



WebHuddle: User Guide

Version: 0.4.9

Preface	iv
I. How to Use WebHuddle	1
1. Introduction	2
1.1. What is WebHuddle?	2
1.2. Why Use WebHuddle?	2
1.3. Getting Help	2
1.3.1. Problems Using WebHuddle	2
1.3.2. Problems Administrating WebHuddle	3
2. How to Start a WebHuddle Meeting	4
2.1. Create Your Account	4
2.2. Start the WebHuddle	4
2.3. Security Prompt	5
2.4. Invite People	6
3. WebHuddle Meeting Features	7
3.1. Desktop Sharing	7
3.2. Recording the WebHuddle	9
3.3. Promote to Moderator	9
3.4. Slide Presentation	9
3.5. Polling and Questions	10
3.6. Q & A	11
3.7. Chat	12
3.8. VOIP (Voice Over IP)	12
4. Before Your WebHuddle: Meeting Preparation (Optional)	13
4.1. Overview	13
4.2. Upload Slides	13
4.3. Managing Scripts	14
4.4. Scheduling	15
5. After Your WebHuddle: Managing Recordings	17
5.1. Introduction	17
5.2. Recording Preview	17
5.3. Recording Download	17
5.4. Publish to a Web Server	17
5.5. Publish to a Network Drive	18
5.6. Merge an Audio Recording	18
II. How to Administrate WebHuddle	19
6. How to Install a WebHuddle Server	20
6.1. Requirements	20
6.1.1. Hardware	20
6.1.2. Software	20
6.1.3. Network	20
6.2. Installation	21
6.2.1. Windows	21
6.2.2. Linux, UNIX, Solaris and Mac OS X	21
6.2.2.1. Java™	21
6.2.2.2. JBoss™	22
6.2.2.3. WebHuddle	22
6.2.2.4. Xvfb (X Virtual Frame Buffer; Optional)	23
6.2.3. OpenOffice.org™ (Optional; Linux and Windows and Mac)	23
6.3. HTTPS/SSL Configuration (Optional)	24
6.3.1. Create the Keystore and Local Certificate	25
6.3.2. Create the Certificate Signing Request	25
6.3.3. Purchase the SSL Certificate	25

6.3.4. Submit Your Certificate Signing Request (CSR)	26
6.3.5. Retrieve Your Signed Certificate	26
6.3.6. Import the Certificates to your Keystore	26
6.3.7. Copy the Keystore and Configure the SSL Connector	27
7. How to Configure a WebHuddle Server	30
7.1. Overview	30
7.2. Audio	30
7.3. Desktop Sharing	30
8. How to Integrate WebHuddle in the Enterprise	32
8.1. Authentication	32
8.1.1. SQL Database Authentication	32
8.1.2. LDAP Authentication	34
III. How to Hack WebHuddle (Under Construction)	37

Preface

Thank you for using WebHuddle! We hope you will agree that WebHuddle is the cross-platform, 100% open source web conferencing solution that's simple, small and secure. WebHuddle has grown thanks in large part to users who share valuable feedback. Please help WebHuddle continue to improve by sharing [http://sourceforge.net/forum/forum.php?forum_id=391245] what you do and don't like about it, what you would change, or any other suggestions.

This document is intended for all WebHuddle audiences: users, administrators, and developers, and is divided in that order.

For any problem you encounter with WebHuddle that this guide does not address, try searching the WebHuddle project forums [http://sourceforge.net/search/?group_id=114433&type_of_search=forums]. If you don't find an answer there, please post details about your issue on the help forum [http://sourceforge.net/forum/forum.php?forum_id=391246]. Posting in the public forums is the most helpful way to communicate, as it makes your issue and solution available to others. If you want to contact WebHuddle via email, use [info \(at\) webhuddle.com](mailto:info@webhuddle.com)

Part I. How to Use WebHuddle

WebHuddle runs in your web browser and requires no installation. To use it, just go to <https://www.webhuddle.com> [<https://www.webhuddle.com>] or any other WebHuddle server. This section introduces WebHuddle and discusses how to use it. If you want to install your own server, see the section [How to Setup a WebHuddle Server](#).

Chapter 1. Introduction

1.1. What is WebHuddle?

WebHuddle is a web conferencing [http://en.wikipedia.org/wiki/Web_conferencing] solution. Using WebHuddle, you conduct meetings, presentations or webinars over the internet. Once in a WebHuddle, you can:

- Share your desktop: others see you what you see on your computer. Imagine someone looking over your shoulder.
- Display and mark-up presentation slides with highlighting tools.
- Issue questions and polls to keep people on their toes.
- Promote any participant to moderator, allowing them to share their desktop.

To see what you can do with WebHuddle, create an account and try it out!

1.2. Why Use WebHuddle?

WebHuddle is the cross-platform, 100% open source web conferencing solution that's simple, small and secure. We won't drone on about it, but here are some of WebHuddle's advantages:

- Cross-Platform [<https://www.webhuddle.com/xplatformpage.jsp>]: Use any Java-enabled computer (Linux, Windows, Unix, Mac)
- Simple [<https://www.webhuddle.com/simplepage.jsp>]: No installation needed -- client runs in web browser
- Small [<https://www.webhuddle.com/sizepage.jsp>]: Thin client (about 125 KB) -- loads quickly
- Secure [<https://www.webhuddle.com/securitypage.jsp>]: All data encrypted by HTTPS protocol
- Open Source: Leverage the many benefits of open source software, including value, transparency, and flexibility. If you find WebHuddle useful you may download it [<http://sourceforge.net/projects/webhuddle>] and freely install it on your own network

1.3. Getting Help

WebHuddle aims to be very simple and reliable, but if you encounter any problem, here's how to get help.

1.3.1. Problems Using WebHuddle

If you encounter issues when using WebHuddle, try searching on your issue in the WebHuddle project forums [http://sourceforge.net/search/?group_id=114433&type_of_search=forums]. If you don't find an answer there, please post details about your issue on the help forum [http://sourceforge.net/forum/forum.php?forum_id=391246]. Useful details include your operating system, your Java version, hardware details such as CPU speed and memory size. As well, post your Java log file if it seems

to have relevant details. On Windows, the Java plugin log file is located here:

```
<User Application Data Folder>\Sun\Java\Deployment\log
```

'<User Application Data Folder>' is as defined here [http://java.sun.com/j2se/1.5.0/docs/guide/plugin/developer_guide/version.html#property]. On Linux or other UNIX-like operating systems, the Java plugin log file is here:

```
<user.home>/ . java / deployment / log
```

1.3.2. Problems Administrating WebHuddle

For issues with installation, configuration, or other administration tasks, search and post on the same forum [http://sourceforge.net/forum/forum.php?forum_id=391246] as listed above. The information to share for administration problems, however, is different. Share the version of WebHuddle that you are running, hardware info, and configuration and log files. Share any configuration file that you have altered, such as

```
<jboss>/server/default/deploy/jbossweb-tomcat50.sar/server.xml
```

Or:

```
<jboss>/server/default/deploy/webhuddle-app.ear/webhuddle-web.war/WEB-INF/web.xml
```

The JBoss server log file usually has clues regarding any problems, so post the first few error messages from it, if you see any. You can find the file here:

```
<jboss>/server/default/log/server.log
```

Chapter 2. How to Start a WebHuddle Meeting

2.1. Create Your Account

To conduct a WebHuddle, you need an account. To create an account at www.webhuddle.com, browse to here [<https://www.webhuddle.com/editProfile.do?action=Create>]. If your school, business, or other organization has its own WebHuddle server, you may already have an account with the same logon and password that you use to logon to email or other applications. If so, you can logon by going to your WebHuddle server and clicking 'Logon'.

When you create an account, the system will generate a random password for you and email it to you. Retrieve the password from your email and logon. This is a good time to update your profile with a password that is memorable to you. To do so, click the 'Profile' link and enter your new password of choice.

2.2. Start the WebHuddle

As soon as you are logged on to WebHuddle you can start a meeting. Just click the 'Begin Meeting' link. A form like the one shown below will appear.

Open My Conference Room

Meeting Name	<input type="text" value="Demo"/>
Meeting Description	<input type="text" value="Quick demonstration of WebHuddle"/>
Meeting Script:	<input type="text" value="Sprint Training"/>
Meeting Password	<input type="password" value="****"/>
(Repeat) Meeting Password	<input type="password" value="****"/>
VOIP (internet audio)	<input checked="" type="checkbox"/>
Record This Meeting	<input checked="" type="checkbox"/>
	<input type="button" value="Submit"/> <input type="button" value="Reset"/>

Most of the fields are self explanatory.

Note that the password field is not for your WebHuddle account password, but for putting a password on a single meeting.

WebHuddle optionally supports one-to-many voice-over-IP (VOIP), meaning one person may speak at a time and the moderator passes the microphone around.

You can also choose to record the WebHuddle for later playback. Note that if your meeting is being recorded, participants will be informed of this fact before joining your WebHuddle. This is the polite thing to do, and required by law in some areas.

The 'Meeting Script' option is how you would select which set of uploaded slides to use, if any. The section 'Uploading Slides' discusses how to upload PowerPoint and other content before starting your WebHuddle.

2.3. Security Prompt

When you start your WebHuddle, you will probably see a security prompt like this:



...or like this:



This is WebHuddle asking for the security permissions it needs for some of its features. In particular, it needs permission to do the screen scraping for desktop sharing. It also needs permission to open your microphone, if you enabled VOIP for your WebHuddle. For these features to work, you need to grant it permission.

Unlike other web conferencing solutions, however, you do not have to say 'Yes'. If you say 'No' to the security popup, WebHuddle will still work. You can use all functions, except for sharing your desktop or broadcasting audio through your microphone (though you can still hear audio).

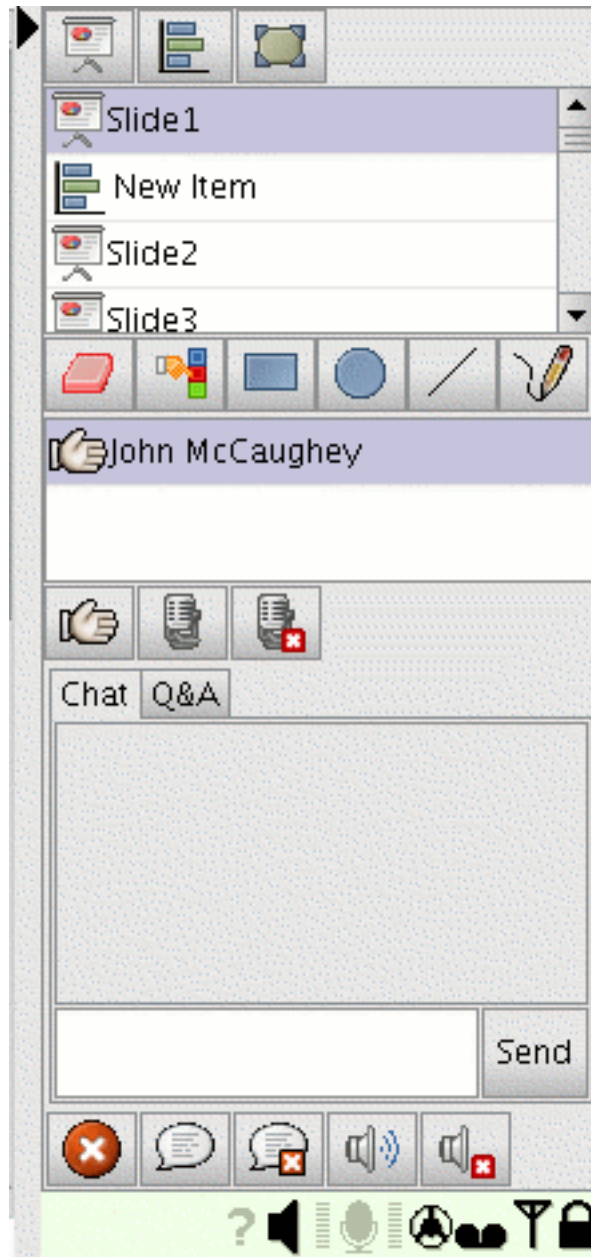
2.4. Invite People

Once you start your WebHuddle, the system will send you an email with a link in it. Forward this email to anyone you want to invite to participate. Invitees just need to click the link or paste it into their browser. They will be prompted for their name or email, and then entered into the meeting. To invite people before the meeting starts, see the section below called 'Scheduling'.

Once people join your WebHuddle, you can start desktop sharing, slide annotation and other features discussed in the next sections.

Chapter 3. WebHuddle Meeting Features

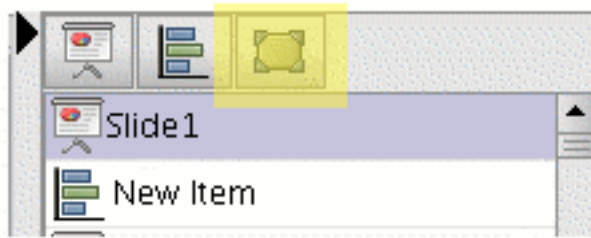
WebHuddle's meeting features include desktop sharing, slide annotation, chat, polling and more. The control area, pictured below, is where the moderator enables the respective features. Here we'll look at how to use the features.



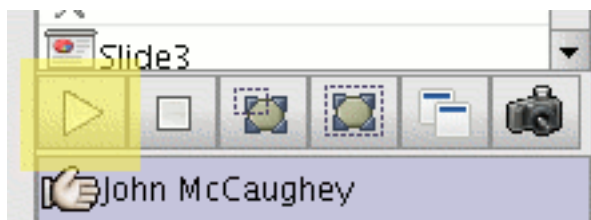
3.1. Desktop Sharing

With desktop sharing you can show anything on your computer to everyone in your WebHuddle. Think of someone looking over your shoulder.

To start sharing your desktop, first click the desktop icon on the top right of the control area. This button is highlighted below.

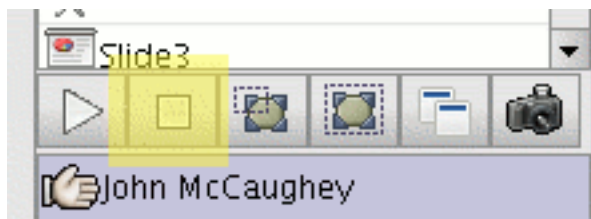


The desktop sharer's view appears in the main content area, with some instructions in the top left. Click the arrow button (pictured below) to start transmitting your desktop to the other participants.

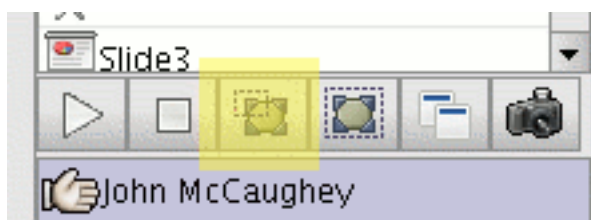


Note that some content doesn't convey well over desktop sharing, such as video or video games. But if you pause your video player or game, still frames may convey well.

To stop sharing your desktop, click the square button pictured below.

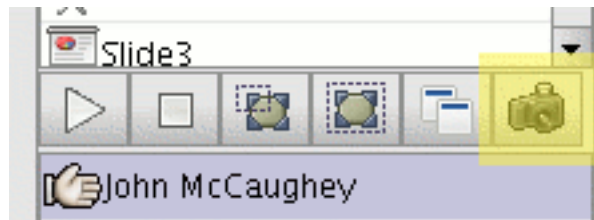


If you don't want to share your entire desktop but instead, just a limited region, click the 'Limited Region' button shown below. A rectangle will appear in the main content area. Drag the bottom or right side of this rectangle to adjust its size, then drag the application you want to share onto the rectangle. WebHuddle will continue transmitting the region where the rectangle was even if you minimize the WebHuddle browser window.



To switch back to sharing you entire desktop, click the 'Entire Desktop' button. Note that the smaller the area that you share, the more quickly screen updates will be sent to participants.

For efficiency, WebHuddle compresses screen updates. The default compression mode is called GIF and it works well for showing applications such as editors and spreadsheets. If you are sharing photographs or other content with many colors, try switching to JPEG mode. Do this by clicking the camera icon pictured below.



If you want another participant to share their desktop, you can promote them to moderator. The next section describes how to promote any participant to moderator.

3.2. Recording the WebHuddle

To record your WebHuddle, enable the recording option when starting or scheduling it. Once all your participants join in and you are ready to start the meeting proper, click the Start/Stop Recording button pictured below.



To see if recording is currently on or off, check the recording status icon highlighted below.



3.3. Promote to Moderator

Unlike most other web conferencing solutions, with WebHuddle you don't need to install special software in order to host a meeting. Thanks to this, you can promote any participant to moderator at any time during your WebHuddle. As the original moderator, you can always take control back.

To promote a participant to moderator, select them from the roster and then click the 'Make Moderator' (with hand icon) button just below the roster. The hand icon will appear next to their name, indicating that they are now the moderator. That participant can now share their desktop, annotate the slides, control the microphone and other features.

To take meeting control back, select yourself in the roster and click the same 'Make Moderator' button.

3.4. Slide Presentation

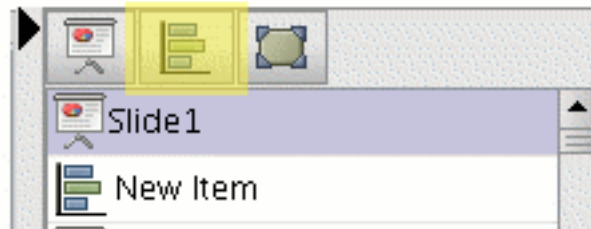
Before your WebHuddle, you can create content such as presentation slides or pre-authored questions and polls. The list of meeting content is just above the roster. If you are not using any slides, one blank slide appears in the content list. You can use this blank slide as a whiteboard.

To advance to a slide, select it in the content list. To highlight a slide, hold down your mouse button and drag your pointer over it. To use a different annotation color or tool, click one of the buttons shown below.



3.5. Polling and Questions

The most effective WeHuddle is when your participants actually participate. To engage your participants, issue a question (also called 'poll') every now and then. You can create polls before your WebHuddle, during it, or both. To create a question during the WebHuddle, click the 'Questions' button at the top of the control area, as pictured below.



Once in the 'Questions' view, the two buttons below will appear in the control area.



If you click the 'New Question' button, a form for authoring your question appears in the main content area. Other participants will not see the question until you finish creating it. You can choose different types of questions, such as True/False, Multiple Choice, and Short Answer. Below is a sample question, before being issued.

New Question:

☒ Multiple Choice
☐ True / False
☐ Yes / No
☐ Short Answer

Question:
Who won the 2006 World Cup?

☐ No correct answer
☐ A Germany
☐ B Brazil
☒ C Italy
☐ D France

Issue Question

When you finish creating your question or poll, hit the 'Issue Question' button. The question will be issued to all participants, including the moderator (you), so confirm it was entered correctly and answer it. Then, as responses are received, a running tally will be displayed of how many are received out of how many total parti-

participants. To end the question, hit the 'End Question' button in the control area. When you end the question, a bar graph will be displayed to all participants, showing the correct answer, if there was one, and the numbers and percentages for each choice. Below is an example:

The screenshot shows a 'Results' window for a poll. At the top, it says 'Results:'. Below that is the 'Question:' field with the text 'Who won the 2006 World Cup?'. There are five radio button options: 'No correct answer', 'A 0/1 (0%) Germany', 'B 1/1 (100%) Brazil', 'C 0/1 (0%) Italy', and 'D 0/1 (0%) France'. The 'Brazil' option is selected, indicated by a filled radio button and a blue bar graph next to it.

3.6. Q & A

Like the polling/questions feature, the 'Questions and Answers' feature facilitates interaction between the moderator and the participants. But this feature goes in the other direction, allowing participants to submit questions or comments to the presenter. To submit a question, any participant simply clicks on the Q&A tab in the bottom part of the control area, shown below.

The screenshot shows the 'Q&A' tab selected in the control area. It has a 'Chat' tab next to it. Below the tabs is a large text input field. At the bottom of the input field, the text 'What is the early termination fee?' is visible. Below the input field are two buttons: 'Next' and 'Share'. At the very bottom is a 'Submit Question' button.

When the participant types their question and then presses the 'Submit' button, a question pending status indicator light goes on in the bottom of the control area, highlighted below.



When you see that participants have submitted questions, you can address them at your convenience. They remain in the order received, and you simply click the 'Next' button to view the submitted question. If you want

to address the question, click the 'Share' button. This will show the question, and who asked it, to all participants. Answer the question verbally. If the question pending status light is still on, click 'Next' again to see additional questions. When the light goes out, there are no more questions pending.

3.7. Chat

On the same panel as the Q&A tab is a Chat tab. Any participant can send text to the entire WebHuddle. Currently, there is no private chat. The moderator can disable and enable chat.

If you want to show a web page to your participants, you can do this through desktop sharing. Participants cannot bookmark any site your browse to, however, as they are only seeing images of it from your computer. To send the entire WebHuddle to a web site, type the URL into chat. Tell participants to double-click on the URL in the chat tab, and a browser will pop up on that page. Now people can bookmark the site and navigate it independently.

Note that some browsers may block the popup function. A future version of WebHuddle will offer a different way to share web content that both avoids the pop-up restriction and removes the need for participants to double click anything.

3.8. VOIP (Voice Over IP)

While a traditional phone conference will usually give the best audio performance, for some WebHuddles you may want to use the one-to-many VOIP option. One person may speak at a time, and the moderator controls the microphone. To give someone the microphone, you select them from the roster and then click the 'Give Microphone' button. To take the microphone back, either click the 'Mute Microphone' button, or select someone else from the roster and give them the microphone.

By default, the speaking person can hear themselves. To turn this monitoring function off, the speaking person clicks the 'Mute my speaker' button on the bottom right of the control area.

Chapter 4. Before Your WebHuddle: Meeting Preparation (Optional)

4.1. Overview

Once you log on to WebHuddle, you can start your meeting right away by clicking the 'Begin Meeting' link. You do not need to upload content, schedule the meeting, install software, or anything else. Some meeting preparation, however, can go a long way toward making your WebHuddle more productive. Preparation includes uploading PowerPoint slides or JPEG images, authoring polls, and scheduling. The next section will discuss how to upload and organize your content before a WebHuddle.

4.2. Upload Slides

WebHuddle supports uploading Microsoft PowerPoint, OpenOffice Impress, GIF, JPEG and other files types before your meeting. Each upload creates a 'script'. Items from one script can be copied into other scripts.

While desktop sharing allows you to show anything on your computer, sometimes it makes sense to upload your content before the WebHuddle. The reasons include:

- Faster performance: desktop sharing uses relatively more bandwidth and can take several seconds to update, depending on the network bandwidth of the presenter and participants
- Annotation: uploaded slides can be marked-up with the highlighting tools
- Better presentation flow: a few well-chosen screen shots can be more effective than making your participants wait while you find and start different applications

To upload slides, log on to your WebHuddle account and click the 'Upload' link. The following form appears:



The form contains the following fields and buttons:

- Content Name**: A text input field.
- Select a .ppt or .sxi file (or a .zip file containing GIF/JPEG images)**: A text input field followed by a **Browse...** button.
- Scale slides**: A dropdown menu with **No scaling** selected.
- Submit** and **Reset**: Two buttons at the bottom.

When you upload content, WebHuddle creates a 'script'. A script is simply a list of slides that can be used for a WebHuddle. When starting or scheduling your WebHuddle, you choose which script to use. The 'Content Name' field in the upload form will be the name of your script.

The 'Select a .ppt...' field is for the actual file to upload. Click the 'Browse...' button to find the file on your computer.

The last field optionally tells WebHuddle to shrink the slides so that participants using laptops or other small screens can see the slides in their entirety. The choices range from 320x240 to 1024x768 pixels.

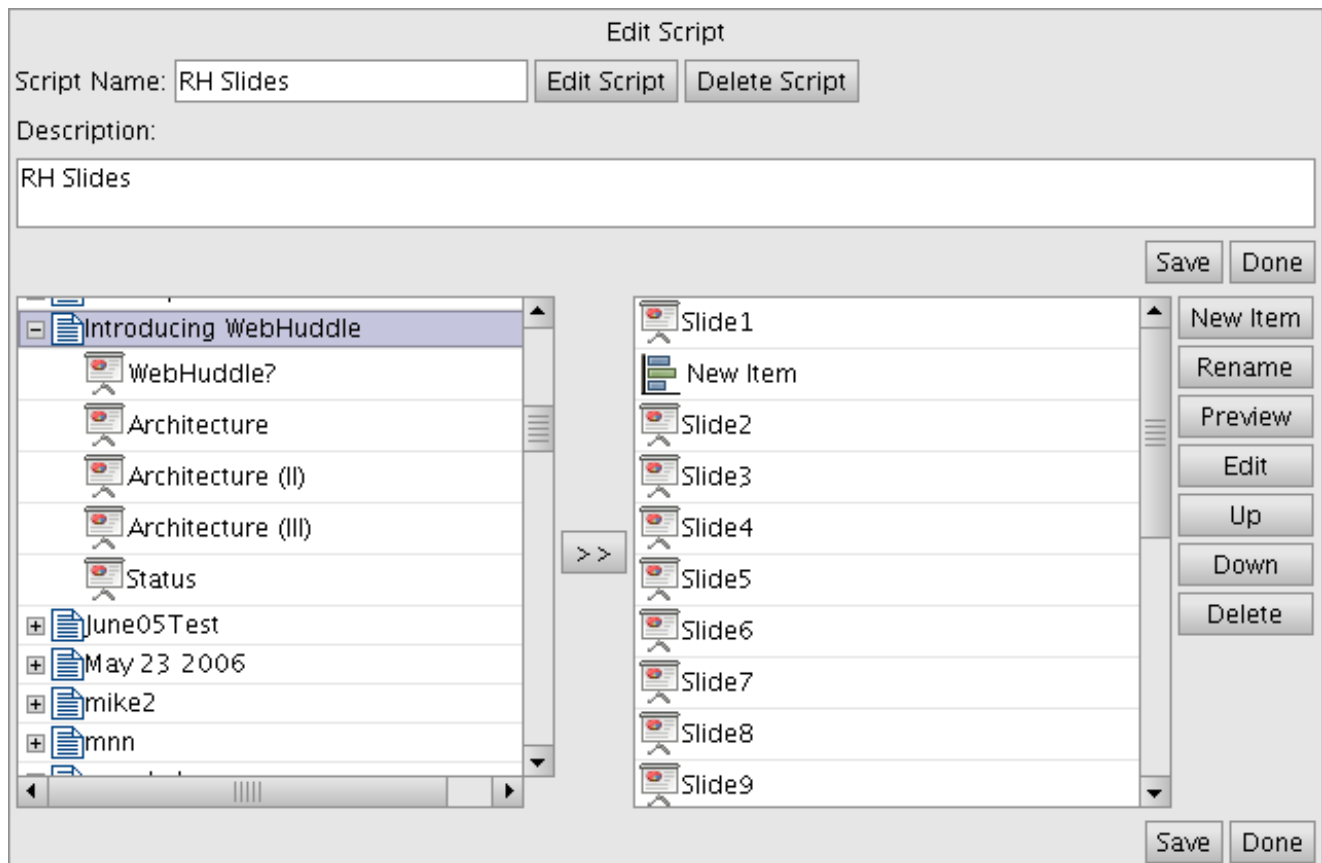
When you click the 'Submit' button, it may take up to a few minutes to upload your content, depending on its

size and the speed of your internet connection. Wait until you see a confirmation message saying that your upload has been processed.

The next section, 'Managing Scripts', explains tasks such as copying items from one script into another, removing or renaming items, inserting questions, and more.

4.3. Managing Scripts

When you upload a PowerPoint file, WebHuddle creates a script, with each slide becoming a single item in that script. To edit or preview a script, click the 'Scripts' link. The interface shown below appears:



The name and description of the current script appears in the top area, and the individual slides in the script appear in the lower right area. The lower left area shows all scripts and their respective slides, so you can copy items from one script into another. For example, you could select the 'Arhitecture (II)' slide in the left area and then click the '>>' button to copy that slide into the current script.

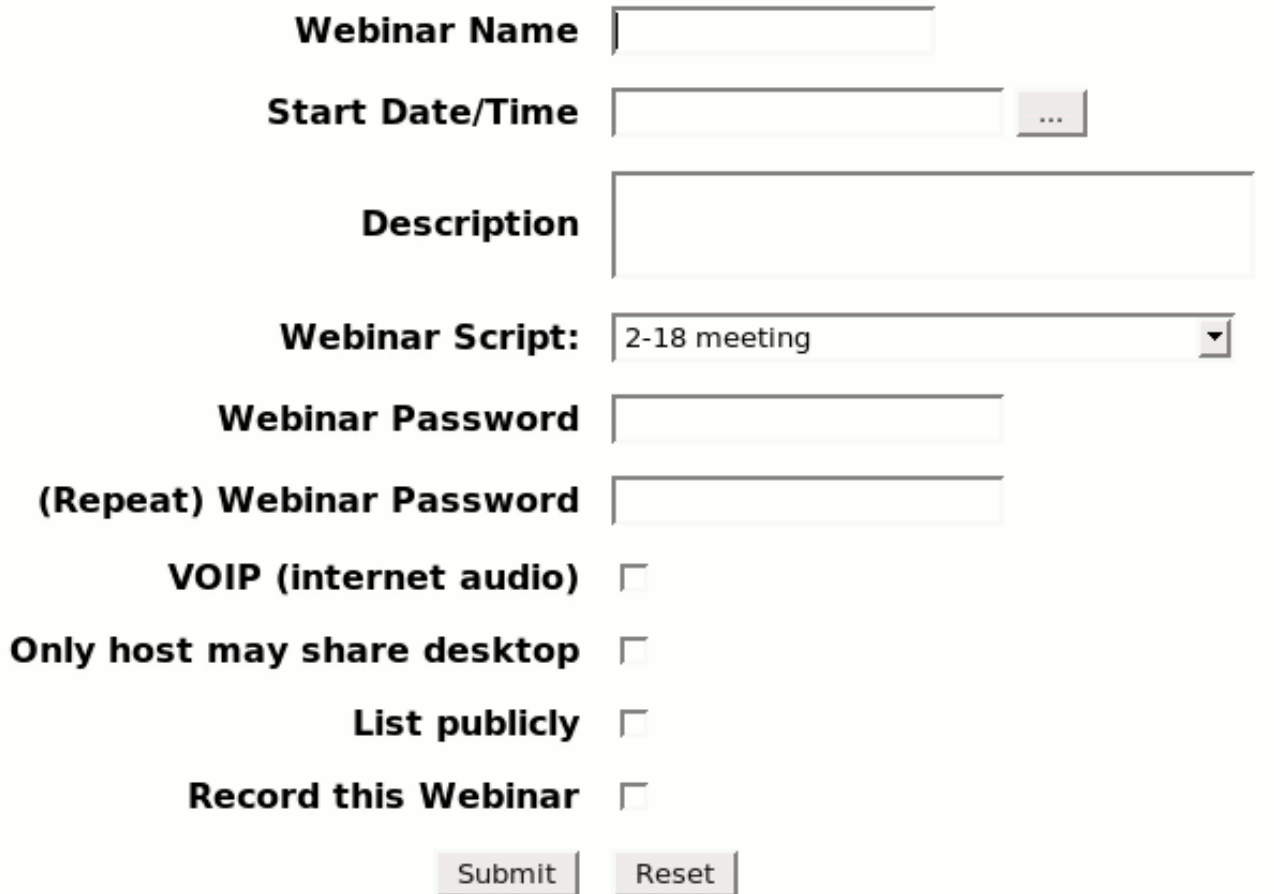
To rename, re-order or delete slides in the current script, select the slide in the right area and click the 'Rename' or other button on the right. You can also insert questions or polls into your script. Just click the 'New Item' button, then type in a name for the question and choose 'questions' as the type. Your new question appears in the script: to actually enter the question, select it in the scripts and click the 'Edit' button. Editing your question here is the same as editing a question on-the-fly, during a live WebHuddle, as described in the Polling and Questions section.

When you finish editing your script, be sure to click the 'Save' button. To edit another script, click the 'Edit Script' button at the top.

4.4. Scheduling

Starting with version 0.4.9, WebHuddle supports some minimal scheduling functions. Basically, you can create a WebHuddle in advance and obtain the link that participants will use to join in when you start it.

To schedule a WebHuddle, click the 'Schedule' link. If you have any pending, scheduled events, they appear in a list. To schedule a new event, click 'Schedule a new WebHuddle.' The scheduling form appears as shown below.



The screenshot shows a web form for scheduling a WebHuddle. The form is titled 'WebHuddle' and contains the following fields and controls:

- Webinar Name:** A text input field.
- Start Date/Time:** A date and time picker with a calendar icon.
- Description:** A large text area for entering details.
- Webinar Script:** A dropdown menu with '2-18 meeting' selected.
- Webinar Password:** A text input field.
- (Repeat) Webinar Password:** A text input field.
- VOIP (internet audio):** A checkbox.
- Only host may share desktop:** A checkbox.
- List publicly:** A checkbox.
- Record this Webinar:** A checkbox.
- Submit:** A button.
- Reset:** A button.

This form is almost the same as the one for creating an unscheduled meeting, described above, except it has two additional fields.

The 'Start Date/Time' field lets you select a date and time for your WebHuddle. Note that this field is for convenience only: you can start your meeting early or late.

The 'List publicly' checkbox means that you want anyone to be able to see the name entry info for your WebHuddle. This is mostly for use when integrating WebHuddle with other applications, such as Content Management Systems (CMSs) or portals. For example, you could have a page in your portal that lists upcoming webinars. To learn more about integrating WebHuddle with other applications, see the section WebHuddle Integration.

After submitting the form, your new WebHuddle appears in the list of scheduled events. Click on it and it will appear with some links near the top, as shown below.

[Start this webinar now](#)

Link for participants to join this webinar (right click to copy):

<https://www.webhuddle.com/j.do?key=808e32777f000001008a93d94fff3ddc&mID=13582>

Webinar Name	<input type="text" value="App Share 2.0"/>
Start Date/Time	<input type="text" value="2007.03.26, 2:00PM"/> <input type="button" value="..."/>
Description	<input type="text" value="Introduction to the new desktop sharing optimizations. "/>

Right click or copy and paste the URL into your scheduling or email software. Anyone can use the link to join the meeting, once you have started it. To start it, select it from the list of scheduled WebHuddles and then click the 'Start this webinar now' link. As with unscheduled events, participants can only enter the event once you, the moderator, have entered it.

Chapter 5. After Your WebHuddle: Managing Recordings

5.1. Introduction

Upload a recorded WebHuddle session to any standard web server and the whole world can see it, in the exact same form as it happened in the live meeting. Like live WebHuddle sessions, viewers only need a Java-enabled browser. On the server side, you don't need a WebHuddle server or any other software. The WebHuddle archive files are fully self-contained, static files. The following sections discuss how to preview, upload, and other recording tasks.

5.2. Recording Preview

To preview any recorded WebHuddle, logon to your WebHuddle account and click the 'Recordings' link. You'll see a list of your recorded sessions and their dates. Click the 'Details' link on one of your recordings. Then click the replay link. If you want to share this recording with others, see the next sections.

5.3. Recording Download

From the 'Recording Details' page, you can download a zip file of your WebHuddle. Just click the 'Download' link and note where you save the file. The file name will be something like 'WHrec_3.zip'. Copy the files in this archive to any web server and people can view your recorded WebHuddle just like any web page. The next sections explain different options for publishing your recording.

5.4. Publish to a Web Server

To publish a recorded WebHuddle to a web server, simply unzip the WebHuddle archive and upload the files to a public directory on your web server. There are many ways to upload files to a web server, but for this example we'll use ftp (file transfer protocol) from the command line. We'll assume you have created a local directory called 'c:\recordings' and unzipped the downloaded WebHuddle archive into it.

```
cd c:\recordings
dir /B
WHrec_3.zip
3
index.php
cd 3
ftp
ftp> open mywebserver.com
User: mylogin
Password: mypassword
ftp> cd public_html
ftp> mkdir upload_demo
ftp> cd upload_demo
ftp> bin
ftp> mput *
mput event.zip? y
...
ftp> bye
```

Be sure to replace the number '3' above with the actual number from your recording archive. Answer 'y' to all the prompts you get after entering the input command. Your recorded WebHuddle is now available at a URL such as:

```
http://mywebserver.com/upload_demo/file_playback.html.
```

5.5. Publish to a Network Drive

Publishing a WebHuddle to a network drive is as easy as unzipping the archive and dragging and dropping the folder. Wherever you put the files, the `file_playback.html` file is the one to reference for playback. In other words, say you put the WebHuddle files here:

```
s:\recordings\3\
```

The link for playing back that recording would be:

```
file:///S:/recordings/3/file_playback.html
```

5.6. Merge an Audio Recording

Because WebHuddle currently only supports one-way audio, many people prefer to use a phone conferencing service for the audio portion of their meeting. If your audio conferencing service supports recording, you can merge the audio recording file with your WebHuddle recording. Right now, WebHuddle only supports a limited number of audio file formats, but will support more formats soon. The formats supported include .wav files with PCM or AU encoding, at 8Khz (8,000 samples per second). freeconferencecall.com [<http://freeconferencecall.com>] and other services use this format.

To merge an audio recording, go to the 'Recordings' page and then click the 'Details' link for the WebHuddle you want. Then click the 'Merge Audio Recording' link. Click the 'Browse' button to find your audio recording. The offset field is the number of seconds that the audio recording started after the WebHuddle recording start. So, if you clicked the 'Start/Stop Recording' button in WebHuddle at 12:05:30 and started the audio recording at 12:05:40, enter '10' for the offset seconds.

After clicking the 'Submit' button, it may take a minute or two to upload and merge your recording. When the process completes, you'll see a message saying that your file was merged successfully. You can now preview the merged recording by playing it back. If the audio is not synchronized correctly, you can merge it again, specifying more or less offset.

Part II. How to Administrate WebHuddle

The following sections describe how to install, configure, and integrate a WebHuddle server. Most people will not need to read this, as you typically use WebHuddle simply by pointing your browser to an existing WebHuddle server. For general information on using WebHuddle, see the section [How to Use WebHuddle](#). If you really do want your own WebHuddle server for your organization or personal use, read on.

Chapter 6. How to Install a WebHuddle Server

6.1. Requirements

The key idea to understanding the requirements for a WebHuddle server is that the more concurrent participants you expect, the more resources you will require. This is most true in the areas of network bandwidth and memory. If you expect to have a maximum of 10 people participating in WebHuddles at any given time, the minimum requirements listed below suffice. But if you want to support hundreds of concurrent participants, you will require more CPU power, but especially, more network bandwidth and memory.

6.1.1. Hardware

- *Memory*: 512 MB of RAM (minimum), 1 GB+ (recommended). Estimate about 256 KB of RAM per concurrent participant.
- *Free Disk Space*: 4 GB (minimum), 40 GB+ (recommended for recorded WebHuddle sessions)
- *CPU*: 500Mhz (minimum), 1 Ghz+ (recommended). Any processor type supported by the operating systems listed below. For example Intel, AMD or other x86, PowerPC, SPARC, etc.

6.1.2. Software

- *Operating System*: Any OS that supports the Java 1.4 (or higher) runtime, such as Linux, FreeBSD, Solaris, Mac OS X, or Windows
- *JBoss 3.2.8*: Available for download from here [http://sourceforge.net/project/showfiles.php?group_id=22866&package_id=16942&release_id=398369] Note: not needed if you are using the WebHuddle .exe installer for Windows.
- *Java Development Kit*: Sun, IBM, or other JDK version 1.4 or higher, available for download from here [<http://java.sun.com/javase/downloads/index.jsp>]. Note: not needed if you are using the WebHuddle .exe installer for Windows.
- *OpenOffice.org (optional)*: If you want to support uploading PowerPoint or OpenOffice Impress files, you need to install OpenOffice.org 2.0 or higher, available for download from here [<http://www.openoffice.org/>].
- *Xvfb (optional)*: If you want to run WebHuddle without a GUI ('headless') on Linux or other X server compatible OS, you need the X window virtual framebuffer (Xvfb) software. If you have the 'apt' package manager, do `apt-get install xvfb` This will probably pull down X server dependencies, but you will not actually need to run X -- you can still run headless.

6.1.3. Network

- *30-100kbps internet bandwidth per participant*: The more internet bandwidth you have, the more concurrent WebHuddle participants you can support. Note that many consumer internet services limit the 'upstream' bandwidth to a fraction of the advertised speed. For example, a typical ADSL service will offer 1.5 Mbps

downstream but only 384kbps upstream.

Your internet connection is typically the most important factor for determining the capacity of your WebHuddle server. With WebHuddle, all data goes through the server; no data is sent directly from participant to participant (peer-to-peer). WebHuddle's server-centric architecture offers several advantages, including better security and complete firewall and proxy transparency, but it does require more bandwidth on the server. So, if you have an ADSL connection with your upstream speed capped at 512 kilobits per second, do not expect good performance in a WebHuddle with 20 people.

If you want to support large WebHuddles but do not have enough bandwidth, consider a service such as RimuHosting.com [<http://rimuhosting.com>]. Rimu Hosting offers virtual private servers appropriate for WebHuddle for about 50 USD per month, and dedicated servers starting at about 120 USD per month.

6.2. Installation

6.2.1. Windows

To install WebHuddle on Windows simply download the installer .exe file from here [http://sourceforge.net/project/showfiles.php?group_id=114433] and run it. You do not need to get JBoss or Java separately as the installer includes them. When the installer finishes, it will open a web browser to the WebHuddle landing page. Click 'Register' to create an account for yourself and then 'Begin Meeting' to try it out.

The only additional steps are both optional: install and configure OpenOffice; configure a commercially signed SSL/HTTPS certificate. If you want your WebHuddle server to support the uploading of PowerPoint slides, see below for details on installing and configuring OpenOffice. If you don't want participants to see any prompts regarding HTTPS certificates, see this section about getting a commercially signed SSL certificate.

6.2.2. Linux, UNIX, Solaris and Mac OS X

Unfortunately, the Linux/UNIX install of WebHuddle is not yet as simple as the Windows install. We hope to make it easier soon, probably via .deb and .rpm packages. For now, however, you need to install Java, then JBoss 3.2.8, and then WebHuddle. As options, also install OpenOffice (to support PowerPoint uploads) and Xvfb (to run headless).

Java™

There are several ways to install Java™, but first check if you already have it. In a command prompt, type:

```
java -version
```

If your computer has Java™, you will probably see output like this:

```
java version "1.6.0"  
Java(TM) SE Runtime Environment (build 1.6.0-b105)  
Java HotSpot(TM) Client VM (build 1.6.0-b105, mixed mode, sharing)
```

If you do not have the Java Development Kit (JDK), we recommend downloading the 'self-extracting file' from here [<http://java.sun.com/javase/downloads/index.jsp>] and using these instructions [<http://java.sun.com/javase/6/webnotes/install/jdk/install-linux.html#self-extracting>] to install it. For simplicity, prefer the self-extracting binary over any .rpm or .deb package because this way you avoid trampling any other

software and don't need root access.

It doesn't really matter where you put the JDK as long as you remember where it is. For the following examples, we'll pretend you installed it to `/home/myname/jdk1.6.0`.

JBoss™

The version of JBoss that WebHuddle supports is 3.2.8. Download it from here [http://sourceforge.net/project/showfiles.php?group_id=22866&package_id=16942&release_id=398369] and then unzip or untar it. To confirm that you have a working JBoss install, try starting it up. Assuming you extracted Java to `/home/myname/jdk1.6.0` and JBoss to `/home/myname/jboss-3.2.8.SP1`, try the following commands:

```
export JAVA_HOME=/home/myname/jdk1.6.0
ln -s /home/myname/jboss-3.2.8.SP1 /home/myname/jboss
cd /home/myname/jboss/bin
chmod +x run.sh
./run.sh
```

You should see a lot of output and finally a message saying that the server started, such as:

```
11:16:07,596 INFO [Server] JBoss (MX MicroKernel) [3.2.8.SP1 (build: CVSTag=JBoss_3_2_8.SP1 date=200
```

WebHuddle

To deploy WebHuddle on Linux or other UNIX-like operating systems, first download the binary and extract the contents. Among the extracted folders will be one called `webhuddle-app.ear`. You will move this entire directory to the JBoss deploy directory. As an example, do these steps with commands like this:

```
rm -fr /home/myname/jboss/server/default/deploy/webhuddle-app.ear
unzip webhuddle-v0_4_9rc19-bin-20070409.1423.zip
cd webhuddle-v0_4_9rc19-bin-20070409.1423
mv webhuddle-app.ear /home/myname/jboss/server/default/deploy
```

Next, you replace the default Tomcat configuration file, deploy the default WebHuddle configuration file and deploy the self-signed, test SSL keystore with commands like this:

```
cp server.xml.sample ~/jboss/server/default/deploy/jbossweb-tomcat50.sar/server.xml
cp webhuddle.properties.sample ~/jboss/bin/webhuddle.properties
cp test-ssl-keystore ~/jboss/server/default/conf
```

Because this SSL certificate is not a commercial one but just a test, self-signed one, participants will receive security warnings when connecting to your WebHuddle server. To avoid this, see the following section 'HTTPS/SSL Configuration'.

If you are running in a non-graphical environment, like `runlevel 2` or `3`, you may need to add the following line to the JBoss startup script (right above the `'# If JAVA_OPTS is...'` line):

```
JAVA_OPTS=-Djava.awt.headless=true
# If JAVA_OPTS is not set try check for Hotspot
```

That's it -- you can now start up JBoss. To get a clean start first stop any running instance and then restart JBoss, as follows:

```
pkill -f jboss
cd ~/jboss/bin
export JAVA_HOME=/home/myname/jdk1.6.0
./run.sh
```

After seeing the `Started in 18s:913ms` message you should be able to create a WebHuddle account and start a meeting. If you encounter any problems, review the above steps and then see the 'Getting Help' section above.

Xvfb (X Virtual Frame Buffer; Optional)

If you want your WebHuddle server to run headless and support PowerPoint uploads, you may need to install Xvfb. This software gives applications a 'dummy' graphics context to use when the server is actually running in non-graphical mode. If your server will run in non-graphical mode, and you are installing OpenOffice.org, you need to configure it to use Xvfb.

To get Xvfb, use your package manager. For example, if you use a Debian-based distro such as Ubuntu, type:

```
sudo apt-get install xvfb
```

This may have some X server dependencies that will be required too, but remember that you will not need to actually run X. To start xvfb, type something like this:

```
/usr/X11R6/bin/Xvfb :1 -screen 0 800x600x24 &
```

6.2.3. OpenOffice.org™ (Optional; Linux and Windows and Mac)

If you want your WebHuddle server to support uploading PowerPoint files, you will need to install and configure OpenOffice.org.s

To install OpenOffice.org using apt, try something like this:

```
sudo apt-get install openoffice.org
```

To install OpenOffice.org on Windows, simply download and run the installer .exe file. Once OpenOffice.org is installed, the configuration steps are the same for Linux, Windows, or any other operating system. The file to edit is

`openoffice2.0.org/share/registry/data/org/openoffice/Setup.xcu`

Find the section `<node oor:name="Office">` and add the following to it:

```
<prop oor:name="ooSetupConnectionURL" oor:type="xs:string">
  <value>socket,host=localhost,port=8100;urp;</value>
</prop>
```

If OpenOffice is running, you need to close it and restart for the settings to take effect. If you are running on Windows, you may need to check your system tray on the bottom right and explicitly close the OpenOffice quick start task. If you are running with xvfb, use a command like the following to start OpenOffice and tell it to use the xvfb graphics environment:

```
/opt/openoffice.org2.0/program/soffice -headless -norestore -invisible -display :1 &
```

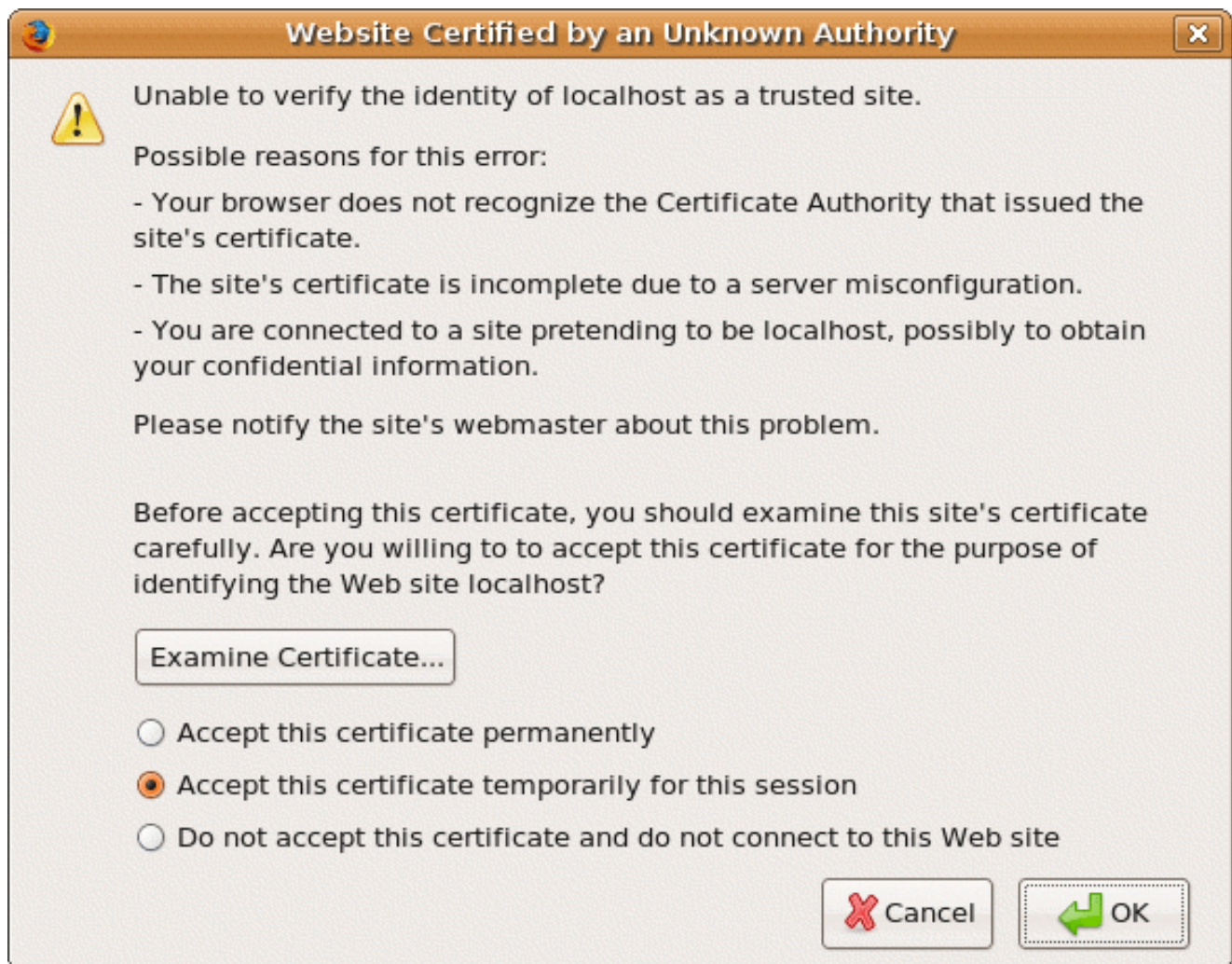
Once OpenOffice is started, confirm that it is listening on port 8100 by typing the following:

```
netstat -an  
OR  
netstat -an | grep 8100
```

The output should include a reference to port 8100. If it does not, OpenOffice is not configured properly for use with WebHuddle. Confirm that your Setup.xcu file is configured properly (see above) and restart OpenOffice. If it still doesn't work, see the 'Getting Help' section above.

6.3. HTTPS/SSL Configuration (Optional)

Because the HTTPS/SSL certificate that ships with WebHuddle is simply a test certificate, users will likely see a security dialog like the following when connecting to your WebHuddle server:



This may be acceptable if you are only using WebHuddle casually or for testing, but otherwise, you probably want the warning to go away. The bad news is that this costs money. The good news is you can purchase commercially signed SSL certificates for about 20 USD. In the following example we'll walk through the process, using godaddy.com [<http://godaddy.com>]'s 'Turbo SSL' offering.

6.3.1. Create the Keystore and Local Certificate

The first step is to create a 'keystore' and populate it with one certificate. The following command accomplishes this:

```
mkdir godaddy
cd godaddy
keytool -genkey -alias tomcat -keyalg RSA -keystore godaddy.keystore
```

When prompted for your first and lastname, enter the domain name of your web site, such as `www.mysite.com`. When prompted for a password, choose one and be sure you can remember it. You will be prompted for a password for the keystore, and later for another password for the certificate you are creating. Tomcat requires that these two passwords *must* be the same, so press 'Enter' in response to the second prompt to use the same password as the keystore.

Here is a sample session after running the above keytool command:

```
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  www.webhuddle.com
What is the name of your organizational unit?
  [Unknown]:  Web Conferencing
What is the name of your organization?
  [Unknown]:  WebHuddle, Inc.
What is the name of your City or Locality?
  [Unknown]:  Mountain View
What is the name of your State or Province?
  [Unknown]:  California
What is the two-letter country code for this unit?
  [Unknown]:  us
Is CN=www.webhuddle.com, OU=Web Conferencing, O="WebHuddle, Inc.", L=Mountain View, ST=California, C=
  [no]:  yes

Enter key password for <tomcat>
      (RETURN if same as keystore password):
```

6.3.2. Create the Certificate Signing Request

The second step is to create a certificate signing request ('CSR'). For this, you also use `keytool`. Make sure you are in the directory containing your keystore created in the step above, and run this command:

```
keytool -certreq -keyalg RSA -alias tomcat -file certreq.csr -keystore godaddy.keystore
```

This command will write a file `certreq.csr` that you will later open in a text editor and copy and paste into the `godaddy.com` SSL request web page. To complete the process you will later also need this same keystore, so now would be a good time to make a backup copy of it.

6.3.3. Purchase the SSL Certificate

To purchase an SSL certificate from `godaddy.com`, you basically give your payment information (they accept credit cards, PayPal and other options) and they give you a credit towards buying the certificate. Note that their process relies on the information in the WHOIS record for your domain.

To verify that you actually control the domain that you are buying the SSL certificate for, `godaddy.com` sends

an email to the 'Administrative Contact' listed in the domain's WHOIS record. So before requesting your certificate, make sure the email listed in your WHOIS domain record is correct. If it isn't, update it with your domain registrar and then wait a few hours for the change to propagate.

When prompted for the type of web server you are using, always choose 'Tomcat'.

6.3.4. Submit Your Certificate Signing Request (CSR)

After paying for your certificate, you will later see a web page on godaddy.com prompting you for your CSR. Open your `certreq.csr` file, created above, in a text editor. Copy and paste the entire contents into the CSR field on the godaddy.com web page. In the prompt near the bottom for the type of web server you are using, choose 'Tomcat'.

6.3.5. Retrieve Your Signed Certificate

After submitting your certificate request, you will receive an email from godaddy.com with a subject line: 'Your SSL Certificate Has Been Issued'. The email contains a link for downloading your certificate. Download and un-ZIP the archive into the `godaddy` directory you created earlier. The actual download is a ZIP file containing your certificate and some other certificates. The other certificates include 'gd_intermediate.crt' and 'gd_cross_intermediate.crt'. The name of your certificate is your domain plus '.crt'.

Your godaddy directory should now contain the original keystore you created to generate the certificate signing request, plus the files from the ZIP file. For example:

```
johnm@johnm-laptop:~/godaddy$ dir
144015187101ac7f59e10112119202b11465294288232814917754080.zip  gd_intermediate.crt
certreq.csr                                                         godaddy.keystore
gd_bundle.crt                                                       godaddy.keystore.bak
gd_cross_intermediate.crt                                           www.webhuddle.com.crt
```

6.3.6. Import the Certificates to your Keystore

You are almost done with setting up SSL! You just need to run a few more commands and the edit a configuration file. First, import the two intermediate certificates to your keystore. These certificates are in your godaddy directory as shown above. Enter this keytool command as a single line:

```
keytool -import -alias cross -keystore godaddy.keystore \
-trustcacerts -file gd_cross_intermediate.crt
```

Then enter this command, also as a single line:

```
keytool -import -alias cross -keystore godaddy.keystore \
-trustcacerts -file gd_intermediate.crt"
```

Finally, enter the following command to import your domain certificate. Be sure to replace `www.mydomain.com.crt` with your domain name!

```
keytool -import -alias tomcat -keystore godaddy.keystore \
-trustcacerts -file www.mydomain.com.crt
```

The output from this last command should say something about private and public key pair being successfully merged. You are now done preparing your SSL keystore and simply need to copy it into your JBoss directory and edit a configuration file, as described next.

6.3.7. Copy the Keystore and Configure the SSL Connector

You can put your keystore anywhere, but we suggest somewhere in your JBoss directory:

```
cp godaddy.keystore ~/jboss/server/default/conf
```

Now you will edit, or replace, the following file:

```
~/jboss/server/default/deploy/jbossweb-tomcat50.sar/server.xml
```

You can either edit the file directly or replace it with `server.xml.sample`, which ships with WebHuddle. Either way, the final result's SSL section should look something like this:

```
<!-- SSL/TLS Connector configuration -->
<Connector
    port="443"
    address="{ jboss.bind.address}"
    maxThreads="300"
    minSpareThreads="25"
    maxSpareThreads="25"
    maxKeepAliveRequests="1"
    scheme="https"
    secure="true"
    clientAuth="false"
    sslProtocol = "TLS"
    keystoreFile="{ jboss.server.home.dir}/conf/godaddy.keystore"
    keystorePass="changeit"
/>
```

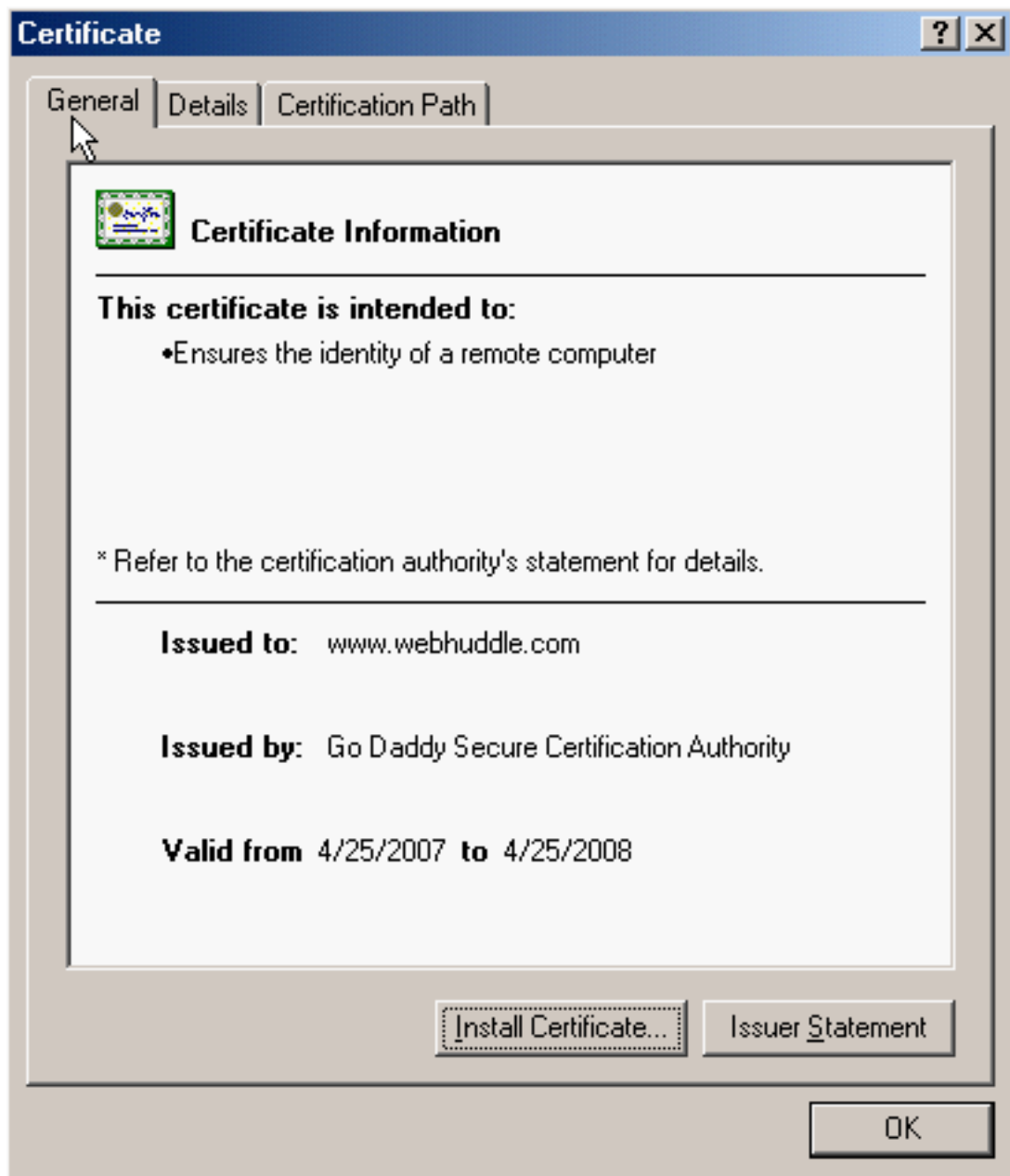
Now restart JBoss and connect to your WebHuddle server using a URL such as the following:

```
https://www.mydomain.com
```

You should now see the WebHuddle home page with no security warnings or prompts. Confirm that you are using your new godaddy.com SSL certificate by double-clicking the padlock icon on the bottom right. You should see a dialog like this:



Or like this:



If you still get the security warning or other error message, something is not configured correctly. Confirm that you followed all the above steps and look at the log file for clues to the problem:

```
~/jboss/server/default/log/server.log
```

If you cannot sort out this or any other issue, see the above section 'Getting Help'.

Chapter 7. How to Configure a WebHuddle Server

7.1. Overview

Besides the installation steps, WebHuddle requires no further configuration. Depending on how you want to use WebHuddle, however, you can adjust several settings to achieve better performance or tighter integration with your environment. For example, WebHuddle's VOIP option supports different options for audio encoding. As well, there are parameters related to bandwidth usage that can be tuned to your network.

WebHuddle configuration options are in a file called `webhuddle.properties`. WebHuddle ships with a `webhuddle.properties.sample` file that you copy to `<jboss>/bin/webhuddle.properties` if you want to customize your installation. Any setting changes take effect after restarting JBoss.

7.2. Audio

By default, the VOIP option uses GSM audio encoding. This codec offers reasonable quality and very low CPU usage. As an alternative, you can configure WebHuddle to use the Speex [<http://www.speex.org/>] codec. This codec supports both 'narrow-band' (phone quality) and 'wide-band' (Skype quality). Both modes sound better than the GSM codec, but they use a lot more CPU power, especially on the speaking person's computer. To enable Speex audio compression, change the `webhuddle.properties` file from this:

```
webhuddle.property.audio.encoder=com.sts.webmeet.content.common.audio.GSMReferenceEncoder
webhuddle.property.audio.decoder=com.sts.webmeet.content.common.audio.GSMReferenceDecoder
#webhuddle.property.audio.encoder=com.sts.webmeet.content.common.audio.SpeexWBEncoder
#webhuddle.property.audio.decoder=com.sts.webmeet.content.common.audio.SpeexWBDecoder
```

to this (for wide-band audio):

```
#webhuddle.property.audio.encoder=com.sts.webmeet.content.common.audio.GSMReferenceEncoder
#webhuddle.property.audio.decoder=com.sts.webmeet.content.common.audio.GSMReferenceDecoder
webhuddle.property.audio.encoder=com.sts.webmeet.content.common.audio.SpeexWBEncoder
webhuddle.property.audio.decoder=com.sts.webmeet.content.common.audio.SpeexWBDecoder
```

or this (for narrow-band audio):

```
#webhuddle.property.audio.encoder=com.sts.webmeet.content.common.audio.GSMReferenceEncoder
#webhuddle.property.audio.decoder=com.sts.webmeet.content.common.audio.GSMReferenceDecoder
webhuddle.property.audio.encoder=com.sts.webmeet.content.common.audio.SpeexNBEncoder
webhuddle.property.audio.decoder=com.sts.webmeet.content.common.audio.SpeexNBDecoder
```

And then restart JBoss so the settings take effect.

7.3. Desktop Sharing

You can configure several aspects of desktop sharing, depending on your preferences and environment. The configuration section for desktop sharing is as follows. Remove the '#' comment character for any line you edit.

```
#webhuddle.property.appshare.kilobits.per.second=150.0
```

```
#webhuddle.property.appshare.minimum.sleep.milliseconds=750
#webhuddle.property.appshare.key.frame.period.seconds=30
#webhuddle.property.appshare.default.compression.type=gif
#webhuddle.property.appshare.default.share.mode=desktop
```

Note that the '`kilobits.per.second`' setting is not a precise, scientific bandwidth measurement but instead a relative setting that you can experiment with. The higher this value is, the faster screen updates will sent (as long as your network can support the bandwidth).

The '`minimum.sleep.milliseconds`' setting is the minimum interval at which the desktop will be scanned for changes. Set this to a lower number to get more frequent updates (but use more CPU and bandwidth).

The '`key.frame.period.seconds`' setting is the minimum frequency at which a key frame (full snapshot) will be forced to be sent. Longer intervals, in general, give better performance, but mean that someone entering the meeting while you are sharing your desktop might not see your desktop until that many seconds later.

The '`default.compression.type`' setting controls which compression mode will be the default. If you typically share photos or other color-rich content, change this to `jpeg`.

With WebHuddle, you can share either your entire desktop or a limited region of it. By default, WebHuddle shares your entire desktop. This makes things simpler for everybody, but the larger shared the region the more bandwidth required. If your participants are typically sharing only a limited region due to bandwidth or performance constraints, you can change this setting to '`region`' to make it the default mode.

Chapter 8. How to Integrate WebHuddle in the Enterprise

8.1. Authentication

If your enterprise already has all its user names and passwords listed in a database or LDAP server, you can configure WebHuddle to use this data for user authentication. This way, there is no need to create accounts for your users: any user in your enterprise database automatically has a WebHuddle account. As well, they can use the same logon and password they use for other applications.

8.1.1. SQL Database Authentication

To configure WebHuddle to use SQL Authentication, 3 files need to be edited. First, edit:

```
<jboss>/server/default/deploy/webhuddle-app.ear/webhuddle-web.war/WEB-INF/web.xml
```

Near the top of this file, there is a section like this:

```
<!-- spring/acegi settings -->
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    /WEB-INF/applicationContext-acegi-security.xml
  </param-value>
<!--
  <param-value>
    /WEB-INF/applicationContext-acegi-security-external.xml
  </param-value>
-->
<!--
  <param-value>
    /WEB-INF/applicationContext-acegi-security-ldap.xml
  </param-value>
-->
</context-param>
```

Remove the comment characters (<!-- and -->) surrounding the section with

```
applicationContext-acegi-security-external.xml
```

and *add* comment characters around the section containing:

```
applicationContext-acegi-security.xml
```

After these changes, the section should look like this:

```
<!-- spring/acegi settings -->
<context-param>
  <param-name>contextConfigLocation</param-name>
<!--
  <param-value>
    /WEB-INF/applicationContext-acegi-security.xml
```

```

        </param-value>
-->
        <param-value>
            /WEB-INF/applicationContext-acegi-security-external.xml
        </param-value>
<!--
        <param-value>
            /WEB-INF/applicationContext-acegi-security-ldap.xml
        </param-value>
-->
    </context-param>

```

Save the modified web.xml to its original location. The next file for editing is:

```
<jboss>/server/default/deploy/webhuddle-app.ear/webhuddle-web.war/WEB-INF/applicationContext-acegi-security.xml
```

First, edit the following section near the top:

```

<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName">
        <!--<value>org.hsqldb.jdbcDriver</value>-->
        <value>com.mysql.jdbc.Driver</value>
    </property>
    <property name="url">
        <value>jdbc:mysql://localhost:3306/externaldatabase</value>
        <!--<value>jdbc:hsqldb:hsqldb://127.0.0.1:1701</value>-->
    </property>
    <property name="username">
        <value>dbuser</value>
        <!--<value>sa</value>-->
    </property>
    <property name="password">
        <value>changeit</value>
    </property>
</bean>

```

If your external JDBC data source for authentication is MySQL, you can leave the 'driverClassName' section as is. Change the 'url' section to reflect the host, port and name of the database with your table of users. Also edit the 'username' and 'password' fields.

To connect to an external database such as MySQL, JBoss needs the corresponding JDBC driver jar file. For our MySQL example, the driver is available here [<http://dev.mysql.com/downloads/connector/j/5.0.html>]. Download the archive file -- it will be called something like "mysql-connector-java..." and extract and copy the actual JDBC driver .jar file into <jboss>/server/default/lib.

The last edit to make in this file is the section 'webhuddleExternalUserDetailsService'. Here is the default value of that section (note that it is just an example -- you will need to edit it):

```

<bean
    id="webhuddleExternalUserDetailsService"
    class="com.sts.webmeet.server.acegi.userdetails.jdbc.WHExternalUserDetailsService"
    >
    <property name="dataSource"><ref bean="dataSource"/></property>
    <property
        name="usersByUsernameQuery"
        value="SELECT
            user_id as external_id,
            login as username,
            passwd as password,
            'true' as enabled,
            f_name as first_name,

```

```

        l_name as last_name,
        email as email
    FROM t_users
    WHERE login = ?"
/>
<property
    name="authoritiesByUsernameQuery"
    value="SELECT login, 'ROLE_USER' FROM t_users WHERE login = ?"
/>
</bean>

```

The part to actually edit is the SELECT statement. WebHuddle expects several named fields, aliased as shown. So, if your table has people's last name in a column called 'last' and their email in a column called 'mail', you would alias these to the column names that WebHuddle understands:

```

SELECT
    user_id as external_id,
    login as username,
    passwd as password,
    'true' as enabled,
    first as first_name,
    last as last_name,
    mail as email
    FROM t_users
    WHERE login = ?

```

You are done with this file and almost there!

If you don't already have a webhuddle.properties file in your <jboss>/bin directory, copy the sample webhuddle.properties.sample file there and change this line:

```
#webhuddle.property.authentication.external=true
```

To this:

```
webhuddle.property.authentication.external=true
```

(It was merely un-commented.) You are done; restart jboss and you should be able to login with your username and password as defined in your MySQL or other table.

8.1.2. LDAP Authentication

Similar to SQL authentication, LDAP authentication requires simple edits to three files. First, edit this file:

```
<jboss>/server/default/deploy/webhuddle-app.ear/webhuddle-web.war/WEB-INF/web.xml
```

Near the top of this file, there is a section like this:

```

<!-- spring/acegi settings -->
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
        /WEB-INF/applicationContext-acegi-security.xml
    </param-value>
<!--
    <param-value>

```

```

        /WEB-INF/applicationContext-acegi-security-external.xml
    </param-value>
-->
<!--
    <param-value>
        /WEB-INF/applicationContext-acegi-security-ldap.xml
    </param-value>
-->
</context-param>

```

Remove the comment characters (`<!--` and `-->`) surrounding the section with:

```
applicationContext-acegi-security-ldap.xml
```

And *add* comment characters around the section with this:

```
applicationContext-acegi-security.xml
```

After these changes, the section should look like this:

```

<!-- spring/acegi settings -->
<context-param>
    <param-name>contextConfigLocation</param-name>
<!--
    <param-value>
        /WEB-INF/applicationContext-acegi-security.xml
    </param-value>
-->
<!--
    <param-value>
        /WEB-INF/applicationContext-acegi-security-external.xml
    </param-value>
-->

    <param-value>
        /WEB-INF/applicationContext-acegi-security-ldap.xml
    </param-value>
</context-param>

```

Save the modified web.xml to its original location. The next file for editing is:

```
<jboss>/server/default/deploy/webhuddle-app.ear/webhuddle-web.war/WEB-INF/applicationContext-acegi-security-ldap.xml
```

First, edit the 'initialDirContextFactory' section, changing the 'constructor-arg' value to point to your ldap server. As an example, you setup for this section might look something like this:

```

<bean id="initialDirContextFactory" class="org.acegisecurity.ldap.DefaultInitialDirContextFactory"
    <constructor-arg value="ldap://ldap.mycompany.com:389/dc=mycompany,dc=com" />
</bean>

```

Now edit the 'userSearch' section, changing the 'constructor-arg index="0"' part to match the values for your ldap setup. As an example, this section would look something like this:

```

<bean id="userSearch"
    class="org.acegisecurity.ldap.search.FilterBasedLdapUserSearch">
    <constructor-arg index="0">
        <value>ou=people,dc=mycompany,dc=com</value>
    </constructor-arg>

```

```
</constructor-arg>
<constructor-arg index="1">
  <value>(mail={0})</value>
</constructor-arg>
<constructor-arg index="2">
  <ref local="initialDirContextFactory" />
</constructor-arg>
<property name="searchSubtree">
  <value>true</value>
</property>
</bean>
```

You are done with this file and almost there!

The last file to edit is webhuddle.properties. If you don't already have a webhuddle.properties file in your <jboss>/bin directory, copy the sample webhuddle.properties.sample file there (drop the '.sample' part) and change this section:

```
#webhuddle.property.authentication.external=true
#webhuddle.property.authentication.ldap.first.name=givenname
#webhuddle.property.authentication.ldap.last.name=sn
```

To this:

```
webhuddle.property.authentication.external=true
webhuddle.property.authentication.ldap.first.name=givenname
webhuddle.property.authentication.ldap.last.name=sn
```

(It was merely uncommented.) You are done; restart jboss and you should be able to login with your ldap user-name and password. If not, confirm that you followed all the above steps and then see the section 'Getting Help'.

Part III. How to Hack WebHuddle (Under Construction)

This section will contain details on WebHuddle's architecture and how to get started making changes or improvements to it. Meanwhile, if you have programming skills and want to help, two areas that need work are: audio performance in general and the CPU efficiency of the JSpeex codecs in particular. The JSpeex codec sounds great but it is a CPU hog. Try profiling it and making it faster.