

Impact Report

Catalyst Systems builds strong business around application lifecycle management

Analyst: [Rachel Chalmers](#)

Sector: [Enterprise Software »»](#)

Date: 5 Apr 2006

Email This Report: [to colleagues »»](#) / [to yourself »»](#)

Integrated development environments (IDEs) have made programming itself less onerous, but generating makefiles (for C and C++ programs) or Apache Ant scripts (for Java) remains slow and error-prone. That's a problem for corporate developers struggling to make applications transparent, compliant and auditable. The answer? Build automation, now emerging as a key link in application lifecycle management. **Catalyst Systems** has been working on its Openmake offering for 11 years, qualifying it as a pioneer in the field.

Impact assessment

The message

Catalyst Systems' Openmake automates the build process, allowing developers to abandon error-prone manual scripts. Build management software appeals to all development organizations, but especially to those struggling toward Sarbanes-Oxley compliance.

Competitive landscape

Codefast Perfectbuild is the closest competitor, but Electric Cloud, Kinook Software, Urbancode, Viewtier and VSoft Technologies all offer takes on build automation.

The 451 Assessment

Code generation is automated, production monitoring and regulatory compliance are automated, so why are developers still writing their own makefiles and Ant scripts? The answer is that in leading development shops, they're not. If the Eclipse application lifecycle framework (ALF) takes off the way Eclipse IDEs have done – and we confidently expect that it will – Openmake and its kindred are poised to make a killing.

Context

Catalyst Systems was founded by consultants Steve Taylor and Tracy Ragan in 1995. The two were engaged by a large bank to troubleshoot its credit card software. Despite strict source-control, the production application kept having performance problems. Taylor and Ragan traced the problem to the menagerie of make scripts written by individual developers to build their own code. They wrote a REXX program for OS/2 to generate standard makefiles; this was the ancestor to the Openmake product.

Today, the company remains entirely self-funded. Its 15 employees – including seven consultants and four engineers – service 40,000 users, paying from \$50 for lifecycle automation seats to \$350 for full build automation seats. Sixty percent of sales come via the channel: **CA Inc** is a reseller, accounting for 40% of sales, and **Serena Software** is an OEM, accounting for 20%. Other partners include **Aldon**, **AccuRev**, **IBM**, **Microsoft**, the Canadian source code management vendor **MKS Inc** and **Sun Microsystems**.

Technology

Openmake exploits a loophole in the development of development tools, if you will. IDEs like Rational and Visual Studio allowed developers to automatically generate application code, but the developers still had to write their own scripts to build that code. Openmake extended automation into the build process.

At its heart, Openmake is a client/server application with Web and command line front ends talking to a Knowledge Base server running on Tomcat. (You can also run the KB server on WebSphere or **Oracle's** OC4J.) Build rules, build parameters, audit reports and build logs are all stored in the KB server. Openmake uses SOAP to transport metadata between the Web or command line client and the KB server.

Once it's installed, Openmake takes upon itself the three chores developers used to have to perform before writing Unix makefiles or Apache Ant scripts. First, it locates the source file for the code to be built. Next, it checks for any necessary third-party libraries. Third, it compiles the code, either for debugging or production.

The software supports more than 200 developer tools out-of-the-box, including Java, J2EE, Rational Rapid Application Developer, Microsoft .NET and Visual Studio, Borland Delphi and JBuilder, native C, C++, COBOL and Assembler, **BEA Systems'** WebLogic and Oracle Developers. Supported target platforms include AIX, **Hewlett-Packard's** HP-UX, **Red Hat** Linux, **SuSE Linux**, Sun Solaris, Windows XP and z/OS.

Strategy

Sarbanes-Oxley compliance and an infusion of venture capital are changing the build management sector and raising the hype, but Catalyst CEO Ragan says there are deeper changes afoot. She's particularly interested in the Eclipse ALF, spearheaded by Serena Software. ALF is a set of SOAP services plus the BPEL (Business Process Execution Language) orchestration engine and an event manager, designed to let developer tools talk to IT production tools.

IBM is excited about ALF, and small wonder. For the first time, there's a framework for a problem identified in (say) Tivoli Composite Application Management for WebSphere to be sent over to a Rational WebSphere Studio Application Developer user, complete with all the relevant context. ALF has little or nothing to do with the IDE per se, so it's a perfect illustration of the evolution of Eclipse into a general-purpose integration framework. A companion project, Corona, lets Eclipse-based developer workbenches talk among themselves.

Openmake was the proof-of-concept build tool for Eclipse ALF. If the framework takes off – and it already has considerable momentum – Catalyst Systems stands to gain from the logical adoption of build automation as another transparent and accountable bridge between development and production management.

Competition

The closest competitor to Openmake is PerfectBuild, from **Codefast**. Like Openmake, it has a repository of information about platforms, tools, processes and preferences, and like Openmake it allows scriptless configuration. ElectricAccelerator from **Electric Cloud** aims to speed the process by executing parallel builds across scalable clusters of commodity boxes; a companion product, ElectricInsight, depicts how the build is structured and run.

Also in the running: Colorado-based **Kinook Software's** Visual Build; the open source CruiseControl from maverick systems integrator **ThoughtWorks**; Anthill Build Server from former game developer **Urbancode**; **Viewtier Systems'** Parabuild; and FinalBuilder, from the Australian software company **VSoft Technologies**.

SWOT analysis

Strengths	Weaknesses
Small and self-sustaining, Catalyst Systems boasts a widely used product and two powerful channel partners in CA and Serena Software. Openmake stands to gain from its association with Eclipse's ALF.	The company is very small and, outside its niche, not very widely known. The low price point for Openmake means the company must generate volume sales, making it heavily dependent upon its channel partners.
Opportunities	Threats
With transparency emerging as the key to Sarbanes-Oxley compliance, and application lifecycle management fast becoming the latest industry fad, we expect interest in build automation to skyrocket.	There's fierce competition within the build automation sector – VSoft's FinalBuilder, in particular, is getting rave reviews. We also expect encroachment from closely related sectors. IDE and source code management vendors will no doubt want a piece of this growth.