



IBM Software Configuration Library Manager Advanced Edition for z/OS, Version 1.1

Highlights

- ***Provides a cost-effective and fully functional centralized z/OS software configuration-management solution available in one easy-to-order package***
- ***Enables developers and administrators to use one central repository to store, manage, build and deploy Java and J2EE applications, as well as traditional applications like COBOL and PL/I, through the use of the same Eclipse IDE interface as other IBM development tooling***
- ***Enables you to easily approve packages, implement and augment security controls, run three-way merges and simplify management of SCLM-based projects***
- ***Allows you to invoke all key SCLM functions using a simplified workstation-based graphical interface in addition to the traditional ISPF-based interface***
- ***Includes valuable functions that you can use with SCLM, such as version comparisons, wildcard searching at the group or type level, identification of package approvers and package-collision detection***

The focus on service oriented architecture (SOA) is spreading rapidly throughout today's business landscape. Increasing complexity in IT infrastructures makes the race to remain competitive ever more challenging. At the same time, enterprises have huge investments in IBM z/OS® platform-based applications. The key to staying ahead of the competition is in your ability to reuse and modernize your existing IT assets while incorporating new technologies.

IBM Software Configuration Library Manager (SCLM) Advanced Edition for z/OS, Version 1.1 provides the solution in one complete and convenient package. Whether you are creating new applications or modifying existing applications, the need to manage integration of new components with traditional z/OS program objects is a critical element of building and deploying modern applications.

SCLM Advanced Edition enables you to use the significant investment you have made in your z/OS platform-based applications while using an Eclipse technology-based integrated development environment (IDE) interface. With SCLM Advanced Edition, you can invoke SCLM function from your Eclipse IDE.

Your source-code modules are critical to your business endeavors. You want to be sure they are stored in a security-rich environment, yet they must be easy to access from anywhere. Managers and developers, whether on-site or remote, need ready and highly secure access to their source-code library to efficiently handle source-code development and approvals. The z/OS platform has long been the system of choice for storing the most valuable assets of organizations across the globe. Now, SCLM Advanced Edition provides security-rich GUI access and approvals for those assets.

SCLM Advanced Edition features

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 functionally rich software 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 graphical metaphors that are intuitive and easy to follow.

SCLM Advanced Edition includes these 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 Administrator Toolkit

SCLM Administrator Toolkit provides a set of tools that 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.

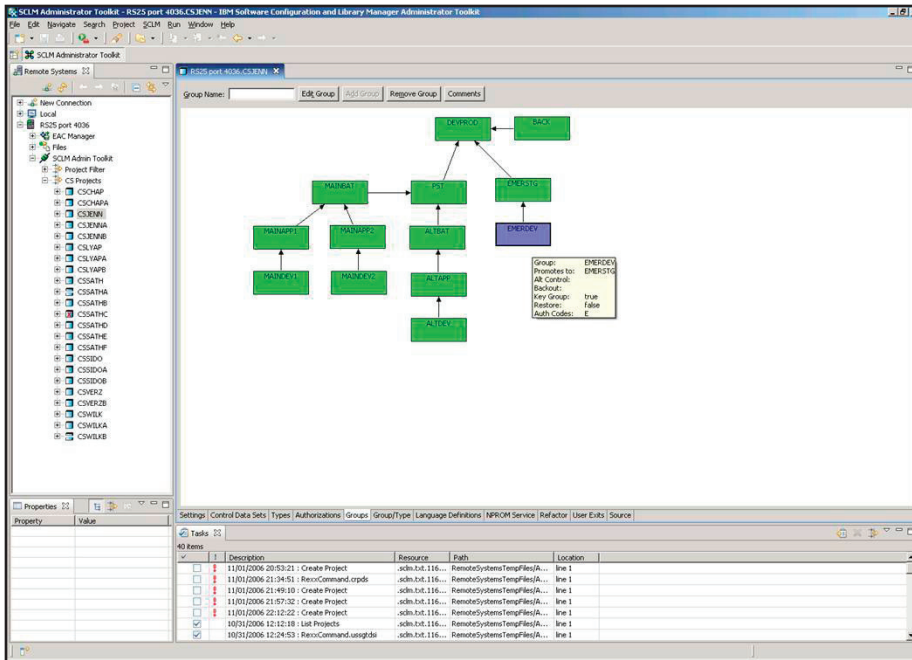
SCLM Administrator Toolkit includes a number of wizards to guide you through the setup and maintenance of your SCLM environment. An EAC wizard works with the EAC component of SCLM Advanced Edition to enable you to create, list, modify and delete IBM Remote Access Control Facility (IBM RACF®) data sets and profiles. The EAC wizard also allows you to 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 Breeze user exits, through the project wizard.

With this release, SCLM Administrator Toolkit, Version 3.1 provides:

- *The ability to parse and build projects using newer SCLM release features.*
- *An improved user interface with enhanced usability.*
- *The ability for power users to have finer-grained control.*
- *The ability to take advantage of current Eclipse features, as well as the ability to integrate with other products, such as IBM WebSphere® Developer for System z.*
- *A user interface that can self-customize to the modification level of SCLM installed.*



SCLM Administrator Toolkit provides both an ISPF- and a workstation-based interface for managing your SCLM projects.

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

- Member locking
- An SCLM administrator
- A control data set where SCLM administrator information is stored
- A description for a language definition
- An IOTYPE of H on FLMALLOC statements, and then you can generate a relevant FLMCPYLB statement that specifies a file resident in a hierarchical file system (HFS)

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

The user interface in SCLM Administrator Toolkit, Version 3.1 includes several improvements as well. These improvements enable you to:

- Use 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.
- Generate comments to be included in the project-definition source code.
- Preserve original comments in the project definition about parsing a project and rebuilding it.

- Avoid having to specify a language name for second and subsequent attempts to migrate the same member into a project.
- Leave language definitions as incomplete without affecting the ability to build a project.

Also, finer-grained control features in SCLM Administrator Toolkit give you 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 Eclipse, Version 3.2.1 and is integrated with WebSphere Developer for System z.

Other features and functions of SCLM Administrator Toolkit include:

Generating project hierarchies

When defining a project life cycle 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.

Adding language definitions

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.

Generating 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.

Cloning projects

Cloning a project enables you to replicate project definitions or entire projects (including your application's source code).

Moving assets

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.

Other functions

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.

Enhanced Access Control for SCLM for z/OS

Security interfaces and user exit points within EAC for SCLM provide granularity and protection for SCLM data sets so that authorized modifications can be made only by a native SCLM user. EAC for SCLM enables you to add new rules that specify exactly which programs can be used to access which data sets, providing flexibility beyond that found in RACF programs to define specific users' read or write access to specific documents. You can use a menu-driven interface to define validation profiles, access to programs, and user or user group privileges.

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 are 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.

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.

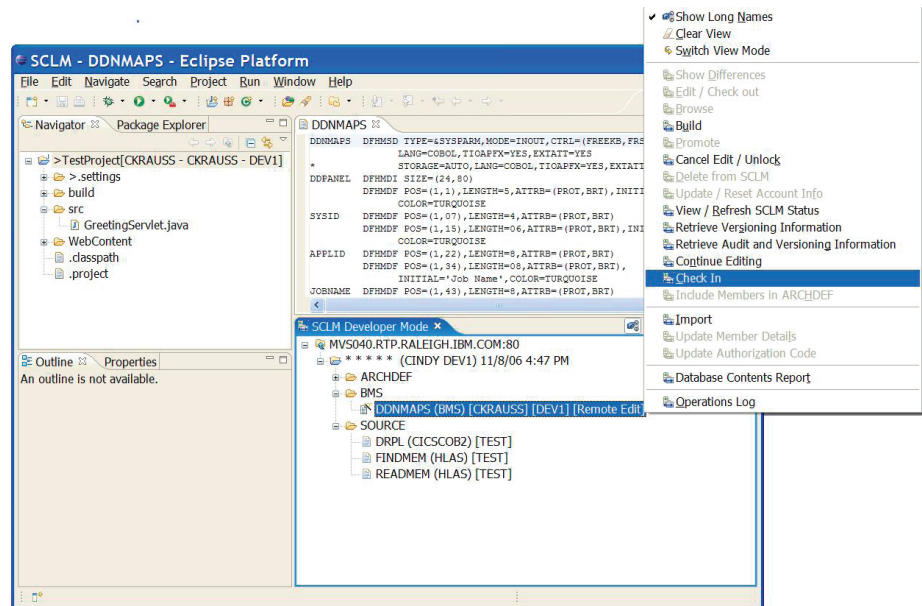
When EAC for SCLM is active, it monitors RACF data-set violations. If a violation occurs for a data set managed according to the EAC for SCLM profiles, then you can use the defined access rules to assign access privileges. If sufficient access privilege is not defined, then a RACF data-set violation 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. 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.

EAC for SCLM provides you with more flexibility and control over access to your SCLM-managed libraries.

SCLM Developer Toolkit

Today's application-development projects use a variety of technologies and platforms. As a result, centralizing the software configuration-management activities of these development projects can avoid or lessen many of the difficulties associated with managing such complex projects. SCLM Developer Toolkit combines the functional richness of the Eclipse technology-based IDE and the proven integrity and dependability of SCLM to give both application developers and SCLM administrators a new interface to a flexible, robust and reliable environment for rapidly and securely modernizing z/OS applications.



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

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 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 that 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 that you can use the IDE services that currently exist for your Eclipse projects while using SCLM for source management.

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 enables Java and Java 2 Platform, Enterprise Edition (J2EE) 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 J2EE objects to IBM WebSphere Application Server for z/OS.

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 they can modify “skeleton” scripts to drive the appropriate build behaviors for SCLM projects.

New features in SCLM Developer Toolkit

SCLM Developer Toolkit, Version 2.1 includes a number of enhancements designed to improve usability and productivity. All SCLM functionality provided in SCLM Developer Toolkit, Version 2.1 is now included as the SCLM Developer Toolkit component in WebSphere Developer for System z, Version 7, replacing the SCLM (VCM) plug-in that was provided with earlier versions of IBM WebSphere Studio Enterprise Developer, Version 5 and IBM WebSphere Developer for zSeries®, Version 6.

Now, users of SCLM Developer Toolkit, Version 2.1 have different mechanisms to transport data between the Eclipse technology-based client and the back-end host SCLM repository and server. When running outside WebSphere Developer for System z, the HTTP transport is used as it currently is in SCLM Developer Toolkit, Version 1. When running SCLM Developer Toolkit under WebSphere Developer for System z, the Remote Systems Explorer (RSE) transport that comes with WebSphere Developer for System z is used instead.

The WebSphere Developer for System z zIDE currently doesn't enforce SCLM workflow controls on members that are under SCLM control. The z/OS projects perspective within WebSphere Developer for System z can now check whether a data set is under SCLM control, and if so, give you the option to browse the file or cancel the operation.

SCLM Developer Toolkit, Version 2.1 provides the ability to associate a default bidirectional (BIDI) format with each SCLM language, and to perform BIDI conversions both for RSE and HTTP transports when the SCLM member BIDI format is different from that used on a client platform. Project-level default BIDI format can be specified for languages that don't have default formats associated with them. Also, the BIDI user interface now enables you to control (view or edit) the BIDI default format of SCLM objects. In addition, SCLM Developer Toolkit, Version 2.1 provides support for storing files with BIDI or double-byte character set (DBCS) files names in SCLM.

J2EE application enhancements include:

- *New Java and J2EE build translators that replace the existing translators, enabling the ARCHDEF itself to be built as is the normal process with other host languages. All components are tied in to the one build map, so a promotion of an ARCHDEF promotes all relevant parts up the hierarchy chain.*
- *High-level ARCHDEF support that is provided in the new language translator. The existing language translators JAVABLD and J2EEBLD are replaced with a new language translator for the build script (J2EEANT) and a new language translator associated with the ARCHDEF (J2EEOBJ).*

- *The ability to deploy J2EE applications without having to depend on the J2EE build service. This new deploy service can be used to invoke the deploy script associated with the application. Also, deploy scripts can be built and promoted up the SCLM hierarchy without automatically triggering application deployment, enabling you to customize these scripts for each level of the hierarchy.*
- *The ability to securely deploy a built object onto a WebSphere Application Server platform using the WebSphere tool suite, or to physically move the built object to a non-WebSphere Application Server target using a defined transport service.*

With this release, SCLM Developer Toolkit includes the ability to optionally add security checks into the SCLM build, promote and deploy processes by securing the processes through security access facility (SAF)-RACF resource definitions.

If the requesting user ID meets the SAF and RACF authority check, then it optionally allows the SCLM build, promote or deploy process to run under a surrogate Time Sharing Option (TSO) user ID for that request. This version also enables the build process to store within a built archive file, a bill of materials in the form of a build map and, optionally, the source code to build it. This capability satisfies audit requirements for the identification of code used to build applications and modules.

SCLM Developer Toolkit, Version 2.1 also includes a number of usability enhancements. Populating a project now provides the project listing in the actual SCLM hierarchical order so that the groups displayed are represented as an SCLM tree structure. You can also change your z/OS password from SCLM Developer Toolkit, either in response to an expired password or as a change to an existing password. And the SCLM customization service configures an SCLM project with a number of defined settings that the SCLM Developer Toolkit client uses to control access to certain SCLM services, as well as enforcing SCLM project policies.

Instead of having to perform operations on members individually if more than one is open, you can use SCLM Developer Kit, Version 2.1 to check in, check out and unlock services on multiple members using a single menu operation. You can also add members to an existing SCLM explorer and developer view. Instead of your having to repopulate the view based on a filter list, SCLM Developer Toolkit now preserves the current list along with the new members. You can also refresh the current view along with the new member list so that the existing member list updates with the current SCLM status.

In addition, previously, if you wanted to refresh a particular view of a project, you had to repopulate. The population dialog remembered the last settings, but if you had multiple projects in your view, the settings might not be correct for what you wanted to refresh. Now, SCLM Developer Toolkit can remember the filter settings for the target project, and has a context menu item that repopulates those stored filters.

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.

Breeze for SCLM for z/OS

Breeze for SCLM for z/OS is designed to enhance the package-processing capabilities of SCLM. It enables you to build reusable packages, assign approvers and automatically notify both approvers and managers using either TSO or e-mail when changes need to be reviewed. A Web interface enables approvers to review the package and package contents, and then approve or veto the package without logging on to TSO.

Just a few years ago, IT managers and mainframe applications developers did not necessarily need Web access or e-mail notification for package approvals, but today, these communication methods are essential tools for most businesses. By enhancing proven SCLM technology, Breeze provides a solution for both developers and managers who want to take advantage of the strengths of SCLM as a powerful development and source-code management tool. 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 the approval of SCLM packages through a Web browser.

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 life cycle. Emergency processing enables you to promote a package to the next stage without requiring the full approval process, yet still helps ensure that all appropriate parties are notified. This critical function helps enable fixes to be integrated into production quickly.

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 that approve general source changes and another set that approves database-application components, such as COBOL with IBM DB2® programs.

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

Not only does Breeze software enable you to streamline in-house approval procedures and provide organization-wide accountability for source changes migrated into production, it is also particularly valuable for facilitating the approval process for developers and managers who are in remote locations or who are on the road.

Breeze for SCLM helps improve the promotion of packages through the software development life cycle.

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, customization of vendor application code. But a common process for managing the consolidation of those changes is necessary. You need to:

- *Identify the components that have changed. Merge Tool enables you to match up base file members with other members having the same name from one or two other files. Based on the matchup of the members, you can easily identify components that require merging.*

- *Analyze the complexity. Merge Tool enables you to 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.*
 - *Merge and review the changes. The merge process generates a marked-up work file that shows inserts, deletions, and overlap or conflict areas.*
 - *Address any merge conflicts. You can edit the work file from within Merge Tool and resolve conflicts before generating your final merge file.*
 - *Create merged source. After you have addressed any conflicts, then you are ready to generate the merge file. Merge Tool reads the work file as input and performs the insertions and deletions indicated in the file to generate the merge file output. The merge file contains the merged source code, which can then be stored in your SCM tool or used as input to a compile.*
 - *Test the merged output. The final step is to compile and link the merged source code and then test the changes.*
- *ISPF Workbench, a simple-to-use ISPF interface that provides access to Merge Tool actions and reports.*
 - *Merge Tool actions, a variety of actions that are available to perform merge tasks. You can determine the degree of merge action based on the complexity of the changes.*
 - *Reports, to aid in the assessment of source consolidation. The statistics file is particularly useful in identifying and sizing a consolidation effort.*
 - *Work file, produced by the merge process that application developers can edit to review the changes and resolve any merge conflicts before generating the final merge result.*
 - *Batch utility, to perform most Merge Tool actions in batch.*

Merge Tool for z/OS can help improve your developers' productivity when they need to consolidate source code.

For more information

SCLM Advanced Edition is an excellent choice for a cost-effective and fully functional centralized z/OS software configuration-management system that supports the development and management of mixed applications. To learn more about SCLM Advanced Edition for z/OS, contact your IBM representative, or visit:

ibm.com/software/awdtools/configmgr/

IBM SCLM Advanced Edition for z/OS, Version 1.1 at a glance

Hardware requirements

For host system

Any IBM System z™ platform capable of supporting the required software

For workstations

- 768 MB RAM required (1 GB or more recommended) when running WebSphere Developer for System z
 - 512 MB RAM required (1 GB or more recommended) when not running WebSphere Developer for System z
 - 512 MB available disk space
 - Processor: Intel® Pentium® III at 800 MHz required; faster recommended
 - Video Graphics Array (VGA) display: 1024x768 minimum
 - TCP/IP connectivity with host mainframe system
-

Software requirements

Host system

- z/OS, Version 1.4.0 or later (Version 1.6 or later recommended)
 - z/OS program temporary fix (PTF) PQ84190 (for Version 1.4 or 1.5 only)
 - z/OS PTF that fixes APAR OA11426 (to support long-name translation)
 - Java, Version 1.4.2
 - Operational UNIX® System Services
 - ANT run time installed in UNIX Systems Services
 - REXX runtime support or alternate libraries
 - HTTP server using basic authentication
 - IBM RACF
 - One of the following if the GUI client is used: OMVS REXEC Service in TCP/IP or RSE daemon
 - IBM XML Toolkit for z/OS, Version 1.7
-

IBM SCLM Advanced Edition for z/OS, Version 1.1 at a glance (continued)

software requirements (continued)

Workstation using GUI

- Operating systems (one of the following)
 - IBM AIX 5L™ on IBM Power PC®
 - Red Hat Linux® Pro, Version 8.0 on Intel x86 with GIMP tool kit (GTK)
 - Microsoft® Windows® XP Professional with Service Pack (SP) 1 or later
 - Windows 2000 Professional with SP4
 - Microsoft Server 2003 Standard
 - Microsoft Server 2003 Enterprise
 - TCP/IP connectivity with host mainframe system
 - Java, Version 1.3 runtime or developer kit installed (used by client installer only)
 - TCP/IP installed and configured
 - Web browser
-

Workstation with 3270 interface

- A 3270-emulator product (such as IBM Personal Communications or IBM Host-on-Demand) that is capable of connecting to your host system and processing the screens generated by ISPF
 - Connectivity with your host system through TCP/IP or Systems Network Architecture (SNA), depending on your network
-

For a comprehensive list of hardware and software requirements for all the components of SCLM Advanced Edition for z/OS or for individual products, visit ibm.com/software/awdtools/configmgr/sysreqs.



© Copyright IBM Corporation 2006

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

Produced in the United States of America
12-06
All Rights Reserved

AIX 5L, DB2, IBM, the IBM logo, PowerPC, RACF, System z, WebSphere, z/OS and zSeries are trademarks of International Business Machines Corporation in the United States, other countries or both.

Intel and Pentium are trademarks of Intel 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.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries or both.

UNIX is a trademark of The Open Group in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries or both.

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