WEBDAV Working Group
Internet Draft
draft-skreddy-enpreq-00.txt
Expires October 16, 1998

Surendra Reddy
Oracle Corporation
April 16, 1998

Requirements for Event Notification Protocol

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or made obsolete 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".

To view the entire list of current Internet-Drafts, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern Europe), ftp.nis.garr.it (Southern Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this document is unlimited. Please send comments to the Distributed Authoring and Versioning (WEBDAV) working group at <w3c-dist-auth@w3.org>, which may be joined by sending a message with subject "subscribe" to <w3c-dist-auth-request@w3.org>.

Discussions of the WEBDAV working group are archived at <URL:http://www.w3.org/pub/WWW/Archives/Public/w3c-dist-auth>.

Abstract

This document describes the requirements for an Event Notification Protocol. The objective is to provide a simple, scalable and highly efficient notification protocol while also providing the appropriate flexibility to meet the needs of both the internet and enterprise environments. Intent of this document is to collect all notification requirements in one place and leverage the work already done in other working groups.

This document is one of a set of documents which together describe all aspects of a new Event Notification Protocol (ENP). ENP is an application level protocol that can be used for distributed event

notification. The full set of ENP documents include:

(1). Requirements for Event Notification Protocol

(2). Model and Semantics Event Notification Protocol

(3). Protocol Specification for Event Notification Protocol

(4). Rationale for the Structure and Model for the Event Notification Protocol

1. Introduction

In a distributed authoring and versioning environment, user may want to monitor the changes performed on various resources created or owned by the user. Similarly, if a PROPFIND operation takes more time to complete the operation, client can choose to register this event to notify the client when the server finishes the PROPFIND rather than client waiting for the server to complete the task. Similarly, if any search operation in DASL takes more time in executing the search, client can register the event with the server so that sever notifies the client when the search is done. These requirements mandate the need for a mechanism to notify events to subscribed users.

There are several different network event notification protocols like CORBA Event Services, X Window System events, SGAP, BSCW, etc. But these services are defined to work with specific architectures and impose large codebase which makes it practically difficult for lightweight notification services.

This document presents a list of features in the form of requirements for a Event Notification Protocol which, if implemented, would improve the efficiency of common event notification mechanisms for Distributed Authoring and Versioning protocol.

2. Terminology

Supplier Events
Supplier events generates event data.

Consumer Events
Consumer events process event data.

Push Model

In the Push model, Event Notification Protocol push event data to consumers.

Pull Model

In Pull model, consumers pull event data from Event Notification Protocol.

3. Event Notification Protocol

3.1. Overview

Event Notification Protocol decouples the communication between communicating processes or events. The event notification protocol defines two roles for the events: the supplier roles and the consumer role. Suppliers produce event data and consumers process event data. Event data are communicated between suppliers and consumers through Event Notification Protocol(ENP). Event Notification Protocol uses push and pull model to initiates communication. The push model allows a supplier of events to initiate the transfer of the event data to consumers. The pull model allows a consumer of events to request the event data from a supplier. In the push model, the supplier is taking the initiative; in the pull model, the consumer is taking the initiative.

The consumer may use either a blocking or non-blocking mechanism for receiving notifications. The consumer can periodically poll the channel for events.

3.2. Examples

(1). The Event Notification Protocol can be used to generate change triggers. When a resource properties or contents are changed, ENP generates events and propagates to all subscribed parties.

(2). Collection may be composed of internal and external members. Document authors are interested in knowing when the value of certain properties or contents of these members have changed. Event Notification Protocol can be used to notify all such changes to all subscribed parties and document authors.

4. Requirements

4.1. Notification Registration

It SHOULD be possible for end users to "register" for notifica-tions of certain types of events.

4.2. Notification Attributes:
It SHOULD be possible to associate attributes with the notifica-
tion request.

4.3. Queued Notification
Notifications which are not necessarily sent immediately, but are
queued for delivery some intermediate network process or for later retrieval.

4.4. Notification with Reliable Delivery
It SHOULD be possible to deliver event notifications in a reliable
manner, assuring fully ordered end-to-end delivery. Guaranteed delivery requires both queued notification and a reliable transport.

4.5. Notifications with Unreliable Delivery
Notifications are delivered via the fundamental transport address
and routing framework, but no acknowledgement or retry is required. Process to process communications, if involved, are unconstrained.

4.6. Quality of Service
Some notification delivery methods may allow users to select qual-
ity of service parameters. These parameters will depend upon the specific delivery method chosen and may include parameters such as priority, security, number of retries, and the like.

4.7. Consumers must be able specify zero or more notification
recipients when submitting an event. When specifying a notifica-
tion recipient, consumers must be able to specify notification delivery method, associated attributes and any other quality of service parameters for the notification recipient.

4.8. It SHOULD be possible to deliver an event notification
through firewalls. However, it need not test to guarantee delivery
of the notification through a firewall before accepting the event registration request.

4.9. A mechanism must be provided for delivering notification to
the submitting client when the delivery of an event notification
to a specified Notification Recipient fails.

4.10. Events work in a distributed environment. Consumers SHOULD be
able either request events or be notified of events, whichever is
more appropriate for application design and performance.

4.11. A supplier can issue a single request to communicate event data to all consumers at once.

4.12. Supplier can generate events without knowing the identities of consumers. Conversely, consumers can receive events without knowing the identities of the suppliers

4.13. Complex events may be handled by constructing tree of event consumers/suppliers checking for successively more specific event predicates.

4.14. Consumers and suppliers SHOULD be able to register with event channels.

4.15. It SHOULD be possible to support event filtering through which event channels deliver events selectively from suppliers to consumers.

4.16. Some applications may require that consumers of an event provide an explicit confirmation of reception back to the supplier.

4.17. It SHOULD be possible to consume events from one or more suppliers and supplies events to one or more consumers.

4.18. Some applications may require that consumers of an event provide an explicit confirmation of reception back to the supplier. Event Notification Protocol SHOULD be able to support this functionality effectively using event attributes.

5. Extensibility

The Event Notification Protocol shall be extensible to facilitate interoperability and prevents implementation collisions.

6. Security Requirements

6.1. It SHOULD be possible to digitally sign the notifications to ensure the integrity of the notifications or origin of the event notifications.

6.2. It SHOULD be possible that the Event Notification Protocol to operate within a secure environment. Wherever possible ENP SHOULD be able to make use of existing security protocols and services. ENP SHOULD not invent new security protocols or services if the requirements described in this document can be met by existing protocols and services.

6.3. ENP shall by definition support event registration and notification from one enterprise to another through firewalls. ENP must be capable of passing through firewalls and/or proxy servers(where enabled by the firewall administrator) preferably without any modifications to the existing firewall technology.

7. Internationalization

7.1. As consumer and producers of events come from all over the world, Event Notification Protocol SHOULD meet internationaliza- tion and localization requirements.

8. References

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

[2] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., and Berners-Lee, T., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997

[3] Postel, Jonathon B., "SIMPLE MAIL TRANSFER PROTOCOL", RFC 821, August 1982

[4] Postel, J., and Reynolds, J., "FILE TRANSFER PROTOCOL (FTP)", RFC 959, October 1985

[5] Alvestrand, H., "IETF Policy on Character Sets and Languages", RFC 2277, January 1998.

[6] Y. Y. Goland, E. J. Whitehead, Jr., A. Faizi, S. R. Carter, D. Jensen, "Extensions for Distributed Authoring on the World Wide Web - WebDAV.", Draft-ietf-webdav- protocol-07.txt, February, 1998.

9. Author's Address

Oracle Corporation
500 Oracle Parkway
M/S 6op3
Redwoodshores, CA 94065
Phone: +1(650) 506 5441
Fax: +1(650) 654 6205
Email: skreddy@us.oracle.com

Expires October 16, 1998