This presentation will focus on Common Secure Interoperability version 2.

Goals

- Discuss the Authentication protocol used for EJB security, namely the open Java™ 2 Enterprise Edition (J2EE) standard CSIv2

The goals for this presentation are to discuss the CSIv2 authentication protocol used for EJB security.

WebSphere Application Server Version 6 supports both the CSIv2 and SAS authentication protocols. SAS is the old IBM protocol and is not discussed in this presentation.

You should complete the WebSphere Version 6 Security Architecture module, or have a good understanding of it, as a prerequisite to this module.

Agenda

- Authentication protocol for EJBs - overview
- CSIv2 - Overview and features
- Configuration
  - Server Side within WebSphere
  - Client Side within WebSphere
  - Client Side for Stand-alone Java/J2EE client applications
- CSIv2 Examples

The agenda for this presentation is to:

Provide an overview of the CSIv2 EJB authentication protocol,
Demonstrate security configuration for server side, client side, and Java clients,
and provide some examples.

Section

Authentication protocol for EJB

This section will discuss authentication protocol for Enterprise JavaBeans.

Authentication protocol for EJB client and server

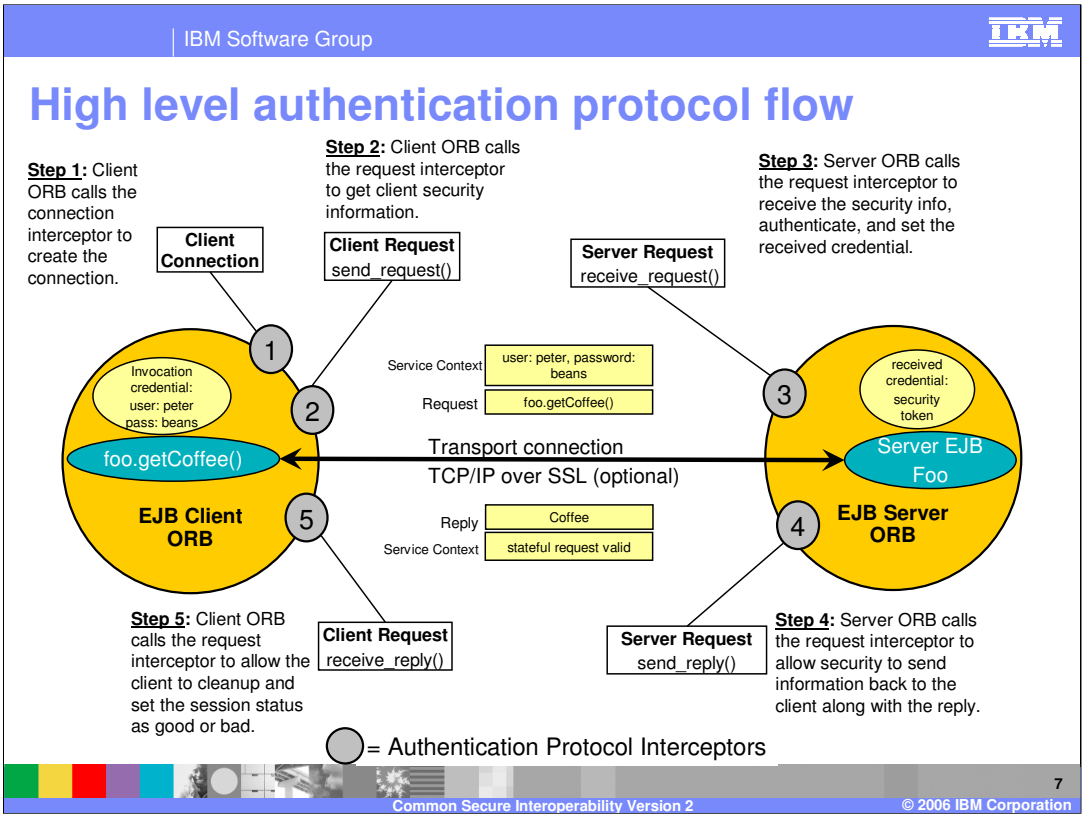
- Authentication protocol determines the level of security and the type of authentication that needs to occur between the EJB client and the EJB for each request in a secure environment
  - It finds the appropriate authentication policy suitable for both the client and the server by coalescing of their configurations
- WebSphere V6 supports 2 authentication protocols:
  - **Common Secure Interoperability Version 2 (CSIv2) - RECOMMENDED**
    - Defined by Object Management Group (OMG) and is part of the J2EE standards
  - Secure Authentication Service (SAS) – for backward compatibility
    - Used by previous levels of WebSphere Application Server
- EJB request and response uses Inter-ORB Protocol (IIOP) services
  - IIOP is a request-and-reply communications protocol used to send messages between two Object Request Brokers (ORBs)
- The EJB authentication protocol used by WebSphere V6 are add-on to IIOP services

EJB clients making secure RMI-IIOP calls to EJBs must provide authentication information. This is done using either the CSIv2 or the SAS authentication protocol supported in WebSphere Application Server Version 6.

Authentication policy decisions

- Based on the EJB client/server configurations, the authentication policy makes the following decisions
  - Connection type - SSL or TCP/IP
  - If SSL is chosen, how strong is the encryption of the data and whether to authenticate the client using client certificates
  - Whether to authenticate the client with a user ID and password or an existing credential
  - Whether to assert the client identity to downstream servers
  - Given the configuration of the client and server, check if a secure request can proceed
    - If the authentication protocol cannot come up with a policy that both the client and the server can satisfy, the request is not sent

The authentication policy is responsible for determining the connection type, encryption strength, authentication type, whether client identity should be asserted downstream, and if the secure request can proceed.

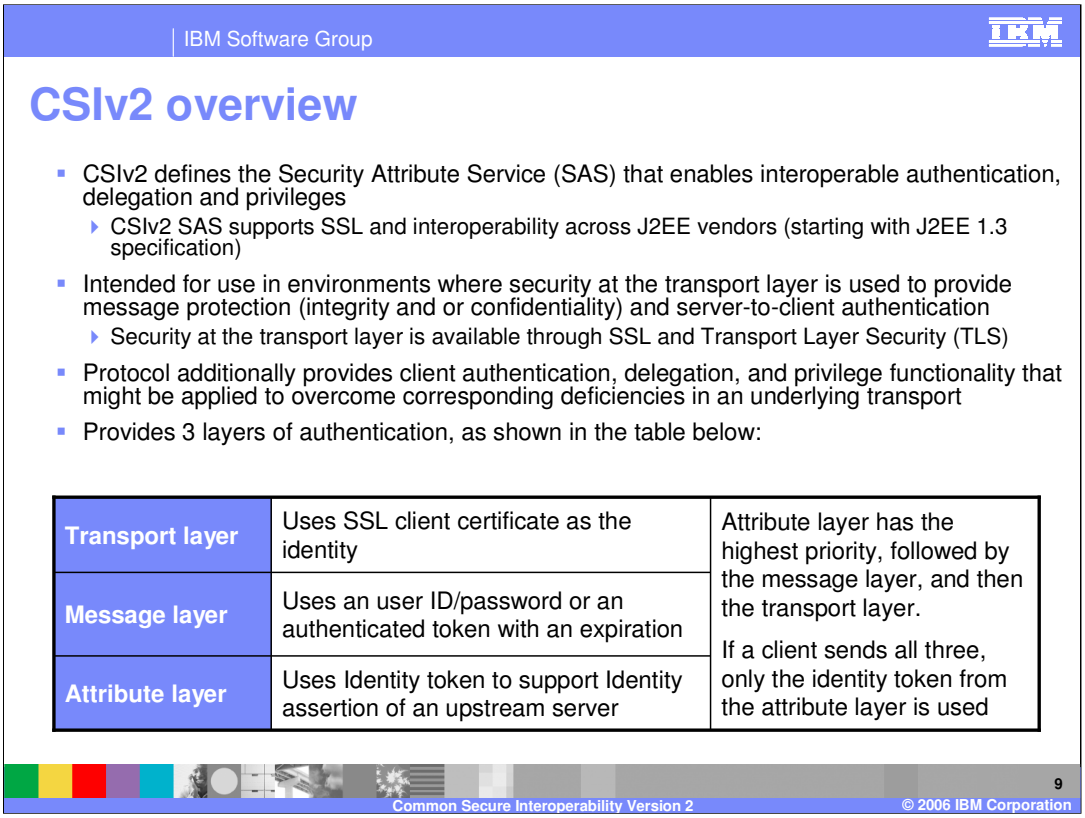


Shown here is a graphic representation of the authentication flow.

Section

CSIv2 – Overview and features

This section will provide an overview of CSIv2 features.



CSIv2 enables interoperable authentication, delegation, and privileges and is intended for use in environments where SSL and TLS are used at the transport layer to provide message protection and server-to-client authentication. The Security Attribute Service in CSIv2 is totally different from IBM SAS authentication protocol.

CSIv2 features

- Features
  - SSL Client Certificate Authentication
  - Message Layer Authentication
  - Identity Assertion
  - Security Attribute Propagation
  - Stateful and Stateless choices
    - Stateful sessions that are used mostly for performance improvements
    - The 1st contact between a client and server must fully authenticate
    - Subsequent contacts with valid sessions reuse the security information
    - The client passes a context ID to the server, and the ID is used to look up the session. The context ID is scoped to the connection, which guarantees uniqueness
    - In cases where you need authentication on every client/server contact, specify Stateless option

This slide provides an overview of the security features offered by CSIv2. Stateful sessions require authentication on the initial contact only and therefore offer better performance. For applications where authentication is needed for every contact, specify the stateless option.

SSL client certificate authentication

- An additional way to authenticate a client to a server using SSL client authentication
- Disable message layer on the client side security (user ID/password) option in the configuration, if the SSL certificate is the identity against which to invoke the method
- If no message layer security exists, then no security context is created and associated with the request – If the server does not find a service context, it checks the server socket for a client certificate chain that contains the client identity
- A credential is created by mapping the identity from the certificate to the user registry
  - For Local OS: The 1st attribute of the DN in the certificate is used to map to the user ID in the registry - Example: For DN "cn=Smith, ou=NewUnit, o=NewCompany, c=us", the user ID is "smith"
  - For LDAP: Either mapping the Subject field in the certification with the EXACT DN name or by matching attributes in the certificate to attributes of LDAP entries
- Advantage: Optimizes authentication performance, because an SSL connection is typically created anyway - Extra overhead of sending the client certificate is minimal
- Disadvantage: Complexity of setting up the keystore file on each client system

SSL Certification Authentication does not occur at the message level, but occurs during the connection handshake using SSL certificates. The advantage of using a certificate is increased authentication performance. The disadvantage is the complexity of configuring each client with the proper keystore file.

Message layer authentication

- Defines the credential information and sends that authentication information, using a token, across the network to a receiving server
  - Token can be user ID/password (Generic Security Services Username Password (GSSUP)) or mechanism-specific format token, like LTPA token
    - Pure Java client uses basic authentication (GSSUP) whereas a Servlet can use basic authentication or LTPA token
    - Cannot use SWAM authentication mechanism, since the tokens are not forwardable in SWAM
- Authentication information is sent with the message inside a service context
- The server knows the mechanism to use when reading and validating the token
  - Each authentication mechanism has an object ID (OID) representing it - OID and the client token are sent to the server
- Optionally, you can set authentication retries and the number of retries

Message layer authentication uses a token to store and exchange credential information with the receiving server. Tokens can contain credentials in either the GSSUP user ID/password format or a mechanism-specific format, such as an LTPA token. You can also specify security policies such as the number of retries allowed.

Identity assertion

- Allows the client identity to be asserted to a downstream server using a CSIv2 identity token - This can be beneficial when there is no common authentication token that can be sent at the message layer for interoperability
- Trust is established between servers prior to using the client identity
- This is commonly used when an EJB or a Servlet on one server calls another downstream server
- The upstream server sets the invocation credential, based on RUN-AS mode of the EJB or Servlet
  - Run-As mode could be originating client identity, the server identity, or a specified different identity (the last one does not apply to a Servlet)
- Upstream server sends an identity token that contains the invocation credential and the upstream server identity (basic authentication or token)
- The downstream server receives the upstream server identity and the invocation identity and does the following
  - Checks if the upstream server (from the upstream server identity) is in its list of trusted servers and if so, authenticates the upstream server
  - Then invocation identity token is mapped to a user – the mapping is based on user registry
  - For Stateful server, this checking is done only once - subsequent requests are made through a session ID

Identity assertion uses a CSIv2 identity token to identify the client to the server, which is helpful when there is no common authentication token. In this case, the trust relationship is established in advance of the request by means of a trusted server list or user registry. This is useful for an EJB or servlet that routinely calls a downstream server. The requesting server sends a token containing an invocation credential, based on the RUN-AS mode of the EJB or servlet. The receiving server checks to see if the token contains valid credentials and if it does, authenticates the requesting server.

Security attribute propagation

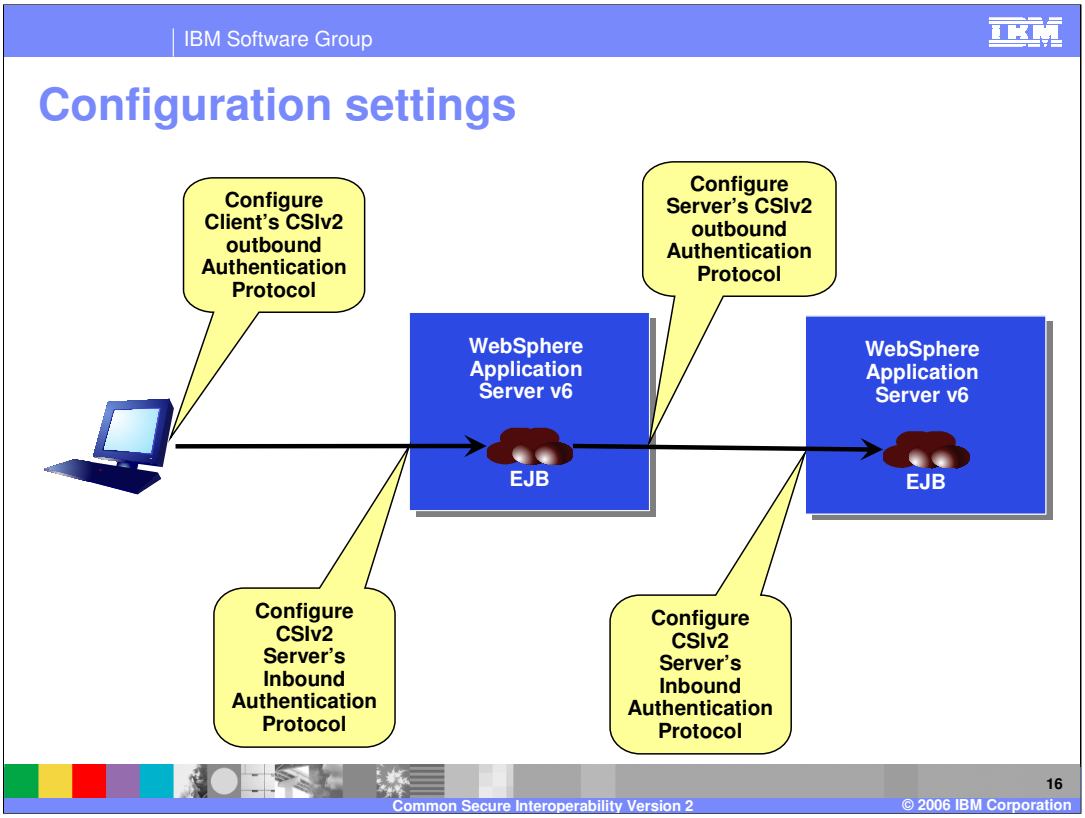
- Transport security attributes (authenticated Subject contents and security context information) from one server to another
  - Attributes might include original caller identity, location, IP address, and so on
  - Java objects in the Subject must be serialized
- Downstream servers do not have to lookup a user registry to get attributes

Security attribute propagation allows authenticated subject contents and security context to be passed from one server to another. Java objects contained in the subject must be serialized. The advantage offered by this feature is that downstream servers do not have to perform a lookup in a user registry in order to get attributes.

Section

CSIv2 or SAS configuration

This section will cover the configuration of CSIv2 or SAS.



This graphic depicts a typical security configuration.

For a stand-alone J2EE client, the authentication protocol (CSIv2 or SAS) is specified by means of properties. Later, some of the key properties will be shown.

For an Application Server receiving a request, Inbound protocol must be configured. For an Application Server running EJB clients and sending EJB request out to another EJB on another server, the Outbound protocol must be configured.

Section

Server side configuration within WebSphere

This section will cover the server side configuration within WebSphere Application server V6.

Server side configuration

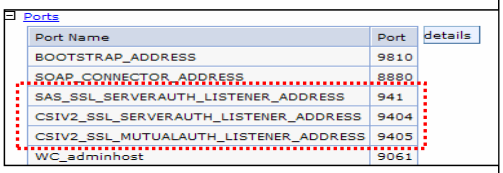
- V6 Security configuration allows you to configure both the CSIv2 and SAS protocols for incoming requests (EJB acting as a server)
  - SAS would be needed if the requests are coming for EJB clients in V4 and below
- When a request is made to the server, the IOR (Interoperable Object Reference) sent back will contain the configurations of CSIv2 or SAS based on what is set for incoming requests
  - You can configure both protocols (CSIv2 and SAS) to work simultaneously

If you are just using CSIv2, do not configure for the SAS protocol. If you do, the SAS interceptor will be called, but do nothing. You would experience better performance by configuring only CSIv2.

Pause this presentation and click the Show-me icon for a demonstration on how to configure CSIv2 and SAS inbound and outbound protocols.

CSIv2 and SAS TCP/IP and SSL inbound ports

- Server opens the following listener port(s), based on the selection
  - SSL-required ➔ SSL port only
  - SSL-supported ➔ Both TCP/IP and SSL ports
  - TCP/IP ➔ TCP/IP port only



Servers > **Application Servers** > *server_name* > **Ports**

Ports	Description
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	CSIv2 Client Authentication SSL Port
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	CSIv2 SSL Port
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	SAS SSL Port
ORB_LISTENER_PORT	TCP/IP Port

For server to dynamically assign port numbers to these ports, specify a value of 0

This slide provides detail concerning port usage for CSIv2 and SAS TCP/IP and SSL.

Section

Client side configuration within WebSphere

This section will cover client side configuration inside WebSphere Application Server V6.

Client side configuration

- V6 security configuration allows you to configure both the CSIv2 and SAS protocols for outbound requests (Applications within WebSphere Application Server are EJB clients)
  - SAS is needed if the requests are made to an EJB running in V4 and below
  - You can configure both protocols (CSIv2 and SAS) to work simultaneously

As previously stated, if you use only CSIv2, do not configure SAS. If you do, the SAS interceptor will get called, but do nothing. You will experience better performance by configuring only CSIv2.

Pause this presentation and click the Show-me icon for a demonstration on how to configure CSIv2 and SAS inbound and outbound protocols.

Section

Client side configuration outside WebSphere

This section will discuss client side configuration outside of WebSphere Application Server Version 6.

Client side configuration for stand-alone client

- Client is running outside WebSphere Application Server in a Stand-alone mode
- Secure client requires configuration properties to determine how to perform security with a server
  - Typically specified in a properties file and using JVM option "-Dcom.ibm.CORBA.ConfigURL"
  - A sample properties files "sas.client.props" is provided
    - Located in *install_root*/properties directory
  - Example:
    - -Dcom.ibm.CORBA.ConfigURL=file:/c:/WebSphere/AppServer/properties/sas.client.props
- The different properties that can be specified in the security properties files are shown in the next few slides

The Client could be an Applet, a Stand-alone Java client or a J2EE application client. Secure clients require configuration properties in order to determine how to authenticate with a server. This is typically done by using the –Dcom.ibm.CORBA.ConfigURL JVM option and specifying the location of a properties file such as sas.client.props.

Properties needed for client configuration

- To enable SSL client certification authentication:
  - Is SSL supported or required
    - com.ibm.CSI.performTransportAssocSSLTLSRequired (true or false)
    - com.ibm.CSI.performTransportAssocSSLTLSSupported (true or false)
  - Is SSL client authentication supported or required
    - com.ibm.CSI.performTLClientAuthenticationRequired (true or false)
    - com.ibm.CSI.performTLClientAuthenticationSupported (true or false)
- Message layer authentication type
  - Is Message Layer authentication supported or required
    - com.ibm.CSI.performClientAuthenticationSupported (true or false)
    - com.ibm.CSI.performClientAuthenticationRequired (true or false)
  - com.ibm.CORBA.authenticationTarget
    - Basic authentication is currently the only valid value
- Authentication retries
  - com.ibm.CORBA.authenticationRetryEnabled (true or false)
  - com.ibm.CORBA.authenticationRetryCount (number of retries)

Listed here are some of the properties that must be specified in the client configuration. These properties determine such things as whether or not SSL and message layer authentication are supported or required and the authentication retries policy.

Properties needed for client configuration (cont.)

- Stateful session between client and server
  - com.ibm.CSI.performStateful (true or false)
- Ciphers for SSL connections
  - 128 bit
    - com.ibm.CSI.performMessageConfidentialitySupported (true or false)
    - com.ibm.CSI.performMessageConfidentialityRequired (true or false)
  - 40-bit
    - com.ibm.CSI.performMessageIntegritySupported (true or false)
    - com.ibm.CSI.performMessageIntegrityRequired (true or false)

Other properties include whether or not stateful sessions are used between the client and the server and the encryption strength.

Section

Summary

This section will provide a summary of the concepts covered by this presentation.

Summary

- CSIv2 defines standards based authentication protocol for EJBs, with some of the main features being:
  - SSL Client Certificate Authentication
  - Message Layer Authentication
  - Identity Assertion
  - Security Attribute Propagation
  - Stateful and Stateless choices

In summary, this presentation has focused on the CSIv2 protocol, which defines the security attribute service that enables interoperable authentication, delegation, and privileges. It also features SSL client certificate authentication, message layer authentication, identity assertion, and security attribute propagation in stateful or stateless modes.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM(logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

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

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

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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

Linux is a registered trademark of Linus Torvalds.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

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

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.