



RadiantOne™

**A Virtual Directory Solution
For Directory and Application Integration**

RadiantOne is a Trademark of Radiant Logic, Inc.

All other products or services mentioned in this document are recognized by their trademarks, or the registered trademarks of their corresponding company or organization. Radiant Logic, Inc. disclaims any responsibility for specifying which trademarks are the property of the various companies or organizations.

The material in this document is subject to change without notice, and it does not represent any obligation on the part of Radiant Logic, Inc. Radiant Logic, Inc. is not responsible for any errors contained inside this document. It is illegal to copy this document unless a license or non-disclosure agreement specifically allows it. This document, or any part of this document, may not be reproduced or transmitted by any means, electronically or mechanically, including photocopying and recording, for any purpose, without the express written consent of Radiant Logic, Inc. No part of this publication may be transcribed, stored in a retrieval system or translated into any language without the prior written consent of Radiant Logic, Inc.

Copyright © 2002 Radiant Logic, Inc.

Radiant Logic, Inc.
1682 Novato Blvd, Suite 300
Novato, Ca. 94947
415.209.6800

www.RadiantLogic.com

Overview

Secure Web access, Portal Authentication and Authorization, User Identity Management, Policy Management and Resource Provisioning: these are just a few of the very sizeable web deployment projects that require an integrated directory. However, as companies optimize their internal organization to better interface with the outside world, they face the following directory challenges:

- Creating directories that provide a common view of people (customers, employees, partners, etc.) and identities (services, devices, etc.), as well as the resources that each is entitled to access.
- Keeping directories in sync with the multiple sources from which they originate (i.e. must be capable of reflecting the agility of the enterprise).
- Defining security, access control, profile management, and resource provisioning architectures in multi-participant networks.

Alas, to the detriment of system administrators, resolving these challenges remains elusive. Directory services deployment and integration is slow, complex, service intensive and expensive. In short, the directory architecture is often the weakest link in an integration and collaboration initiative.

This white paper describes how a new approach based on a technical innovation, the **RadiantOne™ Virtual Directory**, is radically changing the integration equation with a simple, efficient and cost effective solution. By combining the best features of LDAP, XML and RDBMS technology, the RadiantOne™ **Virtual Directory Server (VDS)** transforms directory services from a static network service into a standards-based, service-oriented, real-time interchange that delivers “Context on Demand”™ for the agile enterprise.

First, we shall review the key motives for deploying a standards-based directory infrastructure. The strength of LDAP directories provides crucial advantages in comparison to a centralized database or a registry. We will also examine the important practical problems facing such a deployment, in particular the weaknesses inherent to the current LDAP model and implementation. These limitations severely impair the required integration of directories, databases and applications.

The second part will show how a “virtual directory” can leverage the strength of RDBMS, LDAP and XML technology to solve these problems. We will reveal how Radiant Logic’s VDS implements virtual directory architecture fully addresses the current directory and applications integration challenges.

Why LDAP?

The most efficient way to supply policy management and secure Web access projects with identity information is by creating a central store. One may argue that a “classic” store, such as an RDBMS, will adequately serve this function. The following is a review of the important advantages that make a LDAP directory the most reliable and industry accepted choice for a centralized information management system.

Creating a Central Store Based on a Well-Established, Standard Protocol

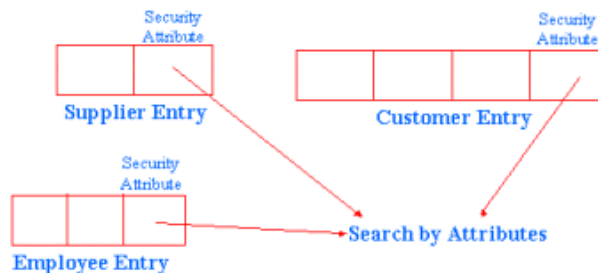
First, a central store being shared by many applications needs to support a common schema/information model and protocol that can be accepted by most applications. Derived from X500, LDAP is a well-established standard that offers a simple protocol and clear information model for a directory service. Due to this standardization, information sharing between LDAP directories is easy and straightforward.

The Hierarchical Structure of a Directory is Self-Disclosing

Beyond centralization, directories have other important advantages. The same way a company’s Org Chart reveals its structure and organization, a directory’s hierarchical structure is self-disclosing. By navigating the directory information tree, a user or program can easily “discover” the critical relationships that reveal information in the right context and facilitate understanding. Both users and applications benefit from this representation of explicit relationships because it provides a relevant, contextual depiction of information.

Attribute Level Searches, Granularity, and Their Importance for Specifying Rules (Security, Policy Management, Profile, and Context Management)

Another, often-overlooked benefit of directories is their use of attribute-level search capabilities. Let us view this from the perspective of a policy management tool. Defining policy rules based on attributes is a lot easier than describing explicitly all the objects that are involved in the rules. For example, it is very simple to state an authorization rule in the manner of, “forbid access to all objects that have a confidential attribute attached.” This policy definition is direct and the directory will answer to the query with a list of the desired objects in one step.



A search on the attribute level will retrieve the three objects without having to specify them explicitly

Now, let us contrast this with the use of a database for the same task. A standard RDBMS must perform searches in accordance with predefined schemas. To perform a query, the

policy management tool is required to know beforehand the underlying schema of the application. This process involves tediously enumerating all the possible objects residing in the database to discover which ones have the attribute desired in the query. As one can imagine, this is far more complex than in the case of the directory.

Summary of LDAP Advantages:

To summarize, LDAP directories provide...

- ... central access based on a well-established standard protocol and information model.
- ...self-disclosing hierarchies which reveal information in its appropriate context.
- ...fine-grain attribute and domain oriented searches that facilitate the implementation of a rule based system.

All of these advantages come at a high cost when trying to deploy and integrate a LDAP directory within an existing IT infrastructure.

The Hidden Cost of Integration and Deployment

Integrating LDAP into a pre-existing IT infrastructure is difficult due to problems intrinsic to LDAP itself and the absence of the tools necessary to support the integration effort. More specifically, a layer of coordination services is required to properly link LDAP with its multiple-application environment.

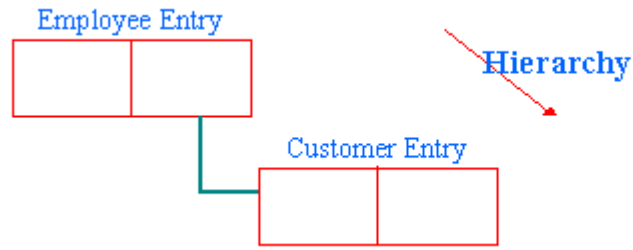
Problems that are specific to LDAP

The Storage Problem

The creators of LDAP performed miracles in simplifying the directory object structure and protocol. However, in their zeal to develop a standards-based directory, they oversimplified LDAP directories by leaving out important database storage features. Many current LDAP implementations are based on a proprietary database that does not contain logging, recoverability, transaction, and events capabilities, features that are key to any RDBMS.

The inability to perform indirect searches: finding an object linked to a specific context.

A strong point for LDAP directories is their ability to search by attributes across objects. However, LDAP sorely lacks the ability to perform indirect searches for a specific object based on attributes belonging to other objects. An example will clarify this point. A search for “Customers that are managed by employee X” cannot be done by a LDAP directory.



Even though within the hierarchy customers are linked to employees, LDAP will not support a search for a set of customers based on a specific employee attribute, “X.”

To support such a search, LDAP would create a **join** operation that merges the two entries and then performs a **search** on the **merged entries**. In order to perform the join, directory designers are forced to artificially add attributes from unrelated entities into the same entry. This “kitchen sink syndrome” leads to poorly designed information models that are difficult to update and maintain.

The Challenges of Integrating LDAP within an Enterprise’s IT Environment

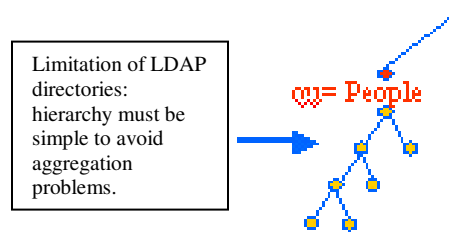
Common schema and standard protocol does not mean flexibility and interoperability.

As a central point, or information hub, a directory is responsible for reconciling different views of data exposed by different classes of applications. This requirement is a big challenge for LDAP directories. Although the directory’s common schema (X500 based) is a positive start, it lacks many key capabilities. The LDAP information tree needs to have the ability to reconcile existing data sources. One should be able to easily extend an existing namespace, and link it back to the rest of an enterprise’s IT infrastructure. This flexibility is the biggest requirement for any namespace derived from an existing informational model. Alas, this flexibility is exactly what LDAP directories are sorely missing. Thus, it is not good enough to just have an information tree based on a standard like LDAP, the information tree must also be able to reflect and evolve with an enterprise’s business.

Shortcomings that lead to a lack of flexibility

The rigidity of the directory tree and namespace stems from several fundamental shortcomings:

1. The LDAP common information model is by definition very generic. The information model is essentially useful for basic categories like people, organization, and organization unit. However, these basic categories do not reflect the diversity of namespaces found throughout the enterprise.
2. The LDAP directory has a relatively static aggregation model that must be kept simple due to difficulties in coding.
3. The fact that LDAP lacks adequate join capabilities means that, as a consequence, the support for federating namespaces is relatively rigid and difficult to manage.



Different information models entail complex synchronization tasks

The fact that LDAP directories have their own standard and repository means they are not interoperable with other applications, databases, and middleware products. Using LDAP to aggregate disparate data sources requires complex transformations and synchronization. Synchronizing complex applications such as SAP™, Siebel™, PeopleSoft™, and Oracle™ is a full time job for EAI project coordinators, however, synchronizing all those data stores with LDAP is an even more complex challenge.

To compensate for this lack of interoperability, most companies tend to include only essential information within the directory in order to limit the need for synchronization and replication with other enterprise applications. However, this completely defeats the purpose of fine-grained authorization and authentication applications where added contextual information is valuable.

Conclusion: IT needs a more advanced and flexible directory service

The Virtual Directory Solution

Radiant Logic's **Virtual Directory Server™** is an innovative implementation of an LDAP compliant directory. Unlike other LDAP implementations that require data to be replicated and synchronized with the data sources, the Virtual Directory does not store any information itself. Instead, the Virtual Directory maps the various data sources and converts data and schemas into XML and LDAP objects. Data sources include relational databases, directories, as well as applications and middleware products. Whatever the

source, the data can be aggregated, reconciled, and mapped by the Virtual Directory into a directory structure.

By eliminating the barriers that separate directory initiatives from the rest of the IT infrastructure, the Virtual Directory benefits are clear. Replication and synchronization scalability issues are eliminated or at least kept to a minimum, and the product allows unlimited directory extensibility with no redesign. Furthermore, by lining up the directory with the rest of the IT infrastructure, RadiantOne™ disintermediates the relationship between a directory and the business environment, enabling directories to effectively mirror the richness of the environment that they represent

The Virtual Directory Server™ brings directory services up to date, enabling them to operate as dynamic “identity and resource” brokers across both organizational lines and corporate boundaries. The RadiantOne™ technology integrates seamlessly with today’s corporate applications (databases, middleware, email systems, etc), and follows today’s standards including LDAP, SQL, XML, JMS, etc. Radiant Logic’s product aggregates, reconciles, queries and updates information from any data source (whether internal or external), makes it appear as a standard LDAP directory with a context-rich directory structure, and allows for multiple dynamic “directory views” based on positions and privileges.

RadiantOne™ Highlights

A key value of RadiantOne™ is the ability to design custom directories around an enterprise’s current data structure, thereby leveraging an the existing information applications in which so much time and money have already been invested. The product approaches the problem of directory integration in three steps.

- RadiantOne™ overcomes the differences in protocol, object model, and architecture that makes data aggregation from multiple sources so difficult.
- The VDS dynamically maps data sources to a directory hierarchy, thereby creating a “virtual directory” that provides a rich representation of a business’ information environment.
- The product facilitates high performance and load balancing for multiple directory views through advanced cache management and dynamic namespace configuration.

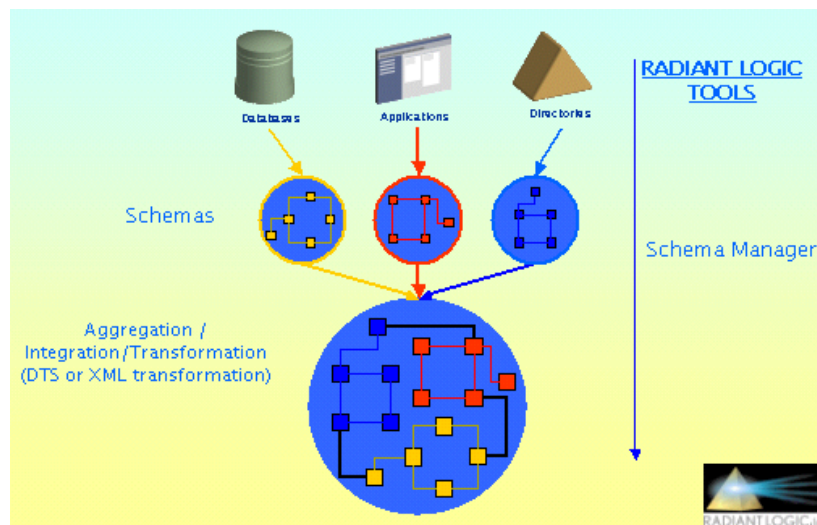
The creation of the virtual directory involves two steps: schema mapping and namespace creation. The RadiantOne™ Digital Context Schema Mapping (DCSM) maps the relationships between data in data sources. LDAP namespace hierarchies are then built on top of this mapping.

Radiant Logic's Intelligent Object Mapping and Cache (IOMC) not only eliminates the need for data transformation, synchronization and replication, but also provides a just-in-time, high performance delivery service of automatically updated data. The combination of the DCSM and IOMC generates a context-rich directory structure and allows for multiple "directory views" based on positions and privileges. Any data source can be transformed dynamically into a directory structure that reconfigures itself on the fly based on the user's and application's navigation within the directory tree.

Overcoming Structural Incompatibility

A major directory integration obstacle is the requirement for synchronizing database and application information with the directory, a process that involves complex mapping and replication. The VDS avoids the trauma of importing data by dynamically mapping from heterogeneous data sources into a directory format. The **Schema Manager**, part of the VDS toolkit, is used for extracting, mapping, and defining schemas from any data source that can be reached using ODBC, JDBC, and LDAPv3. The Schema Manager stores the extracted metadata in configuration files, encodes them in XML, and records their required data source connections.

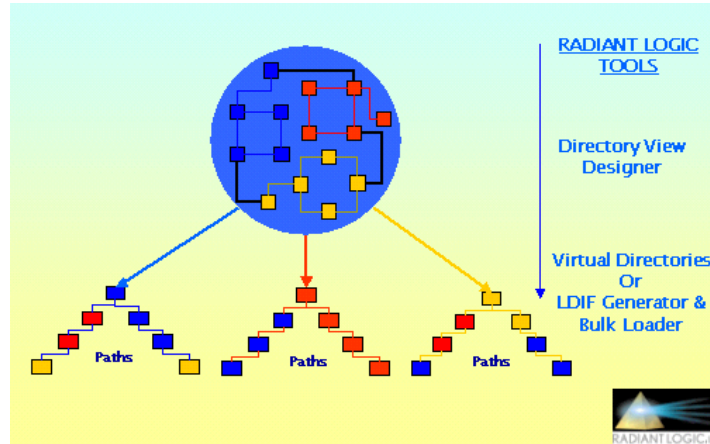
With the Schema Manager, system administrators can create a "unified" schema that represents all possible combinations of information. The Schema Manager manages the mapping of differences between objects and attributes. These schemas can then be combined into a custom directory using the **RadiantOne™ Directory View Designer**.



Designing and Publishing Directory Views

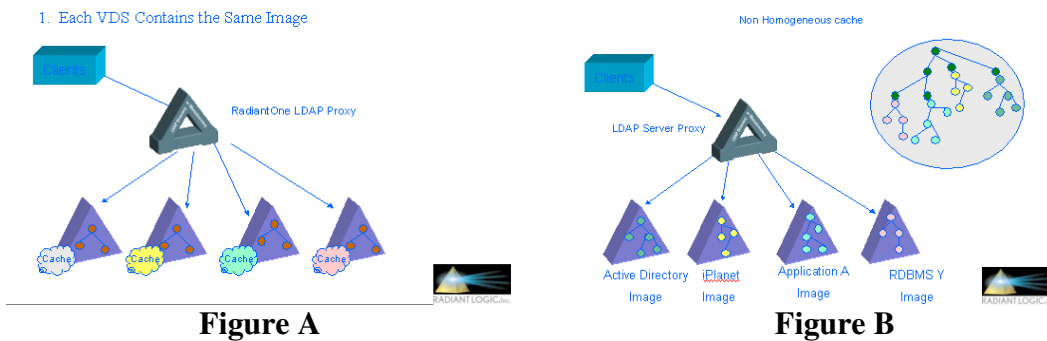
RadiantOne™ also features a **Directory View Designer Tool** that system administrators can use to prototype many different directories before deploying them on the system. The Directory View Designer automatically generates the code associated with a design, and can then be used by business experts to design and modify directories graphically. A "flat" namespace can be deployed based on existing tables, objects, entities, and views.

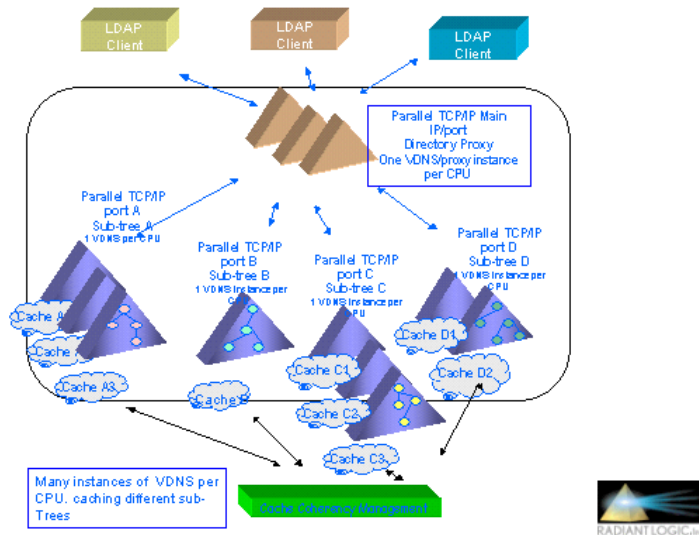
Complex, hierarchical namespaces can also be built based on the relationships that can exist between the different entities in the various data sources. Since it is supported by protocol transformation capabilities, the Directory View Designer can also be used with other directories.



Just-in-time data for an on-demand directory: Cache Management

Mapping a wealth of data from multiple data sources does not by itself ensure the ability for directories to accurately reflect the life of the business. The data source may change due to updates or new relationships. Accuracy requires real-time data aggregation and updates, as well as the automatic recreation of their contextual environment. VDS provides advanced cache management mechanisms. Given that each system is geared toward certain needs, the Virtual Directory cache can be adjusted according to usage patterns. Administrators can configure the cache depending on whether queries are geared toward high volume and availability (**figure A**), depth and richness (**figure B**), or both.

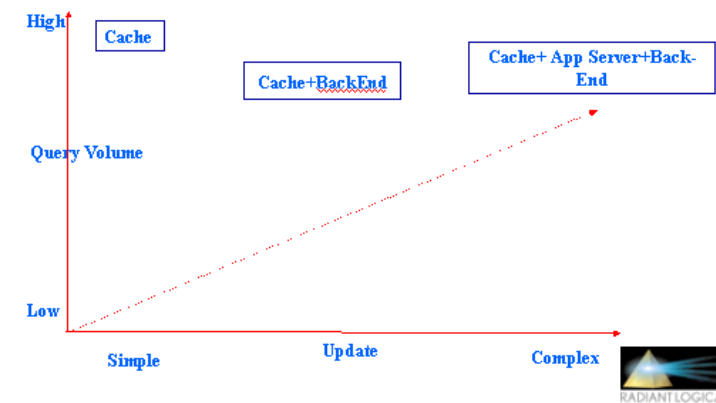




A combination of both cache patterns can also be deployed.

Load Balancing

If the system is essentially used for queries (a classic directory pattern), then load balancing at the level of the cache will bring the best results.



If the system involves a mix of uses a large number of queries and a high level of updates, then both the cache and the back-end need to be load balanced. If the system usage focuses on business logic with a high level of queries and updates, then the load will be distributed between the cache, the back-end, and the application server.