

Revision 1.2 08/18/2009

This document serves as errata to Chapter 12 of Multimedia Networking. This document lists and explains the software's discovered bugs at this point and the workarounds in how to avoid them.

MMUI Software Executables

Some executables may not work on Windows Vista. This is a known issue and may be fixed in the future. Users are advised to operate the software application on Windows XP.

12.1 Speech and Audio Compression Module

Section 12.1.2

The module cannot open the .wav file recorded using the Windows Sound Recorder. The module also cannot open just any .wav file downloaded from online sources. Users are advised to use the speech and audio module to record and save the .wav file; then this file can be opened successfully by the module.

Section 12.1.3

Speech can be recorded at formats 4kHz and 8kHz. However, the 4kHz speech would fail to be encoded any of the speech codecs. Users are advised to use 8kHz for encoding.

The waveform displays may not always show up correctly. Sometimes there will be no waveform displayed. This does not effect the operation of the module. The .wav recording is still encoded successfully as long as no error dialog box shows up.

For MP4 AAC encoding, the text shows "No Mid-Side Cod." This was truncated by the graphical user interface from originally "No Mide-Side Coding." This does not effect the operation of the module.

12.2 Image and Video Compression Module

Section 12.2.2

For MPEG encoding and H-264 encoding of video, the module cannot encode just any .avi file downloaded from online sources. Users are advised to use the image and audio module to record the video and save the .avi file; then this file can be encoded using MPEG and H-264 codecs successfully by the module.

For H-263, the GUI has a potential to crash during the encoding process. The .avi file downloaded from online sources are never successfully encoded, while the .avi file recorded and saved using the image and video are sometimes successfully encoded.

Users are advised to perform H-263 encoding using the .avi file recorded from the module; if it fails, give it a few more attempts.

12.3 Network Sockets Module

Section 12.3.2

After sending and receiving a file, the received and saved file will not have an extension or file type/format. It can be opened successfully with the appropriate application or by adding the appropriate file extension.

Sample Software Codes

All the source codes provided in Sample Codes 12.4, 12.5, 12.6, and 12.7 can be run and compiled in Visual Studio 2005 and 2008. All executables are functional on Windows XP or Vista, with the exception of the AudioUI recording format at 32kHz and 44.1kHz can only be run on Windows XP. Users are advised to operate the software application on Windows XP.

12.4 Creating a MFC Project

No error.

12.5 Encoding/Decoding of Video/Audio

Section 12.5.1.2

The paragraph in the middle of page 513 should be understood first before proceeding. The *Configuration* (top left hand corner on the Property Pages screen) is defaulted to “Active(Debug)”; if users would like the configuration to be applied to all modes such as Released, users need to change the setting accordingly here.

After choosing the Win32 Console Application as template to create a new project, it will go into a Win32 Application Wizard, simply click “Finish” to proceed without changing any parameters.

On page 512, *Project → Project Property* is incorrect in the text. It should be *Project → Property*.

On page 512, *Configuration Properties → General → C/C++ → Additional Include Directories* is incorrect in the text. It should be *Configuration Properties → C/C++ → General → Additional Include Directories*.

Section 12.5.1.5

The MPEG encoder cannot encode just any .avi files – same issue as in the MMUI’s image and video module.

Section 12.5.2 and 12.5.3

The source codes for these two sections, Encoding H.264 Video and Encoding AAC Audio, are not provided to users at this point. These may be provided in a future date.

12.6 Building a Client-Server Video System

The text in this section only guides users through how to create this project by modifying the code from Section 12.4. However, a pre-built code and project is provided to the user (Sample Code 12.6). Users are advised to use the pre-built code to try out this software application.

If users were to modify the code from Section 12.4 to build this software application, the following items should be paid attention to:

12.6.1.4

The *BeginVideoCapture* declaration should be added as a “public” member of the *CVideoIn* class.

12.6.1.5

The *LRESULT CALLBACK VideoStreamCallbackProc* declaration should be added as a “protected” member of the *VideoCaptureDlg* class.

12.6.1.6

The *OnEncodeVideoData* declaration should be added as a “protected” member of the *VideoCaptureDlg* class.

12.6.2.1

The *SockThreadProc* function belongs to the *VideoPlaybackDlg* class, which is not mentioned in the text. The *SockThreadProc* declaration should be added as a “protected” member of the *VideoPlaybackDlg* class.

12.6.2.2

The *OnVideoPacket* declaration should be added as a “private” member of the *VideoPlaybackDlg* class.

12.6.2.3

The *PlayVideoFrame* function belongs to the *VideoPlaybackDlg* class, which is not mentioned in the text. The *PlayVideoFrame* declaration should be added as a “private” member of the *VideoPlaybackDlg* class.

12.7 Creating a P2P Video Conferencing System

12.7.4

The text mentions “There are six accounts, User ID “a” to “e,” on the MCU presence server.” – there were only 5 alphabets. In fact, there are indeed six account – a, b, c, d, e, and f. The text states that the password to each user ID is the same as the user ID alphabet – this is incorrect. The password to all the accounts is simply “a” and it is defaulted to “a” when the application starts up, therefore there is no need to change the password. Users are advised to not edit the password field at all.

For any questions, concerns, and suggestion, please email Isaac Chang at changis@u.washington.edu.