Proposal for an Apache Directory Project

This is the original project proposal for the incubation of the Apache Directory Project sent to the Apache Incubator by Alex Karasulu. Although old (September 2003), it is still worth reading as the original vision of the project. It's a piece of history.

Proposal for an Apache Directory Project

10 Sept 2003, Alex Karasulu (akarasulu@apache.org) LDAPd Group Founder

Section 0 : Rationale

The geometric growth of networks, services, distributed systems, and their components has brought about a revival of interest in directories. Directories are no longer considered commodity products used to store a phone book or list of users. Directories and LDAP have grown into a critical technology that will inevitably become a cornerstone of distributed systems. Several software giants like BEA, SUN, and Microsoft have already demonstrated their belief in this outcome by heavily integrating their products with directories. Directory and naming systems are being used in everything from a simplified form for J2EE configuration to full-blown enterprise directories, within identity management systems and distributed computing platforms like JINI.

The overhead of managing several copies of relatively static information across systems and enterprises is daunting and introduces the potential for data inconsistency. The frequently replicated user and group tables in almost every database driven application alone is a testament to this fact. When multiple systems have to inter operate, they must share access to common information, and directories are the best storage paradigm for rapidly accessing this type of data. Whether the information to be stored, is a component's configuration, a user's security profile, a business rule or a business process, directories are the place to store it for centralized management, and distribution across systems.

To date, a fully compliant pure Java open source LDAPv3 server has not yet been written. This is partially due to the fact that a practical stateful protocol server in Java was virtually impossible until the introduction of the NIO packages in the 1.4 SDK. LDAP, as a stateful protocol, manages client state through a TCP socket connection. The socket connection persists through several client requests and server responses tying them together under a logical session or conversation. Before the introduction of non-blocking IO in the NIO packages of the 1.4 SDK, a thread had to be dedicated to each socket to detect IO. A stateful protocol server written completely in Java, would have needed as many active threads as the number of concurrent clients. Theoretically, scalability would have been very poor; the performance curve would be asymptotic, degrading rapidly with an increase in the number of concurrent connections. To scale pure Java statefull protocol server implementations had to wait until the Merlin release of the JDK.

Furthermore, no embeddable pure Java LDAPv3 server has yet been written and made freely available. We believe the embedding of LDAP servers into existing mail, application and web servers to be the next logical steps in the progression and integration of LDAP. Several commercial examples can already be sited: Microsoft Exchange and BEA Weblogic Server. The management benefits of coupling these servers with an LDAP server are beyond the scope of this proposal and left to the imagination of the reader. However, the synergy between embedded LDAP servers and other flagship Apache servers would lead to greater economies of scale for the Apache server suite and reduce the overhead of management for the Apache software user community.

LDAP directories are specialized databases, however unlike their RDBMS counterparts, LDAP directories are fairly new and for this reason lack rich integration features. High on the list of features lacking in LDAP servers are stored procedures and triggers. Triggers and Java stored procedures would expand the horizon of LDAP in itself even without the benefits of Java. Triggers and stored procedures would alleviate ugly workarounds which currently using polling and meta directories. These features alone would attract groups within the LDAP and Java communities to band together and support the Apache LDAP Project.

LDAP is both a network line protocol and a hierarchical relational database. Some might even argue that LDAP is object oriented. Regardless, the implementation of an LDAP server is not a trivial matter. LDAP code bases written in native languages have to be ported to run on different platforms. Conditional complies in C based implementations were shown to considerably reduce the maintainability of code bases. The complexity of a database, a network line protocol and conditional compilation together have resulted in reduced maintainability for C based LDAP servers. Also the learning curve and hence barrier to entry for developers is greater. These complications would not exist for a Java based LDAP server that is written once and can run anywhere.

The Apache Foundation supports several component container projects which provide the proper foundation for a Java LDAP directory server. Apache also supports several server side application containers which can realize significant functional enhancements by employing an LDAP directory server. Also LDAP requiring organizations today must settle for less than optimal implementations and often have to pay as high as a dollar per entry to directory enable their applications. The Apache LDAP Project would solve these problems by providing a free robust LDAPv3 implementation in a standalone configuration or an embeddable configuration for use in all Apache servers and other third party products.

Our goal is to produce a community of developers with backgrounds in LDAP, X.500, Database Engineering and JNDI tasked with the development of pure Java LDAPv3 Clients, APIs, JNDI providers and LDAPv3 compliant servers with the aforementioned features. The produced software will be ASF licensed. Where applicable the best available ASF/BSD licensed code will be reused and new code will be added to complete the LDAPv3 protocol as defined in RFC 3377 and its associated 8 RFCs listed below:

- RFC2251 Lightweight Directory Access Protocol (v3): LDAP on-the-wire protocol
- RFC2252 Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions
- RFC2253 Lightweight Directory Access Protocol (v3): UTF-8 String Representation of DNs
- RFC2254 The String Representation of LDAP Search Filters

- RFC2255 The LDAP URL Format
- RFC2256 A Summary of the X.500(96) User Schema for use with LDAPv3
- RFC2829 Authentication Methods for LDAP
- RFC2830 Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security

Additionally, the schema objects specified within the following RFCs will be defined for use and partial or full complaince will be met where relavent to the project:

- RFC1274 The COSINE and Internet X.500 Schema
- RFC1804 Schema Publishing in X.500 Directory
- RFC2079 Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)
- RFC2247 Using Domains in LDAP/X.500 Distinguished Names
- RFC2293 Representing Tables and Subtrees in the X.500 Directory
- RFC2294 Representing the O/R Address hierarchy in the X.500 Directory Information Tree
- RFC2307 An Approach for Using LDAP as a Network Information Service
- RFC2377 Naming Plan for Internet Directory-Enabled Applications
- RFC2587 Internet X.509 Public Key Infrastructure LDAPv2 Schema
- RFC2589 Lightweight Directory Access Protocol (v3): Extensions for Dynamic Directory Services
- RFC2596 Use of Language Codes in LDAP
- RFC2649 An LDAP Control and Schema for Holding Operation Signatures
- RFC2696 LDAP Control Extension for Simple Paged Results Manipulation
- RFC2713 Schema for Representing Java(tm) Objects in an LDAP Directory
- RFC2714 Schema for Representing CORBA Object References in an LDAP Directory
- RFC2798 Definition of the inetOrgPerson? LDAP Object Class
- RFC2829 Authentication Methods for LDAP
- RFC2830 Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security
- RFC2849 The LDAP Data Interchange Format (LDIF) - Technical Specification
- RFC2891 LDAP Control Extension for Server Side Sorting of Search Results
- RFC3045 Storing Vendor Information in the LDAP root DSE
- RFC3062 LDAP Password Modify Extended Operation
- RFC3088 OpenLDAP? Root Service An experimental LDAP referral service
- RFC3112 LDAP Authentication Password Schema
- RFC3296 Named Subordinate References in Lightweight Directory Access Protocol (LDAP) Directories
- RFC3383 Internet Assigned Numbers Authority (IANA) considerations for the Lightweight Directory Access Protocol (LDAP)

Section 0.1 : Criteria

We feel that this project has a good chance for success as the following project aspects are carefully considered :

Meritocracy: The project will be meritocratic - the usual Apache meritocracy rules would apply.

Community: The user community for this project is potentially massive. The initial developer community for this project consists of developers from Apache and LDAPd projects. The aim is for this community to grow considerably once this project enters the incubator.

Core Developers: The initial developers are listed further below within this document and consist of some existing Apache committers together with committers from the LDAPd Group.

Alignment: There is clear alignment with many existing Apache projects. From Jakarta projects such as Tomcat, James and Slide for embedding LDAP. The LDAP DSML standards will eventually require the use of several XML projects. Other projects like OpenJms?, which currently are not Apache projects but very well may be in the future will be used for things like replication. Jdbm likewise is used for building Btree based indices for custom databases. Also projects like Apache Xindice may be leveraged for an XML based backend or for DSML enabling subsystems of an LDAP server.

We see the Apache Directory Project as a separate project to existing Apache projects, serving several roles

- Home for Apache Directory Client APIs and Client Scripts
- Home for Apache JNDI Providers for Various Directory Namespaces
- Home for Apache Directory Servers
- May also become home to an ASN.1 Encoding/Decoding? Library

Section 0.2 : Warning Signs

We feel this project has a good chance for success as the following warning signs do not apply to the project we are proposing:

Orphaned products: This project is starting with an existing code base which using lots of the high quality open source code out there which is ASF/BSD licensed.

Inexperience with open source: The initial community is made up of existing Apache and LDAPd Group committers exclusively so the open source experience is there to varying degrees.

Reliance on salaried developers: None of the initial developers are currently paid to work on the Directory Project.

No ties to other Apache products: The Apache Directory Project is complementary to existing technologies at Apache. LDAPd currently uses Avalon and will eventually be used by james. LDAPd will eventually integrate into Geronimo. The Apache Directory Project will integrate with, and possibly subsume, Jakarta Commons Naming.

A fascination with the Apache brand: The committers are interested in developing a healthy open source community around ASF/BSD licensed directory servers and their clients along with other JNDI accessible name spaces. Is Apache the right place? We think it is. Several Apache folks have already shown interest in LDAPd, the seed code base to be used for this project. In fact the LDAPd Group was initially approached by Apache to submit an Incubator proposal. LDAPd Group members voted unanimously to donate the code to Apache in an effort to build the next flagship server at Apache and much more.

Section 1 : Scope of the Project

There are multiple goals for this Apache project:

- Promote a healthy open source community!
- Promote a language agnostic environment for both client side and server side endeavors.
- A one stop location for all directory and naming needs on any platform or any language.
    - client APIs
    - clients
    - JNDI providers for Java and other namespaces.
    - May also include an LDAP protocol correctness and stress testing suite in the future.
- Provide servers in embeddable configurations.
- Provide servers in standalone configurations as UNIX daemons or NT Services.
- Provide support for Directory Server embedding or integration with existing Apache servers like httpd, Slide, Tomcat, James and Geronimo.

Section 2 : Initial Source

The initial code base from which to create the Apache Directory repository is contained mostly at Sourceforge under the LDAPd project accessible here. This code base must assign copyright to the Apache Software Foundation. The code base will need to be licensed under The Apache Software Foundation license. The license file in every subproject will be changed to the Apache license. The current license is already an Apache "style" license, so this change is basically a change to name The ASF as the owner.

The proposed TLP (after incubation) would have an one or more directory servers with multiple datastore backends, a JNDI context for J2EE configuration, JNDI service provider(s) for LDAP and other transports, State and Object Factories suitable for use in the various Context/DirContext? implementations, etc.

Beside source code the LDAPd Group must transfer its IANA assigned ASN.1 Object Identifier to Apache Software Foundation.

Section 3 : ASF Resources to be Created

Section 3.1 : Mailing Lists

- directory-dev@incubator.apache.org

Section 3.2 : CVS Repositories

- incubator-directory

Section 3.3 : Bugzilla or (preferably) Jira

- directory

Section 4 : Initial Set of Committers

All contributors must sign and submit a Contributors License Agreement. The following individuals are the contributors, and have signed and sent the agreement at one time or another:

- Alex Karasulu
- Phil Steitz
- Robb Penoyer
- Noel Bergman
- Jeff Machols
- Henri Yandell
- Wes McKean?

Section 5 : Apache Sponsoring Individuals

- Nicola Ken Barozzi (Apache Incubator)
- Noel J. Bergman (Apache James)

- Henri Yandell (Jakarta Commons Naming)
- Stephen McConnell? (Apache Avalon)
- James Strachan (Apache Geronimo)

Section 6 : Incubation Exit Criteria

We feel this project should exit the incubator to a TLP using the domain name directory.apache.org should the following goals be met:

Technical Goals:

- Clean status with Apaches Continuous Integration System
- Website cross reference to existing Apache literature with respect to rules and regulations
- At a minimum a beta release and/or a series of Release Candidates for a directory server and its clients.
- At least one server that complies with RFC2251

Non-Technical Goals:

- List presence and monitoring in wider Apache communities
- Website cross reference to existing Apache literarature with respect to rules and regulations
- Initial integration plan and cooperation with Geronimo, and James
- Member presence on the licensing@apache list
- More Apache or non-Apache committers who are actively modifying source code