1. Status of this Memo

This document is an Internet-Draft and is subject to all provisions of Section 10 of RFC2026.

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 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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This draft document will be submitted to the RFC Editor as an Informational document. Distribution of this memo is unlimited. Please send comments to the editor <mcs@netscape.com>.

Copyright (C) The Internet Society (1997-2002). All Rights Reserved.

Please see the Copyright section near the end of this document for more information.
2. Abstract

This document defines two controls that extend the LDAPv3 [LDAP] search operation to provide a simple mechanism by which an LDAP client can receive notification of changes that occur in an LDAP server. The mechanism is designed to be very flexible yet easy for clients and servers to implement. Since the IETF is likely to pursue a different, more comprehensive solution in this area, this document will eventually be published with Informational status in order to document an existing practice.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in RFC 2119 [KEYWORDS].

3. General Approach

The approach taken by the Persistent Search mechanism described in this document is to alter the standard LDAP search operation so that it does not end after the initial set of entries matching the search criteria are returned. Instead, LDAP servers keep the search operation going. This provides clients and servers participating in Persistent Search with an active channel through which entries that change (and additional information about the changes that occur) can be communicated.

4. Persistent Search Control

This control may be included in the Controls portion of an LDAPv3 SearchRequest message. The controlType is "2.16.840.1.113730.3.4.3".

   PersistentSearch ::= SEQUENCE {
      changeTypes INTEGER,
      changesOnly BOOLEAN,
      returnECs BOOLEAN
   }

The derefAliases field within the SearchRequest message SHOULD be set to neverDerefAliases (0) or derefFindingBaseObj (2). If any other value is used, the server SHOULD return a protocolError to the client.

Upon receiving this control, a server that supports it MUST process this as a standard LDAPv3 search with the following exceptions:
a) If changesOnly is TRUE, the server MUST NOT return any existing entries that match the search criteria. Entries are only returned when they are changed (added, modified, deleted, or subject to a modifyDN operation).

b) The server MUST NOT return a SearchResultDone message. Instead, the search operation MUST be kept active until it is abandoned by the client or until the client unbinds.

c) As changes are made to the server, the effected entries MUST be returned to the client if they match the standard search criteria and if the operation that caused the change is included in the changeTypes field. The changeTypes field is the logical OR of one or more of these values: add (1), delete (2), modify (4), modDN (8).

d) If returnECs is TRUE, the server MUST return an Entry Change Notification control with each entry returned as the result of changes. This control is described in the next section.

5. Entry Change Notification Control

This control provides additional information about the change the caused a particular entry to be returned as the result of a persistent search. The controlType is "2.16.840.1.113730.3.4.7". If the client set the returnECs boolean to TRUE in the PersistentSearch control, servers MUST include an EntryChangeNotification control in the Controls portion of each SearchResultEntry that is returned due to an entry being added, deleted, or modified.

```plaintext
EntryChangeNotification ::= SEQUENCE {
  changeType ENUMERATED {
    add             (1),
    delete          (2),
    modify          (4),
    modDN           (8)
  },
  previousDN   LDAPDN OPTIONAL, -- modifyDN ops. only
  changeNumber INTEGER OPTIONAL  -- if supported
}
```

changeType indicates what LDAP operation caused the entry to be returned.
previousDN is present only for modifyDN operations and gives the DN of the entry before it was renamed and/or moved. Servers MUST include this optional field only when returning change notifications as a result of modifyDN operations.
/changeNumber is the change number [CHANGELOG] assigned by a server for the change. If a server supports an LDAP Change Log it SHOULD include this field.

6. Protocol Flow

An LDAP client "C" begins a Persistent Search session by sending an LDAP search operation that includes a PersistentSearch control to a server "S":

C -> S A SearchRequest message with a PersistentSearch control attached.

The server optionally responds with entries and references:

S -> C Zero or more SearchResultEntry or SearchResultReference messages, the same as would be returned in response to a regular LDAP search operation. If the changesOnly field of the PersistentSearch control was FALSE, nothing will be returned by the server at this time.

As the server becomes aware of changes to entries that match the criteria presented in the SearchRequest, it sends a series of SearchResultEntry messages to the client:

S -> C Zero or more SearchResultEntry messages. If the returnECs field of the PersistentSearch control was TRUE, each SearchResultEntry includes an EntryChangeNotification control.

The client ends the Persistent Search session by sending an AbandonRequest or by closing the LDAP connection:

C -> S AbandonRequest or connection close.
7. Intended Use

Some of the scenarios that the Persistent Search mechanism described in this document is designed to support are described in this section. Other uses of the mechanism are possible as well, but please refer to the "Implementation Considerations" section for some issues to consider.

7.1. Cache Consistency

An LDAP client application with high performance needs may want to maintain a temporary, local cache of information obtained through LDAP search, compare, or bind operations. To improve performance, the local cache is always consulted before sending a request to an LDAP server. The client application can use Persistent Search(es) against the change-log [CHANGELOG] (if one is available) or against one or more subtrees within the LDAP server to enable it to maintain consistency between the data in its local cache and the data stored in the LDAP server. A Persistent Search request where the changesOnly flag is FALSE can be used if it is desirable to prime the cache; otherwise changesOnly would typically be set to TRUE in the request.

Caches are used for reasons other than performance improvement as well. In some cases, they arise naturally out of a particular application’s design. For example, an LDAP client designed for administration of information held in LDAP servers will undoubtedly generate screen displays that show information gleaned from an LDAP server. The screen display is a cache that is active and visible until the user of the application takes some action that causes different information to be displayed. A refresh button or similar control may be provided to the user to allow them to update the cached display. A Persistent Search request can be used instead by the administrative application to automatically refresh the screen display as soon as the underlying LDAP information changes.

7.2. Synchronization

Some LDAP clients such as those that execute on a portable computer may maintain a partial or complete offline copy of the entries stored in an LDAP server. While connected to the network, such a client can direct all queries to the copy of data it holds and use a Persistent Search to actively maintain the contents of the offline copy (alternatively, the client could direct requests to the LDAP server that is the source of the data). While disconnected from the network, the client must satisfy all queries using its offline copy of the data. When the client reconnects to the network, it can synchronize its own copy of the data with the one stored on the LDAP server and proceed to actively maintain its
offline copy by issuing a Persistent Search with the changesOnly flag set to FALSE against the server’s changelog [CHANGELOG]. A search filter like 
\( \text{“(changeNumber} \geq \text{NUM)”} \) where NUM is an integer one greater than the last change the client processed would be used to limit the entries returned to the set of changes the client has not yet seen.

7.3. Triggered Actions

An LDAP client application may want to take some action when an entry in the directory is changed. A Persistent Search request can be used to proactively monitor one or more LDAP servers for interesting changes that in turn cause specific actions to be taken by an application. For example, an electronic mail repository may want to perform a "create mailbox" task when a new person entry is added to an LDAP directory and a "delete mailbox" task when a person entry is deleted from an LDAP directory.

8. Implementation Considerations

Implementors of servers that support the mechanism described in this document should ensure that their implementation scales well as the number of active Persistent Search requests increases and as the number of changes made in the directory increases.

Each active Persistent Search request requires that an open TCP connection be maintained between an LDAP client and an LDAP server that might not otherwise be kept open. Therefore, client implementors are encouraged to avoid using Persistent Search for non-essential tasks and to close idle LDAP connections as soon as practical. Server implementors are encouraged to support a large number of client connections if they need to support large numbers of Persistent Search clients.

This specification makes no guarantees about how soon a server should send notification of a changed entry to a Persistent Search client. This is intentional as any specific maximum delay would be impossible to meet in a distributed directory service implementation. Server implementors are encouraged to minimize the delay before sending notifications to ensure that clients’ needs for timeliness of change notification are met.

9. Limitations of the Persistent Search Mechanism

The mechanism described in the document has some limitations that make
it unsuitable as a comprehensive LDAP change notification solution.

9.1. No Notification When Disconnected

Changes that occur in the LDAP data store while a client is not connected cannot be detected using Persistent Search. Therefore, disconnected clients that also require accurate synchronization must use another mechanism such as [CHANGELOG] in conjunction with Persistent Search. This is inconvenient and it introduces an undesirable dependency on another mechanism.

9.2. No "Enter Set" or "Leave Set" Notifications

No notification is provided when an entry enters or leaves a result set (the set of entries specified by the client using the search base, scope, and filter). This kind of indication would be useful, for example, when an entry enters or leaves the scope of the result set due to a ModifyDN operation or when a Modify operation acts on a attribute value that is used in the search filter. Without this kind of notification, it is difficult for clients to maintain an accurate cache of the entries they wish to monitor.

10. Security Considerations

In some situations, it may be important to prevent general exposure of information about changes that occur in an LDAP server. Therefore, servers that implement the mechanism described in this document SHOULD provide a means to enforce access control on the entries returned and MAY also provide specific access control mechanisms to control the use of the PersistentSearch and EntryChangeNotification controls.

As with normal LDAP search requests, a malicious client can initiate a large number of Persistent Search requests in an attempt to consume all available server resources and deny service to legitimate clients. For this reason, servers that implement the mechanism described in the document SHOULD provide a means to limit the number of resources that can be consumed by a single client.

11. Copyright

Copyright (C) The Internet Society (1997-2002). All Rights Reserved.

This document and translations of it may be copied and furnished to
others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

12. Bibliography


13. Authors’ Addresses

Mark Smith
Netscape Communications Corp.
447 Marlpool Drive
Saline, MI 48176-1519
14. **Appendix A:** Changes since draft-smith-psearch-ldap-00.txt
   Added a Protocol Flow section.

   Updated the [CHANGELOG] reference.

15. **Appendix B:** Changes since draft-ietf-ldapext-psearch-03.txt

   Add text about appropriate values for the derefAliases field of the SearchRequest.

   Updated the [CHANGELOG] reference.

   Updated the text at the beginning of the document to match the latest Guidelines for Authors of Internet Drafts and to remove reference to the LDAPEXT working group.
Updated Author’s addresses and affiliations.

This Internet Draft expires in August 2002.
1. Status of this Memo..................................................1
2. Abstract..............................................................2
3. General Approach..................................................2
4. Persistent Search Control..........................................2
5. Entry Change Notification Control..............................3
6. Protocol Flow.........................................................4
7. Intended Use..........................................................5
  7.1. Cache Consistency................................................5
  7.2. Synchronization..................................................5
  7.3. Triggered Actions...............................................6
8. Implementation Considerations.................................6
9. Limitations of the Persistent Search Mechanism.............6
  9.1. No Notification When Disconnected...........................7
  9.2. No "Enter Set" or "Leave Set" Notifications................7
10. Security Considerations..........................................7
11. Copyright...................................................................7
12. Bibliography...........................................................8
13. Authors’ Addresses..................................................8
14. Appendix A: Changes since draft-smith-psearch-ldap-00.txt....9