



White Paper

The Extensible Provisioning Protocol

XML Trust Services

Contents

<i>Executive Summary</i>	<i>1</i>
<i>I. Introduction</i>	<i>1</i>
<i>II. Design Goals</i>	<i>1</i>
<i>III. Specifications</i>	<i>2</i>
<i>IV. Examples</i>	<i>2</i>
<i>V. References</i>	<i>3</i>
<i>VI. For More Information</i>	<i>4</i>

Executive Summary

XML (Extensible Markup Language), the flexible data framework that allows applications to communicate on the Internet, has become the preferred infrastructure for e-commerce applications. All of those transactions require trust and security, making it mission-critical to devise common XML mechanisms for authenticating merchants, buyers, and suppliers to each other, and for digitally signing and encrypting XML documents like contracts and payment transactions.

XML Trust Services—a four-component suite of open specifications for application developers developed in partnership with industry leaders including Microsoft, Ariba, WebMethods, and Netegrity—makes it easier than ever to integrate a broad range of trust services into B2B and B2C applications. XML complements Public Key Infrastructure (PKI) and digital certificates, the standard method for securing Internet transactions.

To enable Internet registrars that sell online identity services to access central domain name registry data more efficiently, VeriSign has developed the **EPP** (Extensible Provisioning Protocol]) to support an XML-based domain name management utility. EPP enables VeriSign Global Registry Services' accredited registrar partners to sell domain names, telephone numbers, and other identity assets via EPP, which permits greater information sharing and flexibility and new identification technologies gain acceptance.

I. Introduction

The Extensible Provisioning Protocol (EPP) [1] is a connection-oriented, application layer client-server protocol for the provisioning and management of objects stored in a shared central repository. Specified in the schema notation of the Extensible Markup Language (XML), the protocol defines generic object management operations and an extensible framework that maps protocol operations to objects. A complete set of protocol specifications was recently published with the Internet Engineering Task Force (IETF) as Internet-Draft documents.

II. Design Goals

XML provides a rich set of features that allows communicating peers to create data tags that have semantic meaning in the operating environment shared by the peers. While in general this is a very desirable feature, it introduces an element of instability for protocol designers. Once a protocol has been formally specified, adding new tags to extend the protocol means changes to published specifications. Over time this can lead to a lack of interoperable implementations and specification confusion.

EPP takes a different approach. The base protocol itself is very simple, defining a set of object management features that are not explicitly tied to specific objects. The base protocol is intended to be stable and unchanging to ease development of interoperable implementations. EPP operations are mapped to objects using XML namespaces that provide “hooks” to loosely coupled object specifications so that definitions for

management of new objects can be done outside the base protocol. For example, the protocol can be extended to support provisioning of purchase orders by defining a new specification that defines how purchase order objects are managed.

EPP provides features for session management, object query, and object management. Sessions are established between a client and a server, and once a session is established the client and server exchange commands and responses. Security services are available at both the application and transport layers.

III. Specifications

The EPP protocol suite currently contains a base protocol specification and mappings for three different objects: Internet domain names [2], Internet host names [3], and “contact” identifiers associated with humans and organizations [4]. Specifications for other objects may be developed as needs are identified.

EPP is connection oriented, but transport independent. A specification for transport using the Transmission Control Protocol (TCP) [5] exists; specifications for transport using other protocols or applications frameworks may be produced in the future.

IV. Examples

An EPP client often wants to know if a server “knows” a specific object instance. For example, a client that provisions domain names may want to know if a domain name is already known to the server and thus can not be created anew. An EPP command that performs a query to determine if several domain names are “known” looks like this:

```
<?xml version="1.0" standalone="no"?>
<epp xmlns="urn:iana:xmlns:epp"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
  <command>
    <ping>
      <domain:ping xmlns:domain="urn:iana:xmlns:domain"
        xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
        <domain:name>example1.com</domain:name>
        <domain:name>example2.com</domain:name>
        <domain:name>example3.com</domain:name>
      </domain:ping>
    </ping>
    <trans-id>
      <date>2000-06-08</date>
      <client-id>ClientX</client-id>
      <code>ABC-12345-XYZ</code>
    </trans-id>
  </command>
```

```
</epp>
```

A server's response to this command looks like this:

```
<?xml version="1.0" standalone="no"?>
<epp xmlns="urn:iana:xmlns:epp"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
  <response>
    <result code="1000">
      <text>Command completed successfully</text>
    </result>
    <response-data>
      <domain:ping-data xmlns:domain="urn:iana:xmlns:domain"
        xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
        <domain:name result="known">example1.com</domain:name>
        <domain:name
result="unknown">example2.com</domain:name>
        <domain:name result="known">example3.com</domain:name>
      </domain:ping-data>
    </response-data>
    <trans-id>
      <date>2000-06-08</date>
      <client-id>ClientX</client-id>
      <code>ABC-12345-XYZ</code>
    </trans-id>
  </response>
</epp>
```

V. References

1. Hollenbeck, S., "Extensible Provisioning Protocol", Internet-Draft, November 10, 2000.
2. Hollenbeck, S., "Extensible Provisioning Protocol Domain Name Mapping", Internet-Draft, November 10, 2000
3. Hollenbeck, S., "Extensible Provisioning Protocol Host Mapping", Internet-Draft, November 10, 2000
4. Hollenbeck, S., "Extensible Provisioning Protocol Contact Mapping", Internet-Draft, November 10, 2000
5. Hollenbeck, S., "Extensible Provisioning Protocol Transport Over TCP", Internet-Draft, November 10, 2000

VI. For More Information

To access the components of the EPP specification, visit the following URLs:

- **Base Specification:** <http://www.ietf.org/internet-drafts/draft-hollenbeck-epp-00.txt>
- **Domain Name Mapping :** <http://www.ietf.org/internet-drafts/draft-hollenbeck-epp-domain-00.txt>
- **Host Mapping:** <http://www.ietf.org/internet-drafts/draft-hollenbeck-epp-host-00.txt>
- **Contact Mapping:** <http://www.ietf.org/internet-drafts/draft-hollenbeck-epp-contact-00.txt>
- **Transport over TCP:** <http://www.ietf.org/internet-drafts/draft-hollenbeck-epp-tcp-00.txt>

To learn more about VeriSign's XML Trust Services, see

<http://www.verisign.com/developer/xml/index.html>



VERISIGN, INC.
1350 CHARLESTON ROAD
MOUNTAIN VIEW, CALIFORNIA 9404
WWW.VERISIGN.COM

©2000 VeriSign, Inc. All rights reserved. VeriSign, NetSure, and OnSite are registered trademarks and service marks of VeriSign, Inc. All other trademarks belong to their respective owners. 11/00