



**Rational** software

## IBM Software Configuration Library Manager Advanced Edition for z/OS, Version 1.2

---

### Highlights

---

- ***Provides a cost-effective and fully functional centralized IBM z/OS software configuration management solution available in one easy-to-order package***
- ***Allows developers and administrators to use a single repository to store, manage, build and deploy Java and Java EE applications, as well as traditional applications like COBOL and PL/I***
- ***Simplifies setup and management of SCLM-based projects***
- ***Enhances asset protection with the ability to define authorized access at a more granular level***
- ***Helps improve productivity with rapid access to assets, automated package approval workflows, support for parallel development and enhanced usability features***
- ***Allows practitioners to invoke all key SCLM functions using a simplified workstation-based GUI in addition to the traditional ISPF-based interface***

### **Centralized version control and build support for the IBM z/OS platform**

To improve competitiveness, organizations are modernizing the applications that support key business processes—so essential functions can be redeployed in new ways using the latest business models. For many companies, this means integrating new service-oriented components with traditional IBM z/OS® platform–based program objects. The challenge is to effectively manage modification of z/OS assets for reuse while incorporating new technologies.

Your source-code modules are critical to your business endeavors. You want to be sure they're stored in a security-rich environment, yet they must be easy to access from anywhere. Whether onsite or remote, managers and developers need to get into their source-code libraries to efficiently handle program changes, additions and approvals. The z/OS platform has long been a system of choice for storing the most valuable assets of organizations across the globe. Now, IBM offers security-rich GUI access and approvals for those assets.

IBM Software Configuration Library Manager (SCLM) Advanced Edition for z/OS, Version 1.2 software, when used in conjunction with IBM SCLM for z/OS software, is a cost-effective and fully functional centralized z/OS software configuration management solution available in one comprehensive and convenient package. With SCLM Advanced Edition, you can invoke the SCLM function from Eclipse technology–based integrated development environments (IDEs).

### **Integrate management of distributed and z/OS artifacts with SCLM Advanced Edition**

SCLM is a feature of Interactive System Productivity Facility (ISPF), which is included with the z/OS operating system. If you are a z/OS user, then you already have SCLM. SCLM Advanced Edition provides companies running the z/OS platform with a change and configuration management solution for administering the development of distributed and z/OS artifacts. Source-code management and highly secure production turnover are provided on either an easy-to-use ISPF interface or a workstation-based

GUI that uses standard, easy-to-follow graphical metaphors. SCLM Advanced Edition packages the following products:

- *IBM SCLM Administrator Toolkit*
- *IBM Enhanced Access Control (EAC) for SCLM for z/OS*
- *IBM SCLM Developer Toolkit*
- *IBM Breeze for SCLM for z/OS*
- *IBM Merge Tool for z/OS*

SCLM Advanced Edition, Version 1.2 is being refreshed with the 3.1 version of the SCLM Developer Toolkit. You will find descriptions of the newest features of SCLM Developer Toolkit later in this document.

### **Simplify management of SCLM-based projects with SCLM Administrator Toolkit**

The SCLM Administrator Toolkit provides a set of tools that can make defining and maintaining SCLM projects easier than ever before. You can access administrator functions from either a workstation-based GUI or ISPF panels, enabling you to be effective and productive, regardless of your preferred interface.

#### **Streamlined project setup**

SCLM Administrator Toolkit includes a number of wizards to guide you through the setup and maintenance of your SCLM environment. An EAC wizard works with the EAC component

of SCLM Advanced Edition to enable you to create, list, modify and delete IBM Remote Access Control Facility (IBM RACF®) data sets and profiles. Using the EAC wizard, you can also define, modify, view and delete application profiles; view access and update violations; and change rule files from a centralized console.

A project wizard enables you to select data-set types, specify package back-out parameters and define hierarchy groups. If you are modifying an existing project and want to delete a data-set type or group, the project wizard can, optionally, delete any source-code data associated with that type or group. You can also configure required user exits, including those available in Breeze, through the project wizard.

SCLM Administrator toolkit enhances usability via:

- *The ability to parse and build projects using newer SCLM release features.*
- *Finer-grained control for power users.*
- *Access to current Eclipse features, and the ability to integrate with other products, such as IBM Rational® Developer for IBM System z™ software.*
- *A user interface that can self-customize to the modification level of the SCLM functionality installed.*

In z/OS, Version 1.8 and later systems, SCLM Administrator Toolkit enables you to specify the following SCLM features:

- *Member locking*
- *An SCLM administrator*
- *A control data set where SCLM administrator information is stored*
- *A language definition description*
- *IOTYPE of H on FLMALLOC statements, allowing you to generate a relevant FLMCPYLB statement that specifies a file resident in a hierarchical file system (HFS) when developers deliver code changes*

You can also use SCLM Administrator Toolkit to correctly parse projects using the newer features in z/OS, Version 1.8 or later.

The user interface in SCLM Administrator Toolkit, Version 3.1 has several improvements that enable you to save time when providing necessary project details by:

- *Using wildcards in all member lists to migrate multiple members in a single invocation of the migration wizard and to generate CC and LEC ARCHDEFs in bulk.*
- *Generating comments to be included in the project definition source code.*
- *Preserving original comments in the project definition about parsing a project and rebuilding it.*

- *Eliminating the need to specify a language name for second and subsequent attempts to migrate the same member into a project.*
- *Allowing incomplete language definitions without affecting the ability to build a project.*

Also, finer-grained control features give administrators the ability to:

- *Specify data-set types for PROJDEFS data sets and GROUP/TYPE data sets.*
- *Specify data-set allocation parameters including system-managed storage (SMS) parameters for data sets similar to the ISPF data-set allocation panel.*
- *Specify Virtual Storage Access Method (VSAM)–cluster Freespace parameters.*

SCLM Administrator Toolkit, Version 3.1 is based on the Eclipse 3.2.1 framework and is integrated with IBM Rational Developer for System z software. Other functions in SCLM Administrator Toolkit include automated project hierarchy generation, language definition wizards, automatically generated architecture definitions, project cloning and an asset migration wizard.

#### **Automated project hierarchy generation**

When defining a project lifecycle or hierarchy using the workstation interface, you can drag and drop boxes around the display to represent project groups and build a network of linked

group boxes, much like an organization chart. After the definition is completed, SCLM Administrator Toolkit generates an SCLM project hierarchy. A similar process to define or modify groups is also available at the host using ISPF panels.

#### **Language definition wizards**

You can add, modify or remove language definitions using the language definition wizard. You can create new language definitions from existing definitions, existing Job Control Language (JCL) or manually through the editor.

#### **Automatically generated architecture definitions**

SCLM Administrator Toolkit enables you to automatically generate SCLM architecture definitions from an existing load module or JCL. A manual option for creating architecture definitions is also available.

#### **Project cloning**

Cloning a project enables you to replicate project definitions or entire projects, and it can include your application's source code.

#### **Asset migration wizard**

You can easily move software assets into an SCLM project from existing libraries using the migration wizard. And with the new remote-migration wizard, you can move assets, including those with long file names, from remote systems into SCLM.

#### **Easier project maintenance**

SCLM Administrator Toolkit provides the ability to perform project VSAM and partitioned data set (PDS) allocation, deletion and maintenance tasks. These robust features and functions enable SCLM Administrator Toolkit to help you simplify the maintenance of your SCLM-managed projects.

#### **Improve data protection and access security with Enhanced Access Control**

EAC for SCLM provides you with more flexibility and control over access to your SCLM-managed libraries. Security interfaces and user exit points within EAC for SCLM provide granularity and protection for data sets so that authorized modifications can be made only by a native SCLM user. You're able to add new rules that specify exactly which programs can be used to access which data sets. This flexibility goes well beyond standard RACF programs to define specific users' read or write access to specific documents. You can use a menu-driven interface to define validation profiles, program access, and user or user group privileges.

#### **Granular access control**

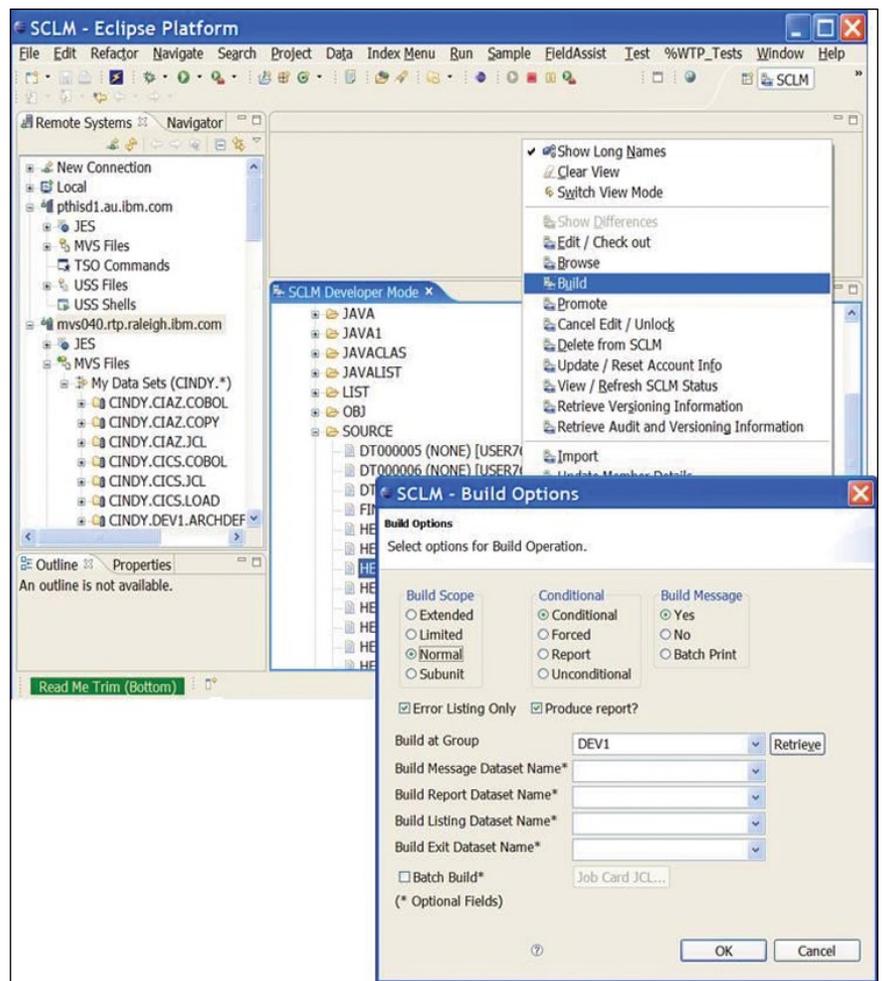
EAC for SCLM augments RACF controls—after normal RACF security controls have been applied—by granting access when a specific set of applications like SCLM is used. The applications can even define various subfunctions of SCLM, so that, for example, an SCLM promote function can be allowed access, whereas an SCLM edit function might be denied access.

Without EAC for SCLM, SCLM users operating in an RACF environment must be granted update access to manipulate data sets managed by SCLM. Otherwise, they would receive RACF data-set violations when attempting to perform various SCLM functions. However, the update access applies even if the data set is accessed using facilities other than SCLM, which enables access to potential users from facilities other than SCLM.

#### Audit trail of access violations

The central concept of EAC for SCLM is that access to SCLM resources is provided when SCLM programs are used, helping to avoid the potential for accidental damage or unexpected changes to SCLM data sets that can result from updates using non-SCLM programs.

When EAC for SCLM is active, it monitors RACF data-set violations. If a violation occurs for a data set managed according to the EAC for SCLM profiles, you can use the defined access rules to assign access privileges. If sufficient access privilege is not defined, then an RACF data-set violation occurs. Like RACF, EAC for SCLM has its own rules database that describes the conditions under which access is granted. These are contained in the rule file, a VSAM key-sequenced data set (KSDS) that you administer through the ISPF dialog.



The SCLM Developer Toolkit provides Eclipse technology-based IDEs with the proven dependability and functional capabilities of SCLM.

From these online panels, you can:

- Define the data sets or generic RACF data-set profiles you want to control.
- Define SCLM and its subfunctions as applications.
- Define the users to be granted access privileges to a profile through an application.
- View violation records collected by EAC for SCLM.

#### Improve developer productivity with SCLM Developer Toolkit, Version 3.1

Most development projects today use a variety of technologies and platforms. Centralizing the software configuration management activities of these development projects can resolve many of the difficulties associated with managing such complex projects.

SCLM Developer Toolkit facilitates collaboration between z/OS and non-z/OS developers and provides investment protection with an architecture that can extend the languages supported by SCLM. The toolkit combines the functional richness of the Eclipse technology-based IDE and the proven integrity and dependability of SCLM to provide application developers and SCLM administrators with an intuitive interface to a flexible, robust and reliable environment for rapidly and securely modernizing z/OS applications.

#### **Shorter learning curves**

For application developers with little or no prior exposure to the z/OS development environment, learning how to develop z/OS applications can take a long time, and a tool that reduces that learning period can significantly improve developer productivity. By removing the reliance on IBM 3270 system access, SCLM Developer Toolkit enables you to concentrate development activities on the rapidly expanding set of Eclipse IDE-based tools by using only a single development environment. The transparent way in which SCLM services can be invoked from the Eclipse IDE means you can use SCLM as the software configuration manager for application development projects, which can potentially shorten developers' learning curves.

Features and services such as "check in" and "check out" are universal software configuration management functions. SCLM Developer Toolkit

delivers these functions, and all other SCLM services, including a large number of software configuration management services such as project synchronization, file comparison and version control, in an Eclipse software configuration management-based, consistent manner. This capability means you can use the IDE services that currently exist for your Eclipse projects while using SCLM for source management.

#### **Support for multiple programming languages**

With its integrated support for storing SCLM files with long names, such as Java™ program files, HTML files, graphics files and other workstation files, SCLM Developer Toolkit enables Java and Java Platform, Enterprise Edition (Java EE) developers to code and test their applications in an Eclipse technology-based IDE and then store their range of application components in SCLM.

You can also use SCLM Developer Toolkit to build and deploy applications within the z/OS environment using SCLM, much as COBOL developers do, because SCLM Developer Toolkit supports deployment of Java and Java EE objects to the IBM WebSphere® Application Server for z/OS platform.

SCLM Developer Toolkit enables you to store source code in almost any language in SCLM. Consequently, the SCLM Developer Toolkit architecture

can extend the level of language support that SCLM can provide through the addition of SCLM build translators. As Eclipse tools are added to the development environment, SCLM Developer Toolkit can support those features. Consolidating your development environment around these open source tools using a proven (and developing) mainframe application environment offers a compelling development scenario. For example, the build services provided with SCLM Developer Toolkit use Ant, the Java technology-based "make" tool, as the principal build tool. Using Ant provides a high level of customizability of the build process. Developers can either generate build scripts or modify skeleton scripts to drive the appropriate build behaviors for SCLM projects.

#### **Enhanced usability features**

SCLM Developer Toolkit, Version 3.1 includes a number of enhancements designed to improve usability. All SCLM functionality provided in SCLM Developer Toolkit, Version 3.1 is included as the SCLM Developer Toolkit component in Rational Developer for System z, Version 7.1. Users of the 3.1 version of SCLM Developer Toolkit have different mechanisms to transport data between the Eclipse technology-based client and the back-end host SCLM repository and server. When running outside Rational Developer for System z, the HTTP transport is used. When running under Rational Developer for System z, the embedded Remote Systems Explorer (RSE) transport is used instead.

Productivity-enhancing benefit	IBM SCLM Developer Toolkit, Version 3.1 capability
Support multiple projects	<ul style="list-style-type: none"> <li>• Display and work with multiple projects residing on different z/OS systems in the SCLM view.</li> <li>• Automatically refresh the existing SCLM view instead of adding a duplicate branch to the tree when you run the populate action multiple times with all options exactly the same.</li> </ul>
Streamline general housekeeping functions	<ul style="list-style-type: none"> <li>• Specify the maximum size of the operations log folder or the maximum length of time to retain operations logs.</li> <li>• Store separate job cards for each group that is used as a build target, a promote source or a deployment source.</li> <li>• Improve logging and messaging for batch build processing to help give clearer error message reporting, making it easy to debug build errors.</li> </ul>
Improve support for non-mainframe programming objects	<ul style="list-style-type: none"> <li>• Enable VSAM RLS (record level sharing) support for shortname/longname table when the shortname/longname file is shared across multiple systems.</li> <li>• Provide fast Java EE build performance that's especially noticeable on project builds of greater than 1,000 files.</li> <li>• Support the SQLJ language type to allow SQLJ projects to be stored and built in SCLM.</li> </ul>
Enhance usability	<ul style="list-style-type: none"> <li>• Display batch output in an editor, allowing you to use editor commands to easily locate data within the batch job.</li> <li>• Find relevant information more easily with color-coded operations logs that allow you to automatically scroll to the first highlighted message; and you can open the log in the default editor by double-clicking in the tree node.</li> <li>• Enhance your ability to customize user preferences with several new options on the preferences page.</li> <li>• Use the key primary ISPF commands in addition to those provided by the LPEX editor.</li> </ul>

*The new SCLM Developer Toolkit features enhance the ability of z/OS and non-z/OS developers to collaborate on complex applications.*

### **Rapidly build and approve reusable packages with Breeze for SCLM**

Breeze for SCLM for z/OS is designed to enhance the package processing and promotion capabilities of SCLM. It enables you to build reusable packages, assign approvers and automatically notify approvers and managers using either time-sharing option (TSO) or e-mail when changes need to be reviewed. Not only can this automation streamline in-house approval procedures and provide organization-wide accountability for source changes migrated into production, it can also facilitate the approval process for developers and managers who are in remote locations or on the road.

### **Administering packages from remote locations**

Just a few years ago, IT managers and mainframe application developers did not necessarily need Web access or e-mail notification for package approvals, but today, these communication methods are essential. IT professionals need the ability to perform administrative tasks from remote locations. Breeze meets this need by incorporating e-mail-based notification of packages awaiting approval with Web access to the package data and the ability to approve or to deny SCLM packages through a Web browser. There's no need to log in to TSO.

### **Automating approval process by package type**

Breeze packages are identified as either standard or emergency. Standard packages require review and approval before they can be promoted to the next stage in the application lifecycle. Emergency processing enables you to promote a package to the next stage without requiring the full approval process; yet it still helps ensure that all appropriate parties are notified. This critical function helps enable fixes to be integrated into production quickly.

Managing the consolidation of changes	IBM Merge Tool for z/OS capabilities
1. Identify the components that have changed	Match base file members with other members having the same name from one or two other files. Based on the matching up of members, you can easily identify components that require merging.
2. Analyze the complexity of the change	Build a preliminary work file to obtain member statistics, including the number of conflicts found. These statistics can give you a better understanding of how many conflicts there are and the resources required for the merge process.
3. Merge and review the changes	Automatically generate a marked-up work file that shows inserts, deletions and overlap or conflict areas.
4. Address any merge conflicts	Edit the work file from within the Merge Tool and resolve conflicts before generating your final merge file.
5. Create merged source	Generate the merge file output by using the Merge Tool to automatically read the work file as input and perform the insertions and deletions indicated in the file. The merge file contains the merged source code, which can then be stored in your software configuration management tool or used as input to a compile.
6. Test the merged output	Compile and link the merged source code, and then test the changes.

*Merge Tool for z/OS speeds the process of consolidating independently coded changes into a working, cohesive whole.*

With Breeze, you can define approvers based on both inventory and location. Defining approvers based on inventory enables you to identify different approvers for different kinds of application data. You might have one set of approvers who approve general source changes and another set who approve database application components, such as COBOL with IBM DB2® programs.

By defining approvers based on the location of the inventory within the application lifecycle, you can further define approvers based on where you are in your development process. For instance, you might require only technical approvals at the first stage of the application lifecycle, and then require management approval for packages to be promoted into production.

### **Support parallel development with Merge Tool for z/OS**

Merge Tool for z/OS provides both project managers and z/OS application developers with the tools they need to identify, analyze and consolidate independently coded changes into a working, cohesive whole. Changes need to be made in parallel for many reasons—an emergency fix, concurrent development, multiple release development or maintenance, or customization of vendor application code. But a common process for managing the consolidation of those changes is necessary (see the above table).



Merge Tool enables project managers and application developers to easily manage the process of consolidating parallel development. The ISPF Workbench, a simple-to-use ISPF interface, provides access to Merge Tool actions and reports. You can determine the degree of merge action based on the complexity of the changes. And you can save time by performing most Merge Tool actions in batch mode.

Merge Tool reports can speed assessment of source consolidation. The statistics file is particularly useful in identifying and sizing a consolidation effort. When conflicts are identified, application developers can easily access, view and edit the work file produced by the merge process to resolve issues before generating the final merge result.

#### **For more information**

To learn more about how the IBM SCLM Advanced Edition for z/OS solution supports the development and management of mixed applications, contact your IBM representative, or visit:

[ibm.com/software/awdtools/configmgr](http://ibm.com/software/awdtools/configmgr)

To see the hardware and software requirements for the IBM SCLM Advanced Edition solution, please visit:

[ibm.com/software/awdtools/configmgr/sysreqs](http://ibm.com/software/awdtools/configmgr/sysreqs)

© Copyright IBM Corporation 2007

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America  
10-07  
All Rights Reserved

DB2, IBM, the IBM logo, RACF, Rational, System z, WebSphere and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.