

Using the open source Portable Miranda instant messaging client to support student-teacher chat sessions from any computer on the Internet

Robin Snyder
Savannah State University
126 Jordan, P. O. Box 20359
Savannah, GA 31404
(912) 356-2716
snyderr@savstate.edu
<http://www.RobinSnyder.com>

Abstract

Portable software can be installed and run from portable media such as USB flash drives - usually without prior installation or support from the local computer staff. This paper (and talk) will discuss (and present) how the free and open source Portable Miranda instant messenger client is being used to allow students (e.g., with a free Yahoo email account) to initiate a chat session with the teacher from any computer on the Internet. Miranda currently supports Yahoo Messenger, Windows Messenger, Jabber, IRC, ICQ, and AIM protocols - all at the same time. Installation, configuration, support, audit logs, add-ons, etc., will be covered.

Introduction

IM (Instant messaging) is a real-time on-line communication method that originated as text-based messages but has since expanded to include audio, video, and file transfers. IM tends to be much more real-time than email and has its roots in online chat systems. To use IM, one needs an IM account on an IM server. One then needs an IM client into which to input messages and see response messages. There are many IM protocols. Here are just a few of the popular IM protocols.

- MSN Messenger
- Windows Live Messenger
- AOL (AOL Instant Messenger)
- Jabber (open source)
- Yahoo! Messenger
- ICQ (I Seek You)
- IRC (Internet Relay Chat)

Each IM system has advantages and disadvantages. Some IM clients, such as the open source Miranda system, discussed here, allow simultaneous access to multiple IM servers running different IM protocols.

This paper (and talk) will discuss (and present) how the free and open source Portable Miranda instant messenger client is being used to allow students (e.g., with a free Yahoo email account) to initiate a chat session with the teacher from any computer on the Internet. Miranda currently

supports Yahoo Messenger, Windows Messenger, Jabber, IRC, ICQ, and AIM protocols - all at the same time.

Portable Applications

A portable application is a software application that can be moved from computer to computer via a storage device such as a USB drive. Such software does not leave a "footprint" on the computer being used. This can be important when using software on, for example, computer lab computers in a University setting. An extensive list of portable applications is available at http://en.wikipedia.org/wiki/list_of_portable_software. The author uses the following portable applications on a regular basis.

- Portable Apps menu and backup system.
- OpenOffice (apps similar to Word, Excel, PowerPoint, Access, etc.)
- Firefox Portable web browser (with many extensions)
- Miranda Portable instant messaging (with many extensions)
- Thunderbird Portable email client (with many extensions)
- Audacity Portable for audio recording/processing
- Portable Python for simple programming for data communications and security courses.

The Portable Apps web site is at <http://www.portableapps.com>. Many portable applications are available for download.

The principle behind a portable application is that everything the application needs is stored in one directory structure. In the case of Windows systems, it means that the Windows Registry is not used. A standardized directory structure allows the portable menu and backup system to "understand" any added portable application.

The directory structure of the portable application system is as follows. Assume that **E:** is the root of the USB flash drive.

- **E:\autorun.inf** contains the commands to automatically start the Portable Menu system - on permission of the user (depending on the autorun settings of the computer).
- **E:\StartPortableApps.exe** is the executable to start the portable application menu and backup system. This system provides convenient access to all installed portable applications.
- **E:\PortableApps** contains the portable application directories. A few of these appear next.
- **E:\PortableApps\FirefoxPortable** contains the Firefox portable system.
- **E:\PortableApps\OpenOfficePortable** contains the OpenOffice portable system.
- **E:\PortableApps\MirandaPortable** contains the Miranda portable system.
- **E:\PortableApps\PortableSnyder** contains the author's portable software. A few of these appear next.
- **E:\PortableApps\PortableSnyder\SecureS.exe** is the author's software for the security and data communications course. A Python IDE is included.
- **E:\PortableApps\PortableSnyder\JavaS.exe** is the author's software for the Java programming course. A simple Java IDE is included.

2007 ASCUE Proceedings

The portable applications menu system, when run, looks for a subdirectory called **PortableApps**. Any subdirectory in the **PortableApps** subdirectory, such as **PortableSnyder**, is searched for executable files, such as **SecureS.exe** and **JavaS.exe**. The title and icon, if any, is extracted from each executable and displayed in the menu system.

The **autorun.inf** file allows the autorun feature of the computer to automatically start the portable applications menu system. This file might appear as follows.

```
[Autorun]
useautoplay=1
Open=PortableApps\PortableAppsMenu\PortableAppsMenu.exe
Action=Start PortableApps
Icon=PortableApps\PortableAppsMenu\PortableAppsMenu.exe
Label=PortableApps
```

Needless to say, both the **autorun.inf** file and individual portable applications can be customized for use in a specific environment. A self-extracting zipped executable can be used to allow students to download and install a customized version of any portable application. Check the license agreement to see if this is permitted. For example, you might be required to have the students download the "standard" software and then download and extract the "customized" changes to the "standard" portable software.

Besides the directory structure, the following are commonly used for packaging portable applications.

The UPX (Ultimate Packer for Executables) is an open source system for packing an executable file so that it requires less space. It is unpacked as needed during runtime. This makes executables take less space which is important for storing applications on USB flash drives. The UPX web site is at <http://upx.sourceforge.net/>.

The NSIS (Nullsoft Scriptable Install System) is an open source system for creating and deploying Windows installers. For portable applications, the full functionality and complexity of NSIS is not needed since the Windows Registry is not used. The NSIS web site is at <http://sourceforge.net/projects/nsis/>.

Together, UPX and NSIS make it easy for a user to download, install, and run portable applications.

Student software

The author has also made much of his software for student use in the author's classes available as portable applications. Here is a short history of the ways in which the author has made software available to students in the past.

Computer hard drive: The software can be put on the computer hard drive. The problem is getting it installed on the hard drive.

Network drives: The software can be put on a network drive to which the student has access. The problem is that this method requires that the student be connected to the network with the correct access rights. This can be tricky to implement and maintain.

Software download: The software can be put on a web server to be downloaded and installed. The author packed the software into a self-extracting executable that used a fixed path on the hard drive. The student then had to find the software the first time. When run, the software would create shortcuts on the desktop for the user.

Terminal services: The software can be put on a network drive that is mapped such that when the student logs in to terminal services using remote desktop, the software is available.

Portable application: The portable application is made available as a self-extracting download that the student extracts to their portable directory - usually on a USB flash drive. The portable menu system then recognizes it.

The portable application method requires the least effort by the author and the least computer support by the local IT staff. To date, this method has worked very well.

Moving portable applications and data is easy. For example, Miranda Portable can be moved from computer to computer without having to install Miranda on each computer. It also means that one can move and have access to the history of messages that are stored on the storage device. Otherwise, that history would be spread over many different computers.

Important: One important note is that students must be taught to not remove their USB flash drive from the computer without closing all open portable applications and then using the "**Safe-ly Remove Hardware**" icon in the tray of the computer. If this feature is disabled, the student can close all open applications and then wait a few seconds to make sure that all changes have been flushed to the USB flash drive. Otherwise, a corrupted database (or other file) might result. In the case of Miranda, there is a program that can be used to attempt to recover a corrupted database.

Important: Another important note is that students should safeguard their USB flash drive so that it is not lost or damaged. USB drives do go bad. A backup is very important. The portable menu system has a backup feature that should be used. The students must be taught to use this system.

If one is using portable applications, changing operating systems is as simple as moving the USB flash drive on which the portable directories reside to the new computer. If one is using the portable applications on the hard drive of a computer, the upgrade consists of copying the portable application directory to the new computer.

Installation and configuration

Although Microsoft has stopped support for Windows systems such as Windows 98 (and earlier), some students may still be using such older systems. This is not a good idea as security vulnerabilities for such systems are well known. But, if such systems are used, Miranda will still work

2007 ASCUE Proceedings

on them but one must use the ANSI versions (i.e., 8-bit character support) and not the UniCode versions (i.e., 16-bit character support).

Installation of Miranda is easy - just extract to the portable applications directory. However, if students save the installation program to their USB flash drive, they will often confuse the installation program with the "**MirandaPortable**" folder that contains the program to start Miranda. Some students will "**install**" Miranda over and over and not understand why it is not "**working**".

Here are some suggestions for configuration. Most of the configuration problems arise when the student does not read the displayed message.

- An initial profile must be created such as "**default**".
- When asked, select "**Cancel**" for any IM account that is not being used, such as AIM, ICQ, etc. These account configuration requests can be later turned off.
- When asked, select "**Cancel**" if there are no other contact lists to be imported.

Once started, the Miranda system starts by appearing in the tray (lower right corner of the desktop on most systems) as a little man. Right click on it and select "**Hide/Show**" to see the contacts tree.

The Miranda system must be configured before it can be used. Select "**Miranda**" (the leftmost icon in the main menu bar) and select "**Options**". There are many options that can be configured. Only a few are covered here.

Plugins: Under "**Plugins**", any plugin that is not needed can be unchecked. For example, unchecking the **aim.dll** plugin will avoid the AIM error message when Miranda is started.

Network: Under "**Network**", login account information can be set for the desired IM systems. Since Yahoo Messenger is "**free**" (i.e., one must tolerate advertisements and marketing monitoring of one's account), the author requires students to get a Yahoo email and messenger account. Since Microsoft Messenger will not allow sending messages to users who are not online, but Yahoo does allow sending messages to users who are not online, the author prefers the Yahoo Messenger system for student use. Under "**Network**", select "**YAHOO**" to set the login options for Yahoo Messenger. One must have a Yahoo Messenger account before this option can be used. The "**ID:**" field is the Yahoo account name without the "**@yahoo.com**" suffix. If Yahoo messages on port **5050** will be blocked by a firewall, the "**Port:**" should be set to port **80**, the standard web port that is not blocked. Other options can be set as desired. It is usually a good idea to check "**Auto-Login to Yahoo**".

For MSN messenger, one should check "**Use HTTP gateway mode**".

One way to go online is to select "**Status**", "**Online**". Miranda then attempts to connect to all IM systems that have been enabled.

Once connected, one can add contacts. To do this, select "**Miranda**", "**Find/Add Contacts**". Select the "**Search:**" system, such as "**YAHOO**" (you must be online to do this), provide an "**ID**" (without the "**@yahoo.com**" suffix), and select "**Search**". When contacts are found, select the

desired contact and select "**Add to list**". Many students will think that finding them automatically adds the contact and will have to redo this step. The chat room tree can be configured as desired.

Customization

Almost any feature of Miranda can be customized - fonts, colors, etc. If not, an plugin probably exists to do what is desired. If not, one can create their own plugin.

One advantage of open source software is that a usually has some way that a programmer can access the system to extend it to do things the system was not originally intended to do. In the case of the Miranda instant messaging system, there are examples on the web of how to create Delphi Pascal (and other) program code to interface to the Miranda software system. Obviously many people have created add-ins for the system so one just needs to dig into the details of how to do it.

Chat sessions

A common problem is that, when accepting a connection, one needs to make that contact permanent. To do this, right-click on the contact in the menu tree and select the appropriate option. Otherwise, the contact will not appear the next time that Miranda is started.

To initiate a chat session, one double-clicks on the desired contact in the contact tree. The chat dialog window opens, and the conversation can start. Or, one can accept an incoming chat request such that the chat dialog window opens.

For group work, the author requires students to contact their group members via IM. The Miranda software system does not currently support groups of members in the same chat room. The IRC system does support group chat. The author required the students to get ICQ accounts and then do a group chat in IRC. However, the University firewall blocked such access and even when permission was granted and the firewall port(s) opened, the access was still blocked. The issue was never resolved. The open source Jabber IM server system might solve the problem.

The author has modified the author's submission system to accept images in GIF or JPEG format along with written text. The free, but not open source software, IrfanView is covered to allow the students to easily grab, crop, save, and submit an image that contains the required content. The required content is such that the author can determine that the students have actually done what they were supposed to do. For example, a requirement might be to use Miranda IM to initiate a conversation with me. The image would require the contact tree window and the message dialog window that shows the conversation. The written requirement might be to discuss how IM can be used in business marketing to enhance communication. The author's submission system makes it easy to collect the submissions, grade them according to a published scoring rubric, and provide feedback to the students via the online submission database. For security, the entire student web system is accessed via a secure web system with login authentication.

Add-ins

There are many add-ins that can be used to extend the functionality of the Miranda instant messaging system. There are hundreds of add-ins for Miranda. The following add-ins for Miranda were found to be useful to extend the functionality of Miranda.

History logs: Miranda maintains history logs of conversations in the database. Some plugins provide convenient access to such history logs. The `msg_export.dll` plug-in is used.

Tabbed chat windows: The `tabsrmm.dll` provides tabbed chat windows. Otherwise, chat sessions with, say, 20 students would result in 20 separate windows on the screen. The tabbed chat windows provides one window with tabbed access to the desired chat session.

Smiley's: The `smileyaddw.dll` provides (Unicode) smiley support. In some cases, the user must do some "extra" work. For example, the smiley's for Yahoo are copyrighted by Yahoo. To use them in Miranda, one can install Yahoo Messenger which installs the smiley's. Then, one can copy the Yahoo smiley's to a directory where Miranda can use them. The same method would be used for any copyrighted but otherwise reusable content that is already on one's computer. As a bonus, the Miranda add-ins allow one convenient graphical access to special Yahoo smiley's that are only available in Yahoo Messenger via their text abbreviations.

Animated smiley's: The `ieview.dll` plugin provides animation for smiley's in the message windows.

Screenshots: The `sendss.dll` plugin is used to allow students to send screenshots to the teacher. This can help in diagnosing problems. However, there are sometimes firewall issues that keep this plugin feature from working properly. The same issues were detected with general file transfers.

Notifications: The `messagenotify.dll` and `notifyanything.dll` plugins are used to provide an easier way to determine that a student has sent a message. The author uses four big monitors connected to one computer and, with so much screen space, it is easy to miss a message window.

Voice recognition

In the introduction, it was stated that one "needs an IM client into which to input messages...". Technologies such as voice recognition can be used to make it easier to input messages. The author has been experimenting with the ScanSoft Dragon Naturally Speaking 8 Preferred voice recognition software. Note that this software has since been taken over by Nuance, at <http://www.nuance.com>.

Once set up and trained for one's voice, the software works pretty well. A good microphone is important. The author uses the USB Samson C02U Recording/Podcasting Pak. One problem with voice recognition software is that it does not always correctly recognize the words you are thinking when you are speaking. There are ways to train the system to make it better at recognition, but this requires investing time in learning the intricacies of the voice recognition system.

Observations

During the Spring semester of 2007, the author has required all students to install and use Miranda IM. Some students have done this. Others avoid it unless forced to use IM. Those students who use IM on a regular basis often ask questions when they get stuck. In many cases, students identify problems with the author's web system such that the author gets feedback in a timely fashion. Future plans include working out issues with screen shot sharing for problem diagnosis and with group chat so that the entire class can be online in the same chat room.

Summary

This paper has discussed aspects of using the open source Portable Miranda instant messaging client to support student-teacher chat sessions from any computer on the Internet. This has allowed the author to provide convenient student access to the author, for class purposes, for many more hours than is provided by conventional office hours. The main problems encountered with using IM were firewall beyond the author's control that blocked the needed ports to use IM. The Yahoo IM system was usable without changing the firewall, but the group chat feature is not available in the Miranda IM system.

References

- [1] Snyder, R. (2007). Simple security programming for students using Portable Python 1st Computer Security Conference (April 12-13, 2007), Myrtle Beach, SC.