



Using the Virtual Identity Server (VIS) to migrate from one LDAP platform to another platform (e.g. from Sun/Oracle or Novell eDirectory to Active Directory/ AD LDS)

Background:

Over time it is quite common that companies wish to migrate from one platform to another platform. This can be a relatively easy change, or can be quite complicated depending upon the platform being changed. With regards to moving from one LDAP directory to another, this is often a very complex and involved migration. In fact, for many organizations it may not even seem feasible given some of the constraints.

Given the rise in popularity of Microsoft's Active Directory over the years, many customers are examining ways to move from one LDAP platform (e.g. Sun, OpenLDAP, eDirectory) to the Microsoft Active Directory platform. Additionally, with the acquisition of Sun by Oracle and Oracle's subsequent price increase, many customers are looking for ways to migrate off of the Sun platform.

This whitepaper will outline some of the challenges and considerations when migrating from one LDAP platform to another. Additionally, this whitepaper will highlight how Optimal IdM's Virtual Identity Server software is uniquely positioned to assist organizations in LDAP migrations. Throughout this whitepaper, "**source**" LDAP directory will be used to reference the customers' existing LDAP directory, while "**target**" LDAP directory will be used to reference the LDAP directory that the customer is migrating towards.

A key concept to keep in mind is that *directory migration* is different than *directory synchronization*. While both move data from one directory to another, the goal directory migration is to move data from the old directory to the new one so that the old directory can be retired. The goal of directory synchronization is to support long term directory coexistence.



Challenges in migrating from one LDAP directory to another:

There are a number of challenges facing an organization that would like to migrate from one LDAP directory to another. Several of these key challenges are explored in greater detail and how the Virtual Identity Server (VIS) addresses each of these.

Schema

Schema is often the first item that comes to mind when looking at a directory migration. For example, there are common differences between the schemas of LDAP directories. For example, Novell's eDirectory utilizes the object class "groupofnames", when referring to a "group", while Microsoft's Active Directory utilizes the object class "Group".

Without VIS

One way to solve this problem is to alter the schema of the target LDAP directory in order to support the old schema of the original LDAP directory. In instances where there is *very little difference* between the two LDAP directories (perhaps a Sun to an OpenLDAP) this *may be* feasible.

There are several key things to consider when attempting to solve this manually.

1. As the number of schema changes increases, the risk of not being able to handle the schema changes increases.
2. The greater the differences between the LDAP directories (e.g. from Sun or eDirectory to Active Directory), the risk of not being able to handle the schema changes increases.
3. Changing the schema of the target directory will alter the behavior for all applications using this directory, which may not be feasible.
 4. Applications are either hard coded to expect a certain schema, or if properly coded they will examine the schema of the given LDAP directory. If the applications examine the directory for the schema, the applications will NOT work with the new target LDAP directory because this information does not look the same between LDAP directories.

With VIS

The Virtual Identity Server (VIS) has out of the box capabilities to manage schema differences across multiple LDAP directories. Configuring objectclass and attribute mappings using an easy point and click interface, allows an administrator to easily map one LDAP directory to another. In fact, these object class and attribute mappings are how VIS translates SQL data, making it appear as LDAP data through VIS. Handling schema changes between LDAP directories is an easy point and click configuration within VIS. VIS can handle these schema changes **without changing one line of code in your applications.**



LDAP Directory Differences

While there are certain specifications and standards for LDAP directories such as being LDAP V3 compatible, there are many other things that are not covered in these specifications. This is what makes each of these LDAP directories look and behave differently. Listed below are just a few of the items that are different among LDAP directories.

- **Paging** – Some LDAP directories support paging, some do not and many of the directories implement this differently. Depending upon how your application is written, it may or may not work with the new LDAP platform.
 - **Without VIS**
 - You will need to re-code each and every application to handle paging differences.
 - **With VIS**
 - VIS can easily solve this problem, allowing your existing code to work against the new LDAP directory **without changing one line of code in your applications**.
- **Directory System Agent (DSA)¹** – The DSA is the process that provides access to the store. The store is the physical store of information on the hard disk. LDAP client applications connect to the DSA using the LDAP protocol. When a client application connects to an LDAP directory, the directory returns information telling the client what type of directory it is and what LDAP controls it supports. This DSA information is different between LDAP directories. Simply stated, when connecting to an Active Directory LDAP directory, it looks very different when compared to connecting to a Sun LDAP directory. They look, feel and act very differently. A non-technical analogy would be to think of an Active Directory speaking English, while a Sun LDAP might speak Spanish.
 - **Without VIS**
 - You will need to re-code each and every application to handle the differences between the LDAP directories.
 - **With VIS**
 - **VIS easily solves this problem, using its directory emulation capabilities.** With directory emulation, VIS can look and act like a Sun/Oracle LDAP directory while using an Active Directory (or any other store) as the target LDAP directory. VIS can be thought of as a translation engine, listening and talking Spanish to client applications, while translating to English on the target LDAP server. **No application changes are required.**

¹ <http://technet.microsoft.com/en-us/library/cc961806.aspx>



- **Namespaces, Directory Information Tree (DIT)** – As mentioned previously, there can be many differences between LDAP directories (even LDAP V3 compatible directories). Another area where there are differences is how the LDAP directories handle namespaces and the Directory Information Tree or DIT as it is commonly referred.

A good example of how this can cause issues in a directory migration is the scenario of application bind accounts. In a Sun/iPlanet/OpenLDAP implementation there is a common service account CN=Directory Manager that is used to gain access to the console. In best practice this account should not be used by applications as a bind account, because it bypasses many of the controls within the LDAP directory and has greater access than the applications need. With that said, many times application developers use this account to have their applications connect to the LDAP directory. If this account is hard coded into the application, a directory migration to an Active Directory for example will be impossible because an Active Directory does not allow for a distinguished name (DN) to not include the full namespace. For example, in Sun it is perfectly acceptable to bind using CN=Directory Manager, with Active Directory it would need the DN to be CN=Directory Manager, DC=something, DC=COM. This is just one example where the application is very tightly coupled to the LDAP directory, making a migration without code changes extremely difficult. Another example is how LDAP directories handle referential integrity and relative distinguished name constraints.

- **Without VIS**

- You will need to re-code each and every application to handle the differences between the LDAP directories. In some instances such as CN=Directory manager this may make the move impossible if there is no access to the source code of the application.

- **With VIS**

- VIS easily solves this problem. For example, VIS can append, *DC=something, DC=COM* to the CN=Directory Manager before processing occurs on the back end LDAP directory. Additionally the CN=Directory Manager can be mapped to an alternative user in the target system. **No application changes are required.**



- **Attribute Control Lists (ACL's)** – How directories implement Attribute Control Lists (ACL's) vary widely. This can be especially difficult in a directory migration where a customer has utilized a feature of a given LDAP directory that is not present in the target LDAP directory.

A good example of this is the ability in a Sun Directory to control what accounts can read or write to a given LDAP attribute. In the Sun Directory, this can be controlled via group membership; meaning only members of an "Admin" group can update or delete a certain attribute. Active Directory, Active Directory Application Mode (ADAM), or Active Directory Lightweight Directory Services (AD LDS), do not offer an easy/manageable way to perform this.

- **Without VIS**

- Significant changes may need to be made in each and every application to handle the differences between ACL features. It is possible that this might not be possible at all in the target LDAP directory, even with making application changes.

- **With VIS**

- VIS easily solves this problem. For example, VIS can easily check group membership and either allow the update to occur (if the user is in the group) or to return the appropriate error message. **No application changes are required.**

- **Other Differences** – Beyond the items mentioned previously, there are many other differences between given LDAP directories. For example, consider if you were migrating from Active Directory to an OpenLDAP. Active Directory has an attribute on a user called memberof. This attribute contains the distinguished names (DN's) of the groups the user is a member of. OpenLDAP does not support this concept.

- **Without VIS**

- In the migration scenario of moving from AD to OpenLDAP, applications would need to change to no longer use the memberof attribute and instead query the directory for the group membership information. This may or may not be feasible.

- **With VIS**

- VIS easily solves this problem, allowing functionality like this to be surfaced to the application. **No application changes are required.**



- **Migrating Passwords** – Another problem with a manual migration is handling migrating passwords. Different LDAP directories often do not use the same hashing algorithms for passwords. Additionally, unless the passwords are stored in clear text or encrypted (where they can be decrypted) it will not be possible to move/migrate the passwords from one LDAP directory to another.
 - **Without VIS**
 - If the passwords are stored in hash as opposed to an encrypted format, it will not be possible to manually migrate the passwords via an export/import. This would then require some other outside process to force users to have their passwords re-set to some new value and to provide this new password to the end users. This may not be feasible from a technical or business perspective and would require costly custom code or other software to accomplish this task.
 - **With VIS**
 - VIS easily solves this problem. This problem is often handled by configuring VIS in the following manner.
 - VIS is configured for a period in time to both the Source and the Target directories.
 - VIS is configured to intelligently route authentication requests based on whether the user has authenticated through VIS. The first time a user authenticates through VIS, VIS will route the authentication to the old Source LDAP directory. Assuming that the authentication was successful, VIS would then set the password on the user object in the Target directory. VIS would then mark this user as being migrated and would route subsequent authentication requests to the Target LDAP directory. After a period of time, the Source LDAP directory is decommissioned.



Conclusion:

Attempting to manually convert from one LDAP directory to another by manipulating the schema, DIT and ACL's of a new target directory is **extremely risky and cannot be guaranteed to be successful**. As the number of applications and the number of differences between the LDAP directories increases, so does the risk that this manual conversion will be successful. Additionally, you will only know "if" you will be successful **after** you have spent the time, money and effort to perform all of the manual changes and tested **each and every** application.

Without using a proven solution such as the Virtual Identity Server, a lot of time and money will be spent without a working solution.

Listed below are some general questions that you may want to consider.

- **How tightly coupled are your applications written to the existing LDAP directory?**

VIS is a black box abstraction layer between the applications and the physical directory, enabling you to quickly and easily make changes without impact to either the application or the directory.

- **Are you sure you know EXACTLY what the applications are doing?**

VIS can quickly and easily produce application profile reports that show you exactly what the applications are doing.

- **Do you have source code to ALL of the applications to make the necessary changes?**

*VIS's emulation and abstraction engine allows you to make the migration **without changing any of the applications source code**.*

- **Assuming you have all of the source code, do you have an idea of the time, money and effort it will take to change the applications?**

VIS provides an easy cost effective mechanism for a directory conversion.

- **With a manual conversion, ALL applications must first be converted before you will know that the applications will work. As the number of applications and customizations increase, so does the risk in manually converting (if possible). Can you afford the risk of this big bang migration?**

VIS enables you to migrate applications slowly over time. VIS is a production tested and proven solution that has been by countless other customers to solve this exact problem.

Conversely, using the built-in and out of the box capabilities of the Virtual Identity Server, converting from one LDAP platform to another can be easily achieved. VIS provides an easy to use and repeatable method that has been utilized by customers around the world to solve the LDAP migration problem.