

## Distribution Completes the Software CM Process

*By: Mary Gillis and Mike Freedland, Emperex Corporation*

In today's fast paced business environment, the ability to quickly modify or create reliable information systems is critical to maintaining a company's competitive edge. The demands on software development departments are enormous. They must deliver higher volumes of feature-rich, error-free software applications, in shorter time frames and using fewer resources. This kind of pressure can easily lead to more application errors. So what can be done to minimize the chance of mistakes and protect your company?

Establishing and following a good Configuration Management (CM) process that includes an Enterprise Software Distribution (ESD) tool is the starting point to error avoidance. This process should take into consideration the entire life cycle of software applications, not just development. Managing an application's components as each release is created, distributed and made obsolete, is just as important as managing the coding or development of a software product.

As software components are compiled and packaged into a release, the value of the CM software becomes most evident, but until we reach the end of the CM process, the benefits of a system that manages the entire software life cycle including Software Deployment tend to be overlooked. The ability to "lock" all components and their dependencies to a release is critical to the guaranteed reproducibility of an application. Changes to a component must be done using a new version, and must not override any component or dependency component that needs to be kept in tact as part of an application. One problem is that most dependencies are not obvious because their references are hidden in the source files. Without a tool that locks each final release, it is almost impossible to know if a source change is "safe" or not.

The market demand for software distribution tools is increasing despite the depression in general IT spending over the past 2 years. Development and IT managers are revisiting their CM and software release processes asking themselves one question: "If my production system fails, how quickly could I rollback to an earlier and working release version?" In many cases, the answer is simply that a sure recovery is not possible. Without a distribution tool to lock down and preserve each product release, it is impossible to be sure that the contents of a post-production release are still valid.

These challenges include:

- **Volume of software releases.** Aside from the overall growth in software spending stated previously, most software vendors have responded to a slow market by further "modularizing" their products into smaller, less expensive components. The result is that while the enterprise may be paying less overall, the frequency of software distribution tasks is expanding rapidly for IT.
- **Security issues and vulnerabilities.** The rash of damaging viruses and worms over the last several years has created an unprecedented need for security patches and upgrades that must be distributed very quickly. The need for efficient software distribution tools, particularly on the desktop, has never been greater.
- **IT cost management issues.** Anxious to control costs, most enterprises have initiated sophisticated asset and software license management efforts to ensure that all assets are used efficiently and no wasted purchases are taking place. In order to execute these initiatives, enterprises need software distribution tools that can deploy applications quickly on an as-needed basis.
- **IT efficiency issues.** In the Windows and Unix environments, there is a renewed effort to bridge the gaps between previously-separate software lifecycle functions, including software development, packaging, distribution, configuration management, performance

## Distribution Completes the Software CM Process

*By: Mary Gillis and Mike Freedland, Emperex Corporation*

management and usage management. The goal is to create a single process for managing the lifecycle of the software, from purchase/development to retirement. In many cases, this integration of processes has necessitated the introduction of new software distribution tools that can work more readily with related tools, such as development, packaging, or applications management systems.

It is important to understand the range of "CM" software, and recognize the limitations of some of the available tools. Vendors with "source control" software sometimes claim to provide a "SCM" solution.

A good CM process will follow the chain of references and protect them from being changed by forcing the creation of a new version using proper security including both approvals and an audit trail and locking each release into a secure repository to provide guaranteed reproducibility. If the reproducibility of each release is not guaranteed, additional downtime results while one tries to locate or repair the parts, before any analysis can take place to fix the original problem.

A good software distribution tool automates manual tasks, another important aspect of error reduction. Automating the software deployment process will save valuable resources of time and man-power and provide release consistency across environments.