WEBDAV Working Group   Surendra Reddy(Oracle)
Internet Draft   Mark Leighton Fisher(TCE)
draft-ietf-webdav-enpreq-01.txt   May 1, 1998
Expires November 1, 1998

Requirements for Event Notification Protocol

Status of this Memo

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 environment, there will be operations thant take so long that the user doesn't want to wait till the completion of the event. For example, 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 report generation event takes significant amount of time to complete the event, user can choose to be notified when the event completes rather than constantly polling or waiting for the event to complete.

Similarly, if any search operation 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 codebases which makes them in practice difficult to use them in 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.

2. Terminology

Events Supplier
    Events Supplier generates event data.

Events Consumer
    Events Consumer 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 the Event Notification Protocol(ENP). Event Notification Protocol can be initiated by either the push or pull model in ENP. 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's properties or contents are changed, ENP generates events and propagates these events all subscribed parties.

(2). Collections may be composed of internal and external members. Document authors are interested in knowing when the value of certain properties or the contents of these members have changed. Event Notification Protocol can be used to send notifications of 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 for some intermediate network process or for
later retrieval. Queued notifications SHOULD be supported.

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 tran-
sport.

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(s).
recipients when submitting a request for event notification. When
specifying a notification recipient, consumers MUST be able to
specify notification delivery method, associated attributes and
any other quality of service parameters for the notification reci-
pient.

4.8. It SHOULD be possible to deliver an event notification
through firewalls. However, guaranteed delivery of the notifica-
tion through a firewall need not be tested 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 exist in a distributed environment. Consumers SHOULD be
able either request events(Subscription Model) or be notified of
events without subscribing(Broadcast Model), whichever is more
appropriate for application design and performance.

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

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

4.13. **Complex events may be handled by constructing tree of events 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 sup-** plier.

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 prevent implementation collisions.

6. Security Requirements

6.1. **It SHOULD be possible to digitally sign the notifications to ensure the authenticity and integrity of the notifications.**

6.2. **It SHOULD be possible for 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 SHOULD support 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. Because of these requirements, ENP SHOULD use UTF-8[RFC2044] as its native character set.

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-08.txt, April 7, 1998.

9. Author's Address

Surendra Reddy
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

Mark Leighton Fisher
Thomson Consumer Electronics
Indianapolis, IN
email: fisherm@indy.tce.com

Expires November 1, 1998