JBuddy Bot Builder™
Users Guide
Version 1.0
2008 - 2012

1 Introduction

Welcome to the JBuddy Bot Builder Users Guide. We tried to make the JBuddy Bot Builder as intuitive as possible but should you need further information, you'll likely find it contained in this document or the JBuddy Bot Framework Users Guide.

Although the JBuddy Bot Framework makes it simple to develop, test and deploy sophisticated interactive message-oriented bots from the command line, we decided a rich IDE (Integrated Development Environment) would make it even simpler in environments where GUI tools are welcome. The focus of this Users Guide is to describe this new tool, the JBuddy Bot Builder. Built on top of the JBuddy Bot Framework it was designed for ease-of-use and to accelerate bot development, testing, and deployment. We hope you agree that we reached our goal.

Without further ado, let's get started!

2 Getting Started

Depending on your environment, you can launch JBuddy Bot Builder in one of several ways. (A) Native launcher for Windows and Mac OS X users: a double-clickable "JBuddyBotBuilder.exe" on Windows and double-clickable "JBuddyBotBuilder.app" on Mac OS X located in the JBuddyBotBuilder folder. (B) Double-clickable "BotBuilder.jar" located in the JBuddyBotBuilder lib directory. (C) From a command line or shell, change into the JBuddyBotBuilder lib directory, and type the following: java -cp BotBuilder.jar

Mac OS X Users: Although the pictures within this Users Guide are based on JBuddyBotBuilder within a Windows XP environment and appear different on a Mac, the functionality of the application should remain identical.

After starting the application, you will see the JBuddy Bot Builder main window (shown below in Image 2). At startup, no bots are running. In order to start a bot, you must load a Bot Definition File which is simply an xml file containing instructions and definitions for your bot and it's behavior based on user input. A Bot Definition File can be created or modified using a plain text/XML editor. Consult the JBuddy Bot Framework Users Guide, xmldocs and javadocs for in-depth documentation on elements of a Bot Definition File.
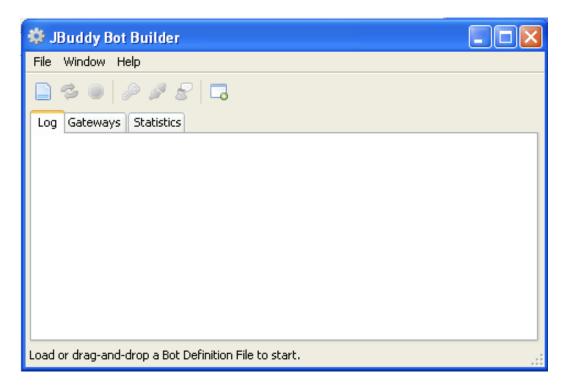

Image: 2. Main Application Window

2.1 Loading a Bot Definition File

To load a Bot Definition File, click the Load File button in the application toolbar. A file chooser will be opened, which you can use to select the Bot Definition File you wish to run (usually named "bot.xml"). If there are no errors, the bot will automatically start. A file can also be loaded by dragging it into the application window. Demos are included within the Bot Framework installation directory. Simply dragging a demo's bot.xml file into the JBuddy Bot Builder's main window will start the bot.

2.2 Running Multiple Bots

You can run multiple bots concurrently by simply clicking on the New Window Button in the application toolbar or selecting the New Bot Window menu item from the File menu. A new window will open that looks identical to the main window that appeared when the JBuddy Bot Builder was initially started. As was the case on startup, you must either open a new Bot Definition File within this window or drag a Bot Definition File to this window in order to load and start another bot.

2.3 The Log Panel

Log messages for the associated bot are shown in the log panel of the Bot's main window. The log also shows Bot Definition error messages, and can be used to fix syntax errors or other problems. Log contents can be saved to a file or cleared by using the "Save Bot Log..." and "Clear Log" menu items in the File menu respectively, or by right clicking in the Log panel.
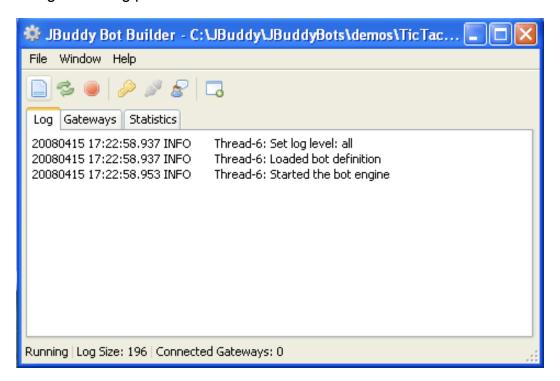

Image: 2.3. Log Panel

2.4 Hint Dialogs

JBuddy Bot Builder shows hint dialogs at times, to help you along or to show available options for a feature. Image 2.4 shows a Bot Definition Error "Hint Dialog." These hints can be ignored individually by checking the "Do not show" check box on the dialog. To reset all ignored hints, choose "Reset Hints" from the "Help" menu.

Image: 2.4. A Hint Dialog

2.5 Reloading a Bot

Changes you make to a Bot Definition File are not automatically detected by the application. To load any recent changes to the file, be sure the file has been saved, then click the Reload button in the application toolbar. If the Bot Definition File contains errors, you will be notified by a pop-up error hint (see Image 2-4-A) unless you have checked [X] Do not show this message again, and the modified Bot Definition File will not be reloaded. If the bot is connected when you click reload, it will remain connected through the reload unless the client definition within the Bot Definition File has changed significantly. This allows for on-the-fly fixes to a production bot.

2.6 Stopping a Bot

Stopping a bot ends all User Sessions (see Chapter 3) and disconnects all Client Gateways (see Chapter 4) for this bot. To stop a bot, click the Stop button in the application toolbar.

3 User Sessions

Like many applications, the JBuddy Bot Framework supports the notion of a user session, defined as the interaction between a user (human or non-human) and the service as well as the session "state". In the JBuddy Bot Framework, User Sessions are managed by the Bot Engine (the runtime environment for a particular bot). When developing and testing a bot prior to deployment, it is important to be able to interact with the bot as an end user would to ensure a consistent and correct user experience. The JBuddy Bot Builder makes this easy by providing an IM "chat window" emulator for all of the supported IM protocols. You can see what a user session will look like to real users, without actually connecting the bot to an IM service.

3.1 Starting a User Session

To start a user session, click the "Start User Session" button in the application toolbar. The New User Session window will be opened (see Image 3.1). Select an IM protocol from the drop-down list selector, enter a username, and new message then click the Send button or hit return on your keyboard.

Image: 3.1. New User Session Window

3.2 Interacting with a Bot

The User Session window looks and behaves like an Instant Messaging application chat window, although initially it may not look that way. Once an initial message is sent to the bot, the New User Session window will change appearance and display a history area (upper) and an input area (lower). The history will reflect the user's submission as well as the bot's response to this submission and any rich content returned by the bot (bold, italic, etc) will be formatted to reflect the differences between IM services (see Image 3.2). The User Session toolbar supports basic IM features such as rich content attributes, emoticons, and hyperlinks.
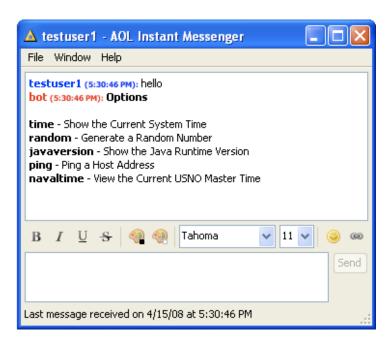
Image: 3.2. User Session Window

3.3 The Chat History Log

The chat history log displays sent and received messages for the current User Session. Log contents can be saved or cleared. Messages can be indented, and message time stamps can be shown or hidden. To access history log options, choose the "Save Bot Log..." or "Clear Log" menu items from the File menu or right-click the chat history log.
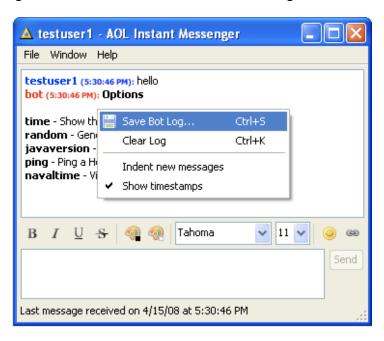

Image: 3.3. History Log Options

3.4 WYSIWYG Features

Not all features are the same for each protocol; for example, not all protocols support background colors or hyperlinks, and some may have different emoticon sets. The User Session window is WYSIWYG ("What You See Is What You Get"). Toolbar features not available for the current protocol are disabled, and the chat log displays only the currently supported rich content attributes. This allows you to see how your bot will behave to end users once it is connected to IM networks. This sometimes results in a different user experience for certain users, as demonstrated below using the "Windows Live Messenger" (aka MSN) protocol (see Image 3.4).
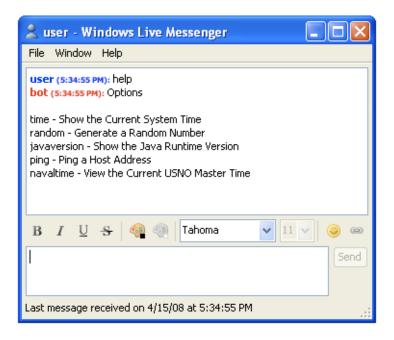

Image: 3.4. WYSIWYG Example

4 Client Gateways

Once started, the Bot Engine typically waits for messages. Messages are brought to the Bot Engine through Client Gateways. Bot responses are typically returned to the user through the same Client Gateway in which the user's message was received. JBuddy Bot Builder helps manage Client Gateway connections, allowing you to bring your bot online and make it available to end users. Client Gateway settings are defined per bot and are loaded from a Bot Definition File.

4.1 The Gateways Panel

Upon loading a Bot Definition File, available Client Gateways are displayed in the Gateways panel. Each Gateway has a status icon that shows when it is disconnected (gray), connecting (yellow), or connected (green). The icon can be clicked to connect or disconnect the Gateway. When a Client Gateway is connected, the bot becomes available for users on the specific protocol.
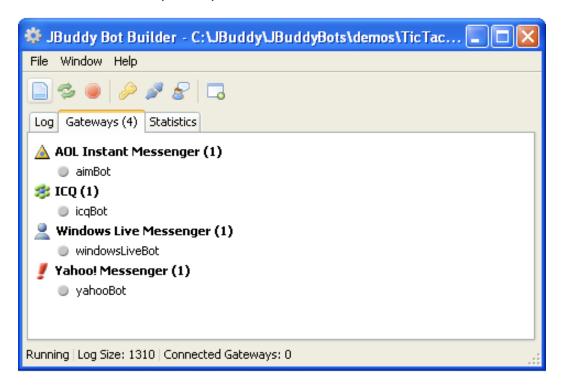

Image: 4.1. Gateways Panel

4.2 Connected Mode

Connected Mode automatically connects Client Gateways. Gateways are reconnected if a connection is lost. Also, after adding new Gateways to the Bot Definition File and reloading the bot, the Gateways are connected and become available in the gateways panel, without disrupting other connections. To enable Connected Mode, click the associated button in the application toolbar. For convenience, if you load a Bot Definition File with Client Gateways available, the application will ask if you'd like to connect them. Upon choosing "Yes", Connected Mode will be enabled.

Image: 4.2. Gateways Available

4.3 Pass Keys

Pass keys provide an extra layer of protection to the Bot Definition File, by encrypting and hiding plain-text Client Gateway passwords. The bot uses the pass key to decrypt the passwords and allow the Gateways to connect. If a Bot Definition File contains encrypted passwords, set the bot's pass key by clicking the Set Pass Key button in the application toolbar. You will be prompted for the bot's pass key (see Image 4.3.A).

Image: 4.3.A. Pass Key Prompt

The JBuddy Bot Builder will also prompt you for a pass key automatically if you load a Bot Definition File that uses encrypted passwords (see Image 4.3.B).

Image: 4.3.B. Pass Key Needed

4.4 Encrypting Passwords

The JBuddy Bot Builder provides a graphical front-end to the command-line Encrypt Password Utility included with the JBuddy Bot Framework. To use it, choose "Encrypt Password..." from the "File" menu (see Image 4.4.A).
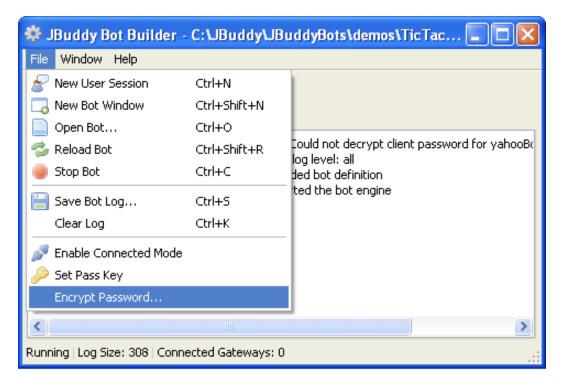

Image: 4.4.A. Encrypt Password Menu Item

The following window will be shown.

Image: 4.4.B. Encrypt Password Utility

To encrypt a password, type a password and an optional pass key. The encrypted value will be shown, and can be conveniently copied to the system clipboard for pasting into the Bot Definition File for use within the client element. You will also need to add the encryptedPassword="true" attribute to the client so that the Bot Engine will recognize that the password is encrypted and prompt you for the Pass Key.

5 Statistics

JBuddy Bot Builder keeps track of certain statistics for a bot and its associated Client Gateways, such as the number of user sessions and sent/received messages.

5.1 The Statistics Panel

The Statistics panel displays global bot statistics, which account for user sessions in all of a bot's Client Gateways. Statistics are updated automatically as the bot runs. Each running bot will have its own global bot statistics within its own Statistics panel.

Note: The FREE version of the JBuddy Developer Tools which includes the JBuddy Bot Framework and JBuddy Bot Builder is limited to three (3) concurrent chat sessions and three actively connected gateways. If you exceed three concurrent chats within a fifteen minute period, you will see a warning message and that user will be unable to interact with the bot until one of the other active users stops interacting with the bot and fifteen (15) minutes of idle (nothing sent or received) occurs. This is a limit only of the FREE version of these tools. Please see the other User's Guides and READMEs for more information on this subject.
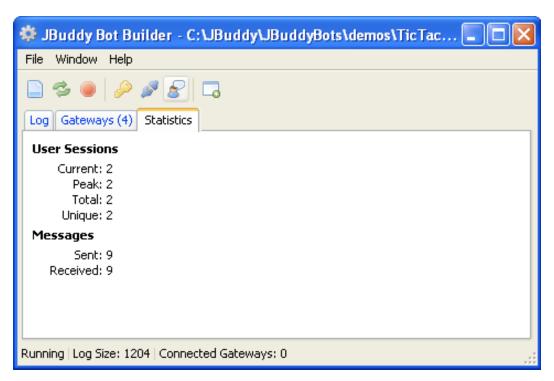

Image: 5.1. The Statistics Panel

5.2 Client Gateway Statistics

Statistics can also be shown for each individual Client Gateway. To view a Client Gateway's statistics, hover over it with your mouse pointer in the gateways panel (see Image 5.2).
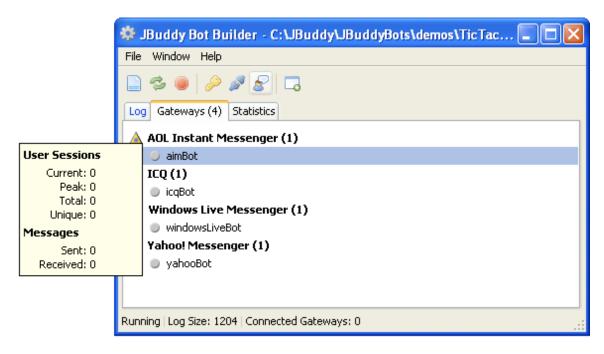

Image: 5.2 Client Gateway Statistics

6 The System Console

The System Console window captures output from the system's standard output and error streams. This includes messages written to Java's System.out and System.err streams, as well as uncaught system exception traces. To open the System Console, choose "Show System Console" from the "Window" menu. Console output is shown in the System Console log. Black signifies standard output messages, while red signifies standard error messages such as exceptions. Log contents can be selected and copied or saved or cleared using the "Save Bot Log..." or "Clear Log" menu items of the File menu respectively, or by right-clicking the log.
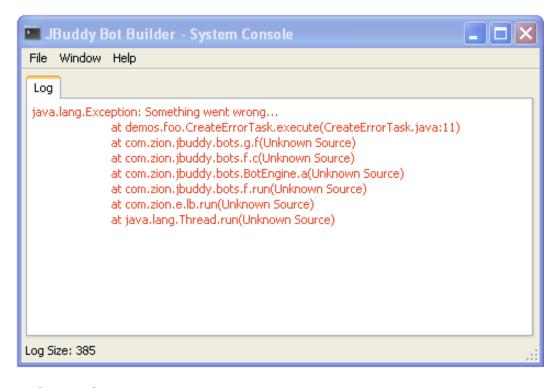
Image: 6. System Console Log