Version 3.1

USER'S GUIDE

CONTENTS

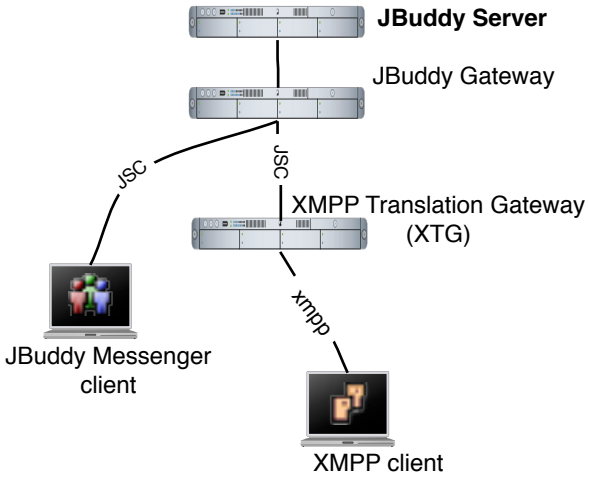
WHAT'S NEW

XMPP Client Support

JBuddy Message Server version 3.1 adds an optional special connectivity option called the XMPP Translation Gateway or XTG (see Figure below). XTG adds connectivity support for XMPP clients to JBuddy Message Server. This means that in addition to the cross-platform and very capable JBuddy Messenger client, JBuddy Message Server version 3.1 allows XMPP clients to connect and send and receive messages with other connected clients. This means users are no longer limited to using JBuddy Messenger as a desktop IM client with JBuddy Message Server. There are many different XMPP clients available to use.

At this time, sending and receiving files and multi-user chat is not supported for XMPP clients, nor is federated messaging with other XMMPP servers. These features may be supported in future releases.

XMPP Translation Gateway



INTRODUCTION

Target Audience

This JBuddy Message Server Users Guide is intended primarily for two kinds of readers. The first group consists of IT staff who will be supporting and maintaining the JBuddy Message Server environment in a “system administration” capacity where central control of all IM activity is desired. The second group includes people who have been given authorization to administer their own accounts.

What This GUIDE Covers

We provide a brief history of the Instant Messaging (IM) industry, describe IM concepts important to the use of features in the JBuddy Message Server. We cover installation of a new server, configuring accounts, and using the server in several common scenarios. We cover the optional Instant Help system which uses JBuddy Message Server for a versatile live click-to-chat web help desk solution with help agents using IM.

Instant Messaging History and Concepts

As far back as the dawn of computer networks, people have desired to communicate with each other. Back in the early 1980’s a unix program called talk was available on nearly all unix based computers. Talk allowed two users to exchange text messages with each other over the network (MESSAGING). In order to determine if another user was online, a second unix program was often used called rwho. Rwho would gather who else was logged into a unix computer on the LAN. After determining that the user you were interested to talk with was logged in to a remote computer (and presumably sitting at the computer) (PRESENCE), talk would be used to send an invitation to the remote user and invite them to chat. This invitation was often a short beep and a invitation message within the remote users’s terminal (this was before GUIs were available). If the remote user agreed to accept the invitation to talk, his terminal would be divided top and bottom with his text appearing in one half of the terminal and the remote user’s text in the other half of the terminal. Thus, instant messaging in it’s most primitive form was born. By 1988 a program called Internet Relay Chat (IRC) was available which allowed multiple users connected to the same IRC server to carry on a GROUP CHAT.

By the early 1990’s cell phones were growing in popularity in Europe and short text messages (aka SMS) were being passed between thousands of cell phone users. SMS still lacks presence, however in Europe, your cell phone is rarely off during the day and the SMS would be delivered to your cell phone as soon as it was turned on and an indication would appear on the phone’s display and possibly flash a light, vibrate, or emit a audible beep.

SMS capable cell phone

By the mid to late 1990’s, the land rush of the internet dot-com era was underway and several large internet service providers (ISPs) were running free public IM services each of which could not communicate with the other IM services. The four largest public IM services at that time

were ICQ (I seek you), AOL Instant Messaging service (AIM), MSN Messenger Service , and Yahoo Messenger .


AIM


ICQ


MSN


Yahoo

By 2003, worldwide adoption of IM continued to grow rapidly with over four hundred million users reportedly using at least one IM service. With corporate scandals and huge financial disasters like Enron and WorldCom surfacing, the legislative branch of government in the United States passed regulatory laws mandating message retention, retrieval and in the case of health information, encryption - not only covering E-Mail, but IM! By 2004 SPIM was reported (unsolicated junk IM) and viruses and worms began to surface, largely using automated IM agents (BOTs) to deliver the message or an infected file. Enterprises realize that with regulatory compliance regulation deadlines looming and the aforementioned risk factors to business, that unmanaged IM is cause for great concern. By 2004, enterprises using IM have begun installing IM firewalls, and IM management software in an effort to lock down and bring IM under central control, security and compliance. Numbered are the days of unauthorized IM use within the enterprise. By 2005, IM is on the top 10 list of action-items within corporate IT departments. Some corporations have even attempted to shutdown any use of IM. By 2006 however, the inevitable trend toward embracing and adopting enterprise IM company-wide with carefully managed access to public IM is underway, which is one of the core features of JBuddy Message Server and described in this Users Guide. As of 2008, we are beginning to see more activity, especially in financial firms for non-human buddies known as IM Bots. Bots can be useful for sending users real-time alerts and allowing them to configure their own alerts via interactive menus. This guide will not cover IM Bots which is covered in depth by Zion's JBuddy Developer Tools products available online at http://www.zionsoftware.com/products/jbuddy/

SERVER ARCHITECTURE

This section will describe JBuddy Message Server architecture and will help you make the most of the platform. In releases prior to version 3.0, JBuddy Message Server supported a simple IM architecture adequate for secure internal IM (EIM). Connectivity to other IM networks was available only through the multi-network enterprise IM client, JBuddy Messenger as shown in Figure 1-1 below.

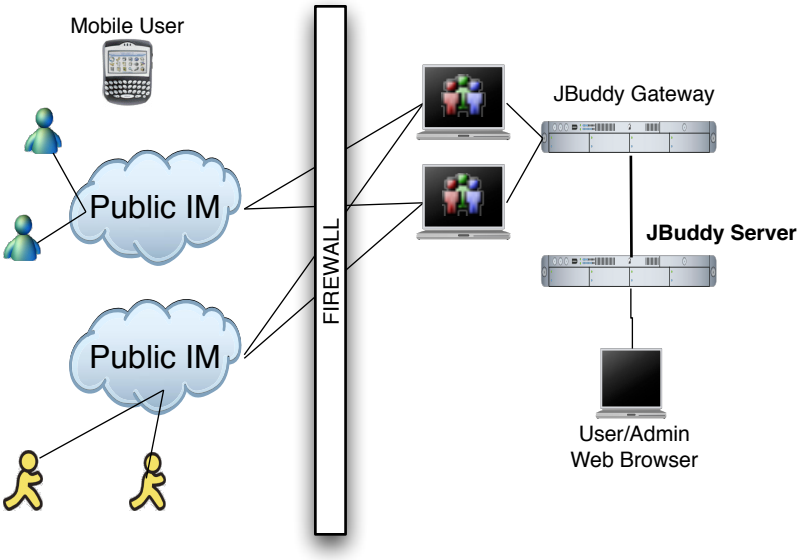


Figure 1-1: JBuddy Message Server Architecture (prior to 3.0)

Beginning with version 3.0, JBuddy Message Server offers substantial benefits to organizations with more complex deployment requirements including IM management and control, operations, user features, mobile uses, and development and system integration.

In Figure 1-2 below, you can see the current JBuddy Message Server 3.x architecture. As in prior releases, it shows how all of the server components remain behind the corporate firewall while communicating with the outside world through the firewall. Beginning in version 3.0, JBuddy Message Server introduced the ability to centrally manage and log all external communications, as well as route both public IM and enterprise IM to specific users including mobile E-Mail users. Version 3.0 added support for an optional message gateway called Instant Help described in a subsequent chapter. Version 3.1 introduces connectivity for XMPP clients through the optional XMPP Translation Gateway (XTG), allowing choice in IM clients used with the JBuddy Message Server platform. By centrally managing external connectivity, the enterprise achieves greater control and reduces potential security threats and misuse while at the same time, bringing all users including mobile users under message archival for regulatory compliance. Connectivity to public IM can still be achieved directly through the JBuddy Messenger IM client if external file transfers are allowed at your organization.

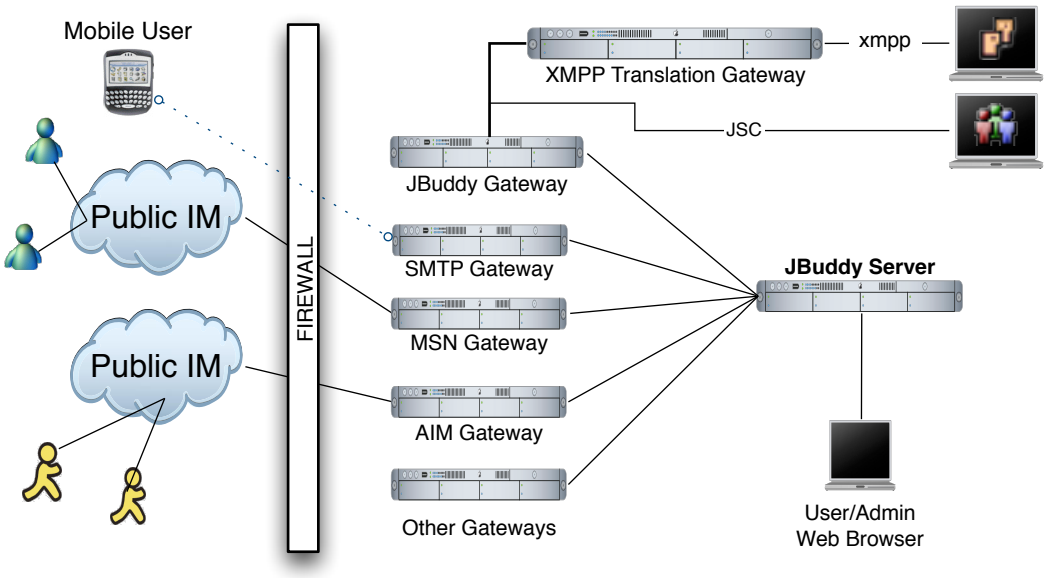


Figure 1-2: JBuddy Message Server Architecture version 3.1

Public IM is represented on the left side of Figure 1-2 by the MSN and AIM icons connected to their respective Public IM network clouds. A Blackberry icon represents an enterprise mobile user with an E-Mail enabled wireless device. This mobile user may or may not play a role in a JBuddy Message Server deployment depending on server configuration. On the right side of Figure 1-2 (inside the corporate firewall), several components which make up the JBuddy Message Server architecture are shown. The JBuddy Message Server consists of several components or services including an authorization / load balancing server, and multiple message gateways.

Presence Services & The Database

The Database server was omitted from Figure 1-2 but plays a critical role in the overall server architecture by persisting IM account data, routing rules, buddy lists, privacy settings, message archival for regulatory compliance purposes, and buddy presence. The presence stored in the database can form the foundation for other useful applications such as the display of presence information to a live web page or with email as a presence enabled URL as depicted in Figure 1-3 below. These applications are made possibly by leveraging the live presence updated in the database and using the Presence Service.

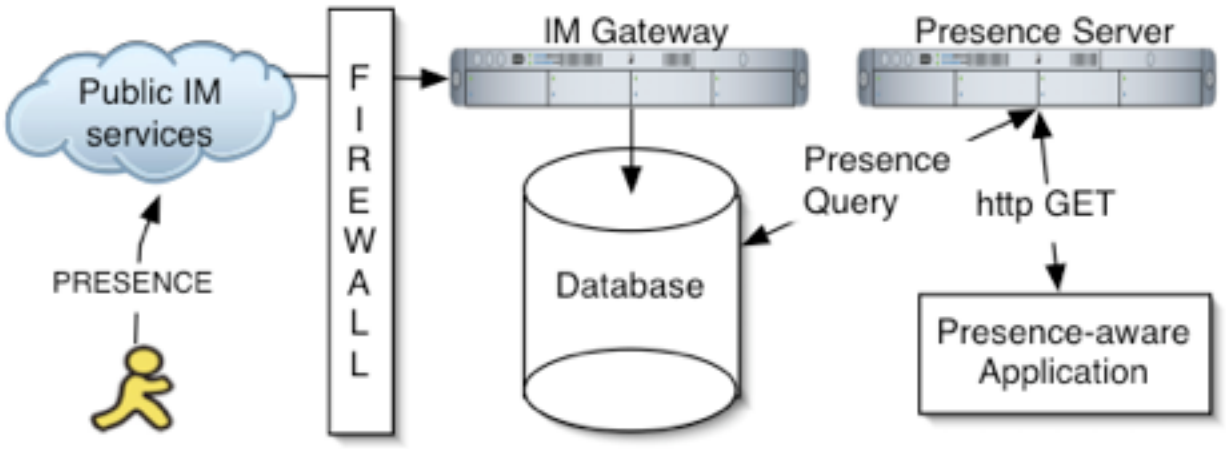


Figure 1-3: Leveraging Presence in Applications

External connectivity management and control is one of the major enhancements to the server in the 3.x architecture. To facilitate central management and mobile messaging, a message routing engine along with server-side public IM gateways, enterprise IM gateways, and other gateways including an SMTP message gateway are available as configuration options. To facilitate web based live support, version 3.0 also introduced a sophisticated Instant Help message gateway shown in Figure 1-4 below.

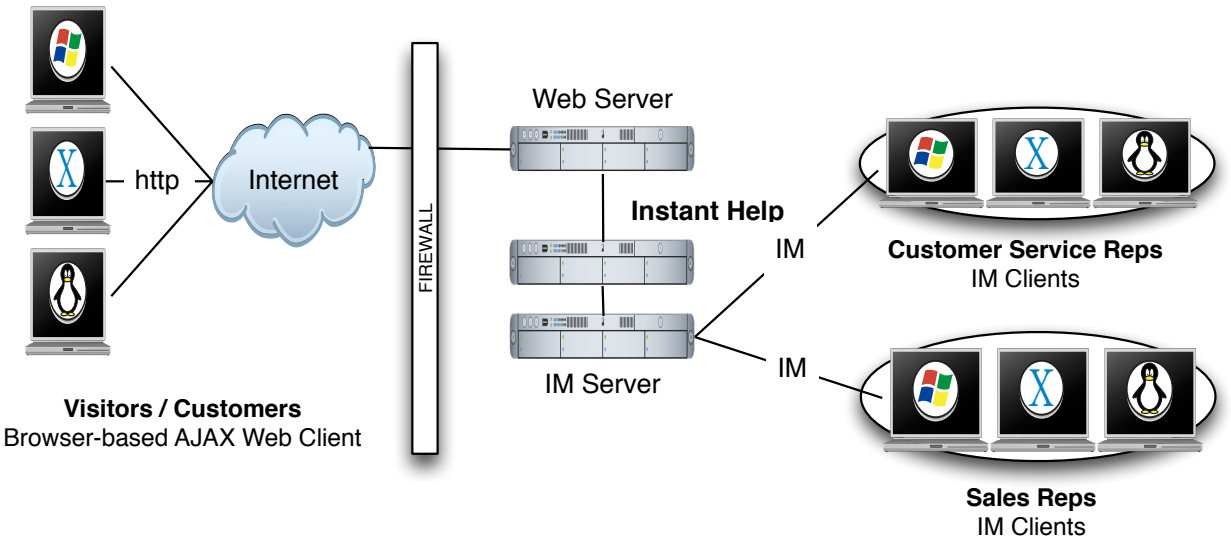


Figure 1-4: Instant Help for web-based live-chat support with IM

INSTALLATION

System Requirements

In order to run the JBuddy Message Server installation program as well as the server itself, you must have a Java Virtual Machine (JVM) version 1.4.x or newer installed. By JVM, we are referring to either Java Runtime Environment (JRE) or Java 2 Standard Edition (J2SE) 1.4.x or newer. If you plan to enforce SSL login and message encryption, you must have JVM version 5.x or newer installed prior to running the server. The JBuddy Message Server installation program as well as JBuddy Message Server itself are written entirely in Java and should therefore run on any JVM 1.4.x or JVM 5.x enabled modern operating system. That said, the server startup and shutdown scripts and any operating system “services” used to run the JBuddy Message Server utilize a free, open source, native library called Java Service Wrapper (JSW) to facilitate better native operating system integration such as with Windows Services. If you plan to operate the JBuddy Message Server software on an unsupported operating system, you may be able to compile a JSW library appropriate for your platfrom by obtaining the source from http://wrapper.tanukisoftware.org/ . If necessary, the JBuddy Message Server 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

You may use JRE 1.4.x or 5.x obtainable by from http://www.java.com/ by following the download instructions for basic demo or small deployment scenarios, however for best performance and scalability, we recommend using the full featured J2SE 5.x obtainable from http://java.sun.com/j2se/ by follow the download instructions. The J2SE edition supports the -server flag which is the intended use of JBuddy Message Server. You may also experimentally try another JVM implementation from vendors such as IBM, Apache, and Blackdown.

Which Java

During installation, if the Installer encounters more than one JVM, 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 JVM, you will need to edit the values above. Also if you desire to use the -server flag for performance reasons, you will need to use a Java version that includes support for this feature. On Windows and OS X, the typical JRE 1.4.x does not, but on Linux it does. You can determine if your JVM environment supports this 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 an XML installation script which can be used later for an

automated (non-graphical) installation such as on a remote Linux or Unix server. If this is your situation, you need to enter information applicable to the remote host when prompted during the Installer. To run the headless installation, copy the installer jar file and xml install script saved 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 as follows:

java -jar JBuddyMessageServerInstaller-3.x.xxxx.jar JBuddyServerInstallScript.xml where the JBuddyServerInstallScript.xml is whatever name you saved at the end of the GUI install.

Graphical Installation

After JVM 1.4.x or 5.x is properly installed, double click on the JBuddyMessageServerInstaller-3.x.xxxx.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 JBuddyMessageServerInstaller-3.x.xxxx.jar. Once launched you should see a small Language Selection dialog appear similar to Figure 2-1 below:

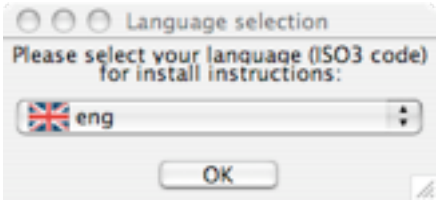


Figure 2-1: Installer Language Selection Dialog

After selecting eng (English) language and clicking OK, the Installation Welcome window appears similar to Figure 2-2:

Figure 2-2: Installation Welcome

After clicking Next, the Installation Information window appears similar to Figure 2-3:

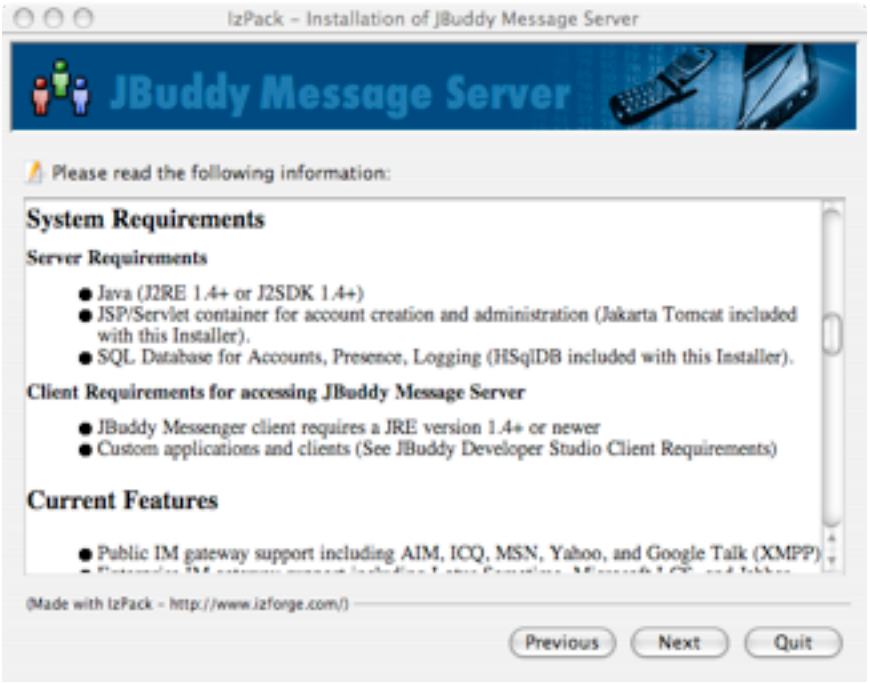


Figure 2-3: Installation Information

After clicking Next, the License Agreement window appears similar to Figure 2-4. You must accept the terms of the license agreement before the Next button will permit proceeding to the next screen.

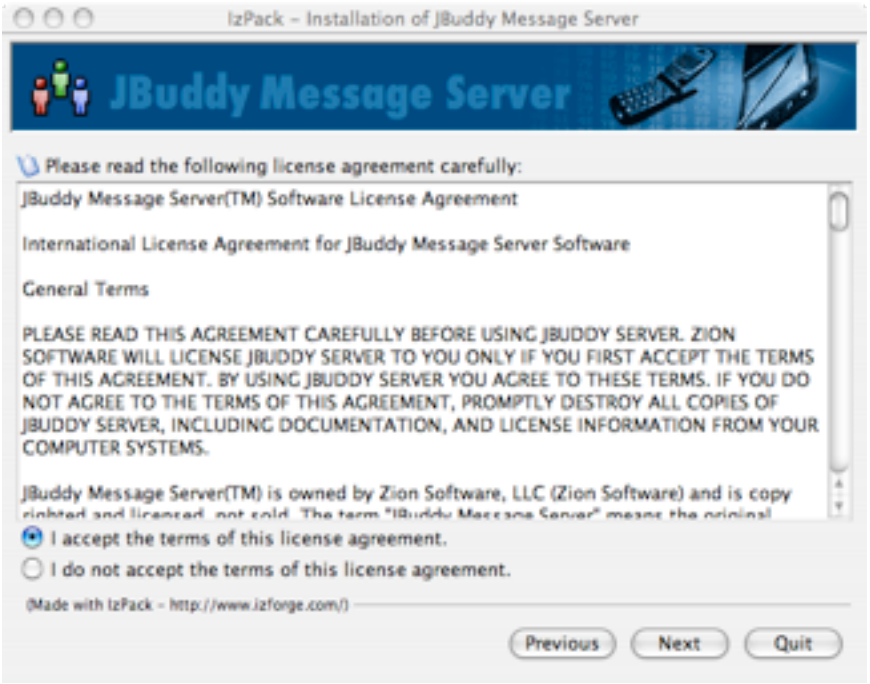


Figure 2-4: Installation License Agreement

After accepting the license agreement terms and clicking Next, the installation path must be specified as in Figure 2-5 below. Once Next is chosen an Message dialog appears to inform you that the directory will be created (or if it already exists).

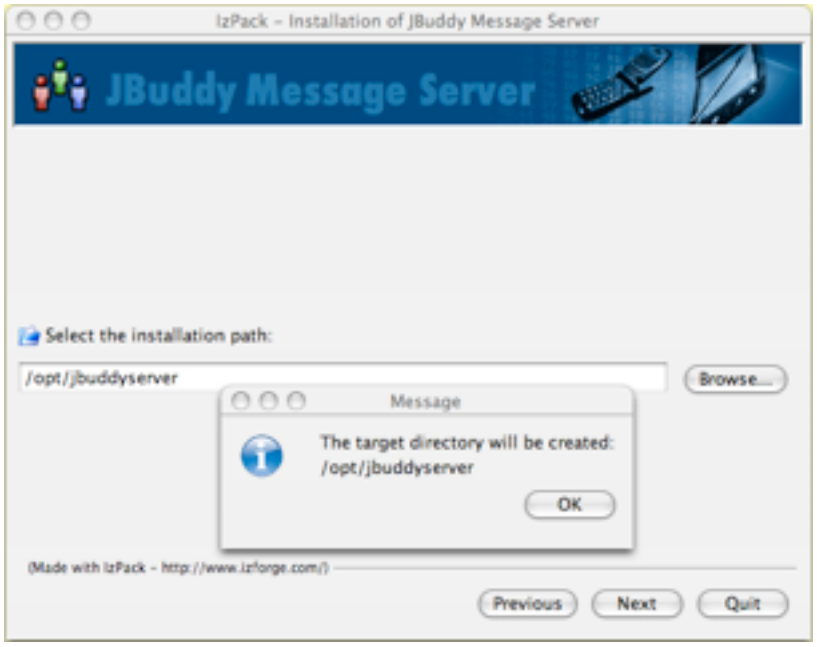


Figure 2-5: Installation Location

After clicking OK to the Message dialog, the Package Selection window will appear similar to Figure 2-6 below:

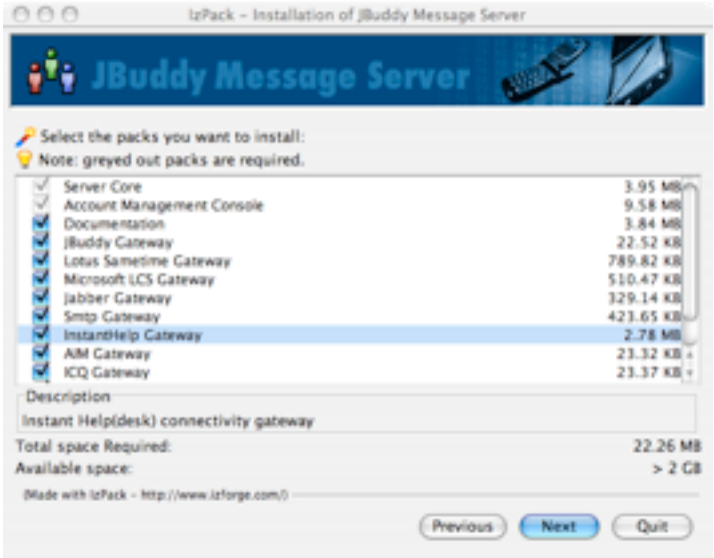


Figure 2-6: Installation Package Selection

The Package Selection window shows mandatory and optional “packs” that can be installed. Clicking a individual pack changes the description shown. For example, if you don’t have plans to leverage Lotus Sametime or ICQ, you can uncheck the pack(s) and it will not get installed. Just because you install a particular pack doesn’t mean you have the proper license to enable

that feature. An improvement of the Installer would only enable the products packs which you hold a valid license to use. Once you are satisfied with your packs selection, click Next to proceed.

After clicking Next, the Installation Configuration window will appear similar to Figure 2-7 below. Server Hostname is the IP address or fully qualified domain name of the machine where JBuddy Message Server will operate once installed. If it will ultimately live on a remote headless machine, enter that host name or IP here.

Figure 2-7: Installation Configuration (part 1)

The Sys Admin UserId is a login userid for administering the JBuddy Message Server. It is not related to operating system authorization credentials. Confirm the chosen password by reentering it in the Confirm Password input field. The Database Secret field is also a password field of sorts. It is used to encrypt all user account passwords with 192 bit Triple DES encryption into the database. It should be noted that although the account passwords are very securely encrypted, the Database Secret must be accessible by the server at runtime to unencrypt the passwords on demand and therefore, utmost care should be taken to protect the JBuddy Message Server configuration files from unauthorized access or view.

After clicking Next, the second Installation Configuration window will appear similar to Figure 2-8 below. E-Mail From is the email address used for emails sent from the JBuddy Message Server such as for lost passwords. The Smtp Host is the fully qualified hostname of the SMTP mail server that JBuddy Message Server will use to deliver the email. If the SMTP mail server requires authorization to login before sending email, the Smtp User and Smtp Password should be entered. If not, leave them blank.

Figure 2-8: Installation Configuration (part 2)

Figure 2-9A: SMTP Configuration

After clicking Next, any optional Installation Configuration windows will appear similar to Figure 2-9 A-D if you selected the SMTP Gateway, Lotus Sametime, Microsoft LCS or Jabber packs respectively during the Installation Package Selection window (Figure 2-6).

In Figure 2-9A, the current Installer prompts for an IMAP Host, IMAP User and Password, however the configuration file for the SmtpMsgGateway.properties file may be configured for POP email retrieval by changing the correct properties in the future.

If you selected any other enterprise IM packs to be installed such as Lotus Sametime or Microsoft LCS or Jabber, you will be prompted to provide the hostname of the EIM server similiar to Figure 2-9B, 2-9C, 2-9D) below.

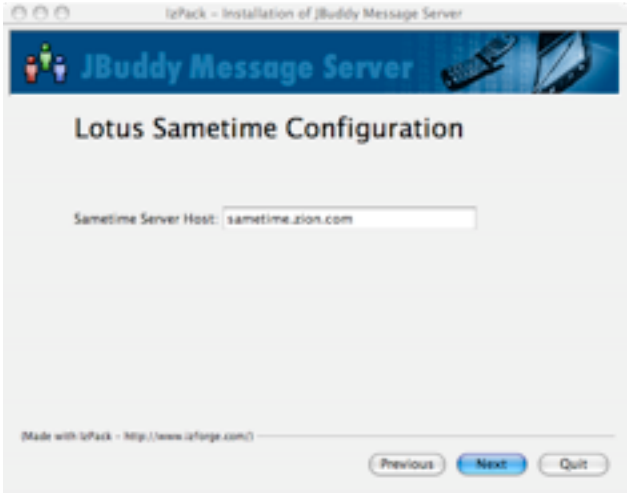


Figure 2-9B: Lotus Sametime Configuration

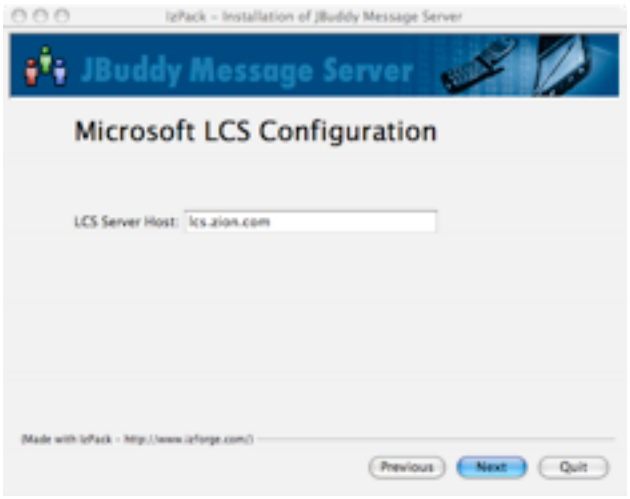


Figure 2-9C: Microsoft LCS Configuration

Figure 2-9D: Jabber Configuration

After clicking Next, the Installation Process (Figure 2-10) will begin and you will see progress bars as the packs are written to the directory specified in the Installation Location window (see Figure 2-5). Once this process is complete, the Next button will become active.

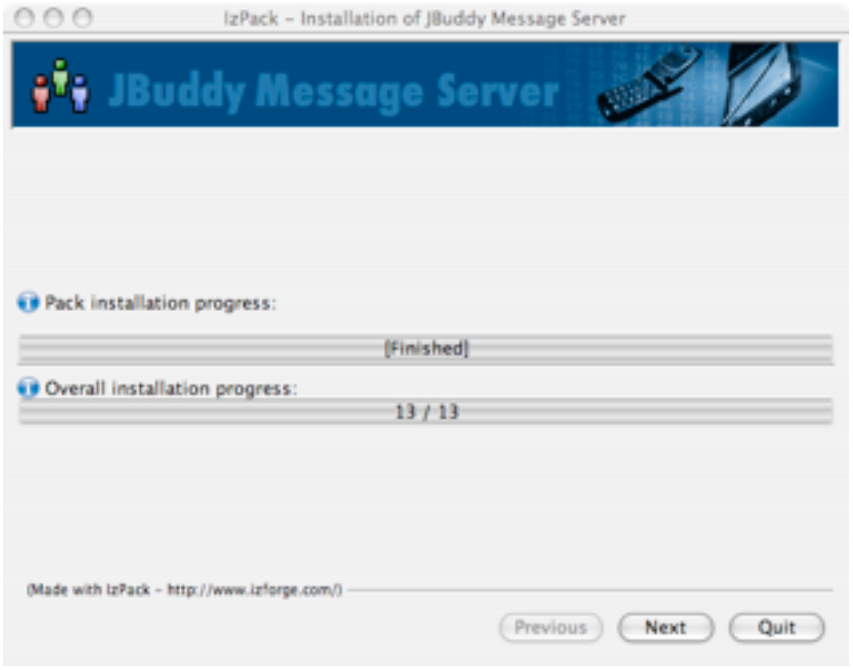


Figure 2-10: Installation Progress

After clicking Next, the Installation Post Install Processing window will appear similar to Figure 2-11 (Mac OS X) below. If the Installation is believed to be successful, the Next button will be enabled once more.

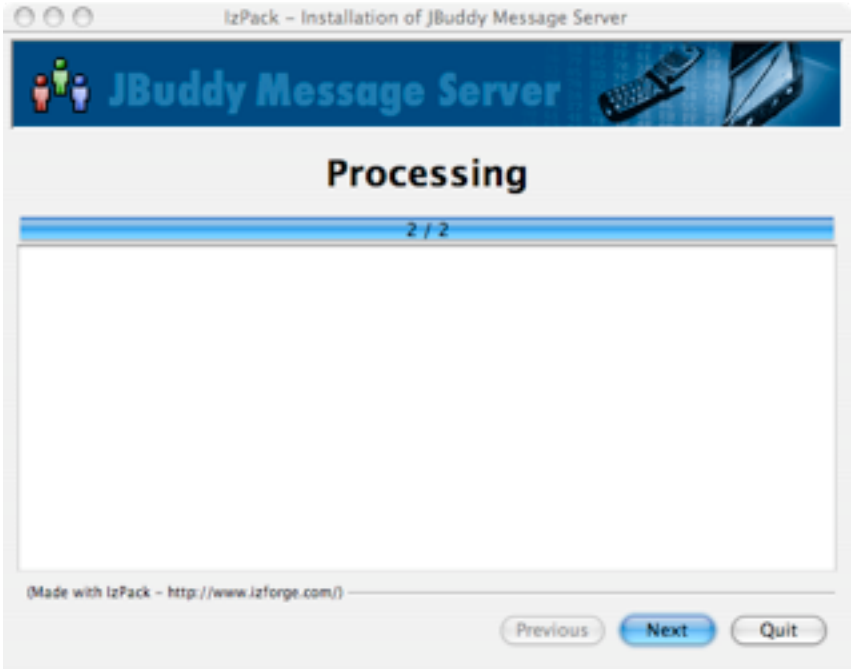


Figure 2-11: Installation Post Install Processing (Mac OS X)

On Windows, the Installer will attempt to register the JBuddy Message Server and JBuddy Message Server Admin Console as services with the operating system and start them. The log out-

put from Post Install Processing will appear in the window as well (Figure 2-12). If the Installation is believed to be successful, the Next button will be enabled once more.

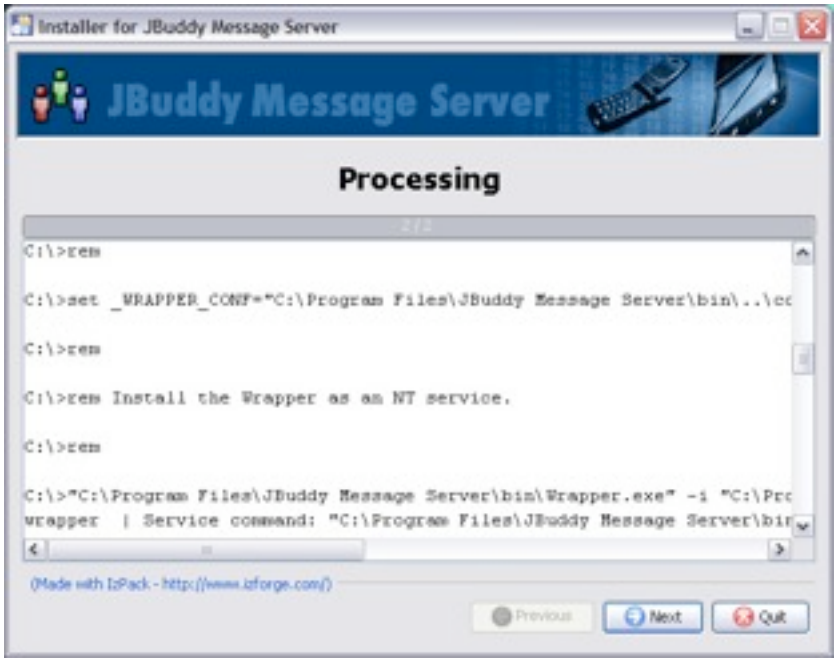


Figure 2-12: Installation Post Install Processing (Windows)

After clicking Next, the final Installation window will appear as shown in Figure 2-13 below. You can save this GUI installation as an XML script for a repeatable non-gui installation, by clicking the “Generate an automatic installation script” button before finishing.

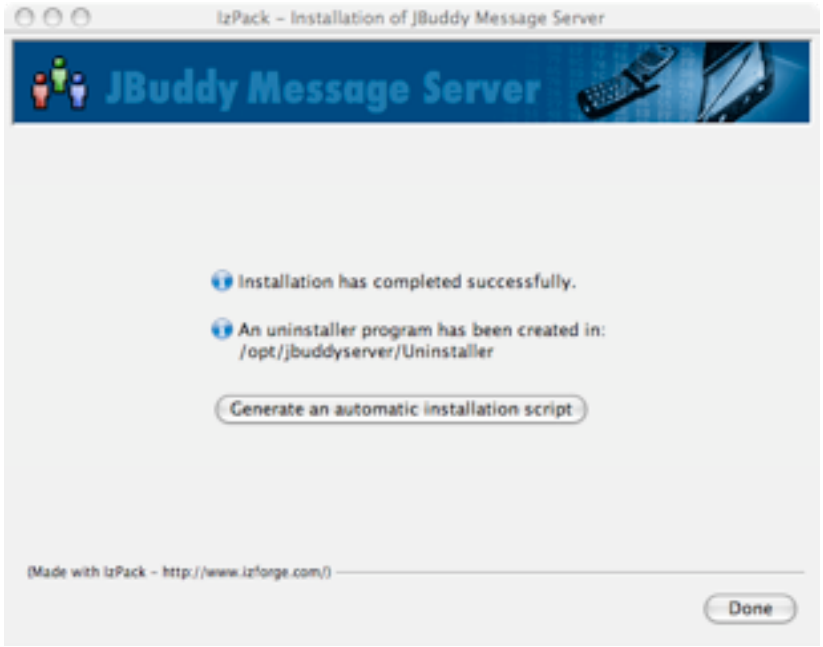


Figure 2-13: Installation Success / Save Installation Script

SERVER ADMINISTRATION

JBuddy Message Server consists of two servers and one or more message gateways. The servers are JBuddy Message Server and JBuddy Message Server Admin Console. The JBuddy Message Server Admin Console runs as a web application with an instance of Jakarta Tomcat 5 which is a web server with Java Servlet and JSP support. The JBuddy Message Server Admin Console is required for configuring accounts, routes, viewing message archives, etc and also supports the Instant Help Manager, Presence Servlet and Messenger Logging Servlet.

Windows

On Windows 2003 Server and above and XP Pro, both services are installed and started automatically in the background as "windows services". Log output shown in the Installation Post Install Processing window (Figure 2-12) would typically indicate the success or failure of the services launch during installation. At any point after the installation, you can view the current running status of the JBuddy Message Server services on a Windows Machine by viewing the Windows Services Panel, (Extended) (Figure 3-1). If the Status is “Started” this indicates the service is running. To shutdown the server on Windows, select the JBuddy Message Server service and click stop the stop button. The JBuddy Message Server Admin Console will also be stopped as it depends on the JBuddy Message Server service. This may take a minute. To start the Server, select each of the JBuddy services and click the start button. After a few seconds the Services panel should refresh and update the services Status to “Started”. If one or both of the services fail to start the Windows Event Logger will show the failure.

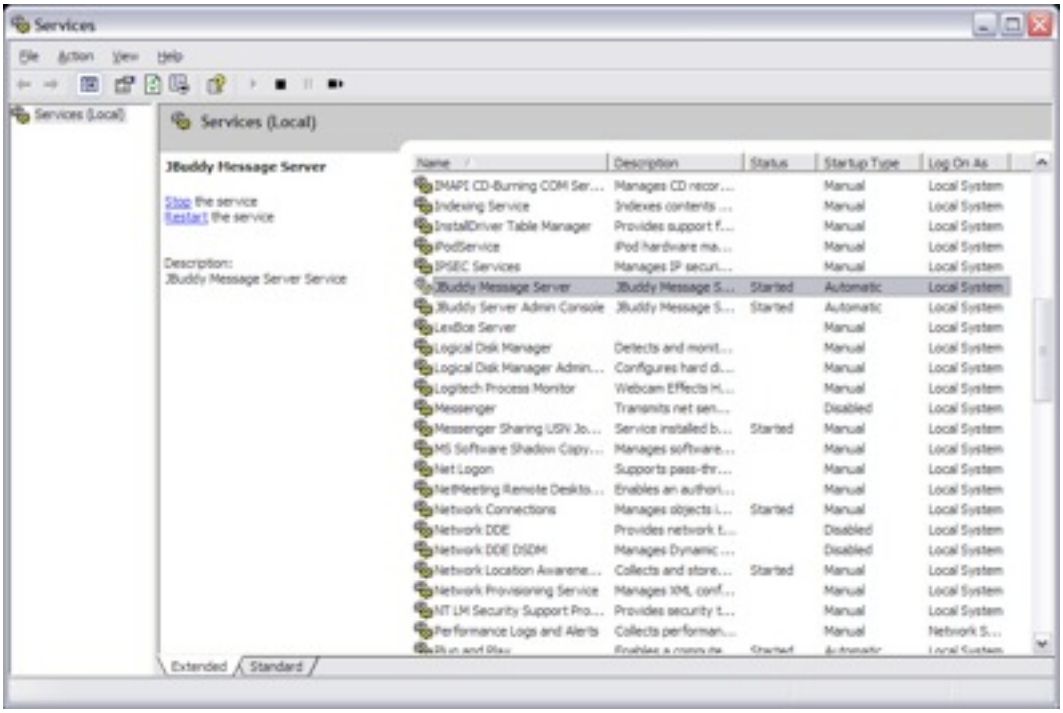


Figure 3-1: Windows Services Panel

Linux, Solaris, Mac OS X, BSD, etc

On Linux, Solaris, Mac OS X, or another flavor of 'unix', the Installation process does not launch the server. You will need to invoke two shell scripts from a command shell or terminal. If having the server available after reboot is desired, add the scripts to the etc server startup scripts. For now, just open command shell or terminal and change to the installation directory. Next enter *./bin/server.sh start* to start the server and *./bin/admin.sh start* to start the admin console. Gateways can be started with *./bin/xxxGateway.sh start* were xxx is the pre-fix of the desired message gateway. Use no arguments to see the other options (stop, status).

Configuration Files

Inside the installation directory is a conf subdirectory which contains JSW and JBuddy Message Server configuration data for servers and gateways. Various configuration parameters may be modified by carefully editing the .conf files and restarting the servers and gateways. A backup copy should always be made prior to editing in case a simple typo is introduced preventing the server or gateway from starting. Important parameters include: **-server** JVM argument to specify the server JVM (no -D is prepended), System properties to enable SSL encryption, message archival, and logging levels, whether to use an external database or not, and many more. Consult the comments within each of the .conf files for details. Below is a table of useful Java System properties that may be set in the .conf files by adding new wrapper.java.additional.X=-D where X is the next highest level and the property name=value is appended after the -D (no space) unless otherwise noted.

Property Name	Property value	Default	Description
-server	N/A	N/A	uses the server JVM instead of the client JVM for server side applications. No -D is prepended.
-XX:AggressiveHeap	N/A	N/A	enables the AggressiveHeap garbage collector in the JVM for high load servers. No -D is prepended.
jbuddy.use_external_db	true	false	Disable embedded HsqlDB engine
jbuddy.disable_message_archive	true	false	Disable message archiving
jbuddy.use_rich_text_in_message_archive	true	false	Archive message in native rich text format instead of plain text
jbuddy.auto_reply_seconds	900	900	Number of seconds of idle to resend auto_reply message

Property Name	Property value	Default	Description
jbuddy.db_key	READ ONLY	DB Secret	This is READ ONLY after installer. Triple DES key used to encrypt passwords.
jbuddy.use_ssl	true	false	Enable SSL login and message encryption (applies to JBuddy Gateway and JBuddy Message Server)
jbuddy.logLevel	0-1023	3	default log level for JBuddy Message Server and **all** components
jbuddy.logThreaded	true	false	Spawn a logger thread to try to order the incoming log requests according to FIFO order

Table 3-1: Server Properties

Additional Server properties are located in various properties files located inside the lib subdirectory.

Logging

Inside the installation subdirectory is a logs directory where both servers store logs. The JBuddy Message Server logs to server.log while the JBuddy Message Server Admin Console writes to the admin.log. The gateways log to xxxx.log where xxxx is the corresponding gateway name. In addition Jakarta-Tomcat may also log other messages to jakarta-tomcat/logs. The logging level of various processes managed by the servers may be changed carefully and the servers restarted. The JSW .conf parameters also specify how large a .log file will grow before it is replace and how many log files will be rotated. The table below outlines various JBuddy specific log levels that can be enabled depending on the desired output. jbuddy.logLevel is bit-wise ANDed with the log level of the log call to determine if it will be logged, therefore you will add the values you want to arrive at a log level. Ie: if you want to log EXCEPTIONS, ERRORS, WARNINGS across all JBuddy processes, you would add the values from the table below to arrive at the logLevel: `jbuddy.logLevel=7`

Log Level Name	Log level	Description
OFF	0	Logging Disabled
EXCEPTION	1	Log Exceptions
ERROR	2	Log Errors
WARNING	4	Log Warnings
DATABASE	8	Database related logs
RMI	16	RMI related logs
SOCKET	32	Socket related logs

Log Level Name	Log level	Description
TRACE	64	Method level logs
DEBUG	128	Full debug logs
LOG_BUDDY	256	Presence related logs
LOG_MESSAGE	512	IM related logs
ALL	1023	Always log

Table 3-2: Log Levels

For more fine-grained control over logging, individual components can have their logLevel set by adding a System property to the MessageServer.properties file after the specified component. For example. To turn on EXCEPTION, ERROR, WARNING, and PRESENCE and MESSAGE logging for the AIM Gateway, we would enter the following using the values from the Log Level Table above:

```
com.zion.messaging.aim.AimMsgGatewayImpl.logLevel=775
```

Database

Inside the installation directory is a hsqldb subdirectory where the HSQLDB database server is installed. This is a simple yet powerful open source relational database written in java. **Using a .conf option, the JBuddy Message Server can be told to use an external database such as Oracle, Sql Server, My SQL, etc rather than HsqlDB if desired.** Presently the JBuddy Message Server controls the start and stop of HsqlDB. JBuddy Message Server communications with the database through a properly configured JDBC driver.

Object Id Server

To provide a database column sequencer that works with any database, JBuddy Message Server includes a process called the ObjectIdServer. This process acts as a object id (primary key sequencer) generator for many of the database tables used by JBuddy Message Server. Therefore, careful care should be made not to manually insert rows into any databse table with an ID field *while the server is running*, otherwise the ObjectIdServer may become out of sync for that database table.

Creating Accounts with SQL

Although we haven't covered User level accounts up to this point, this section will detail how to correctly create User level accounts in the hsqldb database such as for a bulk loading of users by directly updating the hsqldb database with SQL. If you are using an external database by disabling the hsqldb via the config parameter, you can skip steps 2-4 above and create rows according to step 5 and the SQL example provided.

```
1. Shutdown the JBuddy Message Server service.
2. In a command shell or terminal, start only the database (run hsqldb.sh)
3. Run the DatabaseManager command line interface tool:

cd into the hsqldb directory
java -cp lib/hsqldb.jar org.hsqldb.util.DatabaseManager

4. Login to the HsqlDB and change login fields to:
Type: HSQL Database Engine Server
```

```
Driver: org.hsqldb.jdbcDriver
URL: jdbc:hsqldb:hsql://localhost
User: sa Password: <blank>

5. Insert a new database record into the ACCOUNT table for each user. You will need to
increment the ID field for each record inserted. The name field should be lowercase
with no spaces or strange punction but may be formated like a valid email address
firstname.lastname@company.com for example, or flastname or firstl, etc. The U_ID col-
umn must match the account name given for the Sys Admin account or another valid U_ID
from the USER_PROFILE table.

example SQL:

INSERT INTO ACCOUNT (ID, NAME, NICK_NAME, MSP_ID, U_ID, BUDDY_LIST_VERSION, BUDDY_L-
IST_PERMIT_MODE, NUM_MESSAGES_SENT, NUM_MESSAGES_RECEIVED, PASSWORD, KEEP_ALIVE,
AUTO_REPLY_MSG, SERVER_HOST, SERVER_PORT)
VALUES (1, 'bart star', 'bart', 99, 'admin',0,1,0,0, 'password', 'T', 'This conversa-
tion is being logged by JBuddy Message Server', null, null)
```

Network Ports

For enterprise IM using the proprietary JBuddy protocol, the JBuddy Message Server load balancer process listens on port 1424 and the JBuddy Gateway listens on 1425 by default. These can be modified prior to startup by carefully editing the MessageServer.properties file within the lib subdirectory. The load balancer listening on port 1424 redirects connecting clients to connect to the JBuddy Gateway running on port 1425 on the same host or other hosts if JBuddy Message Server is configured for enterprise clustering. The JBuddy Message Server Admin Console runs as a web application within Jakarta-Tomcat and is available by default by pointing your browser to port 8080, although configuring Jakarta-Tomcat to run in parallel with the popular Apache web server on port 80 is very common but beyond the scope of this guide. Other network ports used by JBuddy Message Server may include additional ports opened by Jakarta-Tomcat (see tomcat documentation for details), HsqlDB (defaults to port 9001), and Java's RMI (port 1099 by default). Various other optional message gateways use additional ports and are documented in the section specific to the gateway.

External Access while behind a NAT Router

Normally enterprise IM is configured for privacy and security with no external access permitted. However, if the JBuddy Message Server is used for an internet portal or online community, then external access is required. If the host that JBuddy Message Server is installed on is reachable from the internet with a fully qualified domain name or internet IP, there is no problem. However, if the host that JBuddy Message Server is installed on is only visible from the private network, an additional setting is required. The JBuddy Message Server may be configured to listen for connecting clients from the internet while running on a private IP network such as 192.168.x.x behind a NAT Router by adding the following System Property to MessageServer.properties located in the lib subdirectory:

```
JSC_HOST_DMZ=externalhostname or IP
```

The private hostname or private IP is still required during the installation process since this is the hostname or IP address that the JBuddy Message Server load balancer and the JBuddy Gateway bind to, however with the JSC_HOST_DMZ property set, the JBuddy Message Server load balancer will return the external host name or internet IP address to clients that connect rather than the default hostname or IP that the JBuddy Gateway binds to which would be unreachable from the internet. ***It should be noted that if this setting is utilized, internal clients***

may be unable to connect if they are unable to connect to the external address from inside. *This may be remedied with a proper route entry added to the NAT router, although this is beyond the scope of this guide.*

USER MANAGEMENT

User Profiles

Each user who will have the ability to configure Accounts within JBuddy should have their own User Profile. A User Profile consists of a user id and password and other contact information. The Sys Admin User Id is created during server installation. If central control of all accounts will be handled by the Sys Admin for the JBuddy Message Server, then no additional User Profiles need to be created. The default Central Account Management screen is shown in Figure 4-1A below.

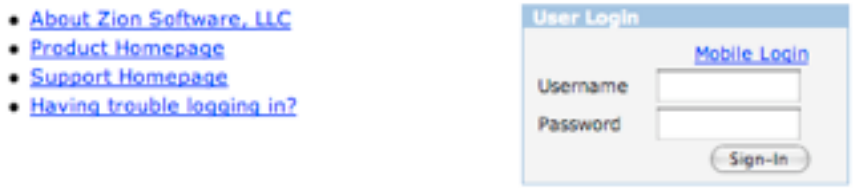


Figure 4-1A: Central Account Management Screen

If each user will be able to manage their own Accounts such as for a hosted service, then each user should to have their own User Profile. Please request the hosted-version of the JBuddy Message Server from Zion Software if this is your intended use. It differs slightly from the non-hosted version (see Figure 4-1B) and provides users with the ability to create their own User Profile. To create their own User Profile, users will click on the Sign Up Now link from the homepage and completing the User Registration form (Figure 4-2).

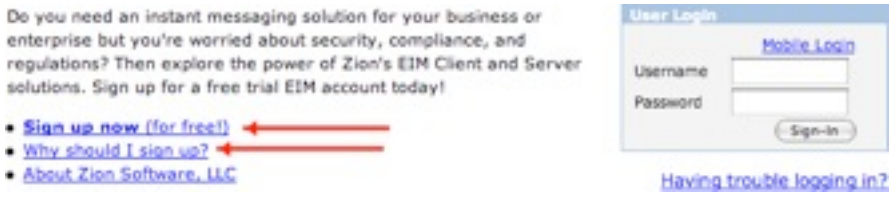


Figure 4-1B: Hosted Account Management Screen

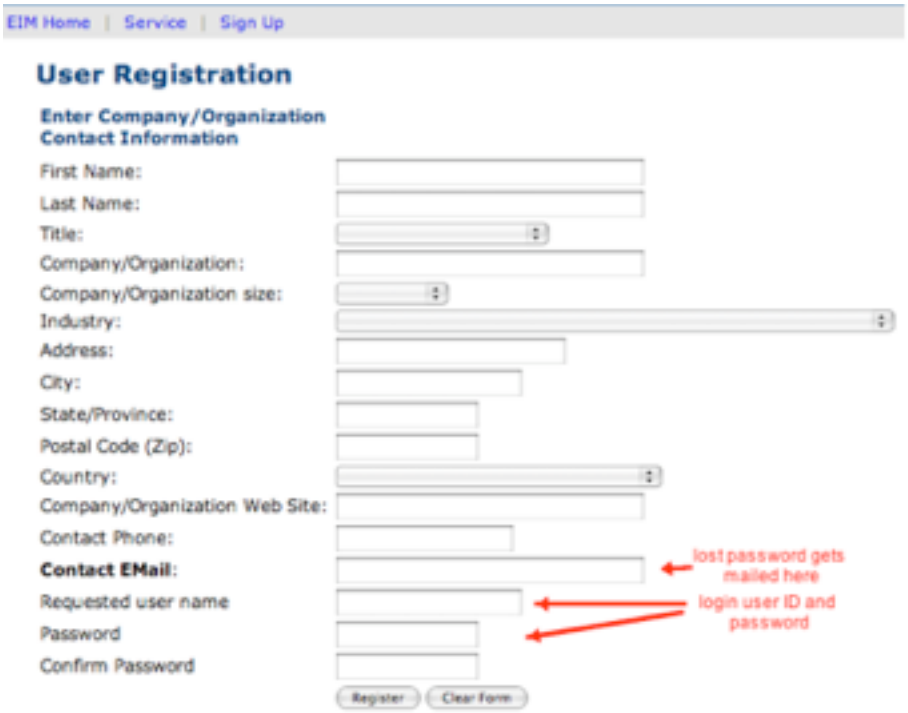


Figure 4-2: User Registration Form

Logging In

Web Login

With a valid user ID and password (either the Sys Admin or one created using the User Registration form in Figure 4-2), a user can login to the JBuddy Message Server Admin console by entering their user ID and password in the User Name and Password fields shown in Figure 4-1, or may login via a WAP enabled mobile device but pointing the mobile device browser to:

WAP Login

http://hostname:8080/JBuddyMessageServer/login.wml

where hostname is the hostname or IP of the server hosting the JBuddy Message Server Admin Console and 8080 is the port where the Admin Console is available (could be port 80 or 443 if accessed through a web server such as apache).

Upon login the user will be presented with the Welcome page (Figure 4-3).

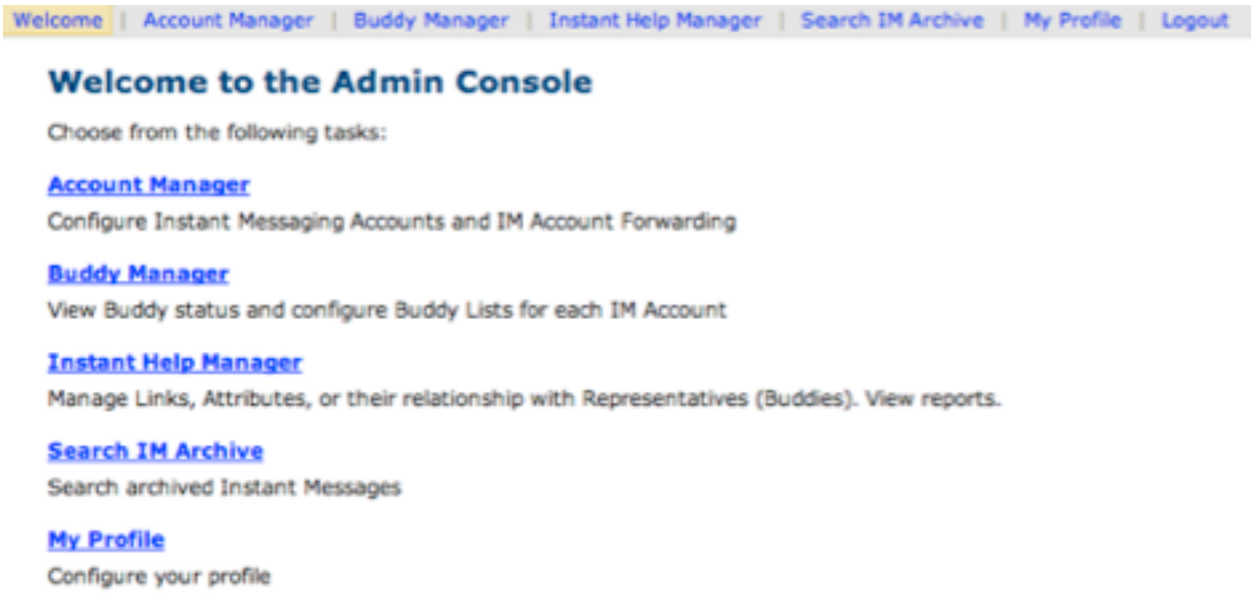


Figure 4-3: Welcome to the Admin Console

From here, the user can access various pages in the Admin Console by either clicking on the links in the Welcome page or clicking on the tabs across the top. Since this is a newly created User, we will begin with the Account Manager.

ACCOUNT MANAGER

IM Accounts

All connectivity in the JBuddy Message Server centers around Accounts. There are IM Accounts and there are other non-IM Message Accounts. A user may have more than one Account since JBuddy Message Server version 3.x is capable of connecting to five (5) public IM networks and four (4) Enterprise IM servers in addition to the other messaging services. After clicking on the Account Manager link or top menu tab, the Account Manager Admin screen appears (Figure 4-4):

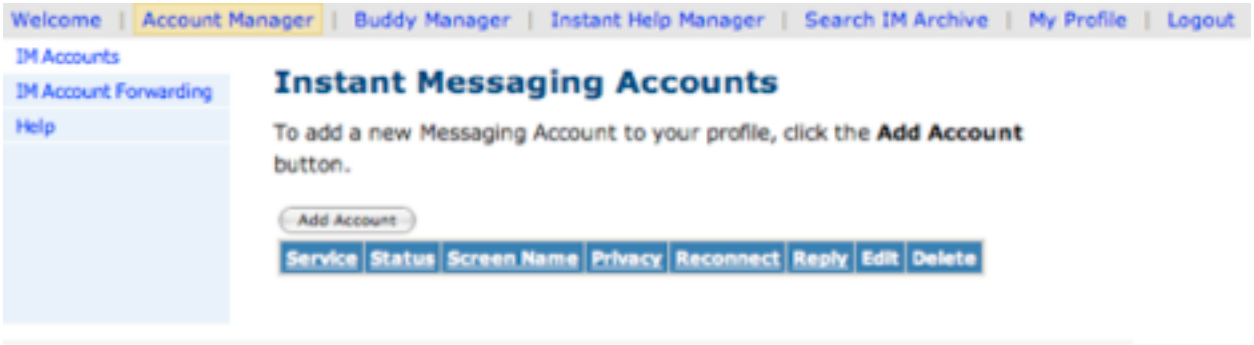


Figure 4-4: Account Manager Admin

Since this is a brand new user profile, there are no accounts visible in the table. After clicking on the Add Account, the New Messaging Service Account screen appears (Figure 4-5):

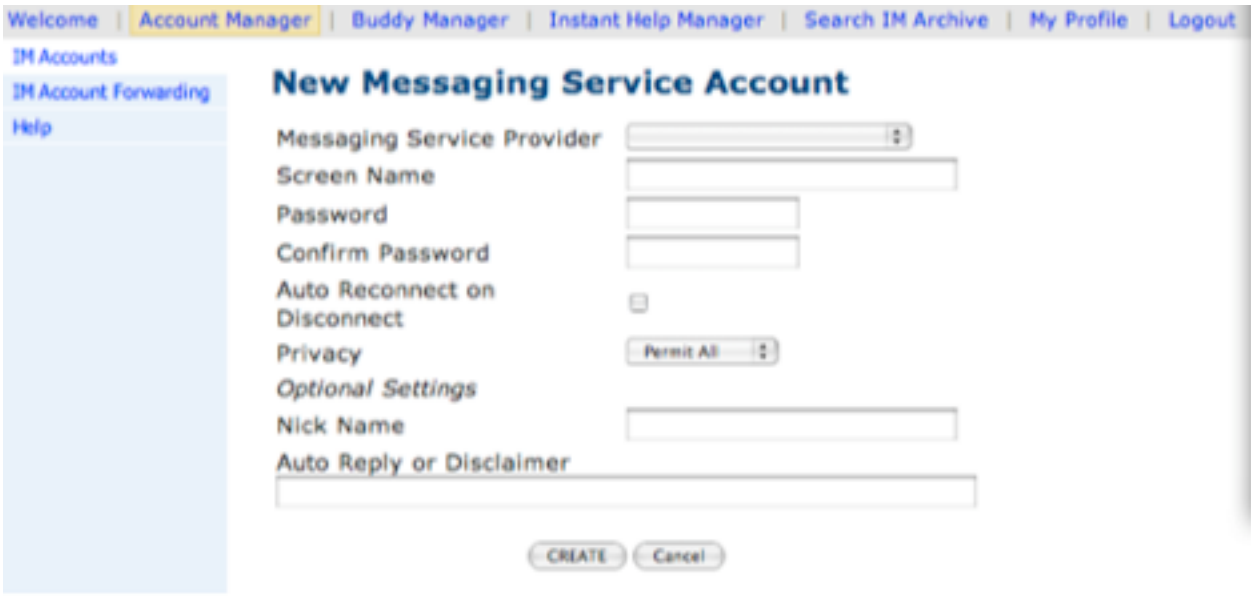


Figure 4-5: New Messaging Service Account

The Message Service Provide drop down list offers a list of public IM, enterprise IM and several other messaging service types. Depending on the license(s) installed, this list may be different. ***If nothing appears in the Message Service Provider drop down list, then JBuddy Message Server has been unable to find a valid license. Please verify that a valid license.lic exists in the lib folder of your JBuddy Message Server installation. This license key is finger-printed and sensitive to any modification.*** After selecting the desired Messaging Service Provider, the default fields may change.

JBuddy Enterprise IM Accounts

In Figure 4-6 below, the JBuddy Message Server drop down choice is selected as shown by the JBuddy icon (red, green and blue people). ***If you are only interested in private, enterprise IM using the JBuddy Message Server, you only need to create JBuddy Message Server IM accounts (even if you intend to connect to the JBuddy Message Server using XMPP Clients using the XTG)***. Once JBuddy accounts are created, all that's required to start using the

JBuddy Message Server for private enterprise IM is to configure the JBuddy Messenger / Pro IM clients to point to this JBuddy Message Server instance (configure the host in the IM client) and login with the above screen name and password. This is discussed in a subsequent chapter on the IM client. If you intend to connect to the JBuddy Message Server using XMPP clients, the XTG should also be running and you will need to configure your XMPP clients to point to the host where the XTG is running.

Figure 4-6: New JBuddy Message Server Account form

Public IM Accounts

In Figure 4-7, we've selected AOL Instant Messenger Messaging Service Provider as indicated by the yellow triangle representing AIM.

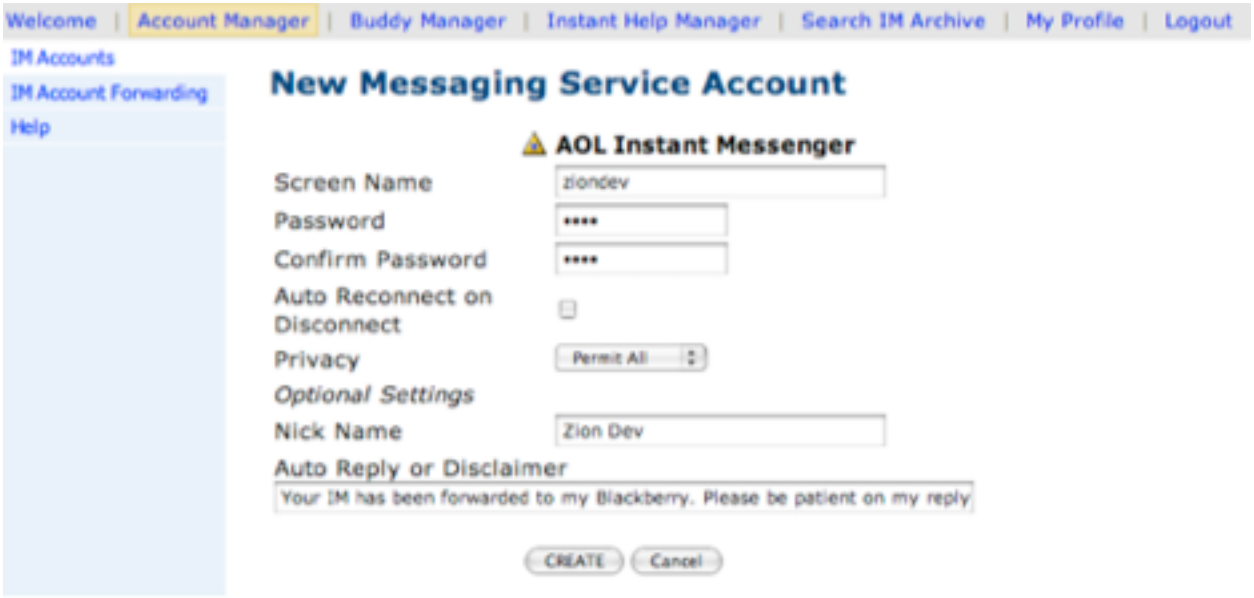


Figure 4-7: New AIM Service Account

The **Screen Name** and **Password** fields are required for all IM messaging accounts and represents a valid screen name of the selected Message Service Provider. The **Password** and **Confirm Password** input fields are where the valid password for this screen name are entered. Optional fields for the public IM and non-JBuddy enterprise IM accounts include the **Auto Reconnect** on **Disconnect** checkbox, **Privacy**, **Nick Name**, and **Auto Reply** fields. The non-JBuddy enterprise IM services including Lotus Sametime, Jabber / XMPP, and Microsoft LCS also require **Server Host** and **Server Port** fields to be provided in order for JBuddy Message Server to know where to connect.

The **Auto Reconnect** checkbox informs the JBuddy Message Server to attempt to maintain an active connection for this screen name. An invalid screen name or password would prevent this. The **Privacy** drop down list specifies how the IM account handles messages from other buddies. *Permit All* allows all messages, *Permit Some* allows messages from the buddies on the Permit / Allow List. The *Deny Some* privacy setting blocks messages from the buddies in the *Deny List* but allows messages from all others. Lastly, the *Deny All* privacy setting prevents messages from everyone which is not often used. The **Nick Name** field is often used by IM services as the *Display Name* seen by other buddies. The **Auto Reply or Disclaimer** field can be used by the server to send an automatic reply to a new conversation with a buddy, potentially to let them know the conversation will be logged, or to let them know the messages are being forwarded to a mobile device.

After creating several message accounts, the Account Manager Admin screen would look similar to Figure 4-8 below:

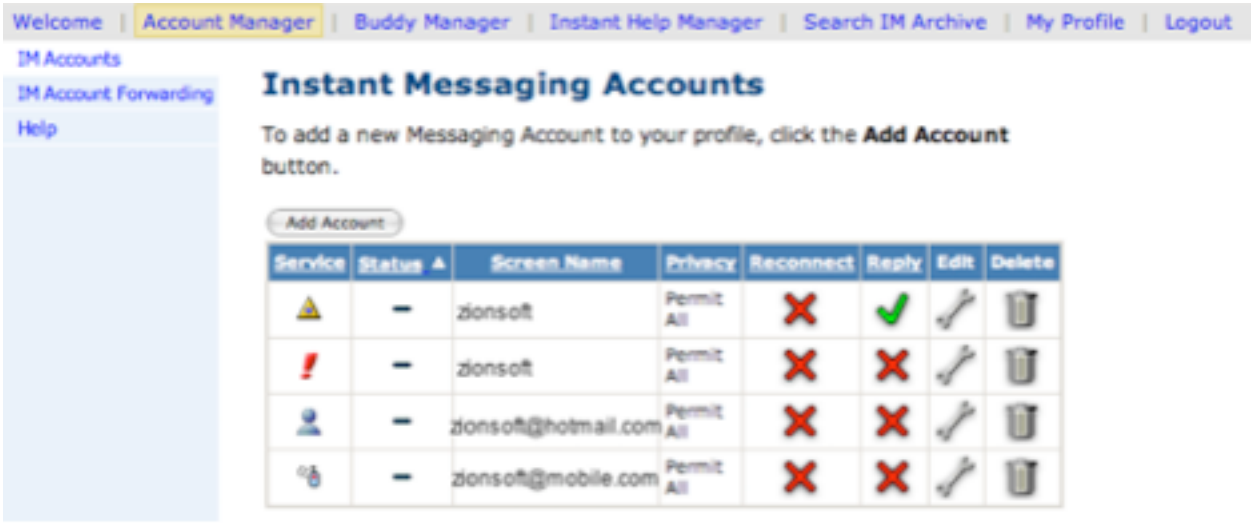


Figure 4-8: Account Manager Admin

Notice in Figure 4-8 how the status is blank for all the services. This is because the accounts are not yet active.

IM Account Forwarding

In order to active the message accounts creating in the IM Accounts screen, they must be associated with another message service. In the JBuddy Message Server Admin Console, this message account association or mapping or routing is termed, 'IM Account Forwarding' and it is available as a sub menu on the left side of the screen in the Account Manager below 'IM Accounts' (Figure 4-9 below).

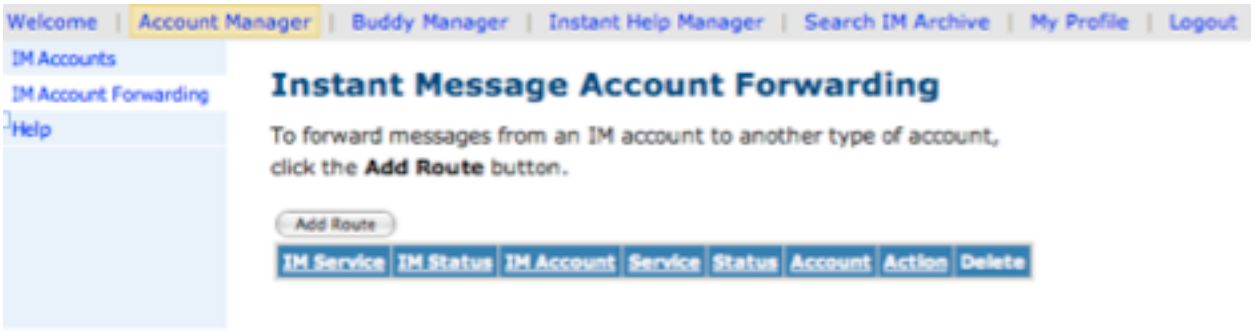


Figure 4-9: Instant Messaging Account Forwarding Admin (no routes).

Adding a Route

After clicking the IM Account Forwarding submenu, the Instant Messaging Account Forwarding Admin screen appears. In a new User Profile, there will be no 'routes' initially. As in Account creation, click the **Add Route** button to specify a new new route between an IM Account and another message Account. After clicking Add Route, a screen similar to Figure 4-10 will appear unless you don't have any Accounts defined, in which case you will get an error screen (Figure 4-11). The top drop down list contains the list of IM Accounts shown with the format: *screen*

name : *message service provider*. The bottom drop down list contains a list of non-IM Accounts that you can associate with the IM Account. This is true except for JBuddy accounts (JSC). This type of IM account will appear in the top list because it's an IM account, however it will also be available as a choice in the bottom drop down list. When a JBuddy account is used in the top drop down list, then it can not be used in the bottom drop down list for that 'route'. Instead a non-IM account can be used to forward JBuddy protocol messaging and presence to a non-IM gateway such as a mobile email account or even the JBuddy Instant Help software. If a JBuddy account is not used in the top drop down list, it can be used in the bottom drop down list. In this scenario, the top drop down list would be a public or enterprise IM and the messages and presence from this public or enterprise IM is forwarded when the JBuddy protocol client (JBuddy Messenger or a JBuddy SDK app such as an IM Bot) is online.

IM Forwarding To Blackberry (e-mail enabled devices)

Forwarding IM accounts to an SMTP account permits an end user to receive IM messages and buddy presence on an email enabled device such as a Blackberry phone or pager. If the routes have been created but not activated prior to leaving the office, the end user can login to the JBuddy Message Server Admin Console using a WAP enabled browser from the device if the end user has authorization credential to login and if the Admin Console is accessible from the device (see WAP Login in the Logging In section above). It should be noted that if IM logging has been enabled, that even messages to and from the mobile user will be logged ensuring full-time regulatory compliance even from the road.

IM Forwarding To Instant Help

Forwarding IM accounts to an Instant Help account permits an IM account to be used as a proxy representative for a real person. As an example, the ZionRep AIM account could be forwarded to an Instant Help account named 'Sales'. When a properly configured Instant Help web link was clicked by a web visitor, the Sales Instant Help proxy would be selected and an IM help invitation would be sent to one or more real IM users through the proxy representative (ZionRep). This scenario is very powerful for distributed live help desk support and will be covered in a chapter of its own.

IM Forwarding To JBuddy Message Server Accounts

Forwarding public or enterprise IM accounts to JBuddy Message Server accounts is used for two scenarios typically:

1. Central control of public IM access. Define which IM accounts and which JBuddy Messenger Users will have access to external protocols. This is primarily when an end-user will be using the JBuddy Messenger / Pro IM client or another JBuddy-enabled IM client on their desktop. (Figure 1-2).

2. Central control and management of IM Bots made available on public IM networks. Bots are written using JBuddy IM toolkits and access the JBuddy Message Server using the JBuddy protocol (JSC) and are made available to public IM users by creating a route to forward public IM account(s) to the JBuddy-protocol connected IM Bot.

Activating IM Forwarding

If the **Enable Forwarding** checkbox is checked, clicking OK will store the active state of this route in the database and the accounts will attempt to connect and go online. For scenario 2 detailed above, the routes state will be active but the IM account will not go online until a remote JBuddy client connects, at which time the route will be provisioned in the server and the IM ac-

count will attempt to go online and begin routing messages and presence to and from the public or enterprise IM client to the JBuddy client.

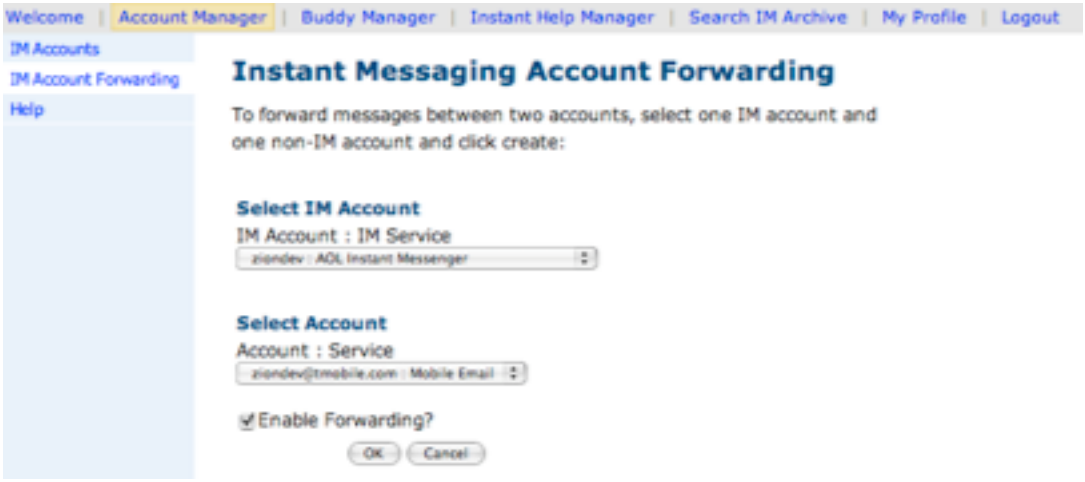


Figure 4-10: Instant Messaging Account Forwarding (create new route)

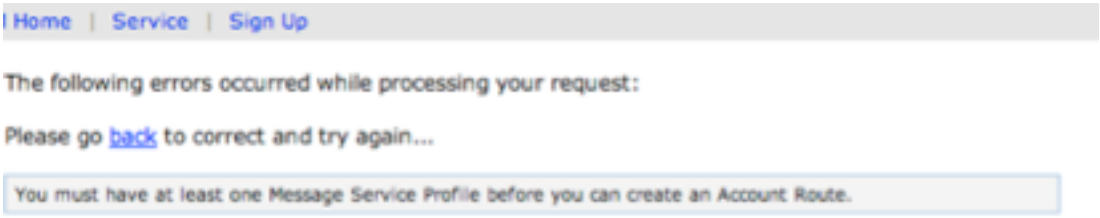


Figure 4-11: Instant Messaging Account Forwarding (error when no accounts exits)

After clicking OK in the Instant Messaging Account Forwarding (create new route) screen, you will return to the Instant Messaging Account Forwarding Admin screen. If **Enable Forwarding** was checked, the *Action* column will show a red stop sign (since clicking it would "stop" the enabled account forwarding). Since activating an account may take some time, the IM Status and Status columns may not refresh by themselves and may have a horizontal bar which normally indicates an offline status (Figure 4-12).

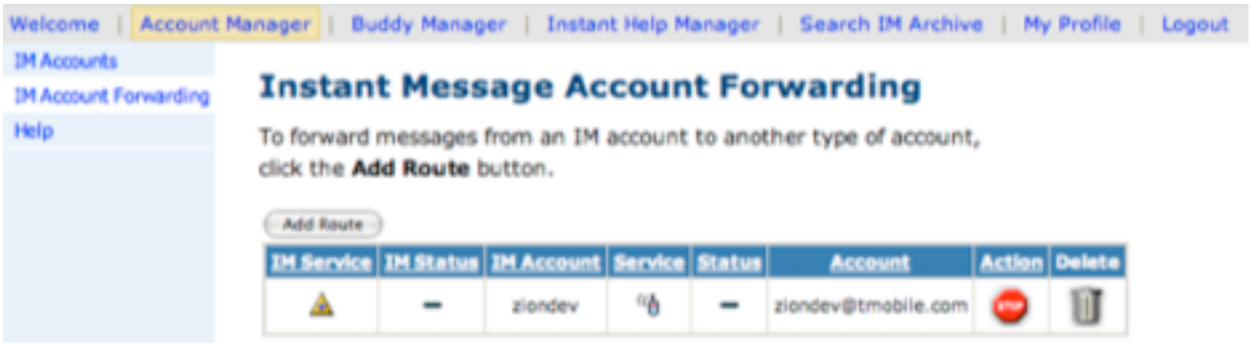


Figure 4-12: Forwarding an AIM account to a mobile email address

In this case, a newly enabled route just needs to be manually refreshed. This can be accomplished by clicking on either the *IM Status* or *Status* column headers to sort by that column. A

side effect of sorting the column is that the statuses also refresh. If they are online, you will see a green bullet icon (Figure 4-13).

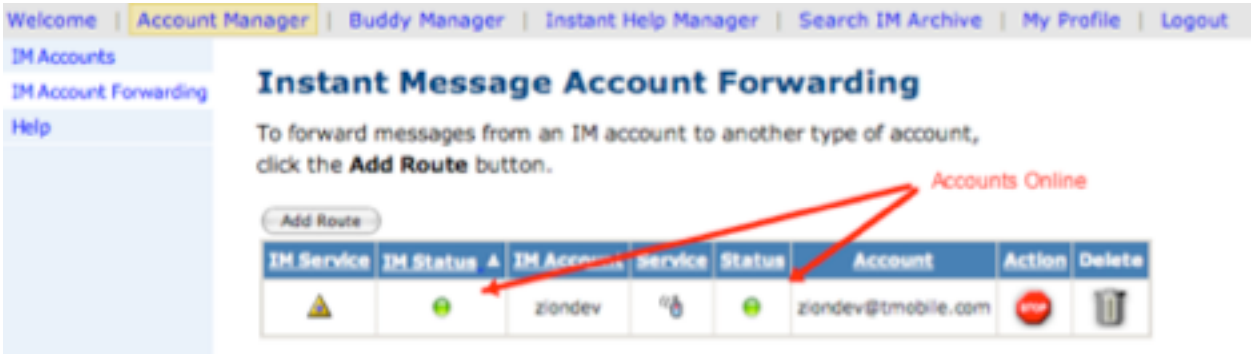


Figure 4-13: AIM Status (online), Mobile Email (online) after column sort

Returning to the IM Account Forwarding screen, we notice the new online accounts status as indicated by the status column with green (online) bullets (Figure 4-14). If a route was not active, instead of a stop sign, you would see a green “Go” button to indicate that you need to press GO to activate it.

DeActivating IM Forwarding

IM Forwarding can be deactivated simply by clicking on the red stop sign or deleting the route altogether. In addition, a mobile users with a WAP enabled browser and authorization credentials can login (see WAP Login in the Logging In section above) to the Admin Console if it’s accessible and de-active any active routes.

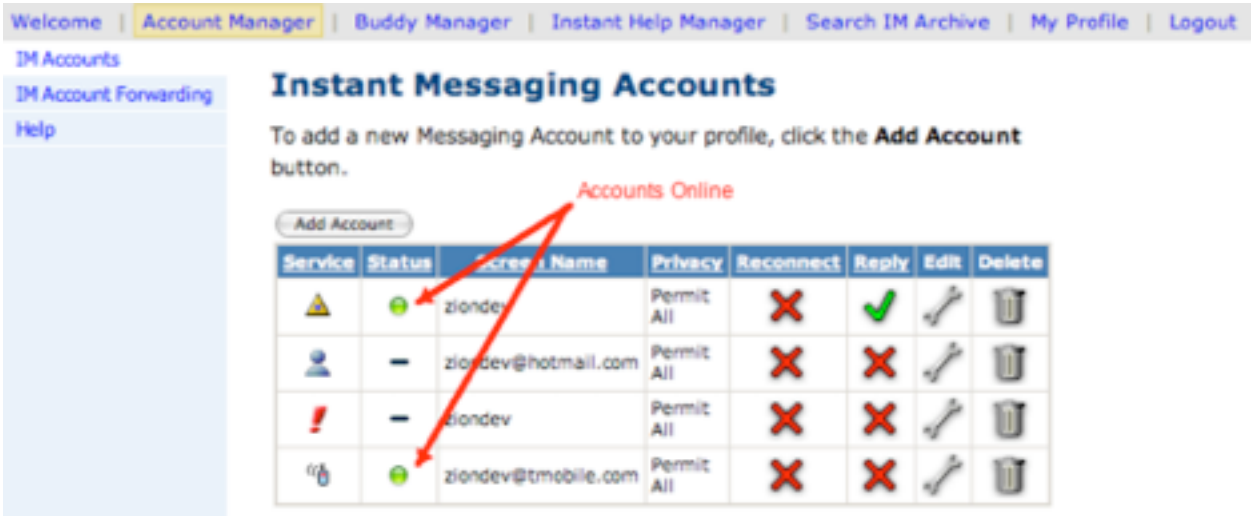


Figure 4-14 Account Manager Admin with account online status

In the next chapter, we will examine the Buddy Manager were the status of buddies is determined by the active, online status of accounts in this section.

BUDDY MANAGER

IM Accounts

On the left of the Buddy Manager page under the Accounts sub-menu is the list of IM Accounts available under this User Profile. If there are no accounts listed you will need to create them first using the Account Manager. The list of IM accounts is displayed as a list under the Accounts submenu (Figure 5-1):

Figure 5-1: Buddy Admin Manager with BuddyList for ZionDev on AIM with Status

Buddy List

After selecting an account on the left in the Accounts sub-menu, the Buddy List for this account is presented in a table in the content area. The data is retrieved from the database and if the account which owns this Buddy List is currently online, the information displayed will represent the current availability (status) of the buddies along with the Status Timestamp when the status change was recorded. The status is displayed as online or as offline or as away or as idle depending on the current status of this Buddy. In addition to the status indicator, the Screen Name and Nick Name as well as the Group Name which the Buddy belongs is listed.

Making Changes to the Buddy List

You can add new buddies, delete buddies and change the Group Name which a Buddy belongs using available actions on the Buddy Manager screen (Figure 5-1). Click the Edit button for a Buddy to change the group that this buddy belongs. Click the Delete buton to remove the Buddy from the Buddy List. Click the Add Buddy button and specify the screen name and group name and then click OK to add a new Buddy to the Buddy List of this account. **Before attempting to make changes to the Buddy List, ensure that this account is currently online** via IM Account Forwarding.

SEARCH IM ARCHIVE

Archiving messages serves no purpose without the ability to search and find specific messages quickly which is why the recent Sarbanes Oxley and HIPPA record retention legislation included time constraints on finding data on demand if a company was under court order or under an SEC audit. This is not a problem with the Search IM Archive screen within the JBuddy Message Server Admin Console (Figure 6-1):

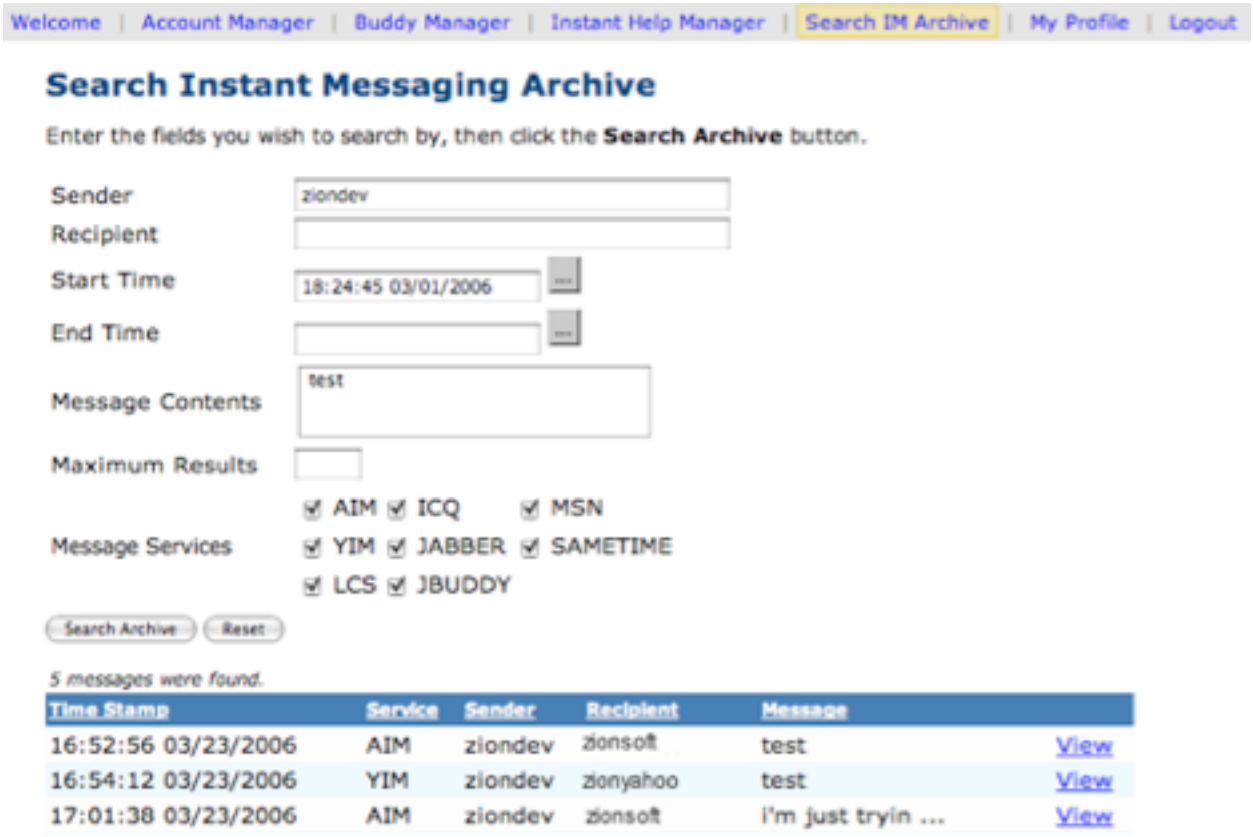


Figure 6-1: Search IM Archive with results

In Figure 6-1,we see a search with several fields specified. At least one Message Service should be checked to indicate which message service to search. In this example, all of the services are checked to indicate a broad search. Start Time and End Time can constrain the search by date and time. Sender and Recipient are the screen names of the message sender and recipient respectively and can be used to narrow the search to a particular user, although in this example, there are two users who have the ziondev screen name; one on AIM and one on YIM. They are different users but might be the same person in real life. The Message Contents uses a SQL (like) %search% approach in order to find partial search word (no matter if it is at the beginning or the end or middle of the message being searched). Clicking on the View link next to the search results will bring up a popup that displays all the details of the instant message.

INSTANT HELP MANAGER

Administration of the Instant Help System

Now that we've covered the core JBuddy Message Server features, we will describe an exciting new technology available as an add-in to the JBuddy Message Server. Clicking on the Instant Help Manager tab brings up the Instant Help Manager Admin screen (Figure 7-1).

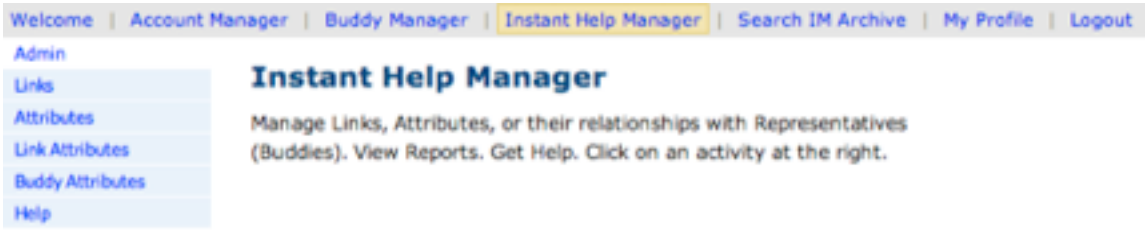


Figure 7-1: Instant Help Manager

Overview

Instant Help is a unique 'help desk' solution built on top of Instant Messaging and offers any company with a website and at least one internet enabled representative with the ability to provide live *Instant Help* to web site visitors. A snippet of HTML is added to the company's web page at various locations. The HTML snippet contains a URL that can be general or very specific depending on the needs of the company. When a visitor clicks the *Instant Help* button (Figure 7-2),

Figure 7-2: Instant Help Web Button

they are prompted for their name and their initial question (Figure 7-3):

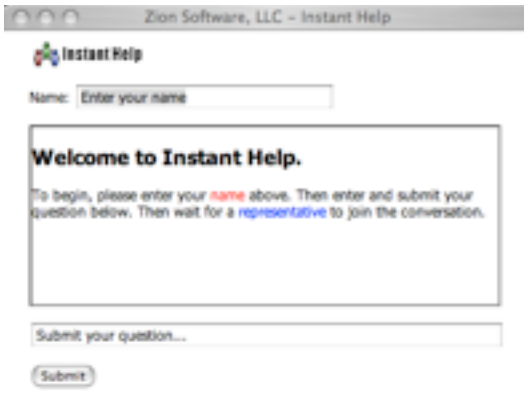


Figure 7-3: Instant Help Visitor (Welcome)

Upon submission, the *Instant Help* server invites (sends a special IM invitation) one or more qualified representatives (rep) to help this visitor. The first invited rep to respond affirmatively is given the opportunity to help the visitor. The rep communicates with the *Instant Help* server over IM and the *Instant Help* server communicates with the visitor within the HTML *Instant Help* Window (Figure 7-4):

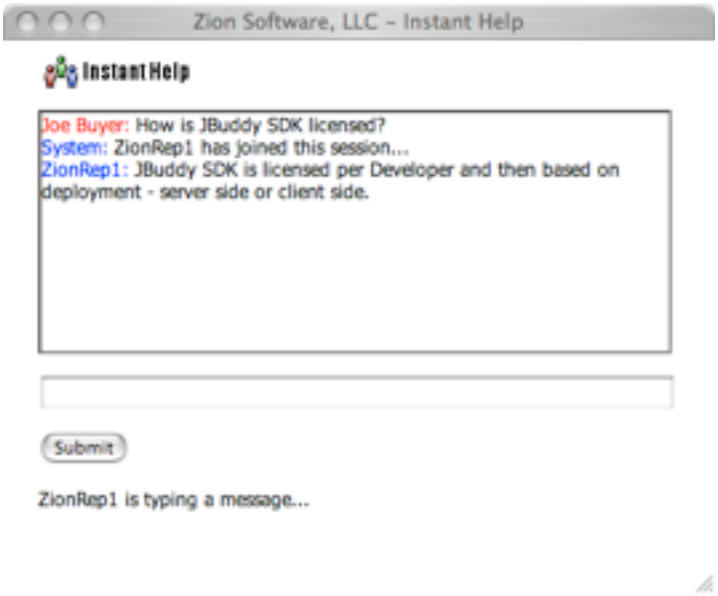


Figure: 7-4: Instant Help Visitor Session

Configuration Overview

Initial configuration of *Instant Help* server involves creating *Instant Help* accounts and IM accounts in the Account Manager. The IM account(s) are then forwarded to *Instant Help* account(s), on a one to one basis. Next, be sure the IM account(s) have Buddies including the buddies who will be the initial reps for the *Instant Help* session. Next, Attributes should be created and assigned to the reps. Finally, Link(s) should be created and Attributes should be assigned. Once all this is complete, all IM accounts forwarded to *Instant Help* accounts should be activated. At this point the HTTP URL may be added to the web pages where help should be available.

Link Manager

A Link represents a HTTP hyperlink. Any place you would like to offer *Instant Help* to visitors such as a company's web page, you can have a unique Link. For example if you have a general store page, you could define a 'store' link. If you wanted to be more specific, you could define an *Instant Help* link for each product. To view existing links, create, edit, or delete defined links, use the Links submenu (Figure 7-5):

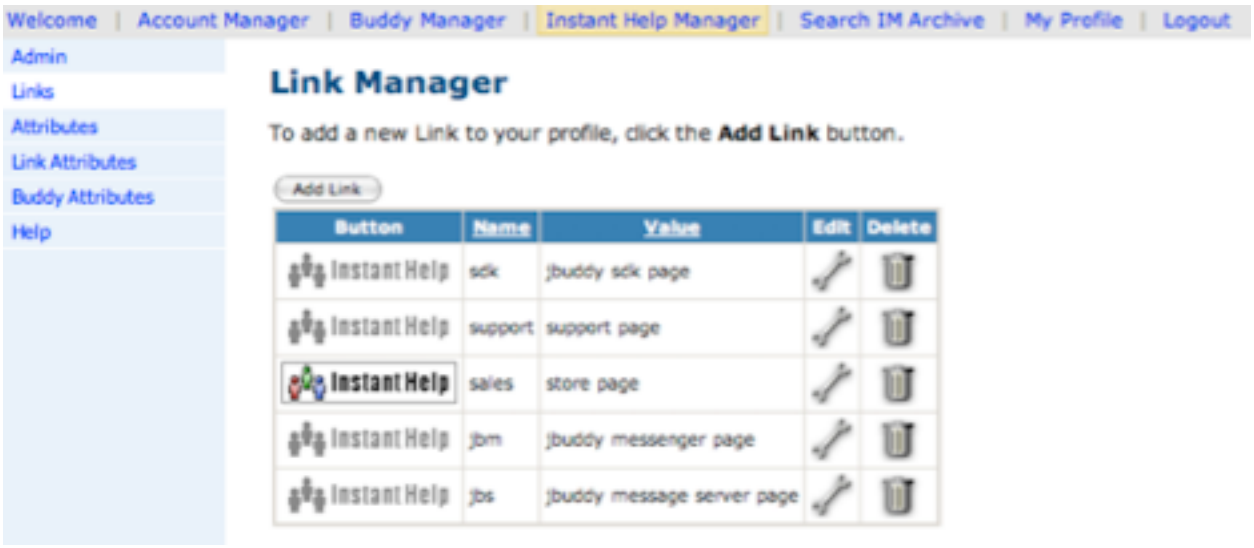


Figure 7-5: Link Manager

Attribute Manager

Names and Values

An Attribute is simply a name and value pair of words that can be associated with another element. Specifically, Links can have Attributes and Buddies can have Attributes linked to them. In the Instant Help system, it is the INTERSECTION of Link Attributes and Buddy Attributes that forms the basis for selecting which Buddies are best suited to help a visitor for a particular link. In the example of a 'Store Page Link' element, we could define the following Attributes for this link:
Name: *org* **Value:** *Acme Inc.*
Name: *url* **Value:** *www.acme.com/store*
If we have product specific links, we could add another attribute specific for the product:

Name: *partid* **Value:** *12345*

To view existing attributes, create, edit, or delete defined links, use the ***Attributes*** submenu (Figure 7-6):

Figure 7-6: Attribute Manager

Predefined Attributes

In this initial release of the Instant Help Server, the following Attributes are predefined (see Table 7-1). In the future additional Attributes may be defined. The Attribute name and value are specified along with the effect it has if associated with a Link or Buddy.

Property Name	Property	Area	Description
dispatchAlgorithm	A whole number	LINK	The number of seconds between sending invitations to help to qualifying reps
dispatchTimeout	A whole numer	LINK	The number of seconds after the last invitation to help is sent before telling the visitor no helper are available
contactFormUrl	URL	LINK	URL to provide to visitor if no helpers are available
contactFormEmail	EMail	LINK	The E-Mail address to send default contact form submissions to
noActivityTimeout	A whole number	LINK	The number of minutes after the last activity that a help session is automatically closed
onlineButtonUrl	URL to graphic	LINK, BUDDY	LINK - URL to online button. BUDDY - URL to online button for Buddy Presence
offlineButtonUrl	URL to graphic	LINK, BUDDY	LINK - URL to offline button. BUDDY - URL to offline button for a Buddy Presence

Property Name	Property	Area	Description
offlineSchedule*	crontab formatted offline schedule	LINK, BUDDY	LINK - governs when the link is offline/ inactive. BUDDY - governs when the buddy is offline/ inactive. This is used for PresenceServlet?buddyId=4 queries. Attribute name or Attribute name prefix (attribute name must start with 'offlineSchedule') for scheduled offline hours in unix crontab format. If no link attributes are equal to or begin with offlineSchedule, the presence of qualified reps will determine the use of offlineButtonUrl or onlineButtonUrl. Attribute Value Format consists of five fields. The fields are separated by spaces or tabs. They are integer patterns that specify the following: minute (0 -59), hour (0 -23), day of the month (1 -31), month of the year (1 -12), day of the week (1 -7 with 1 =Sunday). Each of these patterns may be either an asterisk (meaning all legal values) or an element. An element is either a number or two numbers separated by a minus sign (meaning an inclusive range). Note that the specification of days may be made by two fields (day of the month and day of the week). If both are specified, both are adhered to. For example, 0 0 25-31 * 2 would indicate a scheduled outage between the 25th and 31st of the month, as well as on every Monday. To specify days by only one field, the other field should be set to * (for example, 0 0 * * 2 would run a command only on Mondays). The parsable schedule format may be modified in the future to include an asterisk or list of elements separated by commas. For example, * 0-6 * * 7,1 would mean outage from midnight to 6am on Saturday and Sunday. For now, two records for the named resource would be necessary to specify this schedule.
displayName	name	BUDDY	Rep name to display to visitor if set as an attribute of Buddy (Rep)

Table 7-1: Predefined Attributes

Link Attribute Manager

A Link Attribute is simply an association between a Link and an Attribute with two other elements: *required* and *match weight*. If a rep must have a particular attribute in order to be included in the list of reps that can help, set *required* for the Link Attribute. If some Link Attributes have a higher priority in selecting reps over other Link Attributes, increase the *match weight*. A Link can have zero or more Attributes

associated with it, however in order to find 'reps' to handle an instant help request for this link, at least one Attribute must be associated with the Link. To view, link or delete associations between Links and Attributes, use the ***Link Attributes*** sub-menu (Figure 7-7):

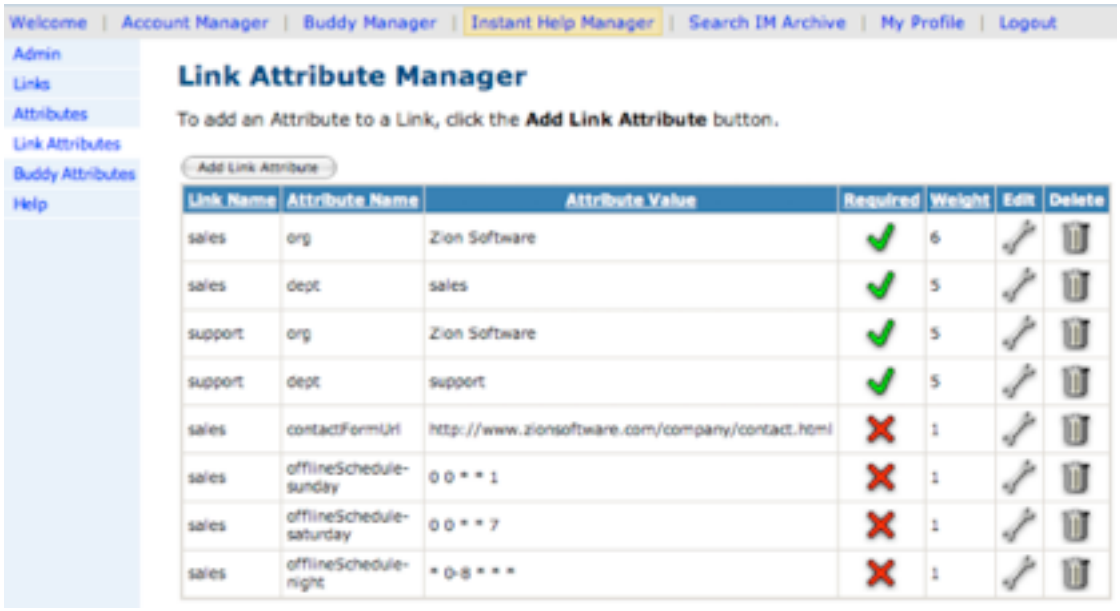


Figure 7-7: Link Attribute Manager

Buddy Attributes

A Buddy Attribute is simply an association between a Buddy and an Attribute with one other element: *proficiency*. The *proficiency* a Buddy has for a particular Attribute determines which Buddies (reps) are prioritized over another Buddy with the same Buddy Attribute. A Buddy can have zero or more Attributes associated with it, however in order to find the best buddies (Representatives) to handle an instant help request for a link, at least one Attribute must be associated with the Buddy. To view, link or delete associations between Buddies and Attributes, use the ***Buddy Attributes*** submenu (Figure 7-8):

Figure 7-8: Buddy Attribute Manager

Reports

Instant Help captures 'help session' data during each session including; how many messages are sent from the visitor, how many messages are sent by the rep, which rep(s) helped the visitor, the link ID the visitor clicked for help, the visitor's given name and given question (initial submission to request help), the referer URL that brought the visitor into the Instant Help system, the type of web browser used by the visitor, the actual messages sent / received (if message logging is enabled), and several other elements - all recorded in the HELP_SESSION database table.

In the near future, the reporting section will include some pre-defined reports which you may find useful. When available, the reports may be viewed using the ***Reports*** submenu.

JBUDDY MESSENGER

Overview

This chapter describes the JBuddy Messenger IM client available from Zion Software as it pertains to configuration with JBuddy Message Server. The main distinction between JBuddy Messenger free and licensed is the ability of the paid license version to connect directly to other enterprise IM servers such as Lotus Sametime, Microsoft OCS/LCS, and Jabber (XMPP). For the purpose of this chapter, JBuddy Messenger will be referred to as “JBM”. As of this writing, JBM is available as a download from http://www.zionsoftware.com/products/messenger/. JBM is a robust enterprises IM supporting rich text, emoticons, file transfer, context history, and much more.

Installation

Before downloading JBM from Zion’s website, be sure you have the latest version of Java installed on your desktop. Java is available as a free download from http://www.java.com/. JBM requires at least Java 1.4+ but recommends Java 5 or newer. Assuming Java is installed properly, you can simply download the JBuddy Messenger archive appropriate for your operating system (Mac OS X, Windows, or Linux / Unix). Next, extract the JBuddy Messenger application from the downloaded archive and simply move it to a location on your computer where you typically store applications.

Account Creation

Once JBM is installed, you will be prompted to create a new IM account. This step doesn't actually provision a new IM account on an IM server, but instead allows the user to specify the login credentials and server configuration necessary to sign on to the server. In order to login to the JBuddy Message Server, the JBM user must already have an account on the server created already (see Figure 4-6). The JBM user will click New Account (JBuddy Server) (Figure 8-1).

Figure 8-1: New Account Selection (JBuddy Server)

Next, the JBM user will specify the Username and Password (Figure 8-2):

Figure 8-2: New Account (JBuddy Message Server)

Next, the JBM user will need to click the Connection tab, uncheck the ***Use default server*** and change the ***Server Address:*** *Host* field to the hostname or IP address where the JBuddy Message Server is running (Figure 8-3). If the Sys Admin has enabled TLS/SSL (secure login), the ***Use secure connection (TLS/SSL)*** must be checked.

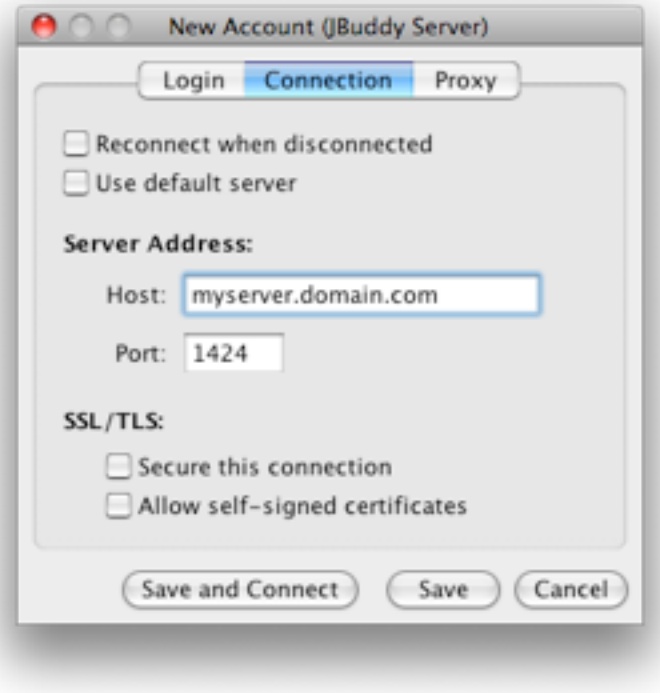


Figure 8-3: Server Host Configuration

Once that is specified, the JBM user can click ***Save and Connect*** and if everything has been specified correctly, the JBM user should connect and see a blank buddy list window. At this point, they can right click on their username and add “buddies” to their “buddy list” by entering other usernames much like is common using public IM such as AOL Instant Messenger and others.