
Understanding LDAP Integration with Clarity

CA Technical Services

© 2007 CA

TABLE OF CONTENTS

Preface	3
Enabling LDAP and Single Sign-On	3
Overview	3
Security Properties for LDAP configuration	4
Encrypting Server Passwords:	7
Mapping Roles in Active Directory to roles in Clarity	7
Resource Management Access Rights	7
About Resources and Roles	8
About the Organizational Breakdown Structure.....	8
An OBS is composed of:	8
About Clarity XOG (XML Open Gateway).....	10
Example: PeopleSoft ERP as the Master for Resource Information.....	10
GEL: Integration Scripting	11
Getting Role information from Active Directory	11
LDIFDE.....	11
CSVDE	11
An approach of importing LDAP roles into Clarity	12
Summary	12
LDAP Overview.....	12

Preface

The following overview is not meant to replace the technical documentation and is provided for informational purposes only. The information provided here is a collection of high-level documentation from different sources provided as a tool to familiarize the process of integrating LDAP with Clarity. Please see official user documentation for more information, or contact your Clarity Representative.

Enabling LDAP and Single Sign-On

If your users use several applications, it can be beneficial to implement a Lightweight Directory Access Protocol (LDAP) interface to authorize user access across all the applications. Instead of storing user access information separately for each application, a central directory server controls access so that users can have one username and password for all applications.

Clarity supports the LDAP v2 protocol (simple) protocol and uses a small subset of LDAP functionality including authentication (clear text or SSL), binding, and searching. Session-based cookies carry a token that is used to access session data and is persisted in the cache for single application environments or in a database for clustered environments. The user's web browser must accept cookies from the Clarity application, which are session-based, so they are never written to disk. When the user logs out, session information in the database and cache that correspond to the cookie are deleted.

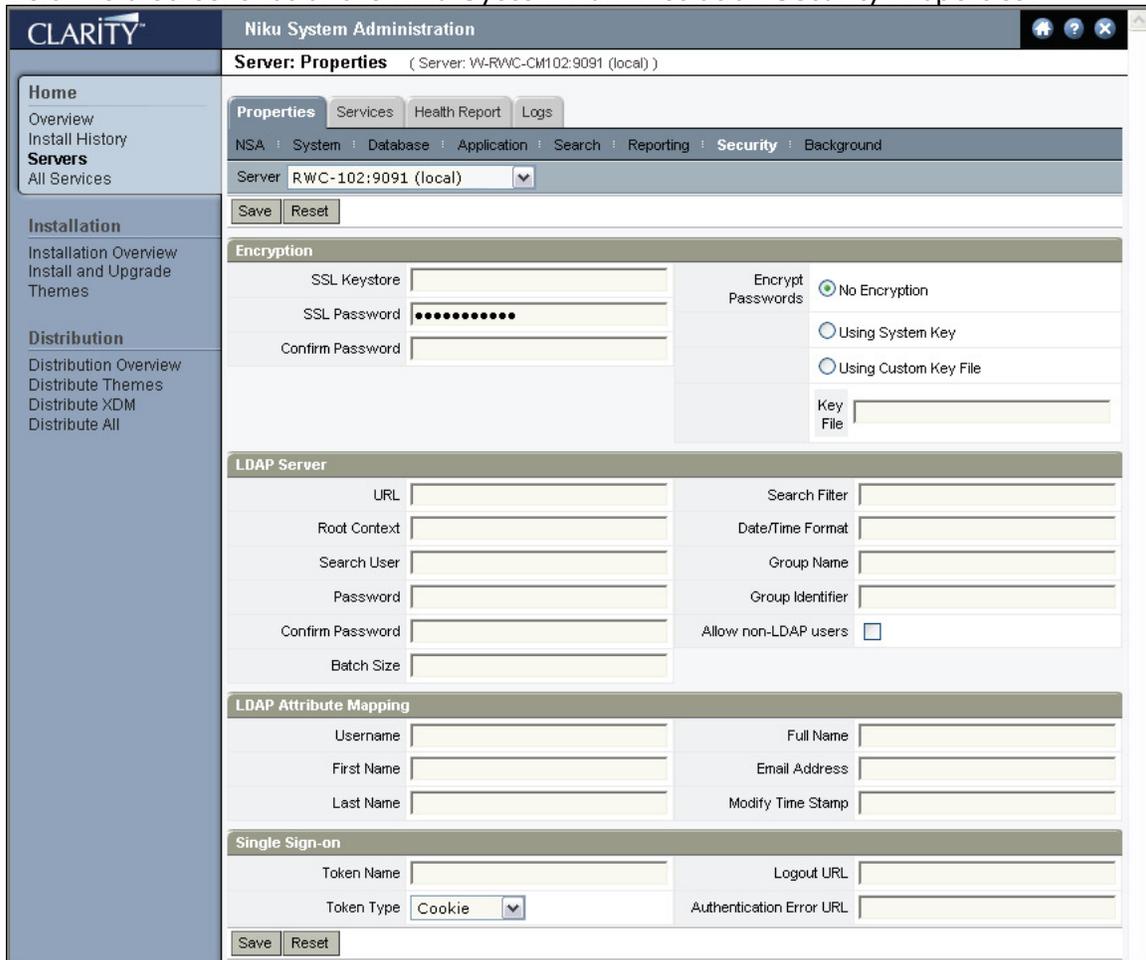
Clarity's *LDAP Synchronize New and Changed Users* job synchronizes LDAP entries. It then stores the last date and time the job ran successfully and stores information in the `MN_DIRECTORY_SERVERS` database table. The next time the background job runs, it synchronizes only recently-created or changed user entries where the timestamp is greater than the value found in the `CMN_DIRECTORY_SERVERS.LAST_SYNC_DATE` property.

If a user is deactivated on the LDAP server, the next time the synchronization job runs the user is deactivated in Clarity. If the user is re-activated on the LDAP server the user will not be re-activated in Clarity; you will need to re-activate the resource (see "Activating and Deactivating Resources," on page 85 of the Administration and Configuration Guide).

Overview

- First, ensure that you can log into the Clarity application without enabling LDAP. Use an existing user or create a new user that exists both in Clarity as well as LDAP.
- Give the user rights to run the Clarity LDAP sync jobs.
- Ensure that you can log into the application using the Clarity password for this user.
- Configure and Enable LDAP through Niku System Administration (NSA). Restart both Application and Background Services.
- Log into the Clarity application for the user using the LDAP password.
- Run the LDAP sync job.
- All new users will get added to the Clarity application.

Below is a screenshot of the Niku System Administration Security Properties:



Security Properties for LDAP configuration

<i>LDAP Server Settings</i>	<i>Description</i>	<i>Synchronized in Cluster</i>	<i>Services to Restart</i>
URL	URL of the LDAP server. For example: "ldap://localhost:389" "ldaps://localhost:636"	Yes	app and bg
Root Context	Root LDAP context. For example, "ou=People,dc=niku,dc=com"	Yes	app and bg
Search User	User that performs any of the directory searches (bind operations). Must be a fully qualified distinguished name of the LDAP user with read rights. For example: uid=nikusearch,dc=niku,dc=com	Yes	app and bg

LDAP Server Settings	Description	Synchronized in Cluster	Services to Restart
Password	Search User's Password.		
Search Filter	<p>Optional LDAP search filter. Default value: "(employee=niku)" Specifying a value in this field enables you to define search criteria and to provide more efficient and effective searches.</p> <p>Niku supports the LDAP search filters as defined in RFC 2254. These search filters are represented by Unicode strings. The following URL describes RFC 2254 in detail: http://www.faqs.org/rfcs/rfc2254.html Section 5 in this document contains several examples. For convenience, some are provided below: To select a user whose first name is "Babs Jensen": (cn=Babs Jensen) To select a user whose first name is not "Tim Howes": (!(cn(=Tim Howes)) To select entries from LDAP of type Person and whose last name is "Jensen" or first name is like "Ben J*": (&(objectClass=Person) (sn=Jensen) (cn=Ben J*)))</p>	Yes	app and bg
Date/Time Format	<p>The Date/Time format of the vendor's directory server. For example, if using Novell eDirectory, use: yyyyMMddHHmmss'Z' MS Active Directory, use : yyyyMMddHHmmss'.0Z'</p>	Yes	app and bg
Group Name	<p>Group Name If specified, authentication and synchronization of users will be done against all the users that are found in this group. It must be a fully qualified distinguished name of the group. For example: cn=QA,ou=Engineering,dc=niku,dc=com</p>	Yes	app and bg
Group Identifier	<p>The name of the attribute that if present signifies that an entity is a group. Different LDAP servers use different attribute names.</p>	Yes	app and bg

LDAP Server Settings	Description	Synchronized in Cluster	Services to Restart
	For example, Novell eDirectory uses the name "uniquemember". MS Active Directory uses the name "member". Default value: "uniquemember".		
LDAP Attribute Mapping			
	*Note: all attributes are required.		
Username	User attribute mapping. Different LDAP servers use different attribute names. MS Active directory uses the name "sAMAccountName". Novell eDirectory uses the name "uid". Default value: "uid"		
First Name	First Name attribute mapping. Default value: "givenName"		
Last Name	Last Name attribute mapping. Default value: "sn"		
Full Name	Full Name attribute mapping. Default value: "cn"		
Email Address	Email Address attribute mapping. Default value: "mail"		
Modify Time Stamp	Modify Time Stamp attribute mapping. Default value: "modifyTimeStamp"		

Encrypting Server Passwords:

Clarity offers DES encryption of server passwords in the properties.xml file. This encryption method requires that you to use a separate password (key) for use when encrypting the actual passwords in properties.xml. You must still keep this unencrypted key secure.

The advantage of using server-side encryption is that you only have to secure this one key, which is stored in a file accessible by the server. The key is only required at server startup. After Clarity is running, the key file can further be secured with another layer of encryption— or if the file resides on removable storage it can be detached and locked in a safe. (For more information, see Encrypting Server Passwords p.52 of the Administration and Configuration Guide.)

Note: Encryption must be off to install Actuate reports.

Mapping Roles in Active Directory to roles in Clarity

Although you can manually assign roles within the Clarity UI, you can also import them through XOG. First, we'll define the resource management access rights.

Resource Management Access Rights

Access to resource management functionality is managed by the use of access rights. Clarity provides rights on a number of levels to offer maximum flexibility and protection to Clarity users.

Clarity provides the following types of access rights:

- **Global.** A global access right is followed by the word "All," as in Resource - Edit All. The suffix "All" means that you can perform the action the global right gives you to all instances of that object in Clarity. For example, the Project - Edit - All right allows you to edit all of the projects in Clarity. Similarly, the right Resource - Edit - All allows you to edit the profiles of any resource in Clarity.
- **Instance.** An instance-level access right is not followed by the word "All," as in Resource - Edit or Approve Timesheet. Resource instance rights are granted per instance of a resource. This means that you can edit only the instance you have been granted access to. For example, if your Clarity administrator has granted you Resource - Edit access to the resource David Smith, you can edit that resource's profiles. In this case, you would not be able to edit any other resource profiles unless you had instance-level or global-level access to them. Most users have only instance-level rights to certain objects.
- **Group.** Instance and global access rights can be granted at the group level, so that if you are a member of a group, you will receive whatever access rights the group has been granted.
- **OBS.** Instance and global access rights can be granted at the OBS unit or department level, so that if you are member of an OBS unit or department, you will receive whatever access rights your OBS unit or department has been granted.

Access rights can be granted by Administrators, resource managers, and project managers. When you are granted access rights, you should be notified in some way. If you are unsure of your access rights, please contact your Clarity Administrator,

resource manager, or project manager. (For more information, see Using Resource Management)

About Resources and Roles

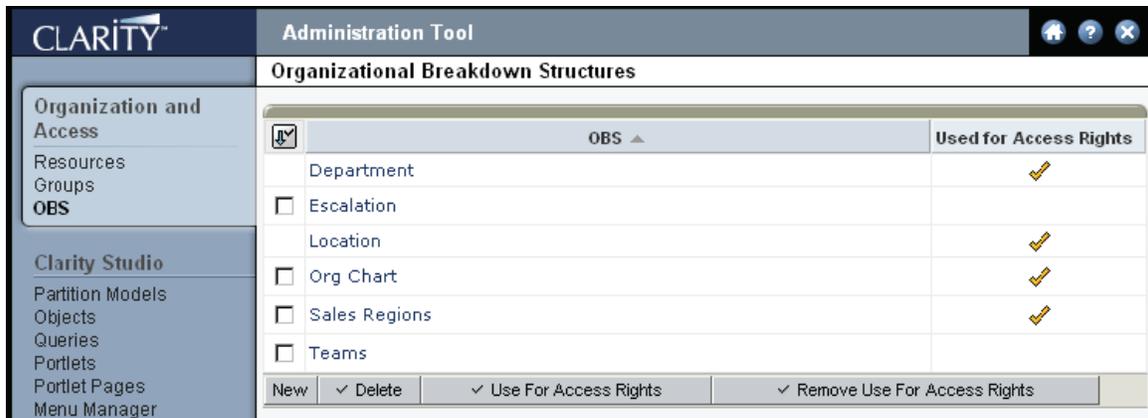
In Clarity, a resource is a person or object (such as equipment or rooms) that is used to fill a role (perform a task). For example, Robert Smith writes code for a project on which you work. In Clarity, Smith is a resource whose role is "Developer". Clarity allows you to create simple or detailed resources and role profiles. You can create roles and assign resources to them; associate resources or roles with projects, skills, documents, and calendars; or edit resource allocations to projects. You can also use Clarity's Administration Tool to change resources and roles. (For more information, see Encrypting Server Passwords p.82 of the Administration and Configuration Guide.)

About the Organizational Breakdown Structure

The organizational breakdown structure (OBS) is a hierarchical representation of your company's structure. An OBS can be used to align projects and investments, resources, and most Clarity objects. With an OBS, you can:

- Grant resources access to object instances based on their OBS membership.
- Support financial setup.
- Associate collections of resources with partitions.
- Categorize objects for filtering and reporting purposes.

To access OBS, select OBS from the Organization and Access menu in Clarity's Administration tool. The Organizational Breakdown Structures page appears.



An OBS is composed of:

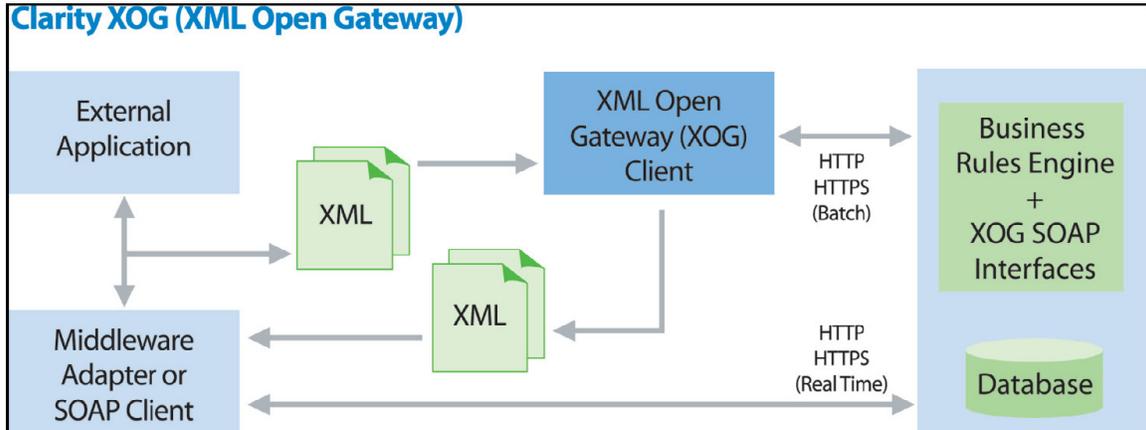
- **OBS type.** The model for the OBS, such as a company's geographical office locations, organizational chart, project types, and resource pool.
- **OBS levels.** The structure of the OBS hierarchy. For example, an OBS type based on organization chart may have "Company" as level 1, "Business Unit" as level 2, and "Department" as level 3.
- **OBS units.** A unit in the hierarchy. For example, an OBS type based on geographical location may use names of countries, states or regions, and cities as OBS units. An OBS unit can be an ancestor, descendent, or both in relation to other OBS units.

You can move OBS and their descendents to other branches or levels as your

company reorganizes without losing any access to objects, resources, and other interface elements. You can also create multiple OBS hierarchies (levels) within the OBS structure.

About Clarity XOG (XML Open Gateway)

The following is a summary on how to import resource information into Clarity.



The XML Open Gateway (XOG) is Clarity's Web service interface. These Web services are available on the same HTTP and/or HTTPS port as the Clarity HTML web browser interface. XOG, however, uses Simple Object Access Protocol (SOAP), an open-standard, human-readable, XML-based protocol. Using XOG, you can read and write data objects from Clarity, execute NSQL queries, and execute other server-side actions.

The XOG API is a server-side and a client-side XOG component which is an interface that is accessible on the standard HTTP or HTTPS port using SOAP. You may use the client-side component to call the API. Since the XOG API is a standard SOAP-based Web service interface, the XOG client is not required. You may use any client-side solution to communicate with XOG.

XOG includes a Web Service Description Language (WSDL) file that is downloadable from your Clarity installation. The WSDL described the available XOG services and how to communicate with them. (For more information, see the Clarity Integration Guide).

Example: PeopleSoft ERP as the Master for Resource Information

If your company uses PeopleSoft ERP as the master resource management system and Clarity as the master project management system, you can import resource information from PeopleSoft ERP into Clarity.

To achieve this integration, use a middleware adapter* to track changes in the PeopleSoft ERP system. The middleware establishes a persistent session with the XOG and then funnels requests from all external applications to the XOG.

The logical steps are:

1. Perform a one-time bulk import from the PeopleSoft ERP into Clarity using the XOG or using a SOAP API.
2. Manage resources in PeopleSoft ERP and update Clarity with new or changed resource data.

*The middleware adapter can be created by Clarity Technical Services using GEL.

GEL: Integration Scripting

GEL (Generic Execution Language) is a scripting language that is based on the Jelly libraries from the Apache Jakarta Commons project. You can use GEL to capture or send data in XOG format.

GEL also provides a collection of standard integrations which provide connectors to enterprise applications such as Remedy® Help Desk. With GEL you can invoke and process a variety of data sources:

- **Web services.** GEL can read or write to any SOAP-based web service. This includes Clarity's XOG web services (see description in Chapter 4, "XOG Object Reference," beginning on page 39).
- **File system.** GEL can read or write to any delimited file including those on local disks, network disks or disk arrays. An output from CSVDE would be an example.
- **FTP.** GEL can upload or download to FTP servers.
- **JDBC.** GEL uses JDBC to access RDBMS to read or write data (including the Clarity database).

Getting Role information from Active Directory

LDAP synchronization will only synchronize the following five fields: username, firstname, lastname, fullname and email address. All other information has to be maintained manually. If you want to get this information from LDAP, you cannot use the standard LDAP synchronization; you will have to create an import interface. But first, you will need to get information from the directory using one of the following:

LDIFDE

A Windows domain controller comes with a command-line tool for importing and exporting LDIF files, LDIFDE. Run `ldifde` with no switches to get a list of parameters.

CSVDE

Working with the LDIF format can get a little tedious because it sorts attributes vertically rather than horizontally. If you prefer a more standard spreadsheet layout, use the CSVDE utility. The switches for CSVDE are the same as for LDIFDE.

An approach of importing LDAP roles into Clarity

One way is to create an integration process. A process is a series of operations that you use to accomplish an integration process. Steps are a series of operations and each step performs an action that is intended to move the process toward its completion. If desired, you can group steps.

Integration processes can be disconnected from any specific object. This allows you to schedule integration processes or initiate them in real-time, manually from the GUI, or by XOG Web service request. Real-time integrations enable external applications to send data proactively. The request starts an integration process and then passes the incoming data.

Custom Actions are available for normal process steps which include custom GEL code. These GEL snippets use tag libraries to interact with various data sources and data destinations, and form a very flexible technical integration environment. You can schedule integration processes manually from the Clarity GUI, using an incoming XOG web service request, or you can invoke it from a background job. This allows real-time integrations that send data proactively.

Summary

1. Active Directory export via CSVDE
2. Middleware app that transforms the CSVDE to XOG format XML
3. XOG into Clarity
4. This can all be automated in a process

It is recommended that the above integration is created by Clarity Technical Services

LDAP Overview

- Clarity synchronizes user data from LDAP server. No information is sent from Clarity to the Active Directory.
- There are two synchronization jobs that run from Clarity:
 - Synchronize New and Changed Users
 - Synchronize Obsolete Users
- Clarity supports the LDAP v2 and v3 protocol.
- Clarity uses the following LDAP functionality:
 - Authentication (clear text and SSL)
 - Binding
 - Searching
 - Paged Results (v3 only)
- Clarity supports the following LDAP servers:
 - Novll eDirectory 8.7.1
 - Sun ONE Directory Server 5.x
 - Microsoft Windows 2000 – Active Directory
 - Microsoft Windows 2003 Server – Active Directory