WvStreams

An Easier Way to World Domination

(Abridged Version)

Dave Coombs *Net Integration Technologies, Inc.* 2003 12 22

ACHTUNG!

Mein Deutsch ist schrecklich! Dieses Kurz-Papier wird auf Englisch geschrieben, und der Vortrag wird auch auf Englisch gehalten werden.

WHAT IS IT?

WvStreams is an open-source C++ networking library developed over the last few years with the following major goal: make coding easy, without sacrificing performance. Really.

WHAT CAN IT DO?

WvStreams came about because we wanted to write a really big C++ application to control an entire Linux system acting as an Internet gateway. We decided to abstract away the most common parts of programming by making a library that would act as the framework for our main project, but which we could use in future projects too. Do the hard stuff right once, and then never worry about it again.

So what *is* the hard stuff? What *do* programmers really need? What sort of stuff do we do all the time, that this library should encompass? Here's a list of important ones:

- File I/O. We all read and write files all the time.
- String manipulation and buffers. Admit it, you *still* do this wrong sometimes.
- Maintain sets of things. Have you written a linked list or a hash table? Have you done this more than once? More than twice? I thought so.
- Networking. We need to make and accept TCP connections, send and receive UDP packets, and keep track of multiple instances of these.

- Configuration. Most programs need to store their config options somewhere, and most programmers are too lazy to care if their config system is ugly or inflexible.
- Multitask. Threads are dangerously evil. What to do?

Some of the above things are really hard to do properly. That's why programmers so often cheap out on their config systems, string manipulation, and buffer management, even though those things are important to get right.

That's what WvStreams is really good at. It does most of the hard stuff for you, and presents a clean, friendly API. It takes a bit of getting used to, but we've put lots of work into making the underlying code solid, and we think it was worth the effort. Most people who have seen WvStreams-based code comment to us on how readable it is, and are surprised to find how easy it is to write working programs with it.

We have a philosophy about Justifiable Ugliness: "Any amount of code ugliness is okay if it removes more ugliness than it adds." The insides of WvStreams are quite complex, but using it to write clean, readable, fast code is easy.

WvStreams is *pragmatic*. It's carefully optimized to make you write the fewest possible lines of code to do common tasks. `WvStream::getline()` works just like you want it to work. The standard C++ `iostream::getline()` doesn't.

What is the talk about?

The talk will be an overview of many different things WvStreams is good at, and how they apply to common, everyday programming tasks. I will give short examples in C++, illustrating each aspect of WvStreams that I describe, and showing how it can improve your life.

At the end, I will combine the previous topics to give a few longer examples that are complete, usable programs, and walk through the code. That's not as scary as it sounds, because the whole point is that the code isn't very long.

Relevant links

WvStreams project:	http://open.nit.ca/wiki/?page=WvStreams
Our open-source site:	http://open.nit.ca/
Personal:	http://people.nit.ca/~dcoombs/
Email:	dcoombs@nit.ca