ViERel.12.07.2009

VideoEngine PC 2.5.0

PRODUCT RELEASE INFORMATION

Product:	VideoEngine PC
Release Number:	2.5.0
Release Date:	December 7th, 2009
Previous Release:	2.4.3

INTRODUCTION

This release is a significant update from v2.4.3. There are a number of new features covered in the new features and improvements section below. This release also addresses a number of issues highlighted below in the bug fixes section.

NEW FEATURES AND IMPROVEMENTS

F#1358 Automatic Video Tuning of LSVX at Start-up based on some input parameters

VideoEngine takes information about the system on which it is running to tune resolution parameters in order to maximize the experience of the video session.

F#1415 Native Cocoa rendering (Mac only)

In order to improve performance and facilitate the support of advanced UI features, VideoEngine now supports native Cocoa rendering for Macintosh.

F#1422 Codec independent FEC

VideoEngine now supports RFC #5109 Forward Error Correction (FEC) in a codec agnostic approach for video streams. Previously, FEC was only available for LSVX.

F#1434 Support 320x200 video size

For improved interoperability, VideoEngine now supports the 320x200 video resolution.

F#1466 VideoEngine now supports Windows 7

VideoEngine v2.5 has been thoroughly tested with Windows 7. Windows 7 is now an officially supported operating system.

F#1469 Support 3D rendering (Windows only)

Direct3D support has been added to VideoEngine which will enable it to render video on a 3 dimensional surface. This support is currently only available on Windows through DirectX’s Direct3D.

F#1492 Turn off scene change detection when using H.264

VideoEngine now relies on the scene detection algorithms in the H.264 codec. This change only applies to this codec.

F#1501 Decrease LSVX decoding complexity by signaling other end to reduce sending resolution as a way to reduce CPU complexity on highly loaded machines

In order to avoid a scenario in which a machine that is already struggling under computational loading (either from video or other factor), VideoEngine can use in-band signaling to reduce the far end's transmit resolution in order to reduce the amount of data received for computation on the local machine. This feature is currently only available with LSVX.

F#1532 [GIPS-1635] Include active thread indication in trace files

In an effort to improve the troubleshooting tools available to debug VideoEngine integration, the active thread can be included in the trace files so that any issues can be quickly pinpointed.

F#1551 Add dynamic RTP logging

RTP packets can now be dynamically logged into an RTP file for offline analysis.

F#1563 DirectDraw/DirectShow functionality transparent to user

The goal of this effort was to facilitate the integration of VideoEngine as much as possible. Particularly issues in which DirectDraw failed previously were addressed, and DirectDraw and Direct3D are now enabled by default.

F#1565 Improve VideoEngine Logging - Track one frame the whole way

As part of an ongoing effort to facilitate VideoEngine integrations, logging now includes the ability to track a frame through the process.

F#1566 Improve VideoEngine Logging - Read hardware and driver info from registry and log

As part of an ongoing effort to facilitate VideoEngine integrations, logging now includes the ability to read hardware and driver information from the registry for inclusion in the log. This will help in troubleshooting options within the engine.

F#1576 API for video file duration, codec name and resolution

When using VideoEngine to play a file, it is now possible to gather the file length, codec name and resolution. This feature is modeled after 'GIPSVE_GetFileDuration' in VoiceEngine.

F#1621 JitterBuffer, keep a list of incomplete frames and their size for late estimate updates

When sending HD video with huge key, the first couple of key frames may be decoded before they're complete, resulting in the same artifacts as seen when having packet losses. VideoEngine is able to adjust to this situation by expanding the JitterBuffer. By keeping track of incomplete frames, this adjustment period is much shorter.

F#1622 Extend GIPSVideo_ViewCaptureDialogBox() method to include X, Y window positioning parameters

VideoEngine now allows for the positioning of the property dialog on the desktop by specifying top and left (X, Y) placement parameters.

F#1659 GipsEncryptedTrace.txt not created in the CSF core Logs directory

VideoEngine now accepts a properly UTF-8 formatted filename and directory name for use in the saving of file name.

F#1664 Modifications and Improvements requested for IncomingCodecChanged (VideoChannelCallback)

Removed the dependency of HWnd for IncomingCodecChanged callback.

F#1665 Scaling up the image when SetSendCodec width/height is greater than resolution of the camera

With this feature, there is now no black border when the camera generates a smaller image than is being used by SetSendCodec. The image is instead scalled up.

F#1725 Spread out how our video engine transmits packets instead of transmitting all packets at once

When a large amount of data is required to be transmitted, it is now possible to spread the packets out in order to avoid any impact on the network with the GIPSVideo_SetMaxPacketBurstSize API call.

F#1766 Add timestamp argument to GIPSVideo_IncomingCapturedFrame

This feature will assist with the use of external codec lip synchronization by time-stamping the video frames.

F#1797 provide API to store/read information in GMF file for player to set decoder

Makes it easier to use the GMF file format.

F#1851 Scale down image captured from camera when it is greater than SetSendCodec

When the Camera image is greater than the SetSendCodec value, the image will be scaled down. The previous functionality was to crop the image to make it fit into the desired send resolution.

API CHANGES

1. Removed CodecConfigParameters callback from GIPSVideoChannelCallback class
2. Added GIPSVideo_SetCaptureDelay
3. Added GIPSVideo_GetFileInfo
4. Added GIPSVideo_EnableFrameUpScale
5. Added GIPSVideo_EnableInterpolateScaling
6. Added GIPSVideo_SetMaxPacketBurstSize
7. Added GIPSVideo_EnableFEC
8. Change in GIPSVideo_EnableRTCP
9. Added GIPSVideo_EnableKeyFrameRequestCallback
10. Change in GIPSVideo_IncomingCapturedFrame
11. Change in GIPSVideo_EnableSRTPSend
12. Change in GIPSVideo_EnableSRTPReceive
13. Change in GIPSVideo_EnableEncryption
14. Added GIPSVideo_StartRTPDump
15. Added GIPSVideo_StopRTPDump
16. GIPSVideo_RTPDumpIsActive
17. GIPSVideo_SetThresholdToSignalRemoteResize
18. Change in GIPSVideo_ViewCaptureDialogBox
19. Added GIPSVideo_EnableDirect3D
20. Added GIPSVideo_GetD3DSurface
21. Added GIPSVideo_AddRemoteRenderer for HIViewRef
22. Added GIPSVideo_AddLocalRenderer for HIViewRef
23. Added GIPSVideo_AddLocalRenderer for Cocoa
24. Added GIPSVideo_AddRemoteRenderer for Cocoa

For complete documentation of these APIs, please refer to the latest VideoEngine API document.

BUG FIXES

D#2522 [VIEPC-4] SetCaptureDevice() does not work properly after calling SetCaptureDevice(NULL)

D#4377 usage of payload type's other than 34 for H.263 codec causes error

D#4528 Unrelenting error messages

D#4550 No playout of video untill after a sync (could take one sec)

D#4717 [GIPS-1698] GIPSVideo_SetReceiveCodec() lacks input parameters in API doc

D#4743 JitterBuffer Not releaseing frames for decoding.

D#4827 Crash related to H264 decoder when running ViE 2.4.3 between two computers using network simulation

D#4838 some cameras doesn't work with ViE demo app, but at same time they might work with GIPSTest app, with same library

D#4853 Mac's remote image on win32 using h263 looks bad.

D#4854 GIPSVideo_GetCodec does not return H263-1998 type.

D#4857 Bad picture quality during startup

D#4862 memory leak in Video_RTPSender.cpp if a channel is not deleted properly

D#4892 ViE return -1 when stopping trace

D#4895 failure when add camera in ViE

D#4920 StartRecording outgoing stream doesn't work with preencoded frames(moved from SR00527)

D#4939 Better error feedback when incorrectly setting up lip sync in ViE

D#4957 codec.quality = GIPS_QUALITY_DEFAULT will cause the failure of GIPSVideoGenericEncoder::InitEncode

KNOWN ISSUES AND WORKAROUNDS

GIPS H.264 SVC v3.4.3 Namespace Conflict

The GIPS H.264 SVC v3.4.3 has a namespace conflict with GIPS H.264 v3.3.5. Therefore, the two codecs cannot be run simultaneously. GIPS H.264 v3.3.5 has been shown to provide the best H.264 AVC experience and has full complement of the error concealment and error resilience techniques (see GIPS H 264 ER-EC Technical Notes for more information) and should be used in production deployments. For more information on GIPS H.264 SVC v3.4.3, please see the H.264 v3.4.3 Release Notes.

FEC Needs Further Tuning

F#1422, codec independent FEC is a new feature that needs tuning. It also requires that you lose bitrate available to quality for redundancy.

INSTALLATION AND UPGRADE NOTES

System Requirements

Supported OS:	Windows 2000, Windows XP, Vista, and Windows 7
	Linux (greater or equal to kernel 2.6)
	Mac OS X 10.4, with QuickTime 7, or greater
Processor:	Minimum: Intel Pentium IV 1.0 GHz or equivalent

Global IP Solutions, GIPS, MediaWare, GIPS VoiceEngine, NetEQ, iLBC, iPCM-wb, iSAC, GIPS Enhanced G.711, Remote Extension, REX, Click-To-Talk and associated design marks and logos are trademarks of Global IP Solutions, and may be registered in certain jurisdictions. Patents and Patents Pending, Global IP Solutions, Inc. www.gipscorp.com

Any third-party trademarks referenced are the property of their respective owners. For further information regarding Global IP Solutions trademarks, please see: http://www.gipscorp.com/legal/trademark-and-logo-usage-guidelines.html

For additional contact information, please visit the GIPS URL: http://www.gipscorp.com