Version 1.1

JBuddy LDAP Gateway

User's Guide

Contents

Release Notes

Release 1.1

RELEASE 1.1.120227

- Added support for restricting authenticated users to a specific organization unit (ou) and/or a specific LDAP/AD security group.
- Added support for setting user's nickName (displayName) to LDAP/AD user's firstName lastName and keeping it in sync upon login.
- Added support for looking up Buddies in support of JBuddy Messenger's Add Buddy feature, using a firstName and/or lastName search.
- Added SASL DIGEST-MD5 password encryption to communication with Active Directory and other LDAPv3 based directories.
- Added military-grade password encryption to communication with JBuddy Server components.

Prior Releases

Please refer to the docs/ReleaseNotes.html or the online version available at http://www.zionsoftware.com/support/jmessageserver/ldap/docs/ReleaseNotes.html for the complete release notes of JBuddy LDAP Gateway.

Introduction

Intended Audience

Welcome to the JBuddy LDAP Gateway User's Guide. This guide is intended primarily for two types of readers:

- System Administrators
- Directory Administrators (LDAP or Active Directory)

System Administrators would typically be involved in preparing the hardware and software environment for JBuddy Server installation and ongoing maintenance and administration. Directory Administrators may be a distinct role within the business or handled by System Administrators. They are responsible for the ongoing support and maintenance of the corporate directory and may want to be aware of how JBuddy LDAP Gateway interfaces with the directory for authentication purposes.

Terms

First a clarification on terms is in order.

- 'JBuddy Message Server' also known as simply 'JBuddy Server', means the server-side components making up the enterprise instant messaging solution.
- 'Directory' means any LDAP v3 compatible directory including Microsoft Active Directory, Mac OS X Open Directory, or openldap. Directories are expected to store user account credentials in industry standard form and must support an LDAP 'simple bind' using plaintext passwords.
- 'XTG' means the XMPP Translation Gateway. The XTG proxies a connection from a XMPP IM Client to the JBuddy Server.

Summary

For organizations with an existing directory, the JBuddy LDAP Gateway greatly simplifies user account management. The JBuddy LDAP Gateway runs as a separate service and proxies LDAP bind and search requests using simple auth (username & password) to a directory service on behalf of JBuddy Server. With this in mind, carefully read through this document to learn more about JBuddy LDAP Gateway, architecture, security considerations, installation and administration.

Architecture

To make the most of JBuddy LDAP Gateway, it is important to understand the role it plays in a JBuddy Server deployment. The JBuddy LDAP Gateway is a separate service within the JBuddy Server solution and may be run on the same server or a different server depending on the requirements of the organization. See Figure 1. Also see the Security section below for deployment considerations. Additionally, multiple instances of the JBuddy LDAP Gateway may be deployed in environments where redundancy is critical.

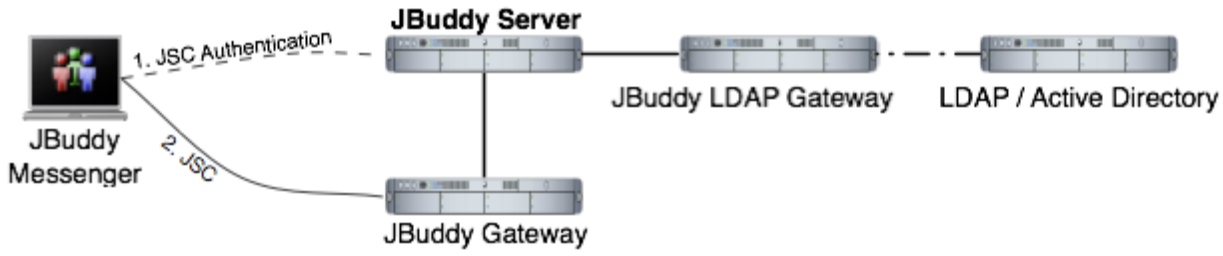

Figure 1: Architecture

JBuddy Server Authentication

If you are only interested in private, enterprise IM using the JBuddy Server, you only need JBuddy IM accounts (even if you intend to connect to the JBuddy Server using XMPP Clients using the XTG). JBuddy IM Accounts can be created one of three ways.

- Automatic
- Manual
- Database bulk load

This guide only covers the Automatic JBuddy IM Accounts since the focus is using the JBuddy LDAP Gateway. Note: **Only the Automatic JBuddy IM Accounts are available for login if the JBuddy LDAP Gateway is used. If the JBuddy LDAP Gateway is not used, only the Manual or Database bulk load created JBuddy IM Accounts are available.**

Authentication with the JBuddy LDAP Gateway

Due to the enhanced miltary-grade encryption added in v1.1, the JBuddy LDAP Gateway version 1.1 must be used with JBuddy Server version 3.3 and newer as well as JBuddy Messenger version 3.3 and newer. When used together, the corporate directory becomes the authoritative source for JBuddy IM Accounts and logging into the JBuddy Server. The same username and password used to login to the computer system is available for use in logging into the JBuddy Server.

Automatic JBuddy IM Account Creation

When a JBuddy Messenger user, a JBuddy SDK-enabled application, or a XMPP Client (connecting through the XTG) attempts to login to the JBuddy Server for the first time, the JBuddy LDAP Gateway makes an LDAP Bind request to the corporate directory with the username and password provided by the user. On the first successful LDAP Bind, JBuddy Server automatically creates a JBuddy IM Account in the JBuddy Server's database, based on

the username provided. In the JBuddy Server database, a new row is created in the Account table. This JBuddy IM Account is owned (linked with a foreign key to the row representing the Sys Admin in the Profile table) by the ‘Sys Admin’ User Profile provided during JBuddy Server installation. The password for this new JBuddy IM Account is set to a random, encrypted value to prevent this Account from being used unless the JBuddy LDAP Gateway is available.

Security Considerations

Client Security

While the primary security consideration for IM clients is password security, it should be noted that unless the JBuddy Server is setup for SSL/TLS connections, communications besides passwords (messages, presence, etc) are sent unencrypted.

JBuddy Messenger v3.3

JBuddy Messenger v3.3 introduced military-grade encryption to safe-guard user passwords even over non-TLS/SSL encrypted network connections. v3.3 must be used with JBuddy Server v3.3 and JBuddy LDAP Gateway v1.1 or newer in order to authenticate securely.

XMPP Clients

The XTG supports SASL “PLAIN” and “DIGEST-MD5” authentication modes. JBuddy Server v3.3 updated the XTG to support authentication with JBuddy LDAP Gateway v1.1, however, the updated XTG currently only supports “PLAIN” authentication mode when used with JBuddy LDAP Gateway v1.1. Currently there is only a true or false setting. Since most deployments will authenticate against a Directory, we have defaulted this to true. If you use local accounts and which to support SASL DIGEST-MD5, switch it to false. See lib/XTG.properties:

```
xtg.use_plain_sasl_mechanism=true
```

Since “PLAIN” SASL sends the user’s password in clear-text over the network, it is strongly recommended that the XTG be configured to support SSL/TLS connections from XMPP clients. See the XTG.properties file for settings in support of SSL/TLS.

Server to Server Security

JBuddy Server to JBuddy LDAP Gateway

The JBuddy LDAP Gateway (as well as all the other optional JBuddy Server Gateways) locates and connects to the JBuddy Server through a Java service called RMI. As part of the JBuddy Server installation, a Java RMI service is launched and it typically listens on port 1099 on the same machine as the JBuddy Server. Typically, JBuddy Server Gateways will be deployed on the same server as the JBuddy Server. Thus the server itself would need to be compromised in order for the communication between the server and gateway to be at risk. The JBuddy LDAP Gateway is provided as a separate installer and therefore could quite possibly be installed on another server, perhaps the server hosting LDAP or Active Directory. Since the JBuddy LDAP Gateway also communicates with the JBuddy Server via the Java RMI service, passwords are sent between these components. In JBuddy Server v3.3 and JBuddy LDAP Gateway v1.1, the passwords are protected even over non-TLS/SSL using military-grade encryption. Other communication such as usernames and buddy search terms are not encrypted. Under normal business environments, this should not pose a security concern.

JBuddy LDAP Gateway and Directory Service

Since the JBuddy LDAP Gateway makes a network connection to the LDAP or Active Directory Service, it’s important to understand the security implications and options available. JBuddy LDAP Gateway v1.1 added an import security update by inducing SASL DIGEST-MD5 support as a better security authentication mechanism. Most Active Directory and LDAP installations

support SASL DIGEST-MD5. Using this security authentication mechanism is strongly recommended over “simple” bind (clear-text password) authentication. If a “simple” bind security authentication mechanism is the only choice available for the LDAP directory used, it is strongly recommended that the directory be configured to support SSL/TLS connections to prevent passwords from being sent unencrypted over the network. To do otherwise would be reckless. The only exception to this would be if JBuddy LDAP Gateway were running on the same server as the Directory in which case the passwords would not be send over a network connection that may otherwise be compromised by a network sniffer. The security of the server hosting JBuddy LDAP Gateway and the Directory would be paramount. When JBuddy products are configured to support TLS/SSL network connections, Java (the underlying technology that JBuddy products are built on), expects a commercially issued security certificate where the Security Certificate Provider is listed in the Java Root Certificates security file. Unless a commercial SSL/TLS certificate is used on the LDAP or Active Directory Service, the java environment must be correctly configured to allow self-signed security certifcates. This is beyond the scope of this User’s Guide. A second, simpler option to secure communication between these services would be to install the JBuddy LDAP Gateway on the same server as the LDAP or Active Directory server. In this way, the communication would be secure as long as the server running the Directory Service itself was not compromised.

Installation

Choice of Installer

The JBuddy LDAP Gateway is available for installation in two forms:

- Within the JBuddy Server installer as an optional install pack
- As a separate JBuddy LDAP Gateway installer

In the first installer form, the JBuddy LDAP Gateway is installed on the same server as the JBuddy Server. In the second installer form, the JBuddy LDAP Gateway alone is installed. This is the form that should be used if you wish to install the JBuddy LDAP Gateway on a different server than the JBuddy Server such as on the same machine as the directory service. The JBuddy Server User’s Guide briefly describes installation in the first form. This guide will cover the second form of the installation in more detail. The Directory specific configuration fields used by both installer forms are the same.

System Requirements

Directory

The JBuddy LDAP Gateway authenticates against a Directory. As stated in the Terms section, ‘Directory’ means any LDAP v3 compatible directory including Microsoft Active Directory, Mac OS X Open Directory, or OpenLDAP service storing user account credentials in industry standard form and supporting an LDAP ‘simple bind’ using plaintext passwords.

Java

To run the JBuddy LDAP Gateway installation program as well as the gateway itself, you must have at least Java Standard Edition (JSE) version 1.4.2.x or newer installed. Note: JSE version 1.4.2.x and version 5.0.x have reached ‘end of service life‘ with support only available for Java for Business subscribers (paying support to Oracle). If you plan to enforce SSL login and

message encryption, you must have JSE 5.x or newer installed before running the server and gateways. **We have tested with Oracle's JSE. We have not validated with other versions of Java such as IBM JRE or the OpenJDK project.** The installer as well as the gateway are written entirely in Java and should therefore run on any modern, operating system supported by Oracle's JSE. That said, the startup and shutdown scripts and any operating system 'services' used to launch, monitor and shutdown the gateway utilize native library called Java Service Wrapper (JSW) version 3.2.0 to facilitate better native operating system integration such as with Windows Services. JSW version 3.2.0 was released with an open source license friendly to commercial software. This version includes 32-bit native wrappers for Windows x86, Linux x86 and PPC, and Solaris SPARC. Newer JSW versions and 64-bit versions are available directly from Tanuki Software http://wrapper.tanukisoftware.org/ however, Zion does not offer support for newer JSW versions at this time. If necessary, the JBuddy LDAP Gateway can be operated without the startup / shutdown scripts by passing the proper arguments directly to the JVM from a script, command shell or terminal. Instructions for custom JSW library compilation and non-JSW startup/shutdown are beyond the scope of this guide.

Installing Java

For best performance and scalability, we recommend using the full featured JSE 6.x or 7.x obtainable from Oracle at http://www.oracle.com/technetwork/java/javase/downloads/index.html. The JSE supports the -server flag which is the default and intended use of JBuddy LDAP Gateway. Further, you should select the 32-bit version of JSE if you are running on a 32-bit OS and the 64-bit JSE if you are running on a 64-bit OS. A 32-bit version of JSW should not prevent you from running 64-bit JSE.

Which Java

During installation, if the Installer encounters more than one Java environment, it will make a 'best guess' on which Java environment to use. It updates .conf files located in the conf directory. Near the top of these files it sets the property 'wrapper.java.command=$JAVA_HOME/bin/java' and then a little lower it updates the classpath wrapper.java.classpath.1= $JAVA_HOME/lib/tools.jar.' The $JAVA_HOME variable is replaced with the 'JAVA_HOME' that the Installer believes is the correct version. If you wish to use a different Java, you will need to edit the values above. The -server flag is enabled by default for performance reasons. The JRE available from www.java.com does not support the -server flag, therefore we recommend the JSE over the JRE. You can determine if your JVM environment supports the -server flag by typing the following in a command shell or terminal: `java –server –version` which should tell you the version of java used as well as if it is the server or client version of the virtual machine.

Command Line Installation

Initially you will need to run the installation on a computer with a graphical interface (see Launching the Graphical Installer below). At the end of the installation, you will be prompted if you wish to save the installation as a XML installation script which can be used later for an automated (non-graphical) installation such as on a remote Linux or unix-based server. If this is your situation, you need to enter information applicable to the remote host when prompted during the Installer. To run a headless installation, copy the installer jar file and xml install script (that you chose to save at the end of the GUI install) to the remote host. Then login to the remote host and from a command shell or terminal, enter the following command:

`java –jar JBuddyLdapGatewayInstaller–1.1.xxxxxx.jar JBuddyLdapGatewayInstallScript.xml` where the JBuddyLdapGatewayInstallScript.xml is whatever name you saved at the end of the GUI install.

Graphical Installation

After a modern JSE is properly installed, double click on the JBuddyLdapGatewayInstaller-1.1.xxxxxx.jar file to launch the installation program. If you prefer, you can launch the Installer by simply entering the following from a command shell or terminal: `java -jar JBuddyLdapGatewayInstaller-1.1.xxxxxx.jar`. Once launched you should see a small Language Selection dialog appear similar to Figure 2 below:

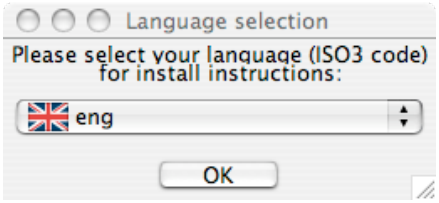

Figure 2: Installer Language Selection Dialog

After selecting eng (English) language and clicking OK, the Installation Welcome window appears similar to Figure 3:

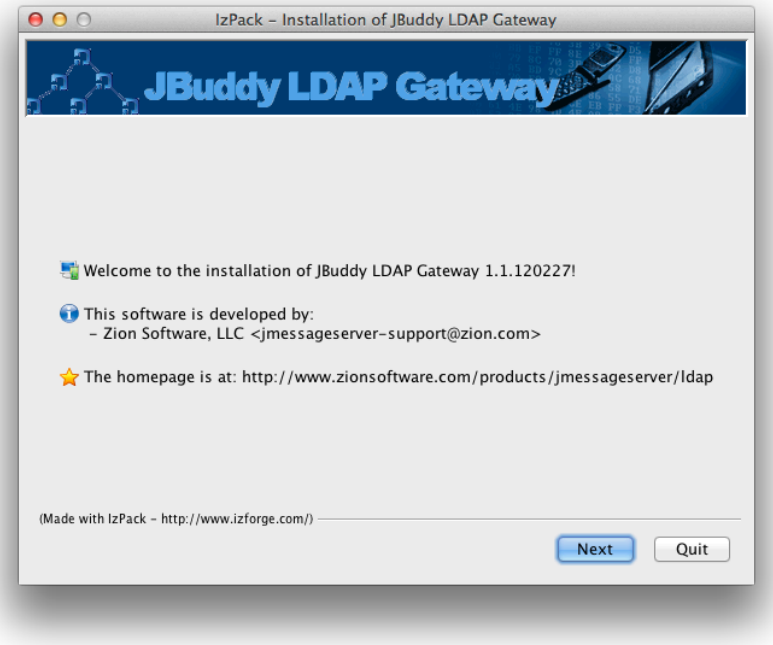

Figure 3: Installation Welcome

After clicking Next, the Installation Information window appears similar to Figure 4:

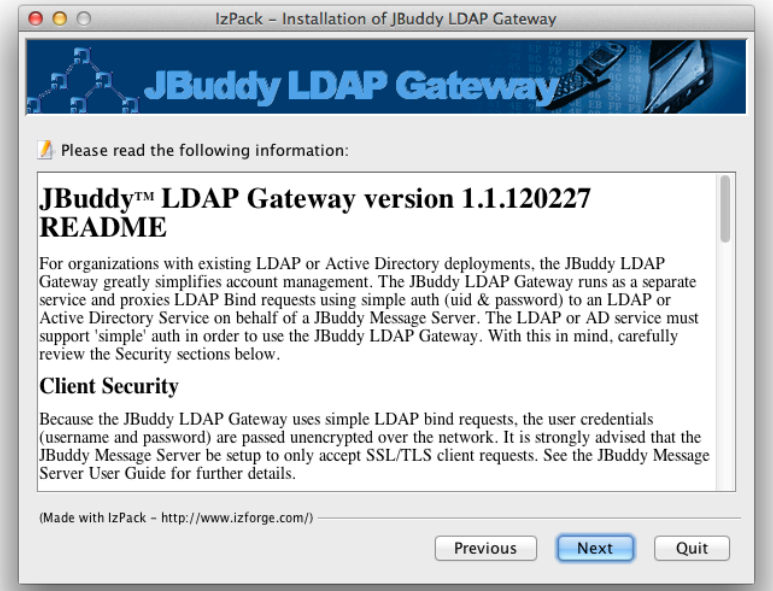

Figure 4: Installation Information

After clicking Next, the License Agreement window appears similar to Figure 5. You must accept the terms of the license agreement before the Next button will be enabled.

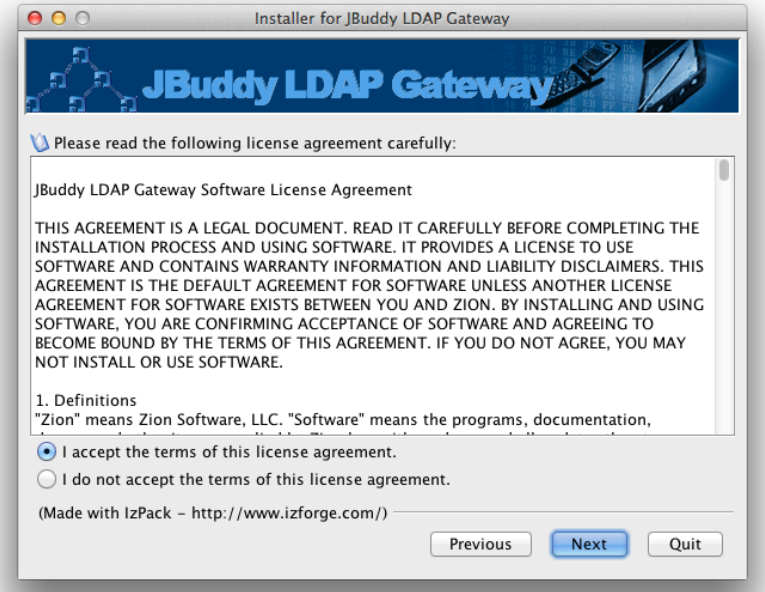

Figure 5: Installation License Agreement

After accepting the license agreement terms and clicking Next, the installation path must be specified as in Figure 6 below. Once Next is chosen a Message dialog appears to inform you that the directory will be created (or if it already exists.)

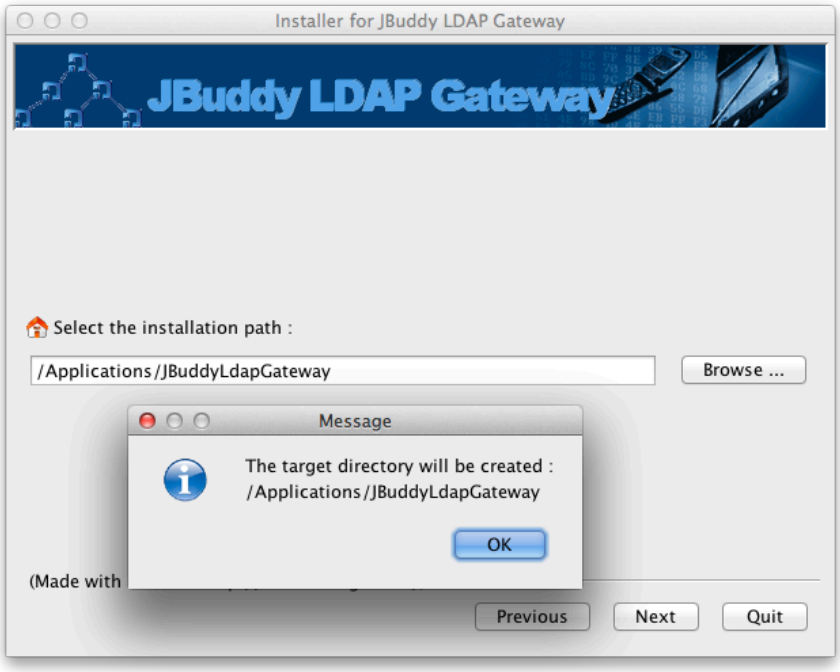

Figure 6: Installation Location

After clicking OK to the Message dialog, the Package Selection window will appear similar to Figure 7:

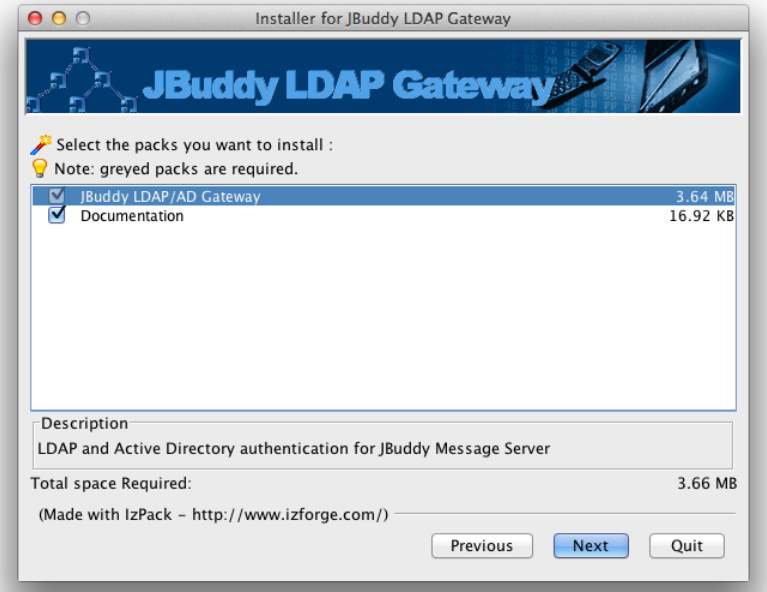

Figure 7: Installation Package Selection

The Package Selection window shows mandatory and optional “packs” that can be installed. Clicking an individual pack changes the description shown. Once you are satisfied with your packs selection, click Next to continue.

The Hosts Configuration window will appear similar to Figure 8 below. JBuddy Server Host is the hostname, IP address, or fully qualified domain name of the machine where JBuddy Server is available for the JBuddy LDAP Gateway to connect. The LDAP Service URI should be left blank if the LDAP DNS SRV records are available (often true for Active Directory). Otherwise, provide the host, IP address or a full uri to reach the directory service.

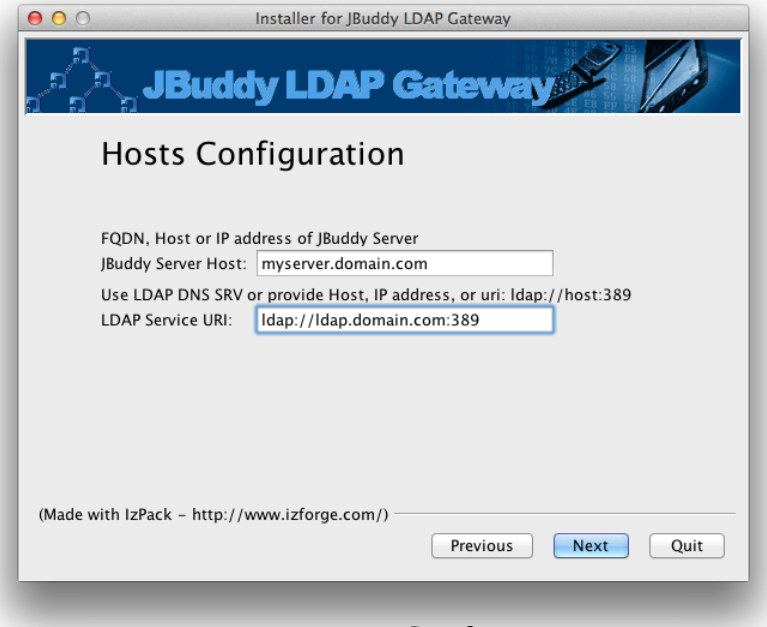

Figure 8: Hosts Configuration

After clicking Next, the Active Directory / LDAP Auth Configuration window will appear similar to Figure 9A below. This window supports basic configuration for both Active Directory and LDAP.

Active Directory Configuration

To configure Active Directory, begin by specifying the Active Directory Domain (first field). If all AD users are permitted JBuddy Server users (ie: no org unit or security group restrictions), then specify the LDAP Auth Base (last field) as the LDAP DN form of your domain. ie: dc=domain, dc=com - both shown in Figure 9A. Then skip the next screen (AD / LDAP Search Configuration shown in Figure 9B). You're done!

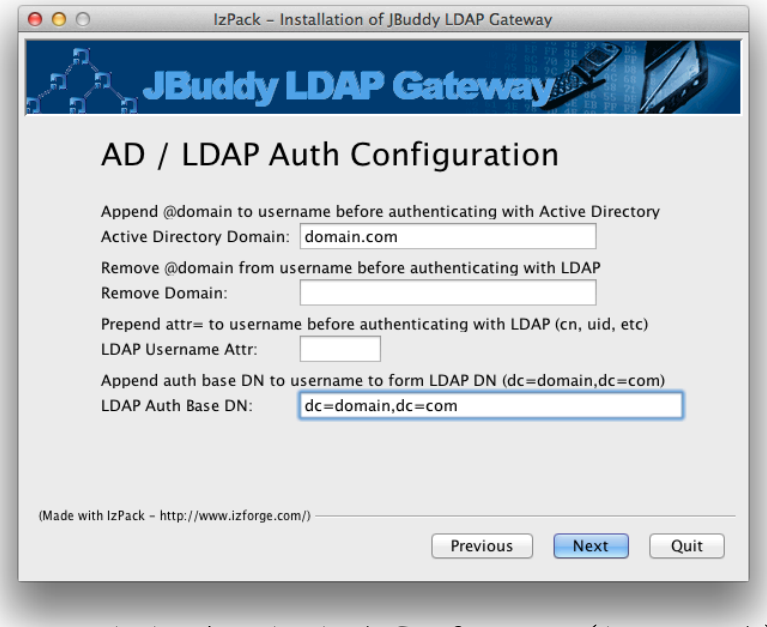

Figure 9A: AD / LDAP Auth Configuration (AD example)

If the permitted JBuddy Server users are a subset of all AD users, you will need to specify additional fields on the AD / LDAP Search Configuration screen (Figure 9B). If you only need to restrict users to an organization unit, then specify the Search Base DN. If you only need to restrict users to a specific security group, then specify the Search Group, and Search Group Base DN (most likely the domain in LDAP DN form. ie: dc=domain,dc=com). If you need to restrict users by both, then fill in all three fields. The Search Group Attribute is not used by AD configuration.

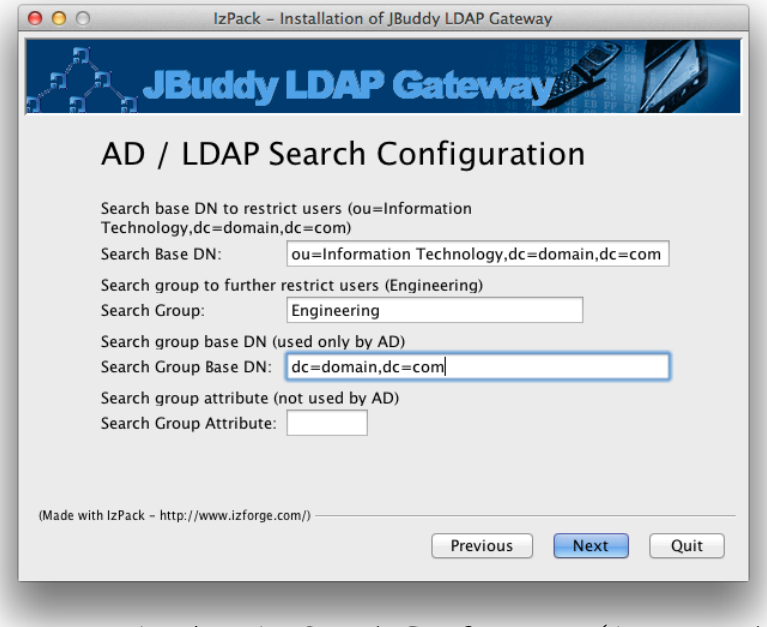

Figure 9B: AD / LDAP Search Configuration (AD example)

LDAP Configuration

To configure for LDAP or Mac OS X Open Directory, the username does not typically contain @domain.com, however in order to leverage the new DNS SRV support allowing the JBuddy client to find the JBuddy Server without troubling the user to provide the host and port, a domain must be appended to the username at the client. In order to successfully authenticate with an LDAP directory, this domain must usually be removed from the username prior to attempting to authenticate with LDAP. Specify the domain portion in the second field. See Figure 10A below. Next, the third and fourth fields on this configuration window also pertain to LDAP configurations. The third field allows the administrator to define the attribute that is prepend to the username prior to authenticating with LDAP. It may be left blank and will default to “uid” as is common for most LDAP servers. The fourth field allows the administrator to specify the remaining portion of the LDAP authentication DN (the auth base). An example is given below illustrating how the second, third and fourth fields combine to modify the username prior to authenticating with the LDAP directory: username = *jack.frost@domain.com*

Remove Domain: *domain.com*

LDAP Username Prefix: *uid*

LDAP Auth Base: *dc=domain,dc=com*

username sent to LDAP: *uid=jack.frost,dc=domain,dc=com*

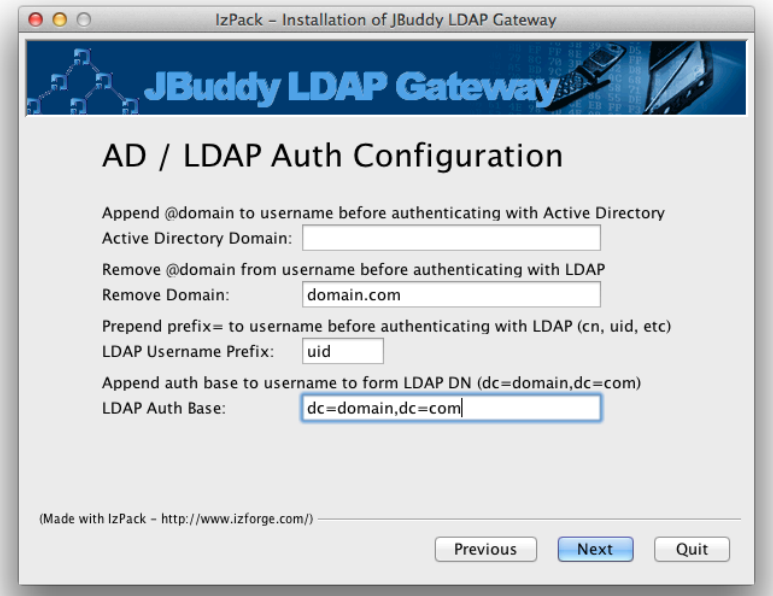

Figure 10A: Active Directory / LDAP Auth Configuration (LDAP example)

If the permitted JBuddy Server users are a subset of all LDAP users, you will need to specify additional fields on the AD / LDAP Search Configuration screen (Figure 10B). If you only need to restrict users to an organization unit, then specify the Search Base DN. If you only need to restrict users to a specific security group, then specify the Search Group, and Search Group Attribute (for Mac OS X Open Directory this is likely “cn”, for other organizations using LDAP, this will be any LDAP field where the restricted group membership is stored. The Search Group Base DN is only used by AD configurations. If you need to restrict users by both, then fill in all three fields. See Figure 10B below.

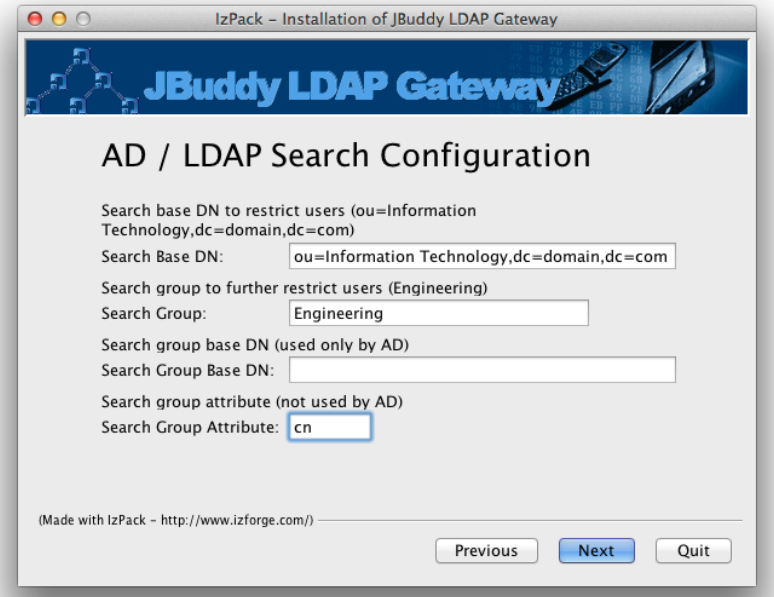

Figure 10B: AD / LDAP Search Configuration (LDAP example)

After completing the AD / LDAP Search Configuration, click the Next button to begin the software installation. See Figure 11.

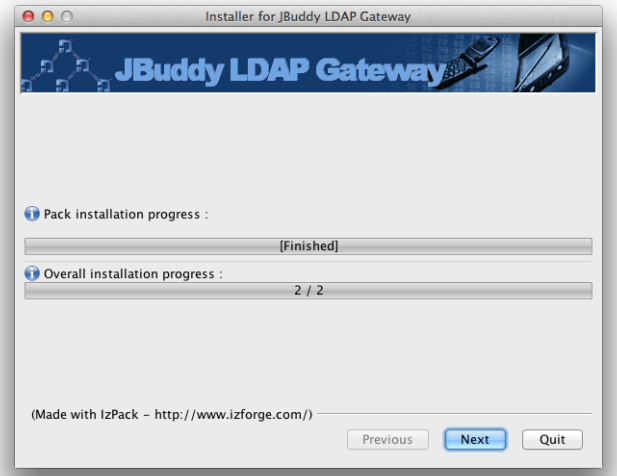

Figure 11: Installation window

After the installation window is finished, click the Next button to proceed to the Post-Processing window. On non-Windows installation, the Post-Processing window will appear but no post-processing needs to be done (see Figure 12.) On Windows installations, the installer attempts to register JBuddy LDAP Gateway as a Windows Service. The Post-Processing window will display log messages as the post-procesing proceeds (successful or errors). Click the Next button to proceed to the final installer window (see Figure 13.)

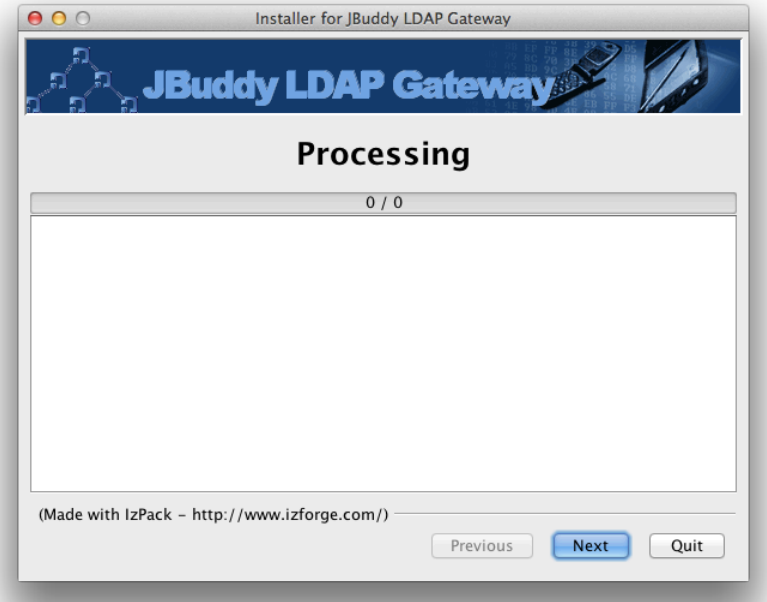

Figure 12: Post-Processing Window (Mac OS X Example)

On this final screen of the graphical installer, you may save the installation script for use with a headless install, described in the Command Line Installation section above or click Done to close the Installer.

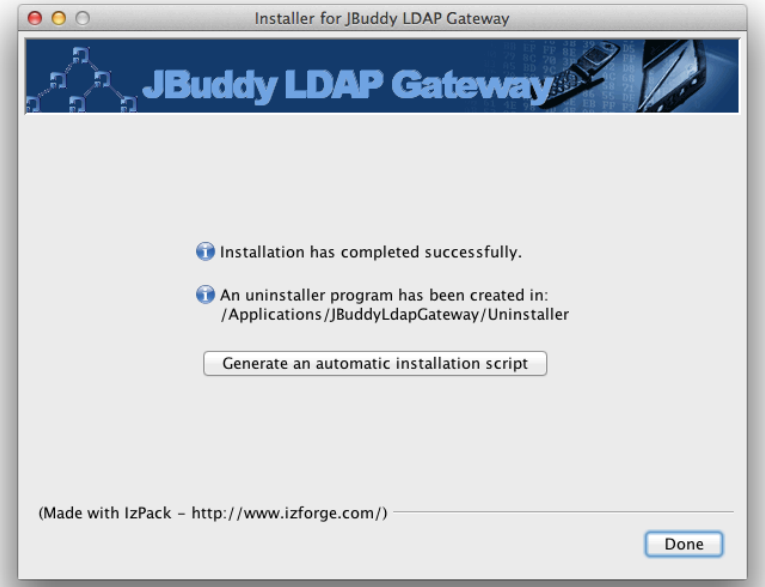

Figure 13: Final Installer window

Additional Settings

Which Directory?

On startup, the JBuddy LDAP Gateway loads the lib/LdapGateway.properties file. There are three properties that help determine what type of Directory is being used.

- com.zion.messaging.ldap.LdapGatewayImpl.DIRECTORY_DOMAIN=*domain.com*
- com.zion.messaging.ldap.LdapGatewayImpl.REMOVE_DOMAIN=*domain.com*
- com.zion.messaging.ldap.LdapGatewayImpl.LDAP_SERVER_OSX=*true or false*

If DIRECTORY_DOMAIN=domain.com, then an Active Directory is assumed. If REMOVE_DOMAIN=domain.com, then an LDAP directory is assumed. If LDAP_SERVER_OSX=true, then Mac OS X Open Directory is determined (the REMOVE_DOMAIN=domain.dom should also be set for Mac OS X Open Directory).

There are several other settings configured during installation and many more which were not configured during installation but will use default values unless uncommented and assigned a value in the LdapGateway.properties file.

Security Authentication

The property *java.naming.security.authentication=* is used to determine which security authentication mode is used to authenticate with the Directory. Unless this property is uncommented the default behavior will be used.

Active Directory Authentication

The default will be SASL DIGEST-MD5. It can be explicately set by uncommented this property and assigning it the value DIGEST-MD5: *java.naming.security.authentication=DIGEST-MD5*.

LDAP Authentication

By default, JBuddy LDAP Gateway will use “simple” bind (clear-text) authentication mode when configured in support of other LDAP (non Active Directory) directories. The security authentication property for “simple” would be: *java.naming.security.authentication=simple.* An SSL/TLS connection to the LDAP server is strongly recommened if “simple” bind is used.

Mac OS X Open Directory

In order to support Mac OS X Open Directory for restricted authentication and restricted add buddy searchs, a special configuration setting must be manually changed. First, stop the JBuddy LDAP Gateway service if it is running, carefully open the lib/LdapGateway.properties file within the JBuddy LDAP Gateway or JBuddy Server installation folder using a text editor. Change the following property from false to true:

```
com.zion.messaging.ldap.LdapGatewayImpl.LDAP_SERVER_OSX=true
```

Finally, save the file, and restart the JBuddy LDAP Gateway service.

Other Settings

In some environments, additional settings, such as limiting search results, must be made, to allow JBuddy LDAP Gateway to operate optimally. These can be made by stopping the JBuddy LDAP Gateway service if it is running, carefully editing the lib/LdapGateway.properties file within the JBuddy LDAP Gateway installation folder, saving the changes, and restarting the JBuddy LDAP Gateway service.

Authentication

Just to recap, JBuddy LDAP Gateway v1.0, JBuddy Server v3.2, and JBuddy Messenger v3.2.111216 introduced authentication against a Directory. An LDAP “simple bind” was used to authenticate a user. JBuddy LDAP Gateway v1.1, JBuddy Server v3.3 and JBuddy Messenger v3.3 introduce military-grade encryption between JBuddy components and also add support for SASL DIGEST-MD5 security authentication with Directories.

Active Directory

For Active Directory, the “userPrincipalName” attribute is expected to contain the user’s fully qualified login name: first.last@domain.com.

LDAP and Mac OS X Open Directory

For LDAP, a fully qualified LDAP DN is required. JBuddy LDAP Gateway transforms the username into this LDAP DN based on the configuration values provided during installation. An LDAP example best illustrates the transformation:

If the JBuddy Messenger user authenticated using the username: name@domain.com and the REMOVE_DOMAIN=domain.com, LDAP_AUTH_BASE=cn=Users,dc=domain,dc=com, and LDAP_USERNAME_ATTR=uid, then the username would be transformed into the LDAP DN: uid=name,cn=Users,dc=domain,dc=com

Restricting Access to JBuddy Server

New for JBuddy LDAP Gateway v1.1, and JBuddy Server v3.3, is the ability to restrict authenticated users by a search base and/or security groups within the Directory.

Active Directory

Active Directory stores group membership using memberOf LDAP attribute and thus, in a single ldap search, we can determine if a user is in the restricted organizational unit and security group.

LDAP

Unfortunately, there does not seem to be a standard way of storing security group membership for LDAP. Many organizations seem to create a new LDAP attribute and store organization-specific values therein. In order to support this, JBuddy LDAP Gateway provides some flexibility. The LDAP_SEARCH_GROUP_ATTR value may contain any LDAP attribute where a grouping may be organized. For example an attribute “role” may hold values such as “student”, “faculty”, “staff”. To restrict JBuddy Server users to only users with a role of “faculty” or “staff”, specify LDAP_SEARCH_GROUP_ATTR=role and LDAP_SEARCH_GROUP=faculty,staff.

Mac OS X Open Directory

Open Directory stores group membership differently than Active Directory. A group stores it’s members under the memberUid LDAP attribute. In order to authenticate a user against Mac OS X Open Directory, the JBuddy LDAP Gateway performs two or three LDAP operations. The first is the simple bind. The second is a basic user search to get the first and last name (for setting this user’s “display name” within JBuddy Messenger buddy lists. If authorized users are a subset of all Open Directory users, JBuddy LDAP Gateway performs a second search to lookup the user’s group membership (which is stored in a group schema. Finally, it verifies the user’s

membership against the list of members in the security group. The users is authorized if this condition is met.

Other Restrictions

It should be noted that the same organizational unit and/or security group restrictions apply to the new Add Buddy Search feature as well. It makes sense that you should not be able to search and fine a buddy who is not a permitted user of JBuddy Server.

Add Buddy Search

New in JBuddy LDAP Gateway v1.1, JBuddy Server v3.3, and JBuddy Messenger v3.3 is the ability for an end user to add “buddies” to their buddy list by performing a search against the Directory. The user can enter one or two search words separated by a space.

LDAP Attributes

The JBuddy LDAP Gateway will perform a search using the LDAP Attributes “givenName” and “sn”, (first name and last name respectively). If the search is two words, the first word will match against first name and the second will match against last name. The two word search is an “AND” search and both words must match. If the search is only one word, then the search is an “OR” search. The single word will be used to match against first name OR last name.

User Supplied Wildcard Search

The user can also include wildcard(s) within the search. The wildcard character is the asterisk (‘*’). If one or more wildcard characters are present in the search, no further modification to the search will take place.

System Supplied Wildcard Search

If the user does not include any wildcard characters in the search, the following modifications will be made by the JBuddy LDAP Gateway in order to help the user find name matches in the directory:

- JBuddy LDAP Gateway will include an “OR” search against the ldap attribute storing the unique user id (as specified in lib/LdapGateway.properties as com.zion.messaging.ldap.LdapGatewayImpl.LDAP_USERNAME_ATTR in order to attempt to match on an exact user. For Active Directory, this is expected to be “userPrincipalName”. For ldap, this is expected to be “uid”: (|(uid=name)...
- JBuddy LDAP Gateway will append a wildcard to the search word(s). If only one search word is used, then the search becomes an “OR” search: (|(givenName=name*)(sn=name*)). If two search words are used, then the search becomes an “AND” search: (&(givenName=firstWord*)(sn=secondWord*)).