

ALTO WG  
Internet-Draft  
Intended status: Standards Track  
Expires: July 27, 2015

W. Roome  
Alcatel-Lucent  
Y. Yang  
Yale  
January 23, 2015

PID Property Extension for ALTO Protocol  
draft-roome-alto-pid-properties-03

Abstract

This document extends the Application-Layer Traffic Optimization (ALTO) Protocol [RFC7285] by defining PID-based properties in much the same way that the original ALTO Protocol defined endpoint-based properties.

Requirements Language

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 [RFC2119].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 27, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- 1. Introduction . . . . . 3
- 2. The Consistency and Inheritance Design Views . . . . . 4
- 3. PID Property Names . . . . . 4
- 4. A Hierarchical View of a Network Map . . . . . 5
  - 4.1. Nested And Overlapping PIDs . . . . . 5
  - 4.2. Definition Of PID Property Inheritance . . . . . 6
- 5. Protocol Specification: New Service Information Resources . . 7
  - 5.1. Full PID Property Map . . . . . 7
    - 5.1.1. Media Type . . . . . 7
    - 5.1.2. HTTP Method . . . . . 7
    - 5.1.3. Accept Input Parameters . . . . . 7
    - 5.1.4. Capabilities . . . . . 7
    - 5.1.5. Uses . . . . . 8
    - 5.1.6. Response . . . . . 8
    - 5.1.7. Example . . . . . 9
  - 5.2. Filtered PID Property Map . . . . . 10
    - 5.2.1. Media Type . . . . . 10
    - 5.2.2. HTTP Method . . . . . 10
    - 5.2.3. Accept Input Parameters . . . . . 10
    - 5.2.4. Capabilities . . . . . 10
    - 5.2.5. Uses . . . . . 11
    - 5.2.6. Response . . . . . 11
    - 5.2.7. Example . . . . . 11
- 6. IRD Example . . . . . 12
- 7. Extensions To Endpoint Property Service . . . . . 13
- 8. Security Considerations . . . . . 14
- 9. IANA Considerations . . . . . 14
  - 9.1. application/alto-\* Media Types . . . . . 14
  - 9.2. ALTO Endpoint Property Type Registry . . . . . 15
- 10. References . . . . . 16
- Authors' Addresses . . . . . 16

## 1. Introduction

A key abstraction introduced by the ALTO Protocol [RFC7285] is a PID (Provider-defined Identifier): a named set of endpoint addresses. For IPv4/IPv6 networks, this set is defined by one or more endpoint address prefixes, or CIDRs [RFC4632].

An ALTO Server defines one or more Network Maps. Each Network Map is defined as a set of uniquely named PIDs, with the constraint that any given CIDR may appear in only one PID. Thus a Network Map logically partitions the space of endpoint addresses into groups defined by the PIDs.

An ALTO Server may define several Network Maps, partitioning the endpoints in different ways. For example, one Network Map might partition endpoints according to geographical locations, with each PID representing the set of endpoints at a given location. Another Network Map might partition the endpoints according to their network capabilities (e.g., CDN delivery protocols such as HTTP or HTTPS). In this case, each PID defined in the Network Map represents endpoints with similar capabilities.

Another abstraction introduced by the ALTO protocol is an Endpoint Property, which is a named value associated with an endpoint. Properties can include data such as the endpoint's ISP or ASN (Autonomous System Number), or the endpoint's latitude and longitude, or the country or continent codes for the endpoint's location. The ALTO protocol defines the mechanism for a client to query a server for these property values, but does not define the Endpoint Properties themselves; that is left to extensions.

Just as Endpoint Properties are useful, it is reasonable to expect that it will be useful for sets of endpoints -- that is, PIDs -- to have properties as well. A PID Property might be an Endpoint Property shared by all endpoints in that PID (e.g., their ISP or ASN). Alternatively, a PID Property could be the aggregation of individual properties of the endpoints in the PID (e.g., the latitude and longitude of their geographic bounding box). Or a PID Property might simply be inherent in the way in which the ALTO Server defined the PID.

However, the base ALTO protocol does not define the concept of PID Properties, or present a mechanism for a client to obtain PID Properties from an ALTO Server. This extension remedies that omission.

## 2. The Consistency and Inheritance Design Views

A key goal for defining PID Properties is that they should be consistent with, and generalize, Endpoint Properties. Specifically, in the base ALTO Protocol, the ALTO Server may associate a set of (name,value) pairs with any endpoint. These sets are the Endpoint Properties. The ALTO Protocol specifies the mechanism for a client to retrieve those Endpoint Properties from a server.

Consider the Endpoint Property PR and all endpoints defined in the PID P. If those endpoints all have the same value for PR, then the PID property PR for P should also have that same value.

For the more general case, assume that P consists of the set of endpoint addresses {E<sub>Pi</sub>}, and E<sub>Pi</sub>.PR is the value of property PR for endpoint E<sub>Pi</sub>. Then we could obtain the value of PR for PID P by evaluating an aggregation function over the endpoint values {E<sub>Pi</sub>.PR}. Typical aggregation functions include average/mean, mode, geo-center, union, bounding box, etc., although of course the actual aggregation function would depend on the semantics of the property.

Complementing the bottom-up aggregation view, we also adopt a top-down inheritance view. That is, if a property PR is not explicitly defined for an endpoint, but the endpoint's PID does define a value for PR, then the endpoint inherits the value of PID property. The concept of inheritance is a simple, but powerful concept to reduce information redundancy.

## 3. PID Property Names

PID Property names MUST follow the same rules as Global Endpoint Property names (Section 10.8.2 of [RFC7285]), and non-private names MUST be registered with IANA.

Furthermore, PID Properties and Endpoint Properties share the same name space, and their definitions MUST be consistent. That is, if PR is used as both a PID Property and an Endpoint Property, then the semantics of PR as a PID Property must be clearly related to semantics of PR as an Endpoint Property. For example, when used as a PID Property, PR could be defined as the average (or some other representative value) of the values of PR for a set of endpoints.

This document will use the type PIDPropertyType to indicate a string denoting a PID Property Name. This type is identical to EndpointPropertyType as defined in Section 10.8 of [RFC7285].

## 4. A Hierarchical View of a Network Map

### 4.1. Nested And Overlapping PIDs

Each PID in a Network Map covers a set of endpoint addresses, and these sets can form a nested hierarchy. As an example, consider the following Network Map:

```
PID p0: [0.0.0.0/0, ::0/0]
PID p1: [1.0.0.0/8]
PID p2a: [1.0.0.0/16]
PID p2b: [1.1.0.0/16]
```

Every endpoint in p2a and p2b is also in the set covered by p1, and every possible IPV4 endpoint is also covered by PID p0. We would like to take advantage of this hierarchy to allow PIDs to inherit properties, just as a class in an object oriented programming language inherits attributes from its parent class. In the example, it is reasonable to expect that PIDs p2a and p2b would inherit any property of p1 that they do not override, and every PID would inherit the properties of PID p0.

Note that when calculating costs, the ALTO Protocol requires every endpoint to be in one, and only one, PID. The ALTO Protocol resolves this by using the "longest match" rule. That is, the ALTO Protocol says when an endpoint address is covered by more than one prefix, the address is in the PID with the prefix with the longest mask. Because any given prefix can be on only one PID, this ensures each endpoint is in only one PID. Thus in our example, 1.0.0.1 is in p2a, 1.1.0.1 is in p2b, and 1.2.0.1 in in p1.

However, while PIDs can nest within other PIDs, they do not necessarily form a simple hierarchy. Suppose we add PID p3 to the previous Network Map:

```
PID p0: [0.0.0.0/0, ::0/0]
PID p1: [1.0.0.0/8]
PID p2a: [1.0.0.0/16]
PID p2b: [1.1.0.0/16]
PID p3: [1.0.255.0/24, 1.1.0.0/24]
```

Some endpoints in p3 are in p2a, and others are in p2b. Thus p3 is partially contained in PIDs p2a and p2b, without being completely contained in either. So we would not expect p3 to inherit properties

from p2a or p2b. But p3 is completely contained in p1 and p0, so it is reasonable to expect p3 to inherit properties from p1 and p0 that are not overridden in p2a or p2b. In short, "PID inheritance" raises issues similar to those of "multiple inheritance" in object oriented programming languages.

#### 4.2. Definition Of PID Property Inheritance

While PID inheritance raises some complications, it is too useful to ignore. Fortunately CIDRs do form a simple "single parent" hierarchy, so we will use CIDR inheritance as the basis for defining PID inheritance. We believe this preserves the useful cases, and avoids the pathological ones.

Note that the ALTO Protocol did not define "PID inheritance" because it was not relevant to the base protocol.

Definition: A parent of CIDR C is any CIDR, in the set of CIDRs for all PIDs in the Network Map, which covers all endpoints covered by C, and whose mask is shorter than the mask of C.

Definition: The immediate parent of CIDR C is the parent of C with the longest prefix. The immediate parent CIDR might not exist, but if it does, it is unique.

Definition: A CIDR C inherits the value V for property PR if the PID containing its immediate parent CIDR defines the value V for property PR, or if its immediate parent CIDR inherits the value V for property PR.

Definition: A PID P has the value V for property PR if PR is explicitly defined with value V in P, or if every CIDR C in P inherits the same value V for PR.

Suppose the following properties are defined for PIDs described above:

```
PID p1:  ISP="Verizon"  country="us"  
PID p2a: ASN="12345"  state="NJ"  
PID p2b: ASN="12345"  state="CT"
```

Then p2a, p2b, and p3 all inherit the "ISP" and "country" properties from p1, because those properties are not defined for p2a or p2b. p3 would inherit "12345" for the "ASN" property, because although p2a and p2b both define that property, they give it the same value. However, p3 would not inherit the "state" property, because it has

different values in p2a and p2b.

## 5. Protocol Specification: New Service Information Resources

This document defines two new ALTO resources. The PID Property Map returns the values of a set of properties for all PIDs in a Network Map. The Filtered PID Property Map returns PID Properties for a subset of PIDs and properties selected by the client. These are analogous to the Full and Filtered Network Maps in the base ALTO Protocol.

An ALTO Server may provide any number of resource of these types, provided that they have unique capabilities. For example, an ALTO Server may offer several Full PID Property Maps, for the same Network Map, as long as each one returns a different set of properties.

Both services are optional. In particular, an ALTO Server may provide a Full PID Property Map without providing a corresponding Filtered PID Property Map, and vice versa.

### 5.1. Full PID Property Map

A Full PID Property Map returns the properties defined for all PIDs in a Network Map.

#### 5.1.1. Media Type

The media type of an ALTO PID Property Map resource is "application/alto-pidprop+json".

#### 5.1.2. HTTP Method

An ALTO PID Property Map resource is requested using the HTTP GET method.

#### 5.1.3. Accept Input Parameters

None.

#### 5.1.4. Capabilities

The capabilities are defined by an object of type PIDPropertyCapabilities:

```
object {
  PIDPropertyType prop-types<1..*>;
} PIDPropertyCapabilities;
```

where "prop-types" is an array with the names of the PID Properties returned by this resource.

#### 5.1.5. Uses

An array with the resource-id of the Network Map which defines the PIDs whose properties are returned.

#### 5.1.6. Response

The "dependent-vtags" field in the "meta" field of the response MUST be an array that includes the version tag of the ALTO Network Map defining these PIDs.

The data component of a PID Properties response is named "pid-properties", which is a JSON object of type PIDPropertyMapData, where:

```
object {
  PIDPropertyMapData pid-properties;
} InfoResourcePIDProperties : ResponseEntityBase;

object-map {
  PIDName -> PIDProps;
} PIDPropertyMapData;

object {
  PIDPropertyType -> JSONValue;
} PIDProps
```

The PIDName and ResponseEntityBase types are defined in Sections 10.1 and 8.4 of the ALTO Protocol [RFC7285].

Specifically, a PIDPropertyMapData object has one member for each PID in the Network Map. The PID's properties are encoded in the corresponding PIDProps object. PIDProps encodes one name/value pair for each property, where the property names are encoded as strings of type PIDPropertyType. A protocol implementation SHOULD assume that the property value is a JSONString and fail to parse if it is not, unless the implementation is using an extension to this document that indicates when and how property values of other data types are signaled.

For each PID in the Network Map, the ALTO Server returns the value defined for each of the PID Properties specified in this resource's "capabilities" list. For efficiency, the ALTO Server SHOULD omit property values that are inherited rather than explicitly defined. The client SHOULD use the algorithm defined in Section 4.2 to

determine inherited property values.

An ALTO Server MAY explicitly define a property as not having a value for a particular PID. That is, a server may say that a property is "defined to have no value", as opposed to the property being "undefined". In this case, if the PID would not inherit a value for that property, then the ALTO Server MUST either omit the property, or else return a "null" value for it. However, if that PID would inherit a value for that property, then the ALTO server MUST return a "null" value. An ALTO Client MUST recognize a "null" value as meaning "do not apply the inheritance rules for this property in the PID."

If the ALTO Server does not define any properties for a PID, then the server MAY omit that PID from the response.

#### 5.1.7. Example

The following example uses the Network Map and PID Properties defined above. Note that the response does not include entries for PIDs p0 or p3. The former was omitted because no properties were defined for it, and the latter because its only properties are inherited.

```
GET /pidprop/full HTTP/1.1
Host: alto.example.com
Accept: application/alto-pidprop+json,application/alto-error+json
```

```
HTTP/1.1 200 OK
Content-Length: ###
Content-Type: application/alto-pidprop+json
{
  "meta" : {
    "dependent-vtags" : [
      { "resource-id": "my-default-network-map",
        "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf62"
      }
    ]
  },
  "pid-properties": {
    "p1" : { "ISP": "Verizon", "country": "us" },
    "p2a": { "ASN": "12345", "state": "NJ" },
    "p2b": { "ASN": "12345", "state": "CT" }
  }
}
```

## 5.2. Filtered PID Property Map

A Filtered PID Property Map returns the values for set of PIDs and properties selected by the client.

### 5.2.1. Media Type

The media type of an ALTO Filtered PID Property Map resource is the same as a Full PID Property Map, and is defined in Section 5.1.1.

### 5.2.2. HTTP Method

An ALTO Filtered PID Property Map resource is requested using the HTTP POST method.

### 5.2.3. Accept Input Parameters

The input parameters for a Filtered PID Property Map request are supplied in the entity body of the POST request. This document specifies the input parameters with a data format indicated by the media type "application/alto-pidpropparams+json", which is a JSON object of type ReqPIDProp:

```
object {
  PIDPropertyType properties<1..*>;
  PIDName          pids<1..*>
} ReqPIDProp;
```

with fields:

**properties:** List of PID properties to be returned for each PID. Each specified property **MUST** be included in the list of supported properties indicated by this resource's "capabilities" field (Section 5.2.4). The ALTO server **MUST** interpret entries appearing multiple times as if they appeared only once.

**pids:** List of PID names for which the specified properties are to be returned. The ALTO server **MUST** interpret entries appearing multiple times as if they appeared only once.

### 5.2.4. Capabilities

The capabilities are defined by an object of type PIDPropertyCapabilities, as defined above.

#### 5.2.5. Uses

An array with the resource-id of the Network Map which defines the PIDs whose properties are returned.

#### 5.2.6. Response

The response is the same as for the Full PID Property Map (Section 5.1.6), except that it only includes the properties and PIDs requested by the client.

Also, Filtered PID Property Map response MUST include all inherited PID property values (unlike the Full PID Property Map, the Filtered PID Property Map response does not include enough information for the client to calculate the inherited values).

#### 5.2.7. Example

The following example uses the Network Map and PID Properties defined above. Note that the response includes the inherited property "ISP" for PID p2a, and "state" and "ISP" for p3.

```
POST /pidprop/lookup HTTP/1.1
Host: alto.example.com
Content-Length: ###
Content-Type: application/alto-pidpropparams+json
Accept: application/alto-pidprop+json,application/alto-error+json
```

```
{
  "properties" : [ "ISP", "ASN", "state" ],
  "pids" : [ "p1", "p2a", "p3" ]
}
```

```
HTTP/1.1 200 OK
Content-Length: ###
Content-Type: application/alto-pidprop+json
```

```
{
  "meta" : {
    "dependent-vtags" : [
      { "resource-id": "my-default-network-map",
        "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf62"
      }
    ]
  },
  "pid-properties" : {
    "p1" : { "ISP": "Verizon" },
    "p2a" : { "ISP": "Verizon", "ASN": "12345", "state": "NJ" },
    "p3" : { "ISP": "Verizon", "ASN": "12345" }
  }
}
```

## 6. IRD Example

Here is an example of an ALTO Information Resource Directory (IRD) which defines several PID Property resources. Note that "full-pid-property-map" returns several PID Properties for all PIDs in the Network Map, while "asn-pid-property-map" returns just the ASN property. The resource "filtered-pid-properties" resource returns values for properties and PIDs selected by the client.

```
...
"resources" : {
  "my-default-network-map" : {
    "uri" : "http://alto.example.com/networkmap",
    "media-type" : "application/alto-networkmap+json"
  },
  "endpoint-property" : {
    "uri" : "http://alto.example.com/endpointprop/lookup",
    "media-type" : "application/alto-endpointprop+json",
    "accepts" : "application/alto-endpointpropparams+json",
    "capabilities" : {
      "prop-types" : [ "my-default-network-map.pid",
                      "priv:ietf-example-prop" ]
    }
  },
  "full-pid-property-map" : {
    "uri" : "http://alto.example.com/pidprop/full",
    "media-type" : "application/alto-pidprop+json",
    "uses" : [ "my-default-network-map" ]
    "capabilities" : {
      "prop-types" : [ "ISP", "country", "ASN", "state" ]
    }
  },
  "asn-pid-property-map" : {
    "uri" : "http://alto.example.com/pidprop/asn",
    "media-type" : "application/alto-pidprop+json",
    "uses" : [ "my-default-network-map" ]
    "capabilities" : {
      "prop-types" : [ "ASN" ]
    }
  },
  "filtered-pid-properties" : {
    "uri" : "http://alto.example.com/pidprop/lookup",
    "media-type" : "application/alto-pidprop+json",
    "accepts" : "application/alto-pidpropparams+json",
    "uses" : [ "my-default-network-map" ]
    "capabilities" : {
      "prop-types" : [ "ISP", "country", "ASN", "state" ]
    }
  }
}
```

## 7. Extensions To Endpoint Property Service

As described in Section 10.8 of [RFC7285], Endpoint Property names may be prefixed with the Resource ID of a Network Map. For such

resource-specific properties, if a value is not explicitly defined for an endpoint, the Endpoint Property Service MUST return the value that a PID Property Map would return for the PID containing that endpoint.

For property names that are not prefixed by a Network Map Resource ID, if a value is not defined for an endpoint, the Endpoint Property Service MAY return the value defined for that property in one of the ALTO Server's PID Property Maps for the PID containing the endpoint.

## 8. Security Considerations

As with Endpoint Properties, PID Properties may have sensitive customer-specific information. If this is the case, an ALTO Server may limit access to those properties by providing several different PID Property services. For non-sensitive properties, the ALTO Server would provide a URI which accepts requests from any client. Sensitive properties, on the other hand, would only be available via a secure URI which would require client authentication.

## 9. IANA Considerations

This document defines additional application/alto-\* media types, and extends the ALTO endpoint property registry.

### 9.1. application/alto-\* Media Types

This document registers two additional ALTO media types, listed in Table 1.

Type	Subtype	Specification
application	alto-pidprop+json	Section 5.1.1
application	alto-pidpropparams+json	Section 5.2.3

Table 1: Additional ALTO Media Types

Type name: application

Subtype name: This documents registers multiple subtypes, as listed in Table 1.

Required parameters: n/a

Optional parameters: n/a

Encoding considerations: Encoding considerations are identical to those specified for the "application/json" media type. See [RFC7159].

Security considerations: Security considerations relating to the generation and consumption of ALTO Protocol messages are discussed in Section 15 of [RFC7285].

Interoperability considerations: This document specifies format of conforming messages and the interpretation thereof.

Published specification: This document is the specification for these media types; see Table 1 for the section documenting each media type.

Applications that use this media type: ALTO servers and ALTO clients either stand alone or are embedded within other applications.

Additional information:

Magic number(s): n/a

File extension(s): This document uses the mime type to refer to protocol messages and thus does not require a file extension.

Macintosh file type code(s): n/a

Person & email address to contact for further information: See Authors' Addresses section.

Intended usage: COMMON

Restrictions on usage: n/a

Author: See Authors' Addresses section.

Change controller: Internet Engineering Task Force (mailto:iesg@ietf.org).

## 9.2. ALTO Endpoint Property Type Registry

The ALTO Endpoint Property Type Registry was created by [RFC7285]. If possible, the name of that registry should be changed to "ALTO Property Type Registry", to indicate that it is not restricted to

Endpoint Properties. If it is not feasible to change the name, the description must be amended to indicate that it registers PID Properties as well as Endpoint Properties.

## 10. References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997.
- [RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", RFC 4632, BCP 122, August 2006.
- [RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.
- [RFC7285] Almi, R., Penno, R., Yang, Y., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, September 2014.

## Authors' Addresses

Wendy Roome  
Alcatel-Lucent/Bell Labs  
600 Mountain Ave, Rm 3B-324  
Murray Hill, NJ 07974  
USA

Phone: +1-908-582-7974  
Email: w.roome@alcatel-lucent.com

Y. Richard Yang  
Yale University  
51 Prospect St.  
New Haven, CT  
USA

Email: yry@cs.yale.edu