by HTTP (e.g., using a
301, 303, 307 response). Clients MUST handle HTTP redirects on the
".well-known" URI. Servers MUST NOT locate the actual timezone
service endpoint at the ".well-known" URI as per Section 1.1 of
[RFC5785].

Servers SHOULD set an appropriate Cache-Control header value (as per
Section 14.9 of [RFC2616]) in the redirect response to ensure caching
occurs or does not occur as needed, or as required by the type of
response generated. For example, if it is anticipated that the
location of the redirect might change over time, then a "no-cache"
value would be used.

To facilitate "context path's" that might differ from user to user,
the server MAY require authentication when a client tries to access
the ".well-known" URI (i.e., the server would return a 401 status
response to the unauthenticated request from the client, then return
the redirect response only after a successful authentication by the

client).

4.3.1.3.1. Example: well-known URI redirects to actual context path

A Timezone server has a "context path" that is "/servlet/timezone".
The client will use "/.well-known/timezone" as the path for the
service process after it has first found the FQDN and port number via
an SRV lookup or via manual entry of information by the user from
which the client can parse suitable information. When the client
makes its initial HTTP request against "/.well-known/timezone", the
server would issue an HTTP 301 redirect response with a Location
response header using the path "/servlet/timezone". The client would
then "follow" this redirect to the new resource and continue making
HTTP requests there.

4.3.2. Initial Synchronization of All Timezones

When a secondary service or a client wishing to cache all timezone
data first starts or wishes to do a full refresh it synchronizes with
another server by first issuing a list operation with
returnall="true". The client should preserve the returned datastamp
for subsequent use. Each timezone in the returned list can then be
fetched and stored locally. In addition a mapping of aliases to
timezones can be built.

4.3.3. Subsequent Synchronization of All Timezones

Periodically a secondary service or a client caching all timezone
data needs to synchronize with another server. To do so it should
issue a list operation with the changedsince parameter set to the
value of the datestamp returned at the last synchronization. The
client should again preserve the returned datastamp for subsequent
use. Each timezone in the returned list can then be fetched and
stored locally.

Note, this process makes no provision for handling deleted timezones.
In general it is bad practice to delete timezones as they may now be
in use by consumers of timezone data.

5. Operation Parameters

All requests require the "action" request parameter to define what
action is required of the server.

Servers MUST support the following request parameters.

5.1. "action" Parameter

   Name: action

   Description: Specify the action to be carried out.

   Value: Any IANA registered operation name.

5.2. "format" Parameter

   Name: format

   Description: Specify the format for the timezone data returned by
      the server as a standard MIME [RFC2046] media-type. If absent the
      iCalendar [RFC5545] format will be returned with the timezones
      contained within a "VCALENDAR" object (i.e., a default media-type
      of "text/calendar").

   Value: A MIME [RFC2046] media-type. The server MUST support the
      following values:

      text/calendar: Return data as "VTIMEZONE" components embedded in
         a "VCALENDAR" object as per [RFC5545].

      application/calendar+xml: Return data using the XML
         representation of iCalendar data as per iCalendar in XML
         [RFC6321].

5.3. "changedsince" Parameter

   Name: changedsince

   Description: Specify the timestamp for a conditional timezone list
      request in order to restrict the result to timezones changed since
      the given timestamp (see Section 6.2).

   Value: An XML [REF] UTC date-time value, typically a value returned
      in a previous "dtstamp" XML element.

5.4. "start" Parameter

   Name: start

   Description: Specify the inclusive start of a period.

      Value: An iCalendar [RFC5545] "DATE" or "DATE-TIME" value. If an
         'end' request parameter is also present, then both the 'start' and
         'end' values MUST be of the same "DATE" or "DATE-TIME" value
         types.

5.5. "end" Parameter

      Name: end

      Description: Specify the exclusive end of a period.

      Value: An iCalendar [RFC5545] "DATE" or "DATE-TIME" value. If a
         'start' request parameter is also present, then both the 'start'
         and 'end' values MUST be of the same "DATE" or "DATE-TIME" value
         types.

5.6. "lang" Parameter

      Name: lang

      Description: Specify the language in which locale specific values
         are to be returned. e.g., when fetching display names, if a
         language is specified, only display names for that language will
         be returned.

      Value: The value follows the specifications in [RFC4646].

5.7. "returnall" Parameter

      Name: returnall

      Description: If present indicates that all timezones should be
         returned. When fetching the list of timezones the default action
         is to omit inactive timezones. The inclusion of this parameter
         causes inactive timezones to be returned in the list (see
         Section 6.2).

      Value: This parameter takes no value.

5.8. "tzid" Parameter

      Name: tzid

      Description: This parameter is used to identify a timezone to be
         targeted by an operation.

Value: A timezone identifier or alias. In some cases the special
   value "*" is used to indicate that all timezones should be
   matched.

5.9. "name" Parameter

   Name: name

   Description: This parameter is used to specify a name for queries.

   Value: A timezone identifier, alias or localized name. This
      parameter is used when searching for matching specifications.

6. Operations

   Servers MUST support the following operations.

6.1. "capabilities" Operation

   Name: capabilities

   Description: This operation returns the capabilities of the server,
      allowing clients to determine if a specific feature has been
      deployed and/or enabled.

   Parameters:

      action REQUIRED, value MUST be "capabilities"

   Response An XML document containing a "capabilities" (Section 7.1)
      element as the root element. Within that element there is an
      informational section describing the server and a section for each
      operation supported by the server.

6.1.1. Example: Get Capabilities

   In this example the client requests the server capabilities.

   >> Request <<

   GET /?action=capabilities HTTP/1.1
   Host: tz.example.com

   >> Response <<

   HTTP/1.1 200 OK
   Date: Wed, 4 Jun 2008 09:32:12 GMT

```
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<capabilities xmlns="urn:ietf:params:xml:ns:timezone-service">
  <info>
    <primary-source>Olson:2011m</primary-source>
    <contact>mailto:tzs@example.org</contact>
  </info>

  <operation>
    <action>list</action>
    <description>List timezone identifiers and localized forms
    </description>

    <accept-parameter>
      <name>lang</name>
      <required>false</required>
      <multi>true</multi>
      <description>Specify desired localized form(s)</description>
    </accept-parameter>

    <accept-parameter>
      <name>changedsince</name>
      <required>false</required>
      <multi>false</multi>
      <description>Limit result to timezones changed since the
        given date
      </description>
    </accept-parameter>

    <accept-parameter>
      <name>returnall</name>
      <required>false</required>
      <multi>false</multi>
      <description>If present inactive timezones will be returned.
      </description>
    </accept-parameter>
  </operation>

  <operation>
    <action>get</action>
    <description>
      Returns one or more timeszones as specified by the
      tzid parameter.
    </description>

    <accept-parameter>
```

```
      <name>format</name>
      <required>false</required>
      <multi>false</multi>
      <value>text/calendar</value>
      <value>application/calendar+xml</value>
      <description>Specify required format for timezone.
      </description>
    </accept-parameter>

    <accept-parameter>
      <name>lang</name>
      <required>false</required>
      <multi>true</multi>
      <description>Specify desired localized form(s)</description>
    </accept-parameter>

    <accept-parameter>
      <name>tzid</name>
      <required>true</required>
      <multi>true</multi>
      <description>Specify desired timezone identifiers
      </description>
    </accept-parameter>
  </operation>

  <operation>
    <action>expand</action>
    <description>
      Expands the specified timezone(s) into local onset and UTC
      offsets
    </description>

    <accept-parameter>
      <name>tzid</name>
      <required>true</required>
      <multi>true</multi>
      <description>Specify desired timezone identifiers</description>
    </accept-parameter>

    <accept-parameter>
      <name>start</name>
      <required>false</required>
      <multi>false</multi>
      <description>
        Specify start of the period of interest. If omitted the
        current year is assumed.
      </description>
    </accept-parameter>
```

```
      <accept-parameter>
        <name>end</name>
        <required>false</required>
        <multi>false</multi>
        <description>
          Specify end of the period of interest.
          If omitted the current year + 10 is assumed.
        </description>
      </accept-parameter>
    </operation>

    <operation>
      <action>capabilities</action>
      <description>Gets the capabilities of the server</description>
    </operation>
  </capabilities>
```

6.2. "list" Operation

   Name: list

   Description: This operation lists all non alias timezone identifiers
      in summary format with aliases and optional localized data. In
      addition, it returns a timestamp which is the current server
      global last modification value.

   Parameters:

      action   REQUIRED, value MUST be "list"

      lang=<lang-code> OPTIONAL, but MAY occur multiple times.
         Specifies the language or languages for localized information.

      returnall OPTIONAL, but MUST occur only once. If present,
         indicates that all, including inactive, timezones should be
         returned in the response. The TZ:inactive XML element will
         flag those timezones no longer in use.

      changedsince OPTIONAL, but MUST occur only once. If present,
         limits the response to timezones changed since the given
         timestamp. MUST NOT be specified with the tzid parameter.

      tzid=<identifier> OPTIONAL, and MAY occur once or more. If
         specified identifies timezone(s) for which information is to be
         returned. If this parameter is specified changedsince MUST NOT
         be specified.

If tzid is specified the dtstamp element will be returned in
the response. This dtstamp is for the entire set of data and
allows the client to determine if it should refresh its full
set.

   Response: An XML document containing a "timezone-list"
      (Section 7.13) element as the root element.

6.2.1. Example: List timezone identifiers

   In this example the client requests the timezone identifiers and in
   addition requests that the US-English local names be returned.

      >> Request <<

      GET /?action=list&lang=en_US HTTP/1.1
      Host: tz.example.com

      >> Response <<

      HTTP/1.1 200 OK
      Date: Wed, 4 Jun 2008 09:32:12 GMT
      Content-Type: application/xml; charset="utf-8"
      Content-Length: xxxx

      <?xml version="1.0" encoding="utf-8" ?>
      <timezone-list xmlns="urn:ietf:params:xml:ns:timezone-service">
        <dtstamp>2009-10-11T09:32:11Z</dtstamp>
        <summary>
          <tzid>America/New_York</tzid>
          <last-modified>2009-09-17T01:39:34Z</last-modified>
          <alias>US/Eastern</alias>
          <local-name lang="en_US">America/New_York</local-name>
        <summary>
        ...
      </timezone-list>

6.3. "get" Operation

   Name: get

   Description: This operation returns a timezone. If a single
      timezone is specified the response MUST contain an ETag response
      header field indicating the current value of the strong entity tag
      of the timezone resource.

If the identifier is actually an alias to a canonical name the
server will return the specification associated with that
identifier. The substitute-alias parameter specifies whether or
not the alias is to be substitued for the identifier.

Parameters:

action REQUIRED, value MUST be "get"

format=<media-type> OPTIONAL, but MUST occur only once. Return
    information using the specified media-type. In the absence of
    this parameter, the value "text/calendar" MUST be assumed.

lang=<lang-code> OPTIONAL, but MAY occur multiple times. If
    present, specifies the language or languages for localized
    information.

tzid=<identifier> REQUIRED, and MUST occur only once. Identifies
    the timezone for which information is returned. The server
    MUST return an Etag header. Alternatively, if a value of "*"
    is given, returns information for all timezones. The "*"
    option will typically be used by servers that wish to retrieve
    the entire set of timezones supported by another server to re-
    synchronize their entire data cache. Clients will typically
    only retrieve individual timezone data on a case-by-case basis.

substitute-alias=<true|false> OPTIONAL and defaults to false. If
    true and the tzid is an alias it will replace the tzid in the
    returned specification. If false, the returned specification
    will have the canonical identifier.

Response: A document containing all the requested timezone data in
    the format specified.

6.3.1. Example: Get timezone

   In this example the client requests the timezone with a specific
   timezone identifier to be returned

      >> Request <<

      GET /?action=get&tzid=America/New_York
                      &format=text/calendar HTTP/1.1
      Host: tz.example.com

      >> Response <<

      HTTP/1.1 200 OK
      Date: Wed, 4 Jun 2008 09:32:12 GMT
      Content-Type: text/calendar; charset="utf-8"
      Content-Length: xxxx
      ETag: "123456789-000-111"

      BEGIN:VCALENDAR
      ...
      BEGIN:VTIMEZONE
      ...
      END:VTIMEZONE
      END:VCALENDAR

6.4. "expand" Operation

   Name: expand

   Description: This operation expands the specified timezone into a
      list of onset start date/time and offset. The response MUST
      contain an ETag response header field indicating the current value
      of the strong entity tag for the expanded data.

   Parameters:

      action REQUIRED, value MUST be "expand"

      tzid=<identifier> REQUIRED, but MUST only occur once. Identifies
         the timezone for which information is returned. The value "*",
         which has a special meaning in the "get" operation, is not
         supported by this operation.

      lang=<lang-code> OPTIONAL, but MAY occur multiple times. If
         present, specifies the language or languages for localized
         information.

start=date or date-time: OPTIONAL, but MUST occur only once. If
   present, specifies the start of the period of interest. If
   omitted, the current year is assumed.

end=date or date-time: OPTIONAL, but MUST occur only once. If
   present, specifies the end of the period of interest. If
   omitted, the current year + 10 is assumed.

Response: An XML document containing a "timezones" (Section 7.21)
   element as the root element.

6.4.1. Example: Expanded XML Data Format

In this example the client requests a timezone in the expanded form.

```
>> Request <<

GET /?action=expand&tzid=America/New_York HTTP/1.1
Host: tz.example.com

>> Response <<

HTTP/1.1 200 OK
Date: Wed, 4 Jun 2008 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx
ETag: "123456789-000-111"

<?xml version="1.0" encoding="utf-8" ?>
<timezones xmlns="urn:ietf:params:xml:ns:timezone-service">
  <dtstamp>2009-10-11T09:32:11Z</dtstamp>
  <tzdata>
    <tzid>America/New_York</tzid>
    <calscale>Gregorian</calscale>

    <observance>
      <name>Daylight</name>
      <onset>2008-03-09T07:00:00Z</onset>
      <utc-offset-from>-05:00</utc-offset-from>
      <utc-offset-to>-04:00</utc-offset-to>
    </observance>

    <observance>
      <name>Standard</name>
      <onset>2008-11-02T07:00:00Z</onset>
      <utc-offset-from>-04:00</utc-offset-from>
      <utc-offset-to>-05:00</utc-offset-to>
    </observance>

    <observance>
      <name>Daylight</name>
      <onset>2009-03-08T07:00:00Z</onset>
      <utc-offset-from>-05:00</utc-offset-from>
      <utc-offset-to>-04:00</utc-offset-to>
    </observance>

    ...
  </tzdata>
</timezones>
```

6.5. "find" Operation

   Name: find

   Description: This operation allows a client to query the timezone
      service for a matching identifier, alias or localized name.

   Parameters:

      action REQUIRED, value MUST be "find"

      name=<text> REQUIRED, but MUST only occur once. Identifies the
         name to search for. Only partial matching is supported.

      lang=<lang-code> OPTIONAL, but MAY occur multiple times. If
         present, specifies the language or languages for localized
         information.

   Response: The response has the same format as the "list"
      (Section 6.2) operation, with one result element per successful
      match.

6.5.1. Example: Find operation

   In this example the client asks for information about "America/
   New_York".

      >> Request <<

      GET /?action=find&name=America/New_York HTTP/1.1
      Host: tz.example.com

      >> Response <<

      HTTP/1.1 200 OK
      Date: Wed, 4 Jun 2008 09:32:12 GMT
      Content-Type: application/xml; charset="utf-8"
      Content-Length: xxxx

   TBD

7. XML Definitions

   XML elements used by this specification are defined here.

7.1. TZ:capabilities XML element

   Name: capabilities

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Root element encapsulating timezone service capabilities
      information.

   Description: This defines the root (top-level) element used as the
      container for capabilities information.

   Definition:

   <!ELEMENT capabilities (info, operation*)>

7.2. TZ:info XML element

   Name: info

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Element encapsulating server information

   Description: This defines the element used as the container for
      information about the servers source of data and contacts.

   Definition:

   <!ELEMENT info ((source | primary-source), contact*)>

7.3. TZ:source XML element

   Name: source

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Supplied in capabilities to define the source for a
      secondary server

   Description: This element contains the URL of the timezones server
      used to provide the data for this server.

   Definition:

   <!ELEMENT source (#PCDATA)>
   <!-- A valid URL -->

7.4. TZ:primary-source XML element

   Name: primary-source

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Supplied in capabilities to provide information about the
      data source for a primary server

   Description: This element contains some descriptive text, for
      example a filename, to identify the source of the data for this
      server.

   Definition:

   <!ELEMENT primary-source (#PCDATA)>
   <!-- Descriptive text -->

7.5. TZ:contact XML element

   Name: contact

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Supplied in capabilities to provide contact information

   Description: This element can contain any useful contact information
      and may appear multiple times.

   Definition:

   <!ELEMENT source (#PCDATA)>
   <!-- A valid URL -->

7.6. TZ:operation XML element

   Name: operation

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Element encapsulating timezone operation description

   Description: This defines the element used as the container for
      information defining an operation and its parameters.

   Definition:

   <!ELEMENT operation (action, description?, accept-parameter*)>

7.7. TZ:action XML element

   Name: action

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Define name of an action

   Description: Specify name of an operation's action.

   Definition:

   <!ELEMENT action (#PCDATA)>
   <!-- A legal action value -->

7.8. TZ:description XML element

   Name: description

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: A description.

   Description: This element specifies a description that refers to the
      enclosing element.

   Definition:

   <!ELEMENT description (#PCDATA)>

7.9. TZ:accept-parameter XML element

   Name: accept-parameter

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Element defining a timezone operation parameter

   Description: This defines the name, type and characteristics of an
      operation parameter.

   Definition:

   <!ELEMENT accept-parameter (name, required, multi,
                               value*, description?)>

7.10. TZ:required XML element

   Name: required

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Specify if timezone method parameter is required

   Description: Specify if timezone method parameter is required. Same
      form as XML boolean, "true", "false", 0 or 1 are used.

   Definition:

   <!ELEMENT required (#PCDATA)>
   <!-- "true" or "false" -->

7.11. TZ:multi XML element

   Name: multi

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Specify if timezone method parameter is multi-valued

   Description: Specify if timezone method parameter is multi-valued.
      Same form as XML boolean, "true", "false", 0 or 1 are used.

   Definition:

   <!ELEMENT multi (#PCDATA)>
   <!-- "true" or "false" -->

7.12. TZ:value XML element

   Name: value

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Allowed values for a parameter.

Description: This element specifies the allowed values for a
   parameter. If present, only the set of valus specified will be
   allowed by the server.

Definition:

<!ELEMENT value (#PCDATA)>

7.13. TZ:timezone-list XML element

Name: timezone-list

Namespace: urn:ietf:params:xml:ns:timezone-service

Purpose: Root element encapsulating timezone list information.

Description: This defines the root (top-level) element used as the
   container for timezone listing.

Definition:

<!ELEMENT timezone-list (dtstamp, summary*)>

7.14. TZ:dtstamp XML element

Name: dtstamp

Namespace: urn:ietf:params:xml:ns:timezone-service

Purpose: Provide a timestamp value.

Description: Provides the UTC date and time when a substantive
   change was made to the timezone data.

Definition:

<!ELEMENT version (#PCDATA)>
<!-- XML UTC DATE-TIME value -->

7.15. TZ:summary XML element

Name: summary

Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Defines a timezone summary.

   Description: This defines the element that provides summary
      information for a timezone in the timezones list.

   Definition:

   <!ELEMENT summary (tzid, last-modified, inactive?,
                      local-name*, alias*)>

7.16. TZ:tzid XML element

   Name: tzid

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: The identifier for the timezone.

   Description: The text value is the identifier of the timezone being
      referred to.

   Definition:

   <!ELEMENT tzid (#PCDATA)>

7.17. TZ:last-modified XML element

   Name: last-modified

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Provide a timestamp value.

   Description: Provides the UTC date and time that the timezone data
      was last modified.

   Definition:

   <!ELEMENT last-modified (#PCDATA)>
   <!-- XML UTC DATE-TIME value -->

7.18. TZ:inactive XML element

   Name: inactive

Namespace: urn:ietf:params:xml:ns:timezone-service

Purpose: Flags an inactive timezone.

Description: If present indicates that the timezone is inactve, that
   is it does not apply to any current area or date/time.

Definition:

```
<!ELEMENT inactive >
```

7.19. TZ:local-name XML element

Name: local-name

Namespace: urn:ietf:params:xml:ns:timezone-service

Purpose: A localized name for the timezone.

Description: Defines one or more localized names that are used when
   a timezone identifier needs to be presented to a user. The xml:
   lang attribute is used to indicate the language associated with
   each value. If multiple names are provided for the same locale
   the preferred name should be flagged with the PREF="true"
   attribute. In the absence of any indication the client is free to
   choose.

Definition:

```
<!ELEMENT local-name  (#PCDATA)>

<!ATTLIST local-name
        xml:lang CDATA #REQUIRED
        pref ("true"|"false") "false" >
```

7.20. TZ:alias XML element

Name: alias

Namespace: urn:ietf:params:xml:ns:timezone-service

Purpose: An alternative identifier for the timezone.

Description: Defines alternative identifiers that can be used for
   the timezone. This feature allows mapping of old identifiers onto
   new.

   Definition:

   <!ELEMENT alias (#PCDATA)>

7.21. TZ:timezones XML element

   Name: timezones

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Root element encapsulating expanded timezone information.

   Description: This defines the root (top-level) element used as the
      container for expanded timezone data.

   Definition:

   <!ELEMENT timezones (dtstamp, tzdata*)>

7.22. TZ:tzdata XML element

   Name: tzdata

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Encapsulates expanded timezone information.

   Description: This element specifies expanded timezone data for the
      range specified in a request.

   Definition:

   <!ELEMENT tzdata (tzid, calscale?, observance*)>

7.23. TZ:calscale XML element

   Name: calscale

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: Identifies the calendar system for the timezone data.

   Description: The calendar system defines how rules, dates and times
      are to be interpreted. Currently only the value "Gregorian" is
      supported, referring to the Gregorian calendar scale. If this
      element is not present in its normal context, then the value
      "Gregorian" MUST be assumed.

Definition:

```
<!ELEMENT calscale (#PCDATA)>
<!-- allowed PCDATA value is "Gregorian" -->
```

7.24. TZ:observance XML element

Name: observance

Namespace: urn:ietf:params:xml:ns:timezone-service

Purpose: Provides data for a single observance period for a
   timezone.

Description: In an expanded timezone, the observance element
   specifies a single timezone observance.

Definition:

```
<!ELEMENT observance (name, local-name*,
                      onset, utc-offset-from, utc-offset-to)>
```

7.25. TZ:name XML element

Name: name

Namespace: urn:ietf:params:xml:ns:timezone-service

Purpose: Name for the observance.

Description: This element specifies which timezone observance is
   being referred to. Typically this will be either "Standard" or
   "Daylight".

Definition:

```
<!ELEMENT name  (#PCDATA)>
<!-- Typically one of either "Standard" or "Daylight" -->
```

7.26. TZ:onset XML element

Name: onset

Namespace: urn:ietf:params:xml:ns:timezone-service

Purpose: Local time onset of the observance.

Description: The onset element defines the local time at which the
   observance takes effect. The form is the same as an XML date-time
   with timezone information disallowed.

Definition:

```
<!ELEMENT onset  (#PCDATA)>
<!-- A local DATE-TIME value -->
```

7.27. TZ:utc-offset-from XML element

   Name: utc-offset-from

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: UTC offset before the start of the observance.

   Description: The utc-offset-from element defines the UTC offset in
      hours and minutes before the start of this observance. The value
      is a string of the form: ('+' | '-') hh ':' mm (':' ss)

   Definition:

```
<!ELEMENT utc-offset-from  (#PCDATA)>
<!-- Positive or negative hours, minutes and optional seconds -->
```

7.28. TZ:utc-offset-to XML element

   Name: utc-offset-to

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: UTC offset after the start of the observance.

   Description: The utc-offset-to element defines the UTC offset in
      hours and minutes at and after the start of this observance. The
      value is a string of the form: ('+' | '-') hh ':' mm (':' ss)

   Definition:

```
<!ELEMENT utc-offset-to  (#PCDATA)>
<!-- Positive or negative hours, minutes and optional seconds -->
```

7.29. TZ:error element

   Name: error

   Namespace: urn:ietf:params:xml:ns:timezone-service

   Purpose: To convey a descriptive error message to the client.

   Description: Section 4.2.1

   Definition:

   <!ELEMENT error ANY>

8. XML Schema

   Following is the XML schema for the timezone service responses.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:tns="urn:ietf:params:xml:ns:timezone-service"
        xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0"
        targetNamespace="urn:ietf:params:xml:ns:timezone-service"
        elementFormDefault="qualified">
  <!-- =============================================================
       Timezones service XML schema
       ============================================================= -->

  <import schemaLocation="../xml.xsd"
          namespace="http://www.w3.org/XML/1998/namespace" />

  <import schemaLocation="../icalendar/iCalendar.xsd"
          namespace="urn:ietf:params:xml:ns:icalendar-2.0" />

  <!-- ************************************************************
                       basic types
       ************************************************************ -->

  <simpleType name="CalscaleType">
    <annotation>
      <documentation xml:lang="en">
        The calendar system defines how rules, dates and times are to be
        interpreted. Currently only the value "Gregorian" is supported,
        referring to the Gregorian calendar scale. If this element is
        not present in its normal context, then the value "Gregorian"
        MUST be assumed.
      </documentation>
```

```
        </annotation>
        <restriction base="string"/>
      </simpleType>

      <simpleType name="DtstampType">
        <annotation>
          <documentation xml:lang="en">
            Provides the UTC date and time when a substantive change was
            made to the timezone data.
          </documentation>
        </annotation>
        <restriction base="tns:UTCDateTimeType"/>
      </simpleType>

      <simpleType name="LastmodifiedType">
        <annotation>
          <documentation xml:lang="en">
            Provides the UTC date and time that the timezone data was last
            modified.
          </documentation>
        </annotation>
        <restriction base="tns:UTCDateTimeType"/>
      </simpleType>

      <complexType name="LocalNameType">
        <annotation>
          <documentation xml:lang="en">
            Defines one or more localized names that are used when a
            timezone identifier needs to be presented to a user.
            The xml:lang attribute is used to indicate the language
            associated with each value.
            If multiple names are provided for the same locale the preferred
            name can be flagged with the pref attribute.
          </documentation>
        </annotation>
        <simpleContent>
          <extension base="string">
            <attribute ref="xml:lang" />
            <attribute name="pref" type="boolean" default="false" />
          </extension>
        </simpleContent>
      </complexType>

      <simpleType name="OnsetType">
        <annotation>
          <documentation xml:lang="en">
            The onset element defines the local time at which the
            observance takes effect.
```

```
        </documentation>
      </annotation>
      <restriction base="dateTime">
        <pattern value="\d{4}\-\d{2}\-\d{2}T\d{2}:\d{2}:\d{2}(\.\d*)?"/>
      </restriction>
    </simpleType>

    <simpleType name="TzidType">
      <annotation>
        <documentation xml:lang="en">
          The text value is the identifier of the timezone being
          referred to.
        </documentation>
      </annotation>
      <restriction base="string"/>
    </simpleType>

    <simpleType name="UtcOffsetType">
      <annotation>
        <documentation xml:lang="en">
          This type specifies the UTC offset in hours, minutes and
          optional seconds.
        </documentation>
      </annotation>
      <restriction base="string">
        <pattern value="(\-|\+)?\d{2}:\d{2}(:\d{2})?"/>
      </restriction>
    </simpleType>

    <complexType name="AliasType">
      <annotation>
        <documentation xml:lang="en">
          Defines alternative identifiers that can be used for the
          timezone. This feature allows mapping of old identifiers
          onto new.
        </documentation>
      </annotation>
      <simpleContent>
        <extension base="string"/>
      </simpleContent>
    </complexType>

    <complexType name="InactiveType">
      <annotation>
        <documentation xml:lang="en">
          The inactive empty element flags timezones that are
          no longer active.
        </documentation>
```

```
      </annotation>
    </complexType>

    <complexType name="BaseResultType" abstract="true" />

    <simpleType name="UTCDateTimeType">
      <restriction base="dateTime">
        <pattern
          value="(\-|\+)?\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(\.\d*)Z"/>
      </restriction>
    </simpleType>

    <!-- *********************************************************
                             error response
        ********************************************************* -->

    <complexType name="ErrorResponseType" mixed="true" >
      <annotation>
        <documentation xml:lang="en">
          Useful messages SHOULD be returned as an error element.
        </documentation>
      </annotation>
      <sequence>
        <any minOccurs="0"/>
      </sequence>
    </complexType>

    <element name="error" type="tns:ErrorResponseType" />

    <!-- *********************************************************
                           capabilities response
        ********************************************************* -->

    <complexType name="CapabilitiesInfoType">
      <annotation>
        <documentation xml:lang="en">
          The element used as the container for information about the
          servers source of data and contacts.
        </documentation>
      </annotation>
      <sequence>
        <choice>
        <element name="source" type="string" />
        <element name="primary-source" type="string" />
        </choice>
        <element name="contact"  type="string"
                 minOccurs="0" maxOccurs="unbounded" />
      </sequence>
```

```
   </complexType>

   <complexType name="CapabilitiesAcceptParameterType">
     <annotation>
       <documentation xml:lang="en">
         This defines the name, type and characteristics of an operation
         parameter.
       </documentation>
     </annotation>
     <sequence>
       <element name="name" type="string" />
       <element name="required" type="boolean" />
       <element name="multi" type="boolean" />
       <element name="value" type="string" />
       <element name="description" type="string" />
     </sequence>
   </complexType>

   <complexType name="CapabilitiesOperationType">
     <annotation>
       <documentation xml:lang="en">
         The element used as the container for information defining an
         operation and its parameters.
       </documentation>
     </annotation>
     <sequence>
       <element name="action" type="string" />
       <element name="description" type="string" />
       <element name="accept-parameter"
               type="tns:CapabilitiesAcceptParameterType"
               maxOccurs="unbounded" />
     </sequence>
   </complexType>

   <complexType name="CapabilitiesType">
     <annotation>
       <documentation xml:lang="en">
           The root (top-level) element used as the container for
           capabilities information.
       </documentation>
     </annotation>
     <complexContent mixed="false">
       <extension base="tns:BaseResultType">
         <sequence>
           <element name="info"
                    type="tns:CapabilitiesInfoType" />

           <element name="operation"
```

```
                          type="tns:CapabilitiesOperationType"
                          maxOccurs="unbounded" />
          </sequence>
        </extension>
      </complexContent>
    </complexType>

    <element name="capabilities" type="tns:CapabilitiesType" />

    <!-- ***********************************************************
                             list response
         *********************************************************** -->

    <complexType name="SummaryType">
      <annotation>
        <documentation xml:lang="en">
          This defines the element that provides summary information for a
          timezone in the timezones list.
        </documentation>
      </annotation>
      <sequence >
        <element name="tzid" type="tns:TzidType" minOccurs="1" />
        <element name="last-modified" type="tns:LastmodifiedType" />
        <element name="inactive" type="tns:InactiveType" minOccurs="0" />
        <element name="local-name" type="tns:LocalNameType"
                 minOccurs="0" maxOccurs="unbounded" />
        <element name="alias" type="tns:AliasType"
                 minOccurs="0" maxOccurs="unbounded" />
      </sequence>
    </complexType>

    <complexType name="TimezoneListType">
      <annotation>
        <documentation xml:lang="en">
          This defines the root (top-level) element used as the container
          for a timezone listing.
        </documentation>
      </annotation>
      <complexContent mixed="false">
        <extension base="tns:BaseResultType">
          <sequence >
            <element name="dtstamp"
                     type="tns:DtstampType" minOccurs="1" />
            <element name="summary"
                     type="tns:SummaryType"
                     maxOccurs="unbounded"  />
          </sequence>
        </extension>
```

```
      </complexContent>
    </complexType>

    <element name="timezone-list" type="tns:TimezoneListType" />

    <!-- ************************************************************
                          expand response
        ************************************************************ -->

    <complexType name="ObservanceType">
      <annotation>
        <documentation xml:lang="en">
          In an expanded timezone, the observance element specifies a
          single timezone observance.

          The utc-offset-from element defines the UTC offset in hours
          and minutes before the start of this observance.

          The utc-offset-to element defines the UTC offset in hours and
          minutes at and after the start of this observance.
        </documentation>
      </annotation>
      <sequence>
        <element name="name" type="string" />
        <element name="local-name" type="tns:LocalNameType"
                 maxOccurs="unbounded" />
        <element name="onset" type="tns:OnsetType" minOccurs="1" />
        <element name="utc-offset-from"
                 type="tns:UtcOffsetType" minOccurs="1" />
        <element name="utc-offset-to" type="tns:UtcOffsetType"
                 minOccurs="1" />
      </sequence>
    </complexType>

    <complexType name="TzdataType">
      <annotation>
        <documentation xml:lang="en">
          This element specifies expanded timezone data for the range
          specified in a request.
        </documentation>
      </annotation>
      <sequence>
        <element name="tzid" type="tns:TzidType" minOccurs="1" />
        <element name="calscale" type="tns:CalscaleType" />
        <element name="observance" type="tns:ObservanceType"
                 maxOccurs="unbounded" />
      </sequence>
    </complexType>
```

```
  <complexType name="TimezonesType">
    <annotation>
      <documentation xml:lang="en">
        This defines the root (top-level) element used as the container
        for expanded timezone data.
      </documentation>
    </annotation>
    <complexContent mixed="false">
      <extension base="tns:BaseResultType">
        <sequence >
          <element name="dtstamp"
                   type="tns:DtstampType" minOccurs="1" />
          <element name="tzdata" type="tns:TzdataType"
                   maxOccurs="unbounded"   />
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <element name="timezones" type="tns:TimezonesType" />
</schema>
```

9. Security Considerations

   Timezone data is critical in determining local or UTC time for
   devices and in calendaring and scheduling operations. As such, it is
   vital that a reliable source of timezone data is used. Servers
   providing a timezone service MUST support HTTP over Transport Layer
   Security (TLS) (as defined by [RFC2818]) with a valid certificate.
   Clients and servers making use of a timezone service SHOULD use HTTP
   over TLS and verify the authenticity of the service being used before
   accepting and using any timezone data from that source.

   Clients that support transport layer security as defined by [RFC2818]
   SHOULD try the "_timezones" service first before trying the
   "_timezone" service. Clients MUST follow the certificate
   verification process specified in
   [I-D.saintandre-tls-server-id-check].

   A malicious attacker with access to the DNS server data, or able to
   get spoofed answers cached in a recursive resolver, can potentially
   cause clients to connect to any server chosen by the attacker. In
   the absence of a secure DNS option, clients SHOULD check that the
   target FQDN returned in the SRV record matches the original service
   domain that was queried. If the target FQDN is not in the queried
   domain, clients SHOULD verify with the user that the SRV target FQDN
   is suitable for use before executing any connections to the host.

10. IANA Considerations

   This document uses one new URN to identify a new XML namespace for
   the response data used in this specification, and defines a new
   registry of "actions" for the timezone service protocol, and defines
   a "well-known" URI using the registration procedure and template from
   Section 5.1 of [RFC5785], and creates two new SRV service label
   aliases.

10.1. XML namespace

   Registration request for the timezone service namespace:

   URI: urn:ietf:params:xml:ns:timezone-service

   Registrant Contact: See the "Authors' Addresses" section of this
   document.

   XML: None. Namespace URIs do not represent an XML specification.

10.2. Service Operations Registration

   This section defines the process to register new or modified timezone
   service operations with IANA.

10.2.1. Service Operations Registration Procedure

   The IETF will create a mailing list, timezone-service@ietf.org, which
   can be used for public discussion of timezone service operations
   proposals prior to registration. Use of the mailing list is strongly
   encouraged. The IESG will appoint a designated expert who will
   monitor the timezone-service@ietf.org mailing list and review
   registrations.

   Registration of new timezone service operations MUST be reviewed by
   the designated expert and published in an RFC. A Standard Tracks RFC
   is REQUIRED for the registration of new timezone service operations.
   A Standard Tracks RFC is also REQUIRED for changes to operations
   previously documented in a Standard Tracks RFC.

   The registration procedure begins when a completed registration
   template, defined in the sections below, is sent to
   timezone-service@ietf.org and iana@iana.org. The designated expert
   is expected to tell IANA and the submitter of the registration within
   two weeks whether the registration is approved, approved with minor
   changes, or rejected with cause. When a registration is rejected
   with cause, it can be re-submitted if the concerns listed in the
   cause are addressed. Decisions made by the designated expert can be

appealed to the IESG Applications Area Director, then to the IESG. They follow the normal appeals procedure for IESG decisions.

10.2.2. Registration Template for Operations

An operation is defined by completing the following template.

Name: The name of the operation. This is also the value of the "action" parameter used in timezone service requests.

Description: A general description of the operation, its purpose, etc.

Parameters: A list of allowed request parameters, indicating whether they are "REQUIRED" or "OPTIONAL" and whether they can occur only once or multiple times.

Response The nature of the response to the HTTP request, e.g., what format the response data is in.

10.2.3. Registration Template for Operation Parameters

An operation parameter is defined by completing the following template.

Name: The name of the parameter.

Description: A general description of the parameter, its purpose, etc.

Value: The format of the parameter value, or an indication that the parameter has no value.

10.3. Initial Timezone Service Registries

The IANA is requested to create and maintain the following registries for timezone service operations with pointers to appropriate reference documents.

10.3.1. Operations Registry

The following table is to be used to initialize the operations registry.

```
+----------------+---------+----------------------+
| Operation Name | Status  | Reference            |
+----------------+---------+----------------------+
| capabilities   | Current | RFCXXXX, Section 6.1 |
| list           | Current | RFCXXXX, Section 6.2 |
| get            | Current | RFCXXXX, Section 6.3 |
| expand         | Current | RFCXXXX, Section 6.4 |
+----------------+---------+----------------------+
```

10.3.2. Operation Parameters Registry

   The following table is to be used to initialize the parameters
   registry.

```
+--------------+---------+----------------------+
| Parameter    | Status  | Reference            |
+--------------+---------+----------------------+
| action       | Current | RFCXXXX, Section 5.1 |
| changedsince | Current | RFCXXXX, Section 5.3 |
| end          | Current | RFCXXXX, Section 5.5 |
| format       | Current | RFCXXXX, Section 5.2 |
| lang         | Current | RFCXXXX, Section 5.6 |
| returnall    | Current | RFCXXXX, Section 5.7 |
| start        | Current | RFCXXXX, Section 5.4 |
| tzid         | Current | RFCXXXX, Section 5.8 |
+--------------+---------+----------------------+
```

10.4. timezone Well-Known URI Registration

   URI suffix: timezone

   Change controller: IETF.

   Specification document(s): This RFC.

   Related information:

10.5. SRV Service Label Registration

   IANA is requested to add "timezone" and "timezones" service labels as
   aliases for "http" and "https" respectively.

11. Acknowledgements

   The authors would like to thank the members of the Calendaring and
   Scheduling Consortium's Timezone Technical Committee and the
   following individuals for contributing their ideas and support: Bryan

Keller, Ciny Joy, Arnaud Quillaud, Jose Edvaldo Saraiva.

The authors would also like to thank the Calendaring and Scheduling
Consortium for advice with this specification.

12. Normative References

[I-D.cheshire-dnsext-dns-sd]
          Cheshire, S. and M. Krochmal, "DNS-Based Service
          Discovery", draft-cheshire-dnsext-dns-sd-10 (work in
          progress), February 2011.

[I-D.lear-iana-timezone-database]
          Lear, E. and P. Eggert, "IANA Procedures for Maintaining
          the Timezone Database",
          draft-lear-iana-timezone-database-02 (work in progress),
          January 2011.

[I-D.saintandre-tls-server-id-check]
          Saint-Andre, P. and J. Hodges, "Representation and
          Verification of Domain-Based Application Service Identity
          within Internet Public Key Infrastructure Using X.509
          (PKIX) Certificates in the Context of Transport Layer
          Security (TLS)", draft-saintandre-tls-server-id-check-14
          (work in progress), January 2011.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail
          Extensions (MIME) Part Two: Media Types", RFC 2046,
          November 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
          Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,
          Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext
          Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for
          specifying the location of services (DNS SRV)", RFC 2782,
          February 2000.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC4646] Phillips, A. and M. Davis, "Tags for Identifying
          Languages", RFC 4646, September 2006.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security

                 (TLS) Protocol Version 1.2", RFC 5246, August 2008.

   [RFC5545] Desruisseaux, B., "Internet Calendaring and Scheduling
              Core Object Specification (iCalendar)", RFC 5545,
              September 2009.

   [RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known
              Uniform Resource Identifiers (URIs)", RFC 5785,
              April 2010.

   [RFC6321] Daboo, C., Douglass, M., and S. Lees, "xCal: The XML
              Format for iCalendar", RFC 6321, August 2011.

   [W3C.REC-xml-20081126]
              Yergeau, F., Bray, T., Sperberg-McQueen, C., Paoli, J.,
              and E. Maler, "Extensible Markup Language (XML) 1.0 (Fifth
              Edition)", World Wide Web Consortium Recommendation REC-
              xml-20081126, November 2008,
              <http://www.w3.org/TR/2008/REC-xml-20081126>.

Appendix A. Changes for version 03

   o Replaced erroneous reference to ISO3036 with reference to RFC4646

   o Update reference to iCalendar in XML (RFC6321)

   o More description of ids/aliases/names

   o Add subsitute-alias parameter for action=get

   o Allow tzid on list

   o Added name request parameter

   o Added find action

Appendix B. Changes for version 02

   o Missed definitions of the inactive element

   o Restrict UtcOffsetFromType, UtcOffsetToType to a pattern - allow
      seconds.

   o Use restricted XML dateTime as base for onset

o   Use restricted XML dateTime for lastmodified and dtstamp

o   Note that 0 and 1 are valid values for an XML boolean.

o   Set pref to a default value of false

o   Server will now set tzid of aliased timezones to the alias name

o   Remove returnaliases option

o   Aliases should not have lang attribute - removed

o   Add text on status codes and an error element

o   Added capabilities info element containing source | primary-source
    and contacts.

Authors' Addresses

    Michael Douglass
    Rensselaer Polytechnic Institute
    110 8th Street
    Troy, NY 12180
    USA

    Email: douglm@rpi.edu
    URI: http://www.rpi.edu/

    Cyrus Daboo
    Apple Inc.
    1 Infinite Loop
    Cupertino, CA 95014
    USA

    Email: cyrus@daboo.name
    URI: http://www.apple.com/