

VETLIB FUTURE

Video Elaboration & Transmission Library



This list refers to VETLib 1.0.2.25, items are sorted by priority, five icons mean that implementation is very complex, one icon means it may be developed by a newbie.

Framework Update



vetProcess (requires *vetThread*)
Manage a single dataflow as a thread;
Exceptions and Error control;
Control Color-space and format (error management);
Callback events (some are really required because of threading);
Serialization;



vetProcessEx (require *vetProcess*)
Manage multiple *vetProcess* objects, support tree flows.



CVS (some depth knowledge about library is required, some clever choices might be taken)



Project may be hosted on sourceforge.net (..subscription, but keeping and external website), they also provide a CVS server.

Components Update



vetThread: Testing and Debugging + Windows implementation;



vetFrameT update: preset for common colorspace (pixel access, conversion, ..)



Debug and update Color-Space Conversions (move from *vetUtility*, create a standalone class, update *vetFrames* conversions);











vetCoder_XVID debugging, encoding (using same lib);



vetMultiplexer update and optimize for *vetProcess*.

Required Components







-  *vetInputDXMovie* for loading any supported movie through DirectShow;
-  Some common filters, WS plugins and sample applications;
-  *vetMath* components (DCT, Matrix Math);
-  Image Statistics (Count, Mean, Norm, Moments, in regions);
-  *vetBufferSequential* (Array Queue);
-  *vetCoder* component for MPEG2-4 encoding (through *FFMPEG*);
-  *vetNetworking* implementation (through *MPEG4IP*);
-  *vetInput* component for FireWire on Linux (*libraw1394*, *libdv*, *libavc1394*).




Applications Update








WORKSHOP

-  Debug..
-  More logging everywhere..
-  WorkShop Plugin System Update:
.NET DLL support (with integrated graphical interface);
-  Application Help (Windows style + pure HTML);

Updates based on *vetProcess* (so update VETLib first..)

- 
 1. Dataflow will be managed by *vetProcess*, build a GUI for this class;
 2. Remove stream control box from filters, create a custom component;
-  MultiThreading Support (Application Thread + *vetProcess*' Threads);
-  Assuming *vetProcess* serializable, Workshop may save and load sessions;

PKGSTUDIO

-  Debug..
-  More logging everywhere..
-  Application Help (Windows style + pure HTML);
-  Ability to open and edit existing components;
-  Ability to validate packages;



A new internal I/O system, based on new interfaces: *vetInput2*, *vetOutput2*, *vetFilter2*, *vetProcess2*, (or updating old classes), components interacts through a *vetLink* component, defining input and output *Pins*, *vetProcess2* would manage connections and the dataflow.

It should be a simplified version of DirectShow system;



Events support;



For windows users VETLib may provide an access to classes and methods similar to Java / .NET frameworks, using namespaces (composition) and some static classes (for fast calls and presets);



Networking in VETLib is still absent! Plan and class design;



An interface to the Intel Open source Computer Vision Library;



A bridge component for sharing filters with DirectShow (first target is to use a *vetFilter* in a DirectShow graph);



New component for *Video4Linux2*;



New component for *DirectX 10* (not released jet);



Driver modules (data source and renderer) for *DirectX* and *VideoForWindow*;



Scripting System for demonstration and teaching;



.NET and Java interaction (for multimedia contents on PDA);



Simple access to I/O on RS-232 and USB;



WorkShop may generate C++ source code of a process.