z/OS Basic Skills Information Center

Security on z/OS

z/OS Basic Skills Information Center

Security on z/OS

Note
Before using this information and the product it supports, read the information in "Notices" on page 13.

This edition applies to z/OS (product number 5694-A01).

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Send your comments through this Web site: http://publib.boulder.ibm.com/infocenter/zoslnctr/v1r7/index.jsp?topic=/com.ibm.zcontact.doc/webqs.html.

Contents

Overview of security on z/OS

An installation's data and application programs must be protected from unauthorized access — both internally (employees) and externally (customers, business partners, or hackers). In working with z/OS®, you need to understand the importance of security and the z/OS facilities used to implement it.

Over time, it has become easier to create and access computerized information. No longer is system access limited to a handful of highly skilled programmers. Information can now be created and accessed by almost anyone who takes a little time to become familiar with the newer, easier-to-use, high-level inquiry languages.

More and more people are becoming increasingly dependent on computer systems and the information they store in these systems. As general computer literacy and the number of people using computers has increased, the need for data security has taken on a new measure of importance. No longer can the installation depend on keeping data secure simply because no one knows how to access the data.

Making data secure encompasses more than just making confidential information inaccessible to those who should not see it. It also includes preventing the inadvertent destruction of files by people who may not even know that they are improperly manipulating data. Good data security practices reduces the likelihood of unauthorized persons accessing, modifying, or destroying data, either inadvertently or deliberately.

Access, in a computer-based environment, means the ability to do something with a computer resource (for example, use, change, or view something). Access control is the method by which this ability is explicitly enabled or restricted. Computer-based access controls are called *logical access controls*. These are protection mechanisms that limit users' access to information to only what is appropriate for them.

Logical access controls are often built into the operating system, or may be part of the logic of application programs or major utilities, such as database management systems. They may also be implemented in add-on security packages that are installed into an operating system; such packages are available for a variety of systems, including PCs and mainframes. Additionally, logical access controls may be present in specialized components that regulate communications between computers and networks.

Chapter 1. Roles in z/OS security

Mainframe environments tend to be well-structured, with formal roles, such as systems programmer, security administrator, and auditor, that are assigned to separate individuals. This separation of duties is a cornerstone of security and mainframe management. In essence, Ability should not exceed Authority.

A significant difference to note, when deploying a mainframe as opposed to a distributed server environment, is the way in which job definitions and roles are defined and how the IT staff is assigned duties, as explained here:

- In a distributed environment, people often handle multiple duties in the interest of efficiency. For example, an operator who has the authority to shut down the system might also have the ability to delete user IDs.

  However, giving staff the authorization for many tasks, while in one sense efficient, opens the door for abusing this power. For example, a database administrator who sold a corporation's information to its competition might have the ability to hide these actions from auditors.
- In a mainframe environment, by contrast, skills are generally more focused on a specific responsibility. That is, there tends to be more separation of duties. Each mainframe support person is a specialist, yet mainframes usually operate with fewer support personnel relative to the size of the user community because of the centralized nature of mainframe management tools. The efficiency derives from the platform architecture, not from people sharing duties.

In the past, it was the mainframe system programmer who, working with management, decided the overall security policy and procedures. Today companies are seeking higher levels of security, so they often appoint a separate security manager. The system programmer might not have direct responsibility for security, other than advising the security manager about new products. Separation of duties is necessary to prevent any one individual from having uncontrolled access to the system.

Who is the security administrator?

The security administrator is the focal point for planning security in the installation. RACF® gives the security administrator (that is, the user defined with the SPECIAL attribute) many responsibilities both at the system level and at the group level.

The security administrator is responsible for:
- Determining which RACF functions to use
- Identifying the level of RACF protection
- Identifying which data RACF is to protect
- Defining administrative structures and users.

A system administrator assigns user IDs and initial passwords and ensures that the passwords are non-trivial, random, and frequently changed. Because the user IDs and passwords are so critically important, special care must be taken to protect the files that contain them.

In z/OS, the security administrator can use *RACF Remote Sharing Facility (RRSF)* to administer and maintain RACF databases distributed throughout the enterprise. This facility:

- Provides improvements in system performance, system management, system availability, and usability
- Helps to ensure that data integrity is retained across system or network failures and delays
- Informs you when key events have occurred and returns output to view at your convenience.

Who is the security auditor?

Security audits are a way of examining a system, policy or process for violations and exposures.

Auditing is the process of ensuring that the information processing system (hardware, software, liveware, middleware, policies, and procedures) complies with the installation security policy. Auditing may be:

- A one-time project such as a snap inspection, or
- An ongoing process pursuant to policies.

Security audits are a catch-all that have been used for actions ranging from checks on physical security to implementation of the information security plan.

The two types of information security audits can be termed preemptive and reactive. As their names indicate, preemptive audits test security controls. Reactive audits respond to potential security breach events.

Who is the system operator?

Console security means controlling which commands operators can enter on their consoles to monitor and control z/OS.

How you define command authorities for your consoles, or control logon for operators, enables you to plan the operations security of your z/OS system or sysplex. In a sysplex, because an operator on one system can enter commands that affect the processing on another system, your security measures become more complicated and you need to plan accordingly.

When implementing console security, the installation can control which commands operators can enter on their consoles to monitor and control z/OS. To do so, the installation uses RACF and the CONSOLxx member in PARMLIB.

For multiple console support (MCS) consoles, you can use the following to control whether operators can enter commands from a console:

- The AUTH keyword on the CONSOLE statement of CONSOLxx
- The LOGON keyword of the DEFAULT statement and RACF commands and profiles.

For extended MCS consoles, you can control what an authorized SDSF or TSO/E user can do during a console session. Because an extended MCS console can be associated with a TSO/E user ID and not a physical console, you might want to use RACF to limit not only the z/OS commands a user can enter, but from which TSO/E terminals the user can enter the commands.

Chapter 2. Security facilities of z/OS

z/OS facilities provide its high level of security and system integrity.

Data about customers is a valuable resource that could be sold to competitors. Thus, the aim of any security policy is to provide users with only their required level of access and to deny non-authorized users access. This is one reason why auditors prefer that users or groups are granted specific access, rather than using universal access facilities. The traditional focus of mainframe security was to focus on stopping unauthorized people from logging on to the system, and then ensuring that users were only allowed access to data on a need-to-know basis. As mainframes have become Internet servers, however, additional security has been required. There are outside threats such as hackers, viruses, and Trojan horses; Security Server includes tools to deal with these.

However, the main threat to company data is often from within. An employee within a company has a much better chance of obtaining data than someone outside. A well-thought-out security policy is always the first line of defense.

Further, z/OS provides a number of system integrity features to minimize intentional or accidental damage from other programs. Many installations run several copies of z/OS and often do not permit general TSO/ISPF users to access the production systems. z/OS security controls can protect the production environment if they are properly configured and prevent a TSO/ISPF user (either maliciously or accidentally) from impacting important production work.

What is SAF?

System authorization facility or **SAF** is an interface defined by MVS™ that enables programs to use system authorization services to control access to resources, such as data sets and MVS commands. SAF either processes security authorization requests directly or works with RACF, or other security product, to process them.

SAF does not require any other product as a prerequisite, but overall system security functions are greatly enhanced and complemented if it is used concurrently with RACF. The key element in SAF is the SAF router. This router is always present, even when RACF is not present.

The SAF router provides a common focal point for all products providing resource control. This focal point encourages the use of common control functions shared across products and across systems. The resource managing components and subsystems call the z/OS router as part of certain decision-making functions in their processing, such as access-control checking and authorization-related checking. These functions are called **control points**.

The system authorization facility (SAF) conditionally directs control to RACF (if RACF is present), or to a user-supplied processing routine, or both, when receiving a request from a resource manager.

What is IBM Security Server?

IBM® Security Server is a set of features in z/OS that provide security.

Security Server provisions include:
- Controlling the access of users (user ID and password) to the system
- Restricting the functions that an authorized user can perform on the systems' data files and programs

Many installations use a package called Security Server, which is commonly referred to by the name of its most well-known component, RACF. Resource Access Control Facility (RACF) is a component of Security Server. It controls access to all protected z/OS resources. RACF protects resources by granting access only to authorized users of the protected resources and retains information about the users, resources, and access authorities in specific profiles.

The following is a list of the security components of z/OS that are collectively known as Security Server:

- DCE Security Server

This server provides a fully functional OSF DCE 1.1 level security server that runs on z/OS.

- Lightweight Directory Access Protocol (LDAP) Server

This server is based on a client/server model that provides client access to an LDAP server. An LDAP directory provides an easy way to maintain directory information in a central location for storage, update, retrieval, and exchange.

- z/OS Firewall Technologies

This is an IPV4 network security firewall program for z/OS. In essence, the z/OS firewall consists of traditional firewall functions as well as support for virtual private networks.

The inclusion of a firewall means that the mainframe can be connected directly to the Internet if required without any intervening hardware and can provide the required levels of security to protect vital company data. With the VPN technology, securely encrypted tunnels can be established through the Internet from a client to the mainframe.

- Network Authentication Service for z/OS

This provides Kerberos security services without requiring that you purchase or use a middleware product such as Distributed Computing Environment (DCE).

- Enterprise Identity Mapping (EIM)

This offers a new approach to enable inexpensive solutions to easily manage multiple user registries and user identities in an enterprise.

- PKI Services

This allows you to establish a public key infrastructure and serve as a certificate authority for your internal and external users, issuing and administering digital certificates in accordance with your own organization's policies.

- Resource Access Control Facility (RACF)

This is the primary component of the Security Server; it works closely with z/OS to protect vital resources.

The topic of security can be a whole course by itself. In this section, we introduce you to the RACF component and show how its features are used to implement z/OS security.

What is RACF?

Resource Access Control Facility or **RACF** provides the tools to help the installation manage access to critical resources.

Any security mechanism is only as good as the management control of the people who access the system. **Access**, in a computer-based environment, means the ability to do something with a computer resource (for example, use, change, or view something). **Access control** is the method by which this ability is explicitly enabled or restricted. It is the responsibility of the installation to see that access controls that are implemented are working the way they are supposed to work, and that variances are reported to and acted on by management.

Computer-based access controls are called **logical access controls**. These are protection mechanisms that limit users' access to information to only what is appropriate for them. Logical access controls are often built into the operating system, or can be part of the logic of application programs or major utilities, such as database management systems. They may also be implemented in add-on security packages that are installed into an operating system; such packages are available for a variety of systems, including PCs and mainframes. Further, logical access controls might be present in specialized components that regulate communications between computers and networks.

To be effective, access control must allow management to adopt the principle of least possible privilege for those resources that are deemed to be highly sensitive. This principle says that access to these resources is controlled in such a way that permission to use them is restricted to just those people whose normal duties require their use. Any unusual use of the resource should be approved by an administrator or manager, as well as the owner of the resource.

Resource Access Control Facility or **RACF** provides the tools to manage user access to critical resources. RACF is an add-on software product that provides basic security for a mainframe system (examples of other security software packages include ACF2 and Top Secret, both from Computer Associates).

RACF protects resources by granting access only to authorized users of the protected resources. RACF retains information about users, resources, and access authorities in special structures called **profiles** in its database, and it refers to these profiles when deciding which users should be permitted access to protected system resources.

To help your installation accomplish access control, RACF provides the ability to:
- Identify and authenticate users
- Authorize users to access protected resources
- Log and report various attempts of unauthorized access to protected resources
- Control the means of access to resources
- Allow applications to use the RACF macros

RACF uses a user ID and a system-encrypted password to perform its user identification and verification. The user ID identifies the person to the system as a RACF user. The password verifies the user's identity. Often exits are used to enforce a password policy such as a minimum length, lack of repeating characters or adjacent keyboard letters, and also the use of numerics as well as letters. Popular words such as "password" or the use of the user ID are often banned.

The other important policy is the frequency of password change. If a user ID has not been used for a long time, it may be revoked and special action is needed to use it again. When someone leaves a company, there should be a special procedure that ensures that the user IDs are deleted from the system.

RACF, with its lists of users and lists of resources, allows management to delegate the authority to the owners of these entities in such a way as to maintain the separation of duties while maintaining a flexible, responsive access control strategy.

The delegation mechanism in RACF and the easy, nontechnical commands that change the relationship of a user to a resource mean that adopting the principle of least possible privilege need not be burdensome nor inflexible when unusual circumstances dictate that access permission should be changed. When an unforeseen circumstance requires a change in access privilege, the change can be made by a nontechnical person with access to a TSO terminal, and management can be alerted to review the fact that the change was made.

Major subsystems such as CICS® and DB2® use the facilities of RACF to protect transactions and files. Much of the work to configure RACF profiles for these subsystems is done by the CICS and DB2 system programmers. So, there is a need for people in these roles to have a useful understanding of RACF and how it relates to the software they manage.

Chapter 3. z/OS and system integrity

z/OS includes features and facilities specifically designed to protect one program from affecting another, either intentionally or accidentally. The ability of an operating system to protect data and itself from unauthorized changes is called *system integrity*.

Protecting the system involves a number of related disciplines:
- Maintenance of system integrity
- Use of the authorized programming facility
- Use of the resource access control facility (RACF),
- Changing system status
- Protecting low storage.

System integrity is defined as the ability of the system to protect itself against unauthorized user access to the extent that security controls cannot be compromised. That is, there is no way for an unauthorized program using any system interface to bypass store or fetch protection, bypass password checking, bypass RACF checking, or obtain control in an authorized state.

An authorized program in the system is one that runs in PSW key 0-7, in supervisor state, or is authorized through the authorized program facility (APF). An unauthorized program is a problem state program that runs in PSW key 8-F.

Installation Responsibility

To ensure that system integrity is effective and to avoid compromising any of the integrity controls provided in the system, the installation must assume responsibility for the following:
- Physical environment of the computing system.
- Adoption of certain procedures (for example, the password protection of appropriate system data sets) that are a necessary complement to the integrity support within the operating system itself.
- That its own modifications and additions to the system do not introduce any integrity exposures. That is, all installation-written authorized code (for example, an installation SVC) must perform the same or an equivalent type of validity checking and control that the system uses to maintain its integrity.

Elimination of potential integrity exposures

System integrity support restricts only unauthorized problem programs. It is the responsibility of the installation to verify that any authorized programs added to the system control program will not introduce any integrity exposures. To do this effectively, an installation should consider these areas for potential integrity exposure:
- User-supplied addresses for user storage areas.
- User-supplied addresses for protected control blocks.
- Resource identification.
- SVC routines calling SVC routines.
- Control program and user data accessibility.
- Resource serialization (for example, through locking).

What is the authorized program facility?

The **authorized program facility** or **APF** is used to allow the installation to identify system or user programs that can use sensitive system functions. To maintain system security and integrity, a program must be authorized by the APF before it can access restricted functions, such as supervisor calls (SVC) or SVC paths. APF helps to avoid integrity exposures; the installation identifies which libraries contain special functions or programs. These libraries are then called APF libraries.

An authorized program can do virtually anything that it wants. It is essentially an extension of the operating system. It can put itself into supervisor state or a system key. It can modify system control blocks. It can execute privileged instructions (while in supervisor state). It can turn off logging to cover its tracks. Clearly, this authorization must be given out sparingly and monitored carefully.

Your installation can use the authorized program facility (APF) to identify system or user programs that can use sensitive system functions. For example, APF allows your installation to:

- Restrict the use of sensitive system supervisor call (SVC) routines (and sensitive user SVC routines, if you need them) to APF-authorized programs.
- Allow the system to fetch all modules in an authorized job step task only from authorized libraries, to prevent programs from counterfeiting a module in the module flow of an authorized job step task.

Many system functions, such as supervisor calls (SVCs) or special paths through SVCs, are sensitive. Access to these functions must be restricted to only authorized programs to avoid compromising the security and integrity of the system.

The system considers a task authorized when the executing program has the following characteristics:

- It runs in supervisor state (bit 15 of the program status word (PSW) is zero).
- It runs with PSW key 0 to 7 (bits 8 through 11 of the PSW contain a value in the range 0 to 7).
- All previous programs executed in the same task were APF programs.

APF-authorized programs must reside in one of the following authorized libraries:

- SYS1.LINKLIB
- SYS1.SVCLIB
- SYS1.LPALIB
- Authorized libraries specified by your installation.

Authorized libraries are defined in an APF list, or in the link pack area. Any module in the link pack area (pageable LPA, modified LPA, fixed LPA, or dynamic LPA) will be treated by the system as though it came from an APF-authorized library.

The installation must ensure that it has properly protected SYS1.LPALIB and any other library that contributes modules to the link pack area to avoid system security and integrity exposures, just as it would protect any APF-authorized library.

APF also prevents authorized programs (supervisor state, APF-authorized, PSW key 0-7, or PKM 0-7) from accessing a load module that is not in an APF-authorized library.

What is storage protection?

Mainframe hardware has a **storage protection** function, which is normally used to prevent unauthorized alteration of storage. Storage protection is also used to prevent unauthorized reading of storage areas, although z/OS protects only small areas of storage this way.

Storage protection works on 4K pages. It deals only with real memory, not virtual memory. When a page of virtual memory is copied from disk to a free page in main storage, z/OS also sets an appropriate storage protection key in that page of main storage.

Storage protection was much more significant before multiple address spaces came into use. When multiple users and jobs were in a single address space (or in real memory in the days before virtual memory), protecting a user's memory from corruption (or inappropriate data peeking) was critical. With z/OS, the primary protection for each user's memory is the isolation provided by multiple address spaces.

Storage protection keys cannot be altered by application programs. There is no way, using the storage protection function, for a normal application program (not an **authorized program**) to protect part of its virtual memory from other parts of the application in the same address space.

An additional storage protection bit (for each 4K page of real memory) is the **page protection** bit. This prevents even system routines (running in key 0, which can normally store anywhere) from storing in the page. This bit is typically used to protect LPA pages from accidental damage by system routines.

Controlling cross-memory communication

In z/OS, **cross-memory communication** allows a program in one address space to communicate with a program in another address space.

With proper page table management by the operating system, users and applications in different address spaces are completely isolated from each other. One exception to this isolation is the common area. Another exception is cross-memory communication.

A number of cross-memory capabilities are possible, but two are commonly used:
- Ability to call a program that resides in a different address space
- Ability to access (fetch, store) virtual memory in another address space.

The first case uses the **program call** (PC) instruction. Here, only a single hardware instruction is needed to call a program in another address space. A common example of this involves DB2, an IBM database management product. Various parts of DB2 occupy up to four address spaces. Users of DB2 can be TSO users, batch jobs, and middleware, such as a Web server. When these users issue SQL instructions for DB2, the SQL interface in the application uses a program call to obtain services from the DB2 address spaces.

Cross-memory programming must be coordinated through z/OS security controls. In practice, almost all cross-memory usage is in major middleware products and is rarely directly used by typical application programs.

Chapter 4. Security for z/OS UNIX

The security administrator needs to prepare RACF to provide security and to define users to RACF. For a user to be a z/OS UNIX® user, the user's default group must be a z/OS UNIX group.

z/OS UNIX provides security mechanisms that work with the security offered by the z/OS system. A security product is required, either RACF or an equivalent security product. If you do not have a security product, you must write SAF exits to simulate all of the functions.

The z/OS UNIX security functions provided by RACF include user validation, file access checking, privileged user checking, and user limit checking. z/OS UNIX users are defined with RACF commands. When a job starts or a user logs on, the user ID and password are verified by RACF. When an address space requests an z/OS UNIX function for the first time, RACF:

- Verifies that the user is defined as a z/OS UNIX user.
- Verifies that the user's current connect group is defined as a z/OS UNIX group.
- Initializes the control blocks needed for subsequent security checks.

To establish data and system security for z/OS UNIX resources, the security administrator and security auditor might need to work together to accomplish the following:

- Managing group identifiers and user identifiers (GIDs and UIDs)
- Allowing all z/OS UNIX users to transfer file ownership to any UID or GID on the system
- Giving superuser authority to users
- Changing superusers from UID(0) to a unique nonzero UID
- Defining RACF groups to z/OS UNIX groups
- Setting up the FILE.GROUPOWNER.SETGID profile
- Setting up sanction list processing
- Maintaining the security level of the system.

The security administrator needs to prepare RACF to provide security and to define users to RACF. For a user to be a z/OS UNIX user, the user's default group must be a z/OS UNIX group.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This book documents information that is NOT intended to be used as Programming Interfaces of z/OS.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

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

Microsoft®, Windows®, Windows NT®, and the Windows logo are trademarks of Microsoft 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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Printed in USA