Redbooks
ibm.com/redbooks

Security in Development
The IBM Secure Engineering Framework

Warren Grunbok

Marie Cole

IBM®

Redguide

Executive overview

IBM® has long been recognized as a leading provider of hardware, software, and services that are of the highest quality, reliability, function, and integrity. IBM products and services are used around the world by people and organizations with mission-critical demands for high performance, high stress tolerance, high availability, and high security.

As a testament to this long-standing attention at IBM, demonstration of this attention to security can be traced back to the Integrity Statement for IBM mainframe software, which was originally published in 1973:

> *IBM's long-term commitment to System Integrity is unique in the industry, and forms the basis of MVS (now IBM z/OS) industry leadership in system security. IBM MVS (now IBM z/OS) is designed to help you protect your system, data, transactions, and applications from accidental or malicious modification. This is one of the many reasons IBM 360 (now IBM Z) remains the industry's premier data server for mission-critical workloads.*

This commitment continues to apply to IBM's mainframe systems and is reiterated at the z/OS Security Server RACF General User's Guide web page.

The IT market transformed in 40-plus years, and so have product development and information security practices. The IBM commitment to continuously improving product security remains a constant differentiator for the company.

In this IBM Redguide™ publication, we describe secure engineering practices for software products. We offer a description of an end-to-end approach to product development and delivery, with security considered.

IBM is producing this IBM Redguide publication in the hope that interested parties (clients, other IT companies, academics, and others) can find these practices to be a useful example of the type of security practices that are increasingly a must-have for developing products and applications that run in the world's digital infrastructure. We also hope this publication can enrich our continued collaboration with others in the industry, standards bodies, government, and elsewhere, as we seek to learn and continuously refine our approach.

Background

IBM is involved in the following core areas of software development:

- Products and solutions for sale
- Operation of solutions and services for its own internal use
- Operation of solutions and services on behalf of clients

To help meet client demand for flexible and full-function information technology solutions, IBM software product development teams design products that integrate with, and operate within, a wide range of operating systems and programming language environments. IBM products, solutions, and services can integrate IBM developed software, open source code, third-party code, and potentially customized extensions or applications into composite products and solution offerings.

The development of IBM products and solutions is distributed across organizations and laboratories worldwide. The magnitude of the secure engineering and process control challenges that are involved in producing high-quality software in such a global development environment is significant.

Given IBM's breadth of products development processes and thus application of secure engineering to that development vary from product to product. Each development team is responsible for applying these techniques and practices to best suit that teams must ensure that IBM provides quality software to it customers.

The key to delivering products and services that meet clients' high expectations is to focus product development execution in the following critical areas:

- Common Development Process
- Secure Engineering Framework
- Continuous Security Improvement model
- Supply Chain Security process
- Security and Privacy by Design

The combination of process, framework, and model integrate with a broader set of externally facing processes that us referred to as *global supply chain management*.

In the remainder of this publication, we provide more information about the elements of secure product development and a secure supply chain. Development teams are encouraged to follow these practices and procedures, and each development team follows variations of what is presented here.

Common development process

Information Technology organizations must employ a common development process to provide consistent management, technical oversight, and accountability across a wide range of hardware, software, services, and solution development projects. To achieve high levels of efficiency, quality, and security, the common development process should be supported by a set of enforceable and measurable standards and directives.

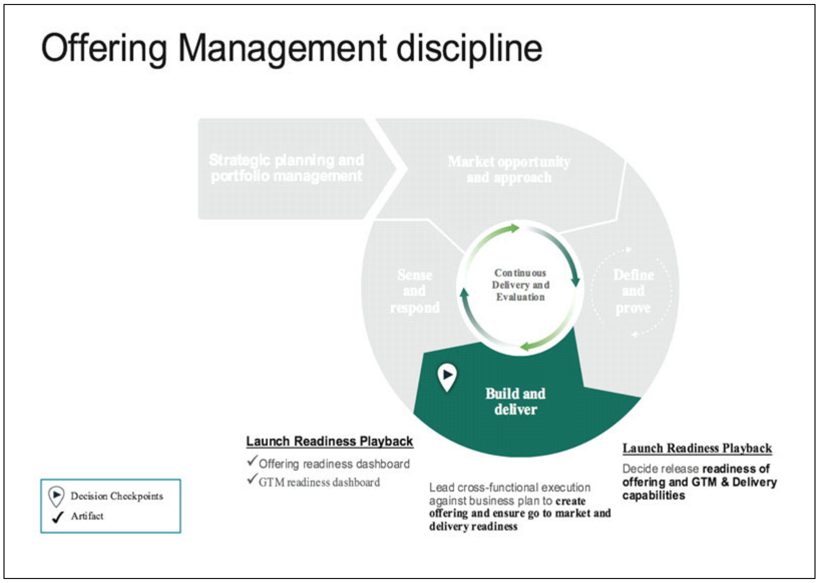
In addition to providing for accountability and control, the common development process enables the coordination of people, technology, and information that are involved in the development lifecycle of components, products, and solutions.

Development projects are outcome-oriented. Each team must have sufficient flexibility to adopt tools and practices that can enhance their ability to deliver, if the results meet the governance criteria and follow the overall development process.

Understanding each products reason for being and how customers use a product is important to also understand any security risks to which a product might be exposed, or risks the product might cause to the customer.

IBM uses a system of product development that is known as Offering Management to try to understand the market early. A series of Market planning stages are used to understand, plan, run, and manage performance throughout the products development lifecycle through a process called Offering Management (OM).

Although the offering is being developed, the OM is also working with the Offering Delivery team to start the offering in the marketplace. The Market Opportunity and Approach, Define and Prove, Build and Deliver, and Sense and Respond stages to understand how offering managers lead their teams and deliver their business results are shown in Figure 1.


Offering Management discipline

Strategic planning and portfolio management
Market opportunity and approach
Define and prove
Build and deliver
Sense and respond
Continuous Delivery and Evaluation
Decision Checkpoints
Artifact

Launch Readiness Playback
✓ Offering readiness dashboard
✓ GTM readiness dashboard

Lead cross-functional execution against business plan to **create offering and ensure go to market and delivery readiness**

Launch Readiness Playback
Decide release **readiness of offering and GTM & Delivery capabilities**

Figure 1 Stages of Offering Management process

Security is considered throughout these stages. The following considerations also are included:

- Will the offering handle sensitive private information?
- How is the product updated?
- Who will have access to the product?

The Offering Manager works with a cross functional team that attempts to address these issues before and after product release.

Checkpoints before availability of the offering help ensure that appropriate testing was completed and that the quality of the offering is acceptable.

Governance of a common development process

In support of the development process, organizations should establish governance in the form of standards, practices, and compliance criteria. Governance for development of hardware, software, and services includes the following important elements:

- Protection of assets
- Product development check points
- Security and quality plans
- Product testing

These elements are described next.

Protection of assets

Because development projects are never isolated from the rest of an organization, it is important to ensure that governance is not limited in scope to development projects.

The IBM Business Conduct Guidelines define proprietary information to include software in object or source code format. Personnel complete an annual mandatory training about these Business Conduct Guidelines, which covers Intellectual Property Protection, Corporate Security Standards, and Export Regulations.

It is a condition of employment that every IBM employee must demonstrate an understanding of, and commit to compliance with, the directives, standards, processes, and practices that are related to their roles in IBM. Protection of proprietary information is one of those responsibilities.

Product development checkpoints

As shown in the lower portion of Figure 1 on page 3, it is recommended that projects within the common development process be separated into phases, such as:

- Strategic planning and portfolio management
- Market opportunity research
- Definition of the problem
- Building and delivery
- Maintenance and product life planning stage

This structure provides an opportunity for the project development team to conduct development checkpoint reviews as the project transitions from one phase to the next. These checkpoints can be used as control points for assessing project risk, expense control, product quality, issue review, product issue remediation, and for project plan synchronization.

For the project to move to the next phase, the project development team should be required to satisfy the success criteria for the prior work, and justify any deviations from the plan, such as a change in scope or content. The projects should be required to address open issues before proceeding to the next project phase.

Although these steps demonstrate a waterfall approach to development, many teams in IBM operate by using a continuous delivery model. They still make use of incorporating these steps into the process to ensure a secure product.

Security and quality plans

Every development project within an organization should require a security plan and a quality engineering plan. These plans detail the technical and audit requirements for asset control, along with the standards and practices for secure engineering practices to be applied in the development process. Each deliverable should describe threat modeling plans, static and dynamic scanning plans, and plans for the ability to maintain the product throughout its life.

Of key importance is the ability to service the product, particularly if components of the product are sourced from a third party. The Offering manager and Project Management teams must outline a service and product lifecycle that aligns with all components. For example, if a new product Foo contains a third-party component Blah, Foo’s service lifecycle cannot be longer than the third party is willing to service Blah.

A key element in the security plan is protection of proprietary information. A team within the organization should be responsible for setting appropriate data classifications and for overseeing the protection of the organization’s proprietary information assets within the development process.

Organizations should require that all product development projects prepare a quality plan that describes how the project meets corporate or external standards. Before availability to customers, a review of the product’s quality results relative to the plan is performed to validate how the project met these standards.

As a specialization of quality engineering, organizations should maintain a common set of practices for secure software engineering. This secure engineering program should establish a measurement system of continuous security improvement as a fundamental part of a secure product development strategy. The secure engineering program should be run in the following parallel and intertwined pillars:

- A mandate for continuous security improvement in technology and manufacturing drives accountability and action.
- A community of software engineers that innovates and share practices and tools for secure product development.
- Integration across products that is achieved through client use cases, scenarios, and end-to-end usage threads in concert with an architectural framework that enables componentization.
- Consumability analysis that looks beyond product defects to the client experience of using and maintaining the offerings.

All of the security and quality plans, practices, and findings should be reviewed in the development phase checkpoint meetings. If agile design practices are used, they should be reviewed at the end of each sprint.

Product testing

All products should undergo a range of tests to verify functional operation in accordance with the official design specifications of component, product, or solution. This process includes verification of the security mechanisms and services incorporated into a component, product, or solution.

IBM development teams perform several levels of testing during development projects, including the following tests:

- Unit Test verifies that a software element, subroutine, or class performs as designed in isolation.
- Component or Function Verification Test verifies that a composite software element operates in accordance with written specifications.
- System Verification Test verifies the integration and operation of components and products within the full solution environment.

Security testing is performed during the Component Verification Test and System Verification Test. Security testing might include automated testing by using tools, such as IBM Rational® Software Analyzer and IBM Rational AppScan® and security testing by using ethical hacking techniques. Use of these tools and the results they produce are considered IBM Confidential information and are not disclosed to ensure that those results cannot be used to attack the eventual product in the field.

Where appropriate, products might undergo outside analysis and testing, including certification as specified by the Common Criteria.

The project development teams should review the results of unit testing, component testing, system testing, security testing, and certification testing during the phase checkpoint meetings. Defects should be returned to the change team within development for rework and subsequent retest and verification by test teams. The project development team leader should hold the authority to prevent the project from progressing from the Development Phase to the Availability Phase until the test exit criteria are met.

Product Lifecycle Management

After a product is made available, attention to security should continue in the product support channel. An organization should establish internal processes that allow for the notification of clients regarding high pervasive (HIPER) fixes that are recommended to be applied.

Organizations should also have internal mechanisms to help ensure that managers of potentially affected products are notified quickly of security issues that might arise. The implementation of a Product Incident Response Team in collaboration with product managers should coordinate the opening of incident records to track responses and fixes to external sources. These external sources help provide guidance for customers, analysts, journalists, and industry experts to understand how best to react to any vulnerabilities that are discovered. Responses can be anywhere from “do nothing” because a product is not affected, to “apply fixes” when a code stream change is required to correct the issue.

Product Lifecycle Management should also consider the components that are used in the product and the maintainability of the product given those components and their lifecycle.

Development process summary

A development process defines the overall steps for developing software and solutions that are delivered to clients of an Information Technology organization. It can provide the structure for conducting development projects, and facilitate compliance with corporate standards and practices. Deploying a development process fosters consistent management, technical oversight, and accountability across a wide range of hardware, software, services, and solution development projects.

The approach to secure development of IBM's products is always being re-evaluated so as to use state of the art techniques and monitoring mechanisms. With many product offerings moving to Software as a Service (SaaS) offerings, how we approach secure development also began to change. Traditional on-premise solutions work with in the Secure Engineering Framework and SaaS offerings uses the Security and Privacy by Design (SPbD) approach (for more information, see "Security and Privacy by Design" on page 14). Both approaches achieve the same goals and use many of the same techniques. These approaches are described next.

Secure Engineering Framework

In addition to a common development process, organizations should give particular attention to the security characteristics of the offerings they create. Organizations might find it useful to create and maintain a set of recommended guidelines and best practices to guide development teams in building more secure software. Each product development organization must apply these guidelines and best practices as they see fit to ensure that the software is developed in a secure manner.

IBM development teams use the practices of Secure Engineering in Development, which aligns with SEF in combination with Secure Engineering by Design. This practice also takes into account the correct handling of any data that is processed by SaaS offerings.

The Secure Engineering Framework (SEF) is intended to provide guidance that helps ensure that software is secure by design, in implementation, and in deployment. The global nature of software development activities today necessitates the application of secure engineering principles across global development teams, regardless of their physical location.

The SEF that we describe in this publication includes the following sections:

- "Education and awareness"
- "Project planning" on page 9
- "Risk assessment and threat modeling" on page 9
- "Security requirements" on page 10
- "Secure coding" on page 11
- "Test and vulnerability assessment" on page 11
- "Documentation" on page 13
- "Incident response" on page 13

We also include a section on securing code repositories and build systems. The following material was selected from the framework as an example.

Education and awareness

Unfortunately, examining the current state of the technology industry reveals that many security exposures occur because development organizations are unaware of the root causes of these security vulnerabilities and their effect. Development teams frequently work under tight deadlines and are under pressure to squeeze in as many product features as possible. When the importance of software security is not understood, it can become an afterthought or viewed as an impediment to productivity.

Offering managers

Offering managers are responsible for building market focused sustainable businesses. They research the market to understand the size, opportunities, competitors, user needs, and specific problems that need solutions. By using that information, they create a minimum delightful experience (MDE) to resolve the problem and test it with sponsor users until they create a market "winner".

The focus is to win in the market by providing the best user experience to the clients in the shortest amount of time. Offering managers are responsible to ensure that each offering was evaluated against the Secure Engineering practices and Security and Privacy by Design, whichever is appropriate.

Offering managers should think like business owners who are responsible for determining offering revenue models and held accountable for delivering business results. The offerings must integrate all the IBM capabilities (Services, Hardware, and Software) and consider the ecosystems, go to market (including routes to market), and enablement requirements. This holistic view of the offering must begin at the earliest stages of design and development to seamlessly deliver an exceptional offering experience. IBM is a large institution, so the offering manager must lead the collaboration of cross functional teams to set the detailed direction, resource envelope, and priorities across disciplines.

Development managers

Development managers should understand the effect of security issues on the users of their applications. They must have general knowledge of the technical nature of these issues. In addition, they should learn about secure development frameworks and methodologies, such as threat modeling and risk assessment, and learn how to implement these frameworks and methodologies during the different phases of their product'[s software development process.

System and software architects

System and software architects benefit from training on the technical nature of security issues and training on secure coding principles and techniques. They should learn the security features and issues that are related to their development platform so that they can design solutions that meet the security requirements.

System and software architects should understand threat modeling and risk assessment techniques so that they can be applied to their products. They should articulate the potential threats to their designs.

Developers

Developers are responsible for writing code (including microcode in systems) that is free of security vulnerabilities, viruses, malicious code, back doors or trap doors, and other potential weaknesses. Developers are also responsible to help ensure that configuration and integration of components within a larger product or offering does not introduce or facilitate security vulnerabilities. This process requires that they understand the coding mistakes that lead to security issues and the principles for secure coding, and being able to test their own code.

Quality assurance personnel

Testing specialists must understand the security issues of which they must be aware. They can also benefit from training on application security testing techniques and methodologies and training on different security testing tools.

Project planning

Each development group should start planning for security from the beginning of the Offering concept phase. Doing so allows for the OM to evaluate the needs for Secure Engineering best practices and any certifications and standards adherence. It also avoids expensive rework as a result of security vulnerabilities that are discovered late in the development cycle.

It is generally accepted that fixing defects earlier in any development cycle is more cost-effective than finding and fixing defects later in the development cycle. This idea is also true for security-related defects. Further, avoiding the loss of confidence by customers is an added incentive to find and fix security-related issues during the development of products.

Further, security analysis and testing should be integrated into each major phase of the product's development cycle, regardless of the methodology that is used.

Regardless of the software development process that is followed, be it waterfall-based, iterative, or agile, every product development team spends time evaluating requirements, designing, coding, testing, and maintaining. The time and scope might vary greatly between the methods that are used, but the basic phases still exist with associated security practices.

During project planning, the development team should account for security analysis, requirements, design, testing, and documentation work. A checklist of basic items in a development plan includes the following items:

- Are the right people, with the right skills, on or available to the development team to perform the security work?
- Was a security risk assessment and architectural review performed?
- Are new security features needed or is it necessary to modify existing features?
- Is a test plan in place and are tools available to perform security testing?
- Did the development team gather the latest information about security threats and vulnerabilities in the technology and the target operating environments for the component, product, or solution?
- Was adequate time factored into the schedule for security testing and fixing any security vulnerability issues found?
- Is a documentation plan available that includes sections that are related to security and securing the offering?

Organizations should be sure to require that applications with higher risk of exposure make an increased investment in creating a security plan for the project.

Risk assessment and threat modeling

Threat modeling is also a critical part of the SEF and SPbD. Threat modeling allows the development groups to identify potential risks or attacks against an application even before it is built and to make decisions about how to address these risks.

After threats are identified, they are ranked in importance and addressed according to a risk profile. Some threats should be addressed in the internal design of the component, product, or solution. However, some threats can be addressed by proper configuration and integration, or might require more components or management processes to adequately control risks. In many, if not all cases, residual risk might exist in deploying and operating the components, products, and solutions.

Although it is not in the scope of this publication to specify or document threat modeling, the following general flow is used:

1. Identify the assets.
2. Identify the potential threats.
3. Assign an impact for each threat.
4. Determine the probability of compromise.
5. Rank the risks.
6. Define mitigating counter-measures as needed.

Although threat modeling is often documented as a point-in-time step of the design process, incremental value is obtained if it is treated as a continuous process within the development cycle. The development team might want to revisit the risk assessment and threat model for each new release of software, or when new risks and threats are discovered.

Security requirements

As with functional requirements and performance requirements, security requirements are needed to help ensure that security is built into the application from the start of product inception. During the Offering management initial offering stages, the Offering manager completes the Secure Engineering checklist, which is reviewed by each IBM Brand security review board to explain how each of the SEF framework items can be met. Any deviations are approved by the systems Business Information Security Officer.

Security requirements define what new security features are required and how existing features should be changed to include necessary security properties. The objective of security requirements is to help ensure that the application can defend itself from attack.

The SEF and SPbD suggests the following categories for security requirements and provides examples for each category:

- Auditing and logging
- Authentication and authorization
- Session management
- Input validation and output encoding
- Exception management
- Cryptography and integrity
- Data at rest
- Data in motion
- Configuration management

Together, these categories formulate the end-to-end security architecture for the product and therefore should be considered alongside one another (not in isolation). Also, each of the categories has many subtopics within it. For example, under authentication and authorization, several aspects of discretionary access controls and mandatory access controls must be considered. Security policies for the product are an outcome of the implementation decisions that are made during development across these nine categories.

Many security requirements are generic across many types of applications, including embedded systems, thick client software, and web-enabled applications. It is important that the security requirements meet the business requirements of the software.

Secure coding

Most application security vulnerabilities often are caused by one of the following problems:

- The requirements and design failed to include proper security.
- During implementation, vulnerabilities were inadvertently or purposefully introduced in the code.
- During deployment, a configuration setting did not match the requirements of the product on the deployment environment (for example, unencrypted communication allowed over the internet).

Attention to secure coding can prevent vulnerabilities being added during implementation. Secure coding guidelines are usually provided in a separate document that is specific to the development team's environment and chosen source code languages. Detailed information about several topics should be available to developers, including the following examples:

- Data validation
- Output encoding
- Handling of sensitive information
- Avoiding invention of encryption and decryption algorithms
- Exception handling
- Source language-specific development tips

Use of automated security analysis tools is recommended, as is the use of proven certified security components. These tools and components allow developers to perform an analysis of known security issues while emphasizing the use of secure and proven code components.

Careful attention to current trends and requirements, such as correct usage of cryptographic algorithms, must be considered. Many of these requirements can be found in the NIST set of documents. Although numerous, developers and offering managers must be aware of these requirements so that products can be secure when announced and maintain the needed level of security through out their life time.

Test and vulnerability assessment

Testing applications for security defects should be an integral and organic part of any software testing process. During security testing, organizations should test to help ensure that the security requirements were implemented and the product is free of known vulnerabilities.

The SEF and SPbD refers to the MITRE Common Weakness Enumeration (CWE) list and the Common Vulnerability Enumeration (CVE) list for the specific vulnerabilities for which products should be tested. This approach helps ensure that the SEF and SPbD do not get stale with old vulnerability information and allows product development teams to reference a current list of weaknesses and vulnerabilities. This approach, in turn, can help ensure that a relevant security assessment is performed against the most current set of known vulnerabilities.

Creating a security test plan is a critical part of test and vulnerability assessment. This test plan includes the documentation and analysis of the following characteristics of the application:

- Entry points
- Output locations
- Deployment environment

- Product functions and business logic
- Application users, roles, and permissions

The SEF and SPbD recommend performing security analysis by using automation tools before Offering Management decision checkpoints that use the most current test cases and knowledge about threats and vulnerabilities.

The tools that are described next should be used to perform automated analysis of source code, object code binaries, dynamic analysis, and runtime analysis.

In addition to internal tracking of vulnerabilities, IBM participates in the ICASI consortium to exchange early information about non-vendor specific vulnerabilities. This early exchange allows IBM to address these vulnerabilities in a more timely manner.

Source code security analyzers

These tools (for example, IBM Rational Software Analyzer and IBM Rational AppScan Source Edition) can analyze application source code to locate vulnerabilities and poor coding practices. These tools can also trace user input through the application (such as code flow analysis and taint propagation) to uncover various injection-based attacks.

Bytecode security analyzers

These tools can analyze application byte code (relevant for certain languages only) for the same vulnerabilities as described in “Source code security analyzers”. In some scenarios, source code is not available to the tester, and bytecode can be used for the analysis.

Binary security analyzers

Binary analysis is similar to source code analysis. However, instead of evaluating the source code, this analysis examines the application binary. When applications are compiled, the source code is interpreted by the compiler and depends on the environmental components that support it. This dependency on environmental factors can lead to contextual risks for some software deployments.

Dynamic analysis tools

These tools perform analysis of the application as a black box, without knowing its internal operation and source code. Dynamic analysis tools automatically map the application, its entry points and exit points, and attempts to inject input, which breaks the application or subverts its logic. IBM Rational AppScan Developer Edition is this type of tool.

Runtime analysis tools

Strictly speaking, Runtime analysis is not a specific security analysis technique or tool. Runtime analysis is the software development practice that is targeted at understanding software behavior during runtime-including system monitoring, memory profiling, performance profiling, thread debugging, and code coverage analysis.

Runtime analysis is almost always deployed with another type of automated security analysis. For example, you might run a dynamic analysis tool against a web-based application and monitor both the system resources for disk read/write and the application source code for code coverage. Viewing the application from the dynamic analysis perspective (black box) and runtime analysis perspective (white box) during this process can lead to a greater understanding of the potential security issues that might exist; for example, stress testing and denial-of-service (DoS) attacks.

Content analysis tools

Content analysis tools (for example, BlackDuck and OPENScap) review the bill of materials list in a software offering. It determines the makeup of the software and the overall vulnerability profile by comparing the offering against a database with components that are known to have vulnerabilities.

These databases are known as SCAP databases. A good example of this database is the National Vulnerability Database that is maintained by NIST, which is partially based on data that is kept by The Mitre Corporation.

One problem with depending on tools such as these is that the tools often do not consider the usage of the packages and other hardening within the system to limit exposure to vulnerabilities. Therefore, they can be noisy with many false positives. Even so, they can be a valuable tool that can be helpful in keeping your software up-to-date and getting an initial look at your security posture.

Documentation

The SEF and SPbD not only document why security documentation is essential, they provide guidance on how security documentation of the product should be structured.

Development projects within an organization should follow an Information Development Plan that outlines the required documentation for the individuals that are involved in the various roles of installing, configuring, operating, and managing the product or solution.

The SEF and SPbD extend that requirement to consider security-related roles that are associated with the component or product. The security role definitions should include security architect, enterprise architect, system integrator, system auditor, and product assurance evaluator.

Further, the SEF and SPbD recommend that Information Development Plans include considerations for security in the Integration, Deployment, Operations, and Management section of the documentation so that security remains visible and relevant over the range of expected deployment life cycles and roles. Security-related guidance in product documentation should include information about security-related settings for the underlying environment within which the software or solution runs.

Incident response

Product teams follow a defined process for handling security-related incident reports. This process is put in place to help ensure that after an incident is discovered and validated, any other product teams in the organization that might be affected by the vulnerability are informed of the situation so they can begin working on a fix, if necessary.

With complex combinations of component reuse and solution construction, such processes are necessary to ensure that potentially affected products are identified quickly.

Security and Privacy by Design

SPbD at IBM is a simplified and agile set of focused security and privacy practices, including threat models, privacy assessments, security testing, and vulnerability management. SPbD@IBM reinforces our commitment to security and ensures products and offerings have security and privacy embedded into the design. IBM's Secure to the Core campaign embodies SPbD@IBM.

Benefits

Customers expect that IBM products and offerings are built with security and privacy in mind. IBM products that are free from serious security vulnerabilities reduces a customer's risk and total cost of ownership.

By developing products and offerings with security and privacy in mind, IBM saves valuable time and effort retrofitting existing products. IBM's brand reputation is strengthened by staying out of headlines for security and privacy-related incidents.

SPbD@IBM

SPbD@IBM is a *mandatory* practice for all IBM SaaS products and offerings. IBM developed a set of SPbD processes and tools that are required across all business units. SPbD@IBM is flexible to enable each business unit to emphasize elements of security and privacy design to best address the demands of their clients: our customers.

As shown in Figure 2 on page 15, the SPbD assessment process includes the following tasks:

- Threat Model identifies, communicates, and understands threats and mitigations within the context of protecting something of value.
- Privacy Assessment is the process to evaluate new projects, policies, and practices for privacy, confidentiality, or security risks associated with the collection, processing, or disclosure of personal information. It also includes develop measures that are intended to mitigate and eliminate identified risks. In particular, this assessment process must meet GDPR requirements.
- Code Scan helps programmers locate potential flaws and determine areas of improvement within the codebase. Code scans must be performed during development and test, cover IBM developed code, and include Open Source Software.
- Security Tests are a key component of the overall test cycle and are intended to ensure that the development process resulted in secure code and that, where possible, threats identified as part of the threat model were properly addressed. Security testing helps validate that the information system in question protects data and functions as intended.
- Penetration Test (also called pen testing) is an authorized simulated attack on a computer system, application, or IT environment. It can involve automated tools and must involve a form of ethical hacking.
- Vulnerability Management is the process of searching for software vulnerabilities in applications by using an automated security program. Vulnerability scanning can be used to find holes and plug them before they are exploited or to find holes and exploit them.
- Secure Release Process is a two-step process by which the local Business Unit evaluates the Secure Release Readiness criteria for a specific offering, followed by the confirmation of the readiness criteria by Enterprise and Technology Security team.

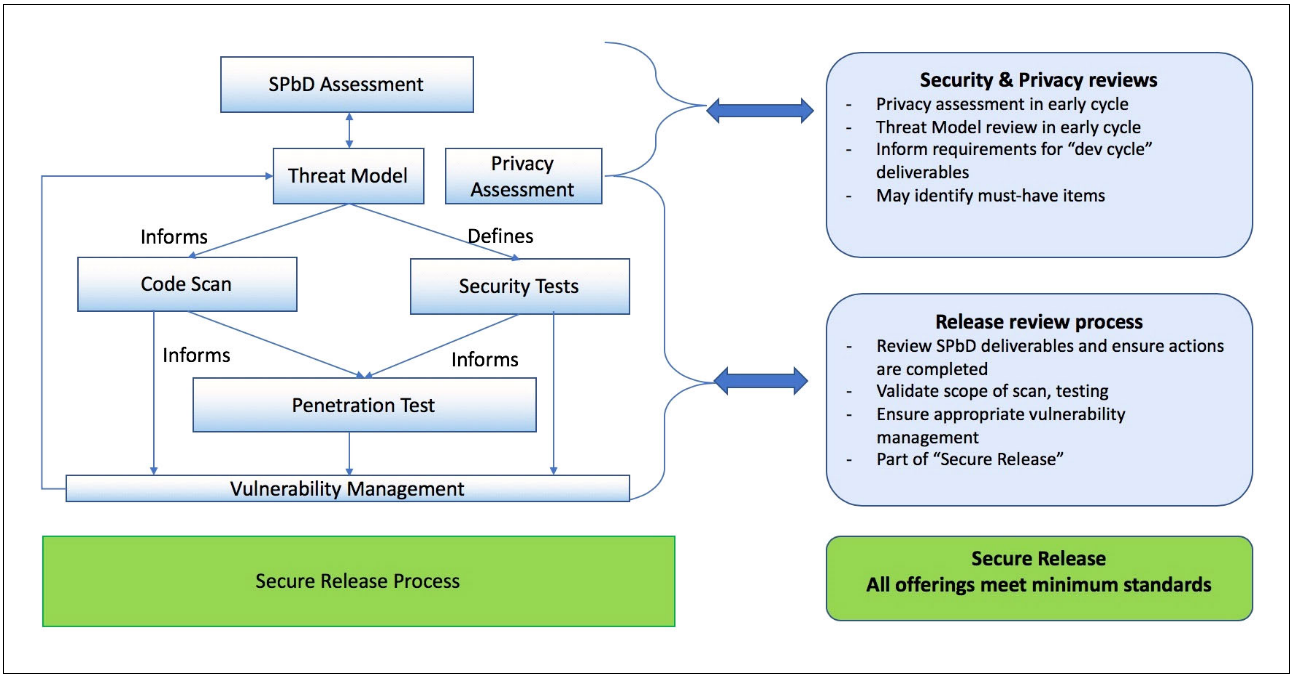

SPbD Assessment
Threat Model
Privacy Assessment
Informs
Defines
Code Scan
Security Tests
Informs
Informs
Penetration Test
Vulnerability Management
Secure Release Process
Security & Privacy reviews
- Privacy assessment in early cycle
- Threat Model review in early cycle
- Inform requirements for "dev cycle" deliverables
- May identify must-have items
Release review process
- Review SPbD deliverables and ensure actions are completed
- Validate scope of scan, testing
- Ensure appropriate vulnerability management
- Part of "Secure Release"
Secure Release
All offerings meet minimum standards

Figure 2 SPbD assessment process

IBM Product Security Incident Response Team overview

The IBM Product Security Incident Response Team (PSIRT) is a global team that manages the receipt, investigation, and internal coordination of security vulnerability information that is related to IBM offerings. IBM PSIRT is a focal point for security researchers, industry groups, government organizations, and vendors to report potential IBM product security vulnerabilities.

This team coordinates with IBM product and solutions teams to investigate, and if needed, identify the appropriate response plan. Customers of IBM offerings should continue to report all product-related issues, including potential security vulnerabilities, to IBM Technical Support. Maintaining communication between all involved parties, both internal and external, is a key component of our vulnerability response process.

IBM Product Security Incident Response Team process

When IBM PSIRT receives a report of a potential vulnerability from a third party, IBM PSIRT logs the issue with the supporting details and provides the tracking number to the vulnerability reporter. IBM PSIRT notifies the appropriate IBM product teams of the potential vulnerability for analysis.

The appropriate product team attempts to reproduce the issue to verify whether it is a vulnerability.

After the initial analysis, the vulnerability undergoes further investigation by the product team to determine the underlying cause and possible methods of use. The team completes the remediation plan for the vulnerability, taking into consideration the affected versions.

In some cases, IBM PSIRT might request more information from the vulnerability reporter to understand the environment in which the vulnerability appears, ways to reproduce the issue, potential use methods, and so on.

After the remediation is available, IBM intends to notify the affected customers about the vulnerability by using targeted communications or issuing a public Security Bulletin. When IBM discloses the vulnerability publicly, the Bulletin includes more information, such as the Common Vulnerability Scoring System (CVSS) Base score and vector, a reference to the assigned Common Vulnerabilities and Exposures (CVE) identifier, remediation for the affected offerings and other relevant links that might cover more information.

The last stage in IBM PSIRT process allows for IBM PSIRT to share findings with our Engineering teams to help minimize similar vulnerabilities in future IBM offerings.

PSIRT response times

IBM’s response times to vulnerabilities vary because of many factors. IBM is committed to ensuring that incidents are responded to in the shortest amount of time while maintaining the highest level of quality for all of this products. IBM also makes every technically reasonable effort to correct the issue. For more information about our PSIRT process, see the IBM Security in Development page of the IBM Security website.

Infrastructure compliance

IBM System’s IT/OT Security Team provides overall IT/OT Security Management System for the division that it supports. The Security Management System (SMS) is an end-to-end security infrastructure that consists of tools, policies, reporting, oversight, and defined management structure. The SMS enables development, support, manufacturing, and operations teams to manage security risk and regulatory compliance at the functional level.

IBM Systems IT/OT Security works in partnership with the various information resources, departments, Internal Audit, Compliance, and Information Technology resources from the CIO/CISO and other divisions to support the corporate mission and goals worldwide.

As part of the overall mission, the IBM Systems Unit Security team provides the following core services:

- Information Risk Management

  IBM Systems IT Security maintains an information security risk management program to evaluate threats and vulnerabilities and assure creation of appropriate remediation plans.

- Information Security Policy and Procedures

  IBM Systems IT Security provides direction for security policies and practices to protect critical resources and services. IBM Systems IT Security also creates division-level security policies, processes, and procedures for approval by executive leadership.

- Information Security Monitoring

  IBM Systems IT Security Management System (SMS) monitors compliance to manage risk to the business unit. In cooperation with the IBM Security threat management teams, IBM Systems IT Security responds to all reports and requests for information.

- Consulting/Advisory

  IBM Systems IT Security provides guidance and assistance for IT security compliance and risk management.

- Security Education, Training, and Awareness (SETA)

  IBM Systems IT Security provides a security awareness program that encompasses policies and procedures, risk avoidance, best practices, and incident response procedures.

Continuous security improvement

Security is a moving target. The landscape of risks, threats, and vulnerabilities is continuously changing.

Throughout the Secure Engineering Framework and Security and Privacy by Design sections, we described the rationale for recommending that product development teams employ a structured development process and a Secure Engineering Framework. In addition, teams should recognize that all of the elements of the Secure Engineering Framework might not be sufficient for improving security on a continuous basis for existing and new projects.

Often, project teams are implementing incremental features and components to large, complex solutions that are in place. To address the complexity of this environment, the third pillar of secure product development is continuous security improvement (CSI).

Product teams should develop components, products, services, and solutions that are as free of security vulnerabilities as possible. They also should strive to continually improve the security characteristics of these offerings.

This requirement defines a set of key performance indicators (that is, measurements and metrics) that are related to security characteristics of offerings. These indicators measure security characteristics and performance of the offering team in achieving their goals for security throughout the offering lifecycle.

Key performance objectives (that is, goals for these measurements and metrics) are set by the offering team and are based on accepted security practices within the organization and the information technology industry, along with goals that demonstrate that the security characteristics of the offering will improve from release to release.

The metrics and progress in achieving the goals are to be reviewed throughout the lifecycle of an offering. The following reviews are included:

- Establishing security quality and risk acceptance goals early in the offering planning process
- Project decision checkpoints in the development process
- Service quality reviews during the post-availability lifecycle
- Selective management reviews
- Offering performance reviews
- Secure release criteria reviews by Security Review Board

Key performance indicators

A continuous improvement framework for security should establish a set of indicators that represents tangible movement from a starting point to a wanted state.

These key performance indicators (KPIs) should capture the results of actions that represent security-oriented goals and achievements throughout the solution lifecycle.

Generally, the KPIs for continuous security improvement fall into the following categories:

- Actions that are taken and results that are achieved in the development process to be reviewed at time of availability.
- Actions that are taken and results that are achieved in the post-release support and maintenance process, including serviceability performance and number of security defects.

In some cases, the continuous security improvement KPIs might be intuitive; for example, the number of security defects for a release of software, or the time to diagnose and resolve a reported security defect. In other cases, the KPIs might be more subjective; for example, the number of security test cases that are run and passed, or the percentage of source code that underwent visual review.

The tables in the following sections highlight some examples of quantitative and qualitative key performance indicators that can be included in a continuous security improvement program.

Development process KPIs

The key performance indicators that are associated with the development process (to be reviewed before offering availability) are categorized by quality, resilience, and integrity.

The pre-release security quality KPIs (see Table 1) are structured to promote best practices for security in development and the use of the best security functionality that is available in the offering operating environment.

Table 1 Security quality KPIs

Security quality KPI	Metric	Improvement trend
Proof that secure engineering practices were followed in development project.	List and work reports	Increasing
Comparison of offering integrity indicators with other offerings in hardware and software operating environment.	Higher/consistent/lower	Increasing
Comparison of offering resilience indicators in comparison to other offerings in hardware and software operating environment.	Higher/consistent/lower	Increasing
Use of the security features of the hardware and underlying software.	Use of best practice/rationale	Increasing

The pre-release security integrity KPIs (see Table 2) are structured to promote offering assurance and integrity.

Table 2 Security integrity KPIs

Security integrity KPI	Metric	Improvement trend
Amount of developed components included in code review.	0% - 100%	Increasing
Amount of external components included in code review.	0% - 100%	Increasing
Amount of offering tested for known vulnerabilities.	0% - 100%	Increasing
Amount of offering with signed code and distribution packages.	0% - 100%	Increasing
Documentation for security features and standards.	0% - 100%	Increasing
Documentation for completed assurance review / regression tests.	0% - 100%	Increasing

The pre-release security resilience KPIs (see Table 3) are structured to promote offerings that can be configured for resilient operation as they are deployed.

Table 3 Security resilience KPIs

Security resilience KPI	Metric	Improvement trend
Completed design documentation for resilient operation.	0% - 100%	Increasing
Completed deployment documentation for resilient operation.	0% - 100%	Increasing
Completed resilience testing (ethical hacking / penetration testing).	0% - 100%	Increasing

Support process KPIs

The KPIs that are associated with post-release support and maintenance track the security quality of the offering in operational environments.

The post-release security quality KPIs (see Table 4) are structured to measure and track the number, type, and severity of security-related defects.

Table 4 Post-release security quality KPIs

Post-release security quality KPIs	Metric	Improvement trend
Time to resolution for post-availability security problems (including CVEs).	Number in hours/ days by incident	Decreasing

Post-release security quality KPIs	Metric	Improvement trend
Percentage of fixes and changes that have undergone and passed code assurance review / regression tests.	0% - 100%	Increasing
Percentage of fixes and changes that have undergone and passed vulnerability tests.	0% - 100%	Increasing
Frequency of post-availability CVE reviews and retests.	Time between reviews	Decreasing

The post-release security serviceability KPIs (see Table 5) are structured to measure and track the time to identify and resolve security-related defects.

Table 5 Post-release security serviceability KPIs

Post-release security serviceability KPIs	Metric	Improvement trend
Number of post-availability security problems reported.	Number/Severity	Decreasing per release found
Number of post-availability CVEs published.		

Offering-specific key performance objectives

Development teams should use the key performance indicators as guidance to set, track, and report on the actions and results related to security of the products and offerings.

At the start of each product delivery cycle, teams should set goals for their development and post-release serviceability KPIs. These goals or objectives are called *key performance objectives* (KPOs). The actions and results of the team are evaluated against the KPOs at various stages and milestones of the release lifecycle. New offerings are expected to set initial KPOs that are consistent with the best practices in their development area.

The development-oriented key performance objectives include:

- Adoption of secure engineering development practices
- Code review coverage
- Extent of vulnerability analysis
- Depth of testing
- Defect remediation
- Documentation of security-related information
- Adoption of ecosystem security features
- Assurance testing
- Review of development practices and risk review of any anomalies

The serviceability-oriented key performance objectives include:

- Frequency of post-availability review of published vulnerabilities and exposures
- Fix code review
- Fix vulnerability testing
- Security regression testing
- Time to remediate and resolve security-related defects

Continuous security improvement summary

The continuous security improvement process is intended to help ensure that the security characteristics of product offerings improve over time and that security characteristics of new product offerings reflect best practices.

An assessment of the actions and the achievements of development teams regarding the security of their offerings should be tracked and evaluated during and after offering availability. The review of the serviceability measurements of an offering helps to validate that attention was given to security in the development process. By having to meet goals that increase in each release, teams are compelled to continually improve their attention to security.

Supply chain security

IBM maintains one of the world's most recognized global supply chain management systems. IBM received numerous awards for innovative supply chain management practices. The IBM Systems Supply Chain is regularly recognized in multiple categories of the prestigious ML 100 Awards and Manufacturing Leaders Awards. The Systems Supply Chain also received recognition from other organizations, such as Supply Chain Leaders in Action, and numerous country-based organizations.

IBM supply chain practices focus on effective management of product design, manufacturing, transportation, fulfillment, import and export, intellectual property management, and customer support. IBM leads the global focus on supply chain security and is a founding member of the Responsible Business Alliance (RBA), formerly the Electronic Industry Citizen Coalition (EICC). The IBM supply chain processes and policies are fully integrated with the standard product development and manufacturing process.

Supplier assurance

Before IBM conducts business with any external supplier, IBM Global Procurement has the responsibility to evaluate and assess the supplier to verify that they meet procurement criteria for qualified suppliers. These criteria include financial solvency, compliance with IBM technology and technical standards, and the ability to meet IBM's requirements.

The criteria for ensuring our suppliers are meeting secure engineering and ethical best practices includes the following assessment:

- Ensure that the supplier is not on an Unapproved or Denied Parties List.
- Supplier must commit to the Responsible Business Alliance (RBA), formerly the Electronic Industry Citizen Coalition (EICC), Code of Conduct.
- Suppliers that provide hazardous waste, special waste, and end of life product disposal services must be in compliance with IBM Corporate instructions.
- Supplier must adhere to the Business Continuity clause requirements to have an enterprise-wide cybersecurity program and sign the IBM Security Letter Agreement (SECLA).
- Suppliers are required to submit to periodic assessments by responding to the Supplier IT Security Assessment.
- A supplier must submit to remediation actions if found to be out of compliance before being reinstated as an approved supplier.

An important element of the supplier assessment process is the Supplier IT Security Assessment. The intent of the Supplier IT Security Assessment is to identify all components that make up the overall supplier risk-offering, process, and business risks. Risk characteristics are identified to help assess the risk severity level. Mitigation strategies are also addressed as part of the assessment process.

Supplier conduct principles

Suppliers are required to adhere to the Responsible Business Alliance (RBA), formerly the Electronic Industry Citizen Coalition (EICC), Code of Conduct. For more information, see the Responsible Business Alliance website.

Effective March 2013, IBM began the use of the EICC, now the RBA, Code of Conduct as the single code with our supply base. The RBA Code supersedes the IBM Supplier Conduct Principles, which was used 2004 - 2013. The RBA Code establishes for our suppliers the minimum social responsibility standards we expect from them as a condition of doing business with IBM.

Our goal is to work with our suppliers to foster full compliance as they, in turn, apply these standards to their extended sources of supply that is engaged in the production of goods and services for IBM. We consider these standards and adherence to them in our selection process and seek ongoing compliance by actively monitoring performance. IBM reserves the right to take action with suppliers that do not comply with the RBA Code and can consider measures, such as reducing or ending business, in accordance with contract terms.

Supply chain security policy enforcement

Suppliers who are directly or indirectly involved with tangible goods shipments to or on behalf of IBM are required to adhere to the requirements of the IBM Business Continuity Clause. The IBM Security Letter Agreement (SECLA) is used to demonstrate commitment to the IBM Supply Chain Security Principles by suppliers that seek a relationship with IBM. These principles include:

- OEM products that carry an IBM logo or are sold by IBM to be used in an IBM system (such as OEM feature cards and adapters) must also meet the same security, export, blue book policies, or compliancies that an internally built or “build-to-print” IBM system must meet. System integration and other testing are performed in the development cycle to ensure proper function.
- Electronic components on an OEM subsystem are covered for quality and performance requirements through the statement of work (SOW), contracts, and other OEM specifications.
- It is the responsibility of an OEM provider to ensure the robustness, stability, performance, and ultimately the execution-time security of the software or firmware they deliver to IBM.
- Access to software or firmware development libraries (including firmware source code and documentation) is controlled by access control lists. Suppliers must be authorized by an IBM manager and must have a need to know before gaining access.

- Any firmware or software that is written for use by IBM is required to have a Certificate of Originality on file for every piece of open source or non-open source code that is picked up and incorporated into one of our deliverables. This requirement is part of the release process for all firmware and software that is developed by IBM. Certificates of Originality are approved by Legal and evaluated against stringent IBM criteria.

Open source

All IBM developers are required to participate in open source training to produce code that is compliant with open source development guidelines. All firmware and software that is produced (internally and by suppliers) is run through an automated tool that identifies potential violations of this policy and each entry is reviewed by project managers and lawyers. This requirement ensures that code contributions or adoption were made by uncontaminated developers and that code is free from potential intellectual property rights violations or inclusion of malicious components.

Employee and contractor assurance

IBM conducts a thorough background check of all suppliers and contractors and requires the same on behalf of approved suppliers. Before placement of supplier personnel at a customer site under a work authorization, for every person (including persons who are not US citizens, green card holders, or permanent US residents), to the fullest extent permitted by applicable law, the supplier is required to perform or have performed a criminal background check covering the counties in which the person was employed or resided for the past five years (or longer as required by applicable law).

Suppliers are not permitted to propose persons who have had a serious criminal conviction (felony) or have been found guilty of an offense involving violence or dishonesty. A supplier relationship cannot be established with IBM unless the proper background checks have been completed according to IBM's GEVS (Global Employment Verification Standard) standard and are on file with the supplier. GEVS also outlines a set of checks that are performed for employees that are brought into IBM by way of and acquisition and for suppliers.

Summary

IBM has long been recognized as a producer of hardware, software, and solutions that are built with high quality, reliability, function, and integrity. IBM accomplishes this recognition through attention to these aspects as teams conceive, design, develop, test, deliver, and service these offerings.

The offering management process, which includes the requirements of Secure Engineering and Security and Privacy by Design can help organizations ensure that appropriate attention to security is paid at all stages of product development. By following a direction of continuous security improvement, product teams work to continually improve the security characteristics of the hardware, software, and solutions we deliver.

The development process and offering lifecycle within an organization are often elements of the larger global supply chain management system that ensures quality and integrity in the products and services that an organization provides worldwide. Attention to security is required across the global supply chain and the development processes to deliver products that have appropriate security characteristics and resistance to vulnerabilities.

Latest updates

The 2018 updates were done on this document to align with current development practices.

Authors

This guide was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO).

Warren Grunbok is a Senior Security Architect who is responsible for applying the latest industry trends, technologies, and leading edge thinking to the security of IBM's systems hardware products. He works closely with industry in the area of vulnerability management, Common Criteria and FIPS. Warren holds an MS in computer science from Marist College and is both a CISSP and a CEH certified ethical hacker. He has been working for the IBM systems group developing operating system software in the areas of high availability, systems management, virtualization, and security for over 30 years. He has several patents in system availability, payment systems, system security, and block chain-related payment systems.

Marie Cole is a Distinguished Engineer in IBM Systems Supply Chain Engineering. She joined IBM in 1984 after completing a B.S. in Chemical Engineering from Rensselaer Polytechnic Institute and then earned an M.S. in Materials Science from Columbia University. She is responsible for Supplier Technical Management strategies to evaluate the quality and reliability of new technologies before they are introduced into IBM Server and Storage systems. She is an internationally recognized expert in electronic packaging materials. Marie has led development teams responsible for microelectronic packaging, card assembly and subsystems with a focus on environmentally friendly materials and processes. She holds several US Patents, is a member of the IBM Academy of Technology, and received several IBM technical awards, including a Corporate Award for Lead-free Soldering Transformation and a Best of IBM Award. Marie is a life member of the Society of Women Engineers and the recipient of a 2017 SWE Spark Award for mentoring.

Thanks to the following authors of the first edition of this IBM Redpaper™ publication for creating the groundwork for this second edition:

Danny Allan
Tim Hahn
Andras Szakal
Jim Whitmore
Axel Buecker

Now you can become a published author, too

Here's an opportunity to spotlight your skills, grow your career, and become a published author, all at the same time. Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us.

We want our to be as helpful as possible. Send us your comments about this book or other IBM Redbooks® publications in one of the following ways:

- Use the online **Contact us** review form:

  ibm.com/redbooks

- Send your comments by email:

  redbook@us.ibm.com

- Mail your comments:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 actual people or business enterprises 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AppScan®
IBM®
Rational®
Redbooks®
Redguide™
Redpaper™
Redbooks (logo) ®

The following terms are trademarks of other companies:

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

REDP-4641-01

ISBN 0738457175

Printed in U.S.A.