

# Jukebox

## Table of contents

1 What is Jukebox.....	2
2 Brief History.....	2

## 1. What is Jukebox

The Jukebox is a framework of reusable components, patterns and idioms related to creation of high performance massive parallel, concurrent and distributed systems.

## 2. Brief History

The project originated in 1990, when the first commercial C++ compiler became available. At that time, it was a combination of a collection framework a-la Smalltalk with the smart pointer infrastructure allowing not to bother with pointers any more.

In 1992, it was ported to MS Windows (3.x at that time). In 1993, a banking operations division support system was written using this project's codebase as a foundation. That was the first time the semaphores and multithreading were introduced.

In 1994, it was ported to OS/2, the semaphore model was redesigned, and true multithreading was achieved for the first time. A countrywide project tracking system was written using Jukebox as a foundation.

In late 1995, Java arrived. At that time it was a sad event, because with it came the realization of the fact that all the collection framework was no longer necessary, so was the smart pointer framework, and most of the semaphore and multithreading code. Well, there's no cloud without a silver lining, the thorough redesign was long overdue anyway, so the project was ported into Java during 1996. Another distributed system, this time related to securities trading, was created on top of Jukebox.

In 1998, I've started working with [Apache JServ](#) team, and eventually sneaked my [logging infrastructure](#) into JServ, and got a nickname: "Captain Log" :) At the same time it has become obvious that the attempt to make Jukebox a "multiprotocol distributed framework" failed, because Sun had more manpower and could release APIs much faster than I did. I'm still aggravated at them for not including an obvious component into the J2EE architecture - yes, I'm talking about a container managed thread abstraction.

In 2000, Jukebox was chosen to become the foundation for the project at [American Express](#). That was the first take at high-load mission-critical application. Surprisingly, no major problems emerged, thus giving me some ego boost :)

Then, starting in 2001, Jukebox was used at another Open Source project, [DIY Zoning](#). This ongoing project is an attempt to scale the framework down, rather than up.

Since then, Jukebox is quietly evolving, shedding parts that are growing up elsewhere - good examples would be the logging subsystem ([log4j](#) is now used) and concurrent metaphors

*Jukebox*

(JDK 1.5 eventually introduced concurrent programming paradigms, but they're still not replacing some of paradigms Jukebox introduced back in mid-nineties).