



Serena's Enterprise Release Option:
Simplifying Complexity



Serena's Enterprise Release Option: Simplifying Complexity

Enterprise Software Change Management (SCM) gives organizations the ability to develop and deploy applications across multiple computing platforms faster, and with greater reliability. It also enables them to protect critical software assets, and ensure the ongoing availability of the applications that drive business, partnerships, and revenue.

The increasing complexity of IT environments has placed new demands on Enterprise SCM systems, as software change must now be managed across geographies, multiple concurrent releases, technologies, methodologies and platforms. Consistency yields efficiency, yet very few organizations have established consistent SCM processes in each of their disparate computing environments, and fewer still have achieved an integrated enterprise infrastructure that gives developers, managers, quality assurance teams, and auditors the flexibility and control they need.

Serena gives organizations an advantage with the most thoroughly integrated multi-platform SCM solution available. Furthermore, Serena has developed the Enterprise Release Option (ERO), to extend its software change management solution to support organizations with multiple releases in motion simultaneously. For release-centric organizations, ERO represents a quantum leap in automation, productivity, auditability and quality assurance.

Bottom Line Benefits

Automating Enterprise SCM yields bottom line dividends by protecting application availability – leading analysts estimate that the cost of downtime can exceed \$1 million per hour (source: Meta Group) – and by providing time-to-market advantages that result in increased market share and the opening of new revenue streams. Minimizing the costs of administering an Enterprise SCM solution will enhance its quantifiable benefits, as will its acceptance by senior management and development teams, and the ease with which it can be tailored to the specified process needs of each organization.

Developer productivity can be maximized by ensuring the Enterprise SCM solution integrates with the IDEs and preferred development tools chosen by each organization. Building the solution on a low-bandwidth architecture inherently designed for cross-platform and Internet communication can minimize the administrative requirements and reduce the need for additional hardware purchases.

The ease with which a solution can be tailored to the specific needs of each organization will depend on the flexibility and control parameters built into the solution. The needs of development teams on different platforms, for instance, may vary greatly, and teams on each platform must be able to optimize their performance within the context of an overall enterprise solution.

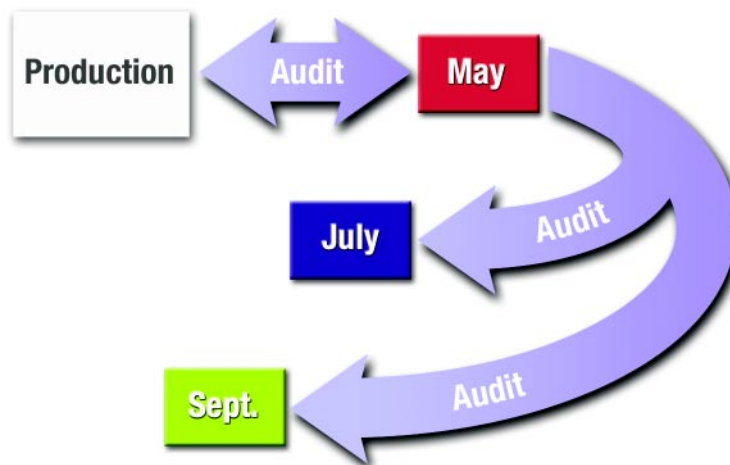
Furthermore, IT organizations may deploy varying development paradigms based on the demands of industry, the criticality of certain applications, or the sheer size of the organization itself. To ensure the quality of the applications being produced, all development activity should occur under the control of the Enterprise SCM solution. To maximize productivity and prevent production regression, communication between development teams must be ensured and supported through the implemented Enterprise SCM solution.

Challenges of Release-Centric Change

Historically, IT organizations that have embraced a release-centric development paradigm have faced the most significant challenges when implementing Enterprise SCM solutions. The presence of more than one release in the same time introduces unique demands on release managers, production control groups and senior management. As release schedules shift, maintaining the interdependencies between components in the development environment becomes both more critical and more difficult. Assuring that the right development teams are working with the right software components becomes an essential aspect of preventing overlays, regression and reworks, and achieving goals.

Solutions that focus on package-centric development approaches address the needs of many organizations. In a package-centric solution environment, all components of a change are grouped together at the beginning of the defined, repeatable, quality-driven development process. Developers and managers then act on the package as whole, comparing (auditing) it to the production version to ensure quality and that the correct component versions are used when changes are made.

In a release-centric environment, developers and managers must maintain the relationships between the components and packages that make up any given application release. Relationships must be ensured between production versions and all release versions in motion. The comparison (audit) must be made between releases, not packages.



Quality assurance for enterprises with multiple releases in motion.

When release schedules change, the Enterprise Release Option of Serena ChangeMan ZMF manages all component relationships, notifies release managers and ensures teams work with the right code, every time.

When release schedules shift and features get dropped from a release (or added), the release manager must make sure all development teams are working with the right components. There is tremendous potential for re-works and regression, and managing the process manually is a considerable challenge. Shifting schedules mean that software components that development teams working on future releases expect to be in production, may or may not actually be there. In a manually managed environment, maintaining the relationships and dependencies between components in all the releases in motion may become virtually impossible. The longer developers work with incorrect component versions, the greater the cost of changing release schedules. Furthermore, without an online audit function, inadvertent use of incorrect component versions poses a threat to production environments and regulatory requirements compliance.

Serena's ERO: A Quantum Leap

Serena's Enterprise Release Option (ERO) simplifies complexity in release-centric environments and delivers dramatic productivity improvements while automating communication and equipping release managers with tools for monitoring activity and component movement. For IT organizations driven by strategic objectives that result in several releases being active simultaneously, ERO aligns development practices with business objectives.

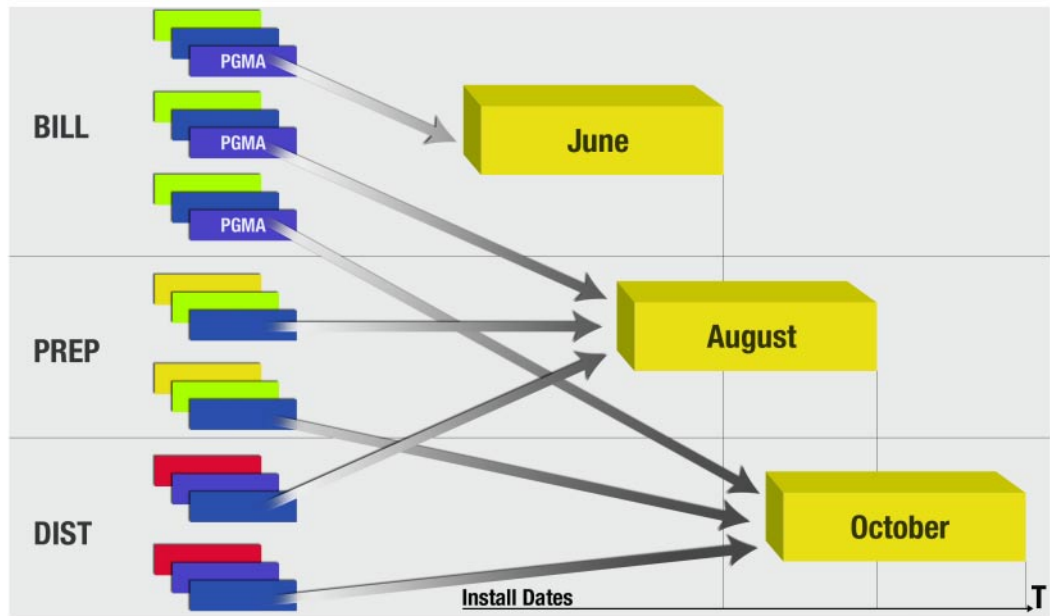
ERO:

- Provides release-centric organizations flexibility, consistency, and simplification benefits
- Protects against overlays and minimizes re-works
- Eliminates manual intervention
- Ensures that the right component versions make it into production every time
- Improves audit performance and compliance
- Replaces the need for extensive customizations to existing SCM solutions
- Simplifies maintenance and upgrades
- Automates communication between release management teams

An Elegant Solution

ERO is an elegant solution for release-centric development environments that simplifies the process of supporting multiple release versions. ERO is an optional module that extends Serena ChangeMan ZMF to streamline the release management process, and supports many packages from many applications in a series of dynamic and dependent releases, complete with audit capability. ERO allows all changes in multiple packages, and across multiple applications, to be managed at the enterprise level.

In complex environments, releases may comprise hundreds of change packages; the same programs may be changed in each release, and/or more than one release may be worked on at a time. A change anywhere in the release schedule can have an impact on any number of releases or release management teams.



Automatically track dependencies and open communication.

Having multiple releases in motion can create complex dependencies between components and releases, which can lead to re-works, regression, and lost productivity. The Enterprise Release Option automatically tracks dependencies and opens communication between release teams.

ERO makes it possible to establish and automate a consistent release management process. Organizations can manage releases as they move from development, testing and finally into production. ERO provides a development path in each release that progressively consolidates application package components into areas until they reside in a single set of area libraries.

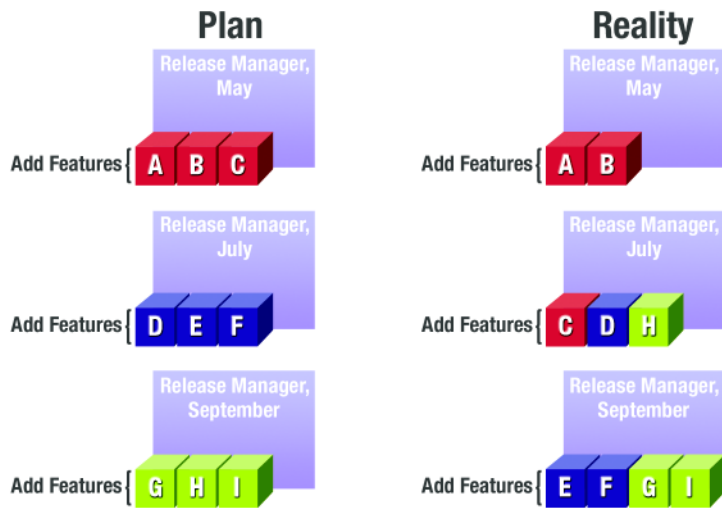
This consolidation guarantees that concurrent development of system objects across the enterprise is detected and resolved to prevent regression or loss of new function. It also provides a series of integrated test environments so that unanticipated effects of component changes can be identified and addressed.

In release-centric environments, development in later releases relies upon components that are scheduled for installation in earlier releases. The use of release libraries in build processes is dynamic. ERO's release audit uses the dynamic relationships between active releases and the contents of each release to evaluate the relationships between components in a release and the contents, both current and anticipated, of baseline libraries. ERO detects relationships that have been invalidated by a change in the relationship between releases or in the contents of a release.

ERO seamlessly manages changes in release order. Releases can be defined, and those scheduled to be installed ahead identified. When builds occur, the subordinate components will be pulled from the correct release automatically.

ERO provides a flexible release architecture. Release managers define releases, areas and applications in ERO, and can assign specific authority to individuals or teams to perform specific functions. The release life cycle is controlled by a set of rules that gives developers the latitude they need for rapid system development, and the rigor required to protect the production environment, all within a single release.

ERO allows the management of multiple releases along a consistent life cycle while providing robust protection and quality assurances.



Scalable release management.

When release schedules change and 'Feature C' moves to 'July', the Enterprise Release Option of Serena ChangeMan ZMF automatically recognizes all the changes, realigns the build processes, and resolves the arising out-of-sync situations.

Key Terms and Concepts

ERO introduces new terms and concepts to Serena ChangeMan ZMF for application in release-centric environments.

Release

A release represents a collection of ChangeMan ZMF change packages that must be installed at the same time. In ERO, a release is a logical set of rules for relating the physical elements of a release, which are the baseline libraries of the applications, the release area libraries, and the staging libraries of the change packages.

Release Application

Applications are joined to a release by an administrator, who also chooses what library types from each application are included in the release. By restricting library types in a release, administrators can build special purpose releases like on-line or batch releases.

Release Area

A release area is a set of libraries that represent a step in the consolidation of the components managed by the release. These libraries are used in SYSLIB concatenations for build processes in release packages, and they are used by release audit to validate release component relationships.

Release Package

A simple change package becomes a release package when it is attached to a particular release and its starting area is defined. The install date for a release package must fall within the range of the install dates defined for the release.

When a package is attached to a release, ERO takes control of building the SYSLIB concatenations for stage, recompile and relink jobs. If components in area libraries must be changed, a developer makes the change in package staging libraries using the familiar Serena ChangeMan ZMF procedures. The component is then checked-in into release again.

Consistent Processes = Quality Code

With ERO, organizations can establish development life cycles that support directly the best practices process for delivering quality code to market fast. The ERO release life cycle overlaps the package life cycle used in the base Serena ChangeMan ZMF product, and typically includes the following steps:

1. Create a Release

ChangeMan ZMF local administrators and ERO release managers make ERO administration entries to create a release. ERO includes robust administration.

2. Attach Change Package to Release

Attaching a change package to a release is the first step in bringing new or changed components into the ERO release life cycle. After attaching a package to a release, the components in the package remain under package control but proceed through the release life cycle.

3. Approve Area Check-in

Check-in approval opens a release area for package or area check-in. The approval process begins with notification to the area check-in approvers.

4. Check-in a Package

Package check-in brings components from a package attached to a release into the starting subsystem area defined for that package. This step begins the integration of the package components with other release components that are in development in other change packages of that release across the enterprise.

5. Check-in an Area

Area check-in copies components from the libraries for one area into the libraries for another area. Check-in advances release components through the hierarchy of areas that progressively integrates release components and passes them through environments for higher order testing.

6. Audit an Area

Serena ChangeMan ZMF ERO maintains the integrity of the components and applications under ERO control through the Release Audit, which is more sophisticated than the package audit delivered with the Serena ChangeMan ZMF base product. Release Audit examines the components in libraries for a particular release area, as well as libraries for other areas in the release, libraries in prior releases and baseline libraries. It evaluates relationships between different versions of the same component and the relationships between components and other components they include, such as copybooks and statically linked load modules, as well as the relationships between the link control cards and the load modules.

7. Block an Area

Blocking an area locks the area down to prevent further changes to area components. When an area is blocked, components cannot be checked-in or retrieved so that the area libraries can be used as testing environments under full control of the area owner who sets appropriate rules for all area functions.

8. Approve Area Check-off

Check-off approval is an administrative function that grants permission to check-in the contents of area libraries to the next area. The process begins with notification to check-off approvers. The requirement for check-off approval is determined by the area approval rule. A check-off approval signifies successful completion of the area testing as a step of the release life cycle.

9. Block a Release

Blocking a release locks down the release and its areas in preparation for install. All areas in a release must be blocked before a release can be blocked, and all packages attached to the release must be approved.

10. Approve a Release for Install

After a release is blocked, all install approvers must enter their approvals before the release is installed. When the last approval is entered, the release status is changed to APR, and installation JCL is built for each package attached to the release.

11. Backing Out a Release

Release back-out submits back-out jobs from the x-node libraries for all packages attached to the release. After all packages have been backed out, the package and the release are in set in BAK status.

12. Reverting Release

Reverting the release clears all release install approvals, unblocks the release, and changes the status of the release from APR or BAK to DEV status. The status of release areas is not changed, and packages attached to the release are not automatically reverted. This allows for selective unblocking of areas and reverting specific packages to rectify the problems before a next attempt to install the release is made.

Advancing Release Management and Enterprise SCM

For organizations to achieve maximum improvements in security, quality and time to market, Enterprise SCM must be viewed as an enabling technology and key infrastructure investment. Enterprise SCM must encompass technology and services to produce immediate infrastructure improvements, as well as a path for continued growth and leveraging of technology investments.

Serena has established itself as the leader in Enterprise SCM, delivering the first integrated multi-platform solution with a single point of control. Only Serena provides native SCM support for both distributed and mainframe platforms in an integrated solution that offers a single point of control and the ability to leverage fully the wide range of IDEs and development tools. Serena's ERO continues the company's leadership in the delivery of enterprise SCM technology by extending its solutions to support even the most complex development configurations and strategies.

Visit www.serena.com to find out more

To learn more about these products, and how to begin implementing comprehensive change management in your enterprise, visit Serena at www.serena.com or write us at info@serena.com.

About Serena Software

Serena Software, Inc. is the Enterprise Change Management (ECM) industry leader. For over 20 years Serena has focused exclusively on providing application change management solutions to the world's leading enterprises, and today its products are in use at over 2,750 customer sites, including 42 of the Fortune 50. Serena leads the way in ECM by providing a single point of control to manage software code and Web content changes throughout the enterprise, from the mainframe to the Web. This ensures application availability and speeds time to market, while reducing development costs. With headquarters in San Mateo, California, Serena serves customers worldwide through local offices and an international network of distributors. www.serena.com.

