WebSphere® software
IBM

Redpaper

Alex Louwe Kooijmans
Mitch Johnson

J2C Security on z/OS

Introduction

This paper describes security options of IBM WebSphere Application Server and Enterprise Information Systems (EIS) when using J2EE™ Connector Architecture (J2C) connectors.

We cover the following topics:

- J2EE Connector Security, in "Security and J2EE connectors" on page 2
- Security when accessing CICS®, in "WebSphere and CICS connectivity" on page 10
- Security when accessing IMS™, in "WebSphere and IMS connectivity" on page 22

Security and J2EE connectors

Enterprise Information Systems (EIS) such as CICS and IMS generally require that a locally defined identity be used within the EIS for resource authorization checking. Here we address various scenarios for providing this authorization identity (or in WebSphere®'s terms, a Connection Identity) from a J2EE application using J2C to access CICS or IMS.

First let us consider the scenarios:

- Security when accessing an EIS is straightforward when the WebSphere application has access to both a user ID and a password. In this case the deployment descriptor can specify application-managed security and the user ID and password are sent as part of the connection request. The EIS authenticates the user ID and password and use this user ID for authorization checks.
- Security is more interesting when the application only has access to a user ID but not to a password. In this case the application needs to assert this user ID to the EIS (*Identity Assertion*) as having already been authenticated, to avoid further authentication by the EIS.
- Finally, the most interesting scenario is when there is no user ID available to the WebSphere application or the runtime container but still a connection identity needs to be asserted to the EIS for EIS authorization checks. In this case the connection identity must be derived by the EIS by means other than those provided by WebSphere (see "Certificate mapping and certificate naming filtering" on page 9).

Before we explore the options available in detail, let us review WebSphere and J2C security-related concepts and terms.

WebSphere terms and concepts

Identity Assertion

As described above, *Identity Assertion* is the sending of a connection request from WebSphere to an EIS with only a user ID provided. WebSphere asserts the user ID has already been authenticated by WebSphere and no further authentication is required by the EIS.

Resource Reference

J2EE applications should access an EIS with a context lookup that uses an indirect JNDI lookup (that is, using a prefix of "java:comp/env"). An indirect JNDI lookup searches for a *Resource Reference* with that name in the global name space. Resource reference information is provided in the EJB™ deployment descriptor and provides a JNDI name of the connection factory (which a deployer can associate with a WebSphere-configured J2C Connection Factory) and other connection parameters such as authentication type of either “application” or “container” and the sharing scope.

Application-managed security

A specification of application-managed security for a resource reference in an EJB or Web deployment descriptor indicates that the application assumes responsibility for providing the connection identity either in a `connectionSpec` property or via a component-managed authentication alias.

Container-managed security

A specification of container-managed security for a resource reference in an EJB or Web deployment descriptor indicates that the runtime container will provide the connection identity. The connection identity is determined by the resource adapter and is derived from a container-managed authenticating alias, the RunAs specification, or by some other means.

> **Note:** If the Resource Reference specifies container-managed security, the application has no control over the connection identity. Even if an application provides a user ID and password in a `connectionSpec`, they will be ignored by the container during the connection request.

Authentication Alias

An “Authentication Alias” is a user ID and password pair that can be associated with one or more J2C Connection Factories. There are two types of J2C Connection Factory aliases, Component Managed and Container Managed. The choice of which alias is used is determined by the Resource Reference in the deployment descriptor. The resource adapter obtains the connection identity and password from the J2C Connection Factory Component alias for application-managed resource references and the Container alias for container-managed resource references.

> **Note:** In our opinion the use of an Authentication Alias for providing a connection identity is problematic from an administrative point of view. Authentication Aliases have to be created and maintained by an administrator and are not flexible for determining connection identities for different applications using the same J2C Connection Factory. Finally, since the password is often subject to expiration, the password component of each alias must be updated on a regular basis (usually after the password has expired).

Thread Identity

When we address J2C connector security, the major issue concerns the connection identity that is used to access the EIS. In IBM WebSphere Application Server for z/OS there is a unique feature called “Thread Identity” support, which is the ability to pass to a J2C-compliant connector the current J2EE Thread Identity, which then can be flowed through to the EIS as the connection identity.

> **Note:** Thread Identity refers to the identity of the Java™ principal within the Subject and is sometimes called the Java Thread Identity or J2EE Identity. This identity is initially established when authentication takes place in WebSphere and optionally may be changed by use of RunAs deployment descriptors.

The important concept to grasp is that within WebSphere you might have many Java threads all executing under different Java principals, and therefore each Java thread will have a different current Thread Identity.

Thread Identity support is applicable only when:

- The J2C resource adapter allows the use of Thread Identity. For the IMS and CICS Connectors this means when they are in local mode configuration with the default principal mapping module,

  > **Note:** We were able to useThread Identity for IMS and CICS Connectors to assert the J2EE identity over remote connections with a customized principal mapping module.

- When res-auth=Container is specified for the Resource Reference defined in the deployment descriptor
- When the connection factory does not specify a Container-Managed Authentication Alias

> **Note:** When a CICS or IMS resource adapter is installed into WebSphere, the adapter declares its level of support to the container. You do not need to set any specific property to “switch on” Thread Identity support for the WebSphere Application Server.

Table 1 on page 5 shows the connectors that can be configured in local configuration mode and are therefore able to exploit Thread Identity support.

Table 1 J2C connectors exploiting Thread Identity Support

Connectors	Thread Identity
IMS Connector local configuration	Allowed
IMS Connector remote configuration	Not Allowed
CICSECI local configuration	Allowed
CICSECI remote configuration	Not Allowed

When the conditions for Thread Identity are met (local configuration and res-auth=container and no authentication alias), WebSphere passes the current J2EE Thread Identity to the Resource Adapter and then the Resource Adapter passes it to the EIS.

Note: WebSphere Thread Security (also know as Connection Manager RunAs Identity or Synch-To-OS Thread) does not apply to J2C connections to CICS or IMS. Thread Security only applies when using JDBC™ to access DB2® or IMS and refers to the ability of WebSphere on z/OS® to do a security context switch of the current thread or J2EE identity onto the RACF® Access Control Element (ACEE) associated with the OS thread TCB. This J2EE identity is subsequently used as the connection identity when accessing a local DB2 or IMS Database subsystem using JDBC.

RunAs identity

The RunAs deployment descriptor setting determines the security identity that is propagated as the RunAs identity from a caller to a called object. You can use this process to switch the J2EE security identity before running a method on another object or connecting to an EIS

Note: When you set the RunAs descriptor for an EJB, it does not affect the security identity under which the methods of that EJB execute. It only affects the security identity used when invoking another object or, in our case, the connection identity used to access an EIS.

There are three types of RunAs mode descriptors:

Caller	Use the identity of the caller
Server	Use the identity of the EJB runtime container
EJBRole	Use the identity assigned to a specific role

RunAs is important in the context of J2EE connectors because you could use the RunAs deployment descriptor to control the user ID that is flowed through to your

EIS. For example, if you have a servlet that invokes various methods of an EJB, which in turn access an EIS, the RunAs deployment descriptor for the methods within the EJB can be configured to use different connection identities on each EIS interaction.

To illustrate, let us assume we are running in a servant with a Server Identity of ASSR1 and that we have defined two EJBRoles using the RACF commands in Example 1.

Example 1 RACF commands define our EJB roles

```
RDEFINE EJBROLE employee   UACC(NONE) APPLDATA('EMPLOYE')
RDEFINE EJBROLE supervisor UACC(NONE) APPLDATA('SUPERV')
```

Note: When using RACF, the EJBROLE profile for the servlet EJB role will contain an APPLDATA field with a RACF user ID and the methods of the EJB will be executed using the security identity specified in the APPLDATA field.

The servlet's deployment descriptor specifies RunAs with an EJBRole of *employee*. The deployment descriptor in the EJB specifies a default RunAs mode of Caller and four methods with RunAs specifications, as shown in Table 2.

Table 2 EJB RunAs specification

Method	Deployment descriptor
*	Caller
getCompanies	Server
getQuotes	Caller
buy	EJBRole(supervisor)
sell	EJBRole(supervisor

Using these deployment descriptors, the servlet will invoke the methods of our EJB (see Figure 1 on page 7) using different connection identities for accessing CICS or IMS.

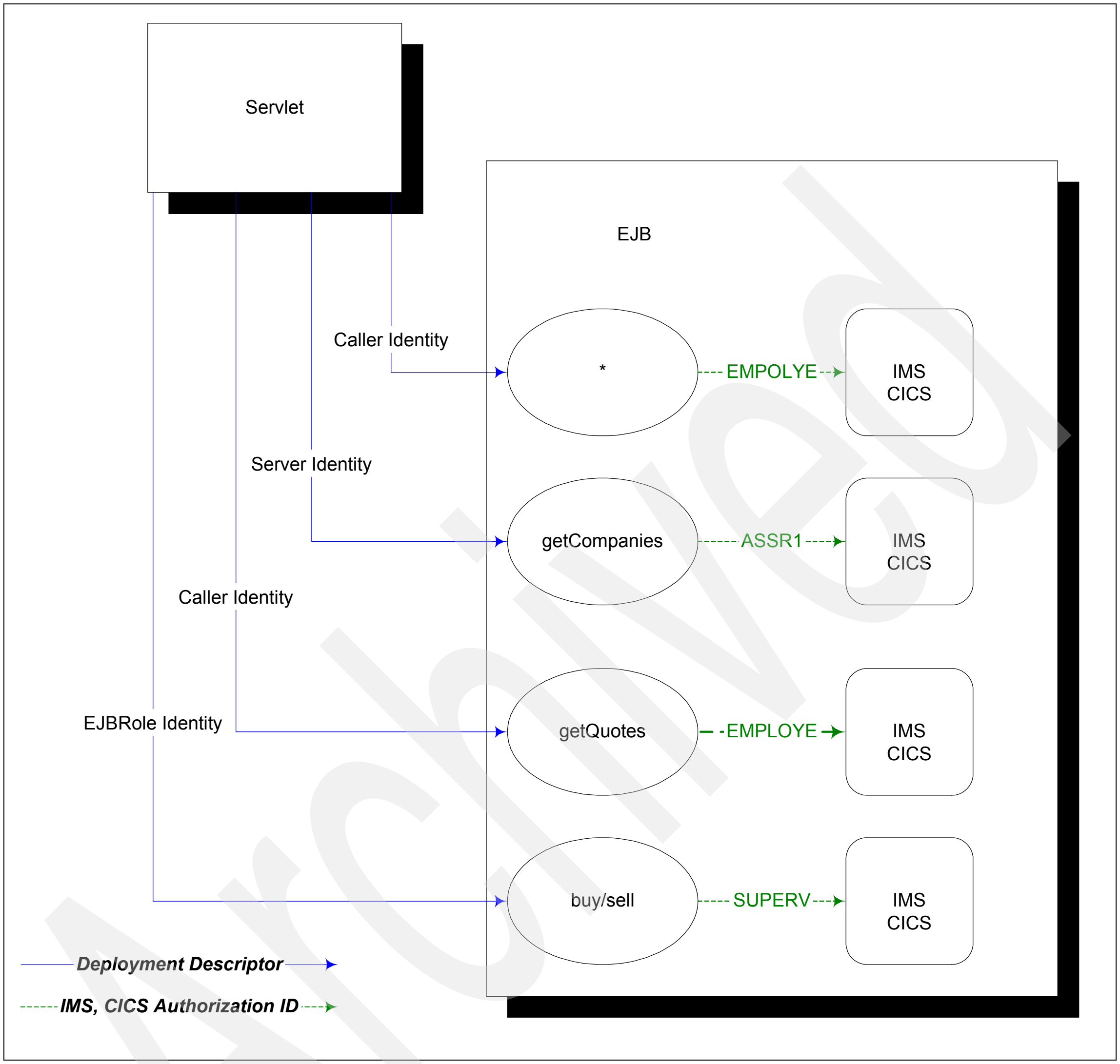


Figure 1   Connection Identities Used to Access CICS or IMS

Method *getCompanies* uses a connection identify of ASSR1 since its deployment descriptor specifies run as Server. Methods *buy* and *sell* use a connection identity of SUPERV since their deployment descriptors specify run as EJBRole *supervisor*. Method *getQuotes* and other methods in the EJB use a connection identity of EMPLOYE since their deployment descriptors specify run as Caller and the servlet invokes the methods with EJBRole *employee*.

In summary, you can use RunAs to provide more flexibility for security in the following situations:

- Initial methods can use MethodPermissions to validate that the caller is authorized for the set of methods needed.
- Downstream method authorizations (which means that the RunAs identity of the current EJB will be used as the Caller identity in methods on any objects called by the current EJB).
- J2EE connectors in conjunction with Thread Identity can use these role-derived identities and the respective APIs to flow a user ID different from the current Java thread identity to the EIS.

J2C reauthentication

J2C reauthentication support in a resource adapter allows the reuse of a connection already in the connection pool when the connection identity changes. Reusing an existing connection in the pool avoids the overhead of establishing a new connection when the only difference is the identity.

CICS and IMS resource adapters support J2C reauthentication following these steps:

- When a connection request is made, the container checks to see if a connection for this factory already exists in the pool and if it is marked as being available and sharable. If the connection identities are the same, the connection is reused.
- If a connection to the EIS is already in the pool and is available but with a different connection identity, the connection is reused with the new connection identity.
- Otherwise a new connection to CICS is created with all of the inherent overhead.

Note that J2C reauthentication requires container-managed security for the resource reference.

Digital certificates and SSL

Digital certificates

Digital certificates are used to validate and identify a request. This validation depends on the participation of a third party or Certificate Authority that generates and signs a digital certificate that is distributed to a requestor by secured means. When a digital certificate is received from a requestor, it is verified by a special certificate known as a Certitude Authority (CA) certificate (which has also been generated and distributed by a CA). If the CA certificate validates the digital certificate as being valid, there is an explicit level of trust of the identity of the requestor.

Certificate mapping and certificate naming filtering

Secured Socket Layer (SSL) communication between WebSphere and CICS and/or IMS enables the encryption of communication and the passing of digital certificates. These digital certificates, which can be self-signed or generated by a Certificate Authority (CA), can be used to authorize access between WebSphere and CICS Transaction Gateway and IMS Connect, and can also be used to assert a connection identity when CTG and IMS Connect are configured to use RACF keyrings.

This is possible because RACF allows the mapping of digital certificates to RACF user IDs. This mapping can be a one-to-one mapping (one digital certificate to one RACF identity; see Example 2) or a many-to-one mapping (multiple certificates to one RACF identity; see Example 3).

- RACF identity determined matching client certificate

  Example 2

  ```
  RACDCERT ID(JOHNSON) MAP IDNFILTER(‘OU=ISSW,O=IBM’)
        SDNFILTER(‘CN=Mitch’) WITHLABEL(‘Mitch’)
  ```

- RACF identity mapped to multiple client certificates

  Filtered by a combination of Subject and Issuers Distinguished Name Criteria.

  Example 3

  ```
  RACDCERT ID(ISSWUSR) MAP IDNFILTER(‘OU=ISSW,O=IBM’)
  WITHLABEL(‘ISSW’)

  RACDCERT ID(IBMUSR) MAP IDNFILTER(‘O=IBM’) WITHLABEL(‘IBM’)
  ```

In Example 2, the user ID JOHNSON is associated with the client certificate since a discrete client certificate already exists.

RACF maps client certificates to a RACF identity. In Example 2, when a client certificate is received with a common name of Mitch and the CA was ISSW in IBM®, the request is mapped to RACF identity JOHNSON, which is issued for subsequent authority checking (the discrete client certificate must not exist).

Certificate Filtering extends Mapping so that distinct certificates do not have to be defined to RACF for every individual user (a major administrative task). In Example 3, RACF mapped certificates issued by an IBM ISSW CA to RACF identity ISSWUSR. A certificate issued by IBM but not by an ISSW CA will be mapped to identity IBMUSR. RACF will map the certificate to the identity that best matches the issuer or subject distinguished name.

SSL authentication and SSL mutual authentication

SSL mutual authentication occurs when a client (a WebSphere J2C connection factory) and a server (CTG or IMS Connect) use a CA certificate to validate each other's personal digital certificates.

WebSphere and CICS connectivity

J2C connectivity from WebSphere to CICS requires CICS Transaction Gateway (CTG) to act as a protocol gateway between WebSphere and CICS.

- CTG converts a request from a Java client and forwards the request as a DPL request to a CICS region using either ECI or EXCI.
- CTG is a Java-based application that can run as an OMVS task on z/OS, as a process on distributed platforms, and within WebSphere on any platform when installed as a resource adapter.
- CTG on z/OS can authenticate a user ID/password with the local SAF (user IDs must have OMVS segments).

> **Note:** The term SAF (System Authorization Facility) refers to any local z/OS security manager. When the term RACF is used, it refers to specific RACF commands or resources. This does not mean to imply that the same functionality is not provided by other security packages.

- CTG on z/OS can authorize a user's access to a CICS region using the SAF's SURROGAT resource class.
- A CTG task on z/OS is trusted by a CICS region more than a CTG process on a distributed platform and is able to assert an identity to CICS while a remote CTG cannot.

The following list summarizes the characteristics of the security interactions between WebSphere and CTG.

- For WebSphere on z/OS, Thread Identity support for local connections is available with the default principal mapping module.
- For WebSphere on distributed platforms and z/OS, Thread Identity support for remote connections requires a custom principal mapping module.
- J2C reauthentication is supported by CTG.
- Identity assertion from WebSphere to CICS is possible when:

  - WebSphere accesses CICS using CTG on z/OS.
  - CTG on z/OS user ID/password authentication is disabled.
  - CICS attachment security from CTG on z/OS to CICS is set to IDENTIFY.

There is a security challenge if you disable CTG user ID authentication or allow CTG to assert an identity to CICS. How can you ensure security and trust between WebSphere and CTG or WebSphere and CICS?

There are options available to address this challenge:

- Mutual authentication of client certificates between WebSphere and CTG
- Enabling of SAF SURROGAT resource checking between WebSphere and CICS
- Enabling of SAF FACILITY resource checking of the EXCI pipe name between WebSphere and CICS
- Enabling SAF FACILITY resource checking of the DFHAPPL name between WebSphere and CICS

WebSphere, CTG and CICS topologies

Now let us summarize WebSphere, CTG and CICS configuration topologies and security considerations for each.

Distributed WebSphere accessing a distributed CTG

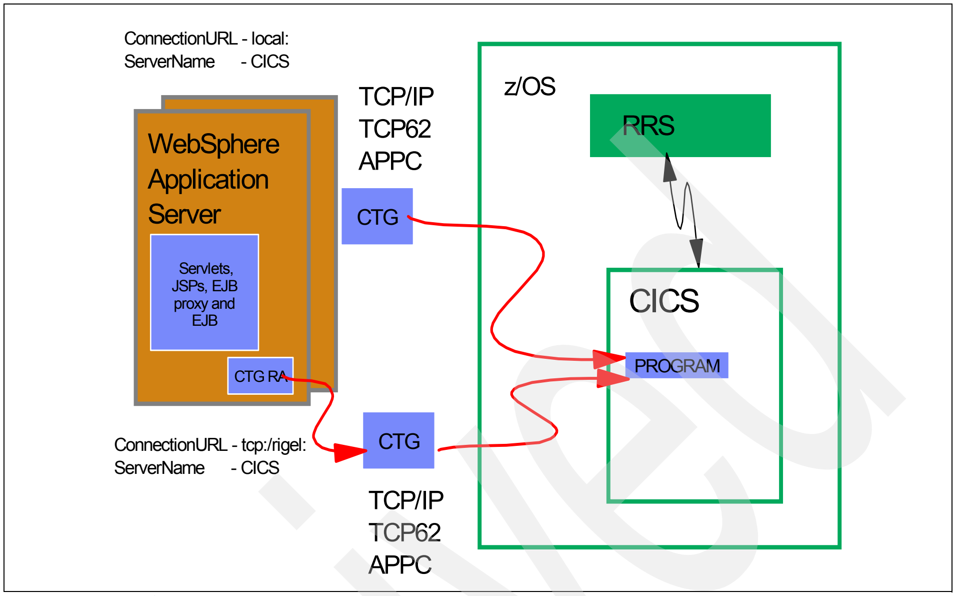


Figure 2 Distributed WebSphere accessing a distributed CTG

Distributed WebSphere accessing CTG on a non-z/OS platform has the following security properties:

- TCP/IP, TCP62, and APPC protocols are supported for communications from a distributed CTG to a CICS region.
- Access to CTG can be controlled by mutual authentication of client certificates when using SSL communication in the J2C connection factory.
- TCP/IP connections from distributed CTG to a CICS region supports LOCAL or VERIFY attachment security.
- TCP62 and APPC connections from distributed CTG to a CICS region support LOCAL, IDENTIFY, VERIFY, PERSISTENT, or MIXIDPE attachment security.
- Identity assertion from WebSphere is not possible even with IDENTIFY attachment security.
- Thread Identity using a password is available using a custom principal mapping module.

Note: 2PC support is not available, but CICS can participate as a last participant in a WebSphere global transaction.

Distributed WebSphere accessing a CTG on z/OS

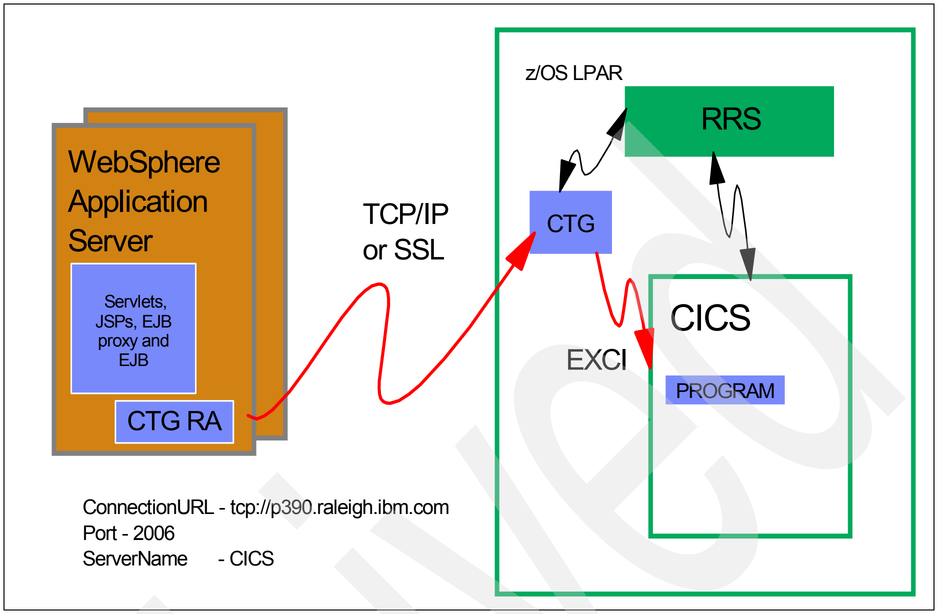

z/OS LPAR
RRS
WebSphere Application Server
Servlets, JSPs, EJB proxy and EJB
CTG RA
TCP/IP or SSL
CTG
EXCI
CICS
PROGRAM
ConnectionURL - tcp://p390.raleigh.ibm.com
Port - 2006
ServerName - CICS

Figure 3 Distributed WebSphere accessing a CTG on z/OS

Distributed WebSphere accessing CTG on z/OS has the following security properties:

- TCP/IP or SSL protocols are supported for communications from a distributed WebSphere to CTG on z/OS.
- Access to CTG on z/OS can be controlled by mutual authentication of client certificates when using SSL communications.
- CTG on z/OS can optionally verify a user ID and password with an SAF (RACF).
- Attachment security from CTG on z/OS to a CICS region can be either LOCAL or IDENTIFY.
- Identity assertion from WebSphere is available if attachment security for the EXCI connection is IDENTIFY and CTG on z/OS user authentication is disabled.
- Thread Identity support is available with a custom principal mapping module.

Note: 2PC support is not available when using pre-CTG for z/OS 6.1, but CICS can participate as a last participant in a WebSphere global transaction. Full 2PC support is available with CTG for z/OS V6.1.

WebSphere on z/OS using an embedded CTG

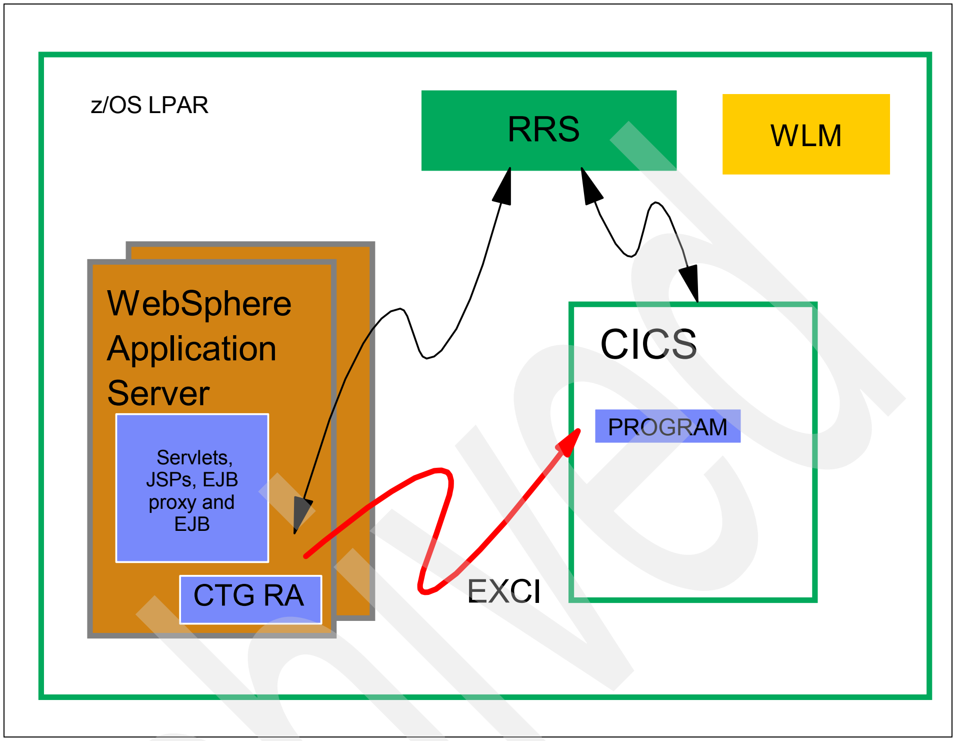

z/OS LPAR
RRS
WLM
WebSphere Application Server
Servlets, JSPs, EJB proxy and EJB
CTG RA
EXCI
CICS
PROGRAM

Figure 4 WebSphere on z/OS using an embedded CTG

WebSphere on z/OS accessing a CICS region directly has the following security properties:

- Attachment security from an embedded CTG on z/OS to CICS can be either LOCAL or IDENTIFY.
- Identity assertion from WebSphere is available if attachment security is IDENTIFY.
- CTG running in the WebSphere on z/OS address space can optionally verify a user ID and password.

Note: Full 2PC is supported for CICS in a WebSphere global transaction.

WebSphere on z/OS accessing a remote CTG on z/OS

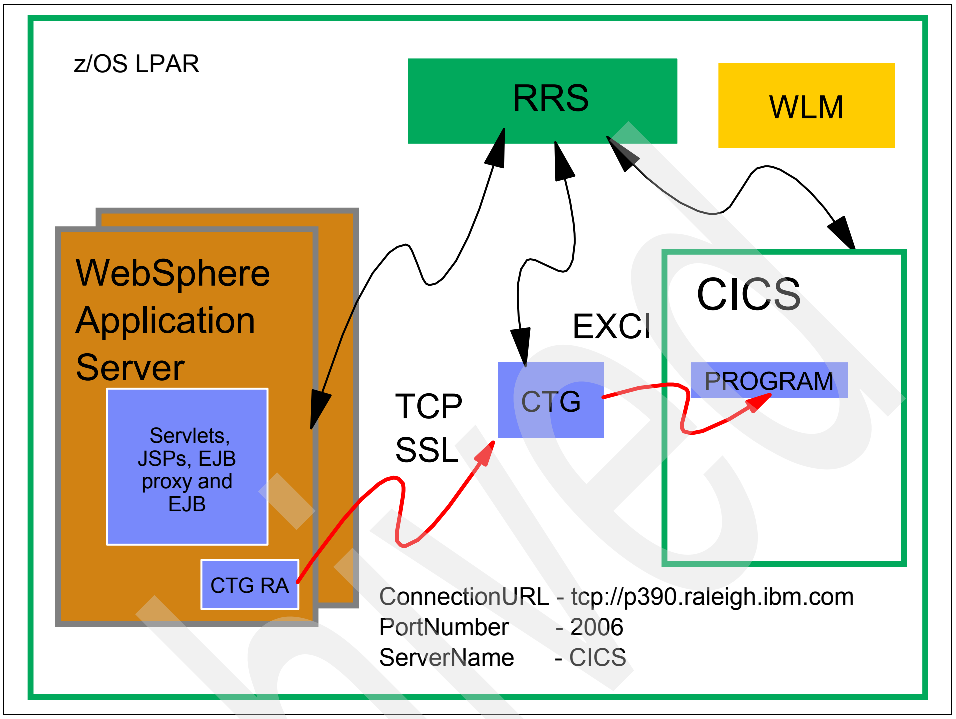

z/OS LPAR
RRS
WLM
WebSphere Application Server
Servlets, JSPs, EJB proxy and EJB
CTG RA
TCP
SSL
CTG
EXCI
CICS
PROGRAM
ConnectionURL - tcp://p390.raleigh.ibm.com
PortNumber - 2006
ServerName - CICS

Figure 5 WebSphere on z/OS accessing a remote CTG on z/OS

WebSphere on z/OS accessing CTG on z/OS remotely has the following security properties:

- TCP or SSL protocols are supported for communications from WebSphere on z/OS to CTG on z/OS.
- Access to CTG on z/OS can be controlled by mutual authentication of digital certificates when using SSL.
- CTG on z/OS can optionally verify a user ID and password with a SAF.
- Attachment security from CTG on z/OS to a CICS region can either be LOCAL or IDENTIFY.
- Identity assertion from WebSphere is available if attachment security is IDENTIFY and CTG on z/OS user authentication is not enabled.

Note: 2PC support is not available when using pre-CTG for z/OS 6.1, but CICS can participate as a last participant in a WebSphere global transaction. Full 2PC support is available with CTG for z/OS V6.1.

Configuring CICS Transaction Gateway for SSL

Next let us explore CTG and WebSphere J2C Connection Factories configuration options and exits that can be used to enable SSL security between WebSphere and CTG.

CTG SSL options summary

CTG in the form of J2C Connection Factories or CTG servers on all platforms support JSSE Keystores created with ikeyman and keytool commands. A CTG V6 server on z/OS supports:

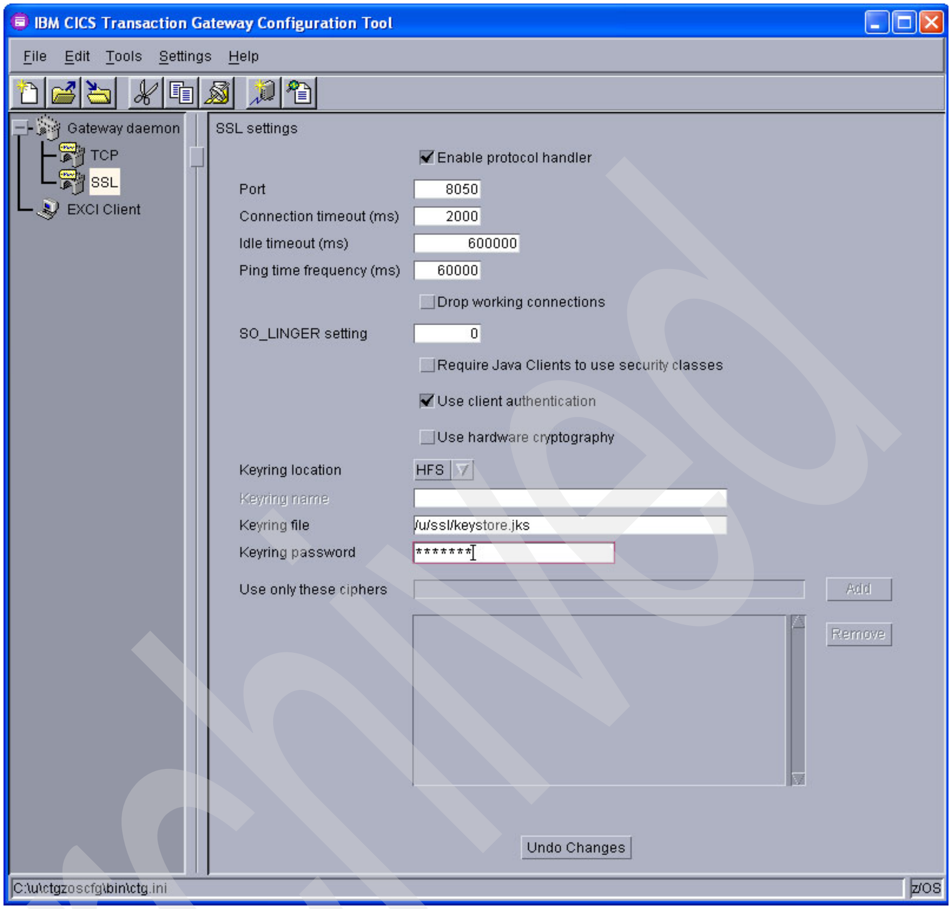
- RACF keyrings created with the RACDCERT command

  This enables identity mapping and Certificate Naming Filtering with SSL.

- ICSF support (requires adding IBMJCE4758 to the list of Java providers) using either:
  - JSSE Keystores created with the **`hwkeytool`** command
  - RACF digital certificates

Figure 6 on page 17 shows how we configured SSL mutual authentication by checking the “Use client authentication” box, and specified a JSSE keystore by providing a file name and password and indicating HFS as a keyring location.

Note: The screen shots of the IBM CICS Transaction Gateway Configuration Tool were made with the ctgzoscfg tool shipped with CTG for z/OS V6.1. The tool was installed on a workstation per the instructions in *CICS Transaction Gateway z/OS Administration Version 6.1,* SC34-6672. The ctg.ini and ctgenvvar files were installed in the appropriate HFS directory on z/OS.


IBM CICS Transaction Gateway Configuration Tool
File Edit Tools Settings Help
Gateway daemon
TCP
SSL
EXCI Client
Add
Remove
Undo Changes
z/OS

SSL settings

☑ Enable protocol handler

Port: 8050
Connection timeout (ms): 2000
Idle timeout (ms): 600000
Ping time frequency (ms): 60000

☐ Drop working connections

SO_LINGER setting: 0

☐ Require Java Clients to use security classes

☑ Use client authentication

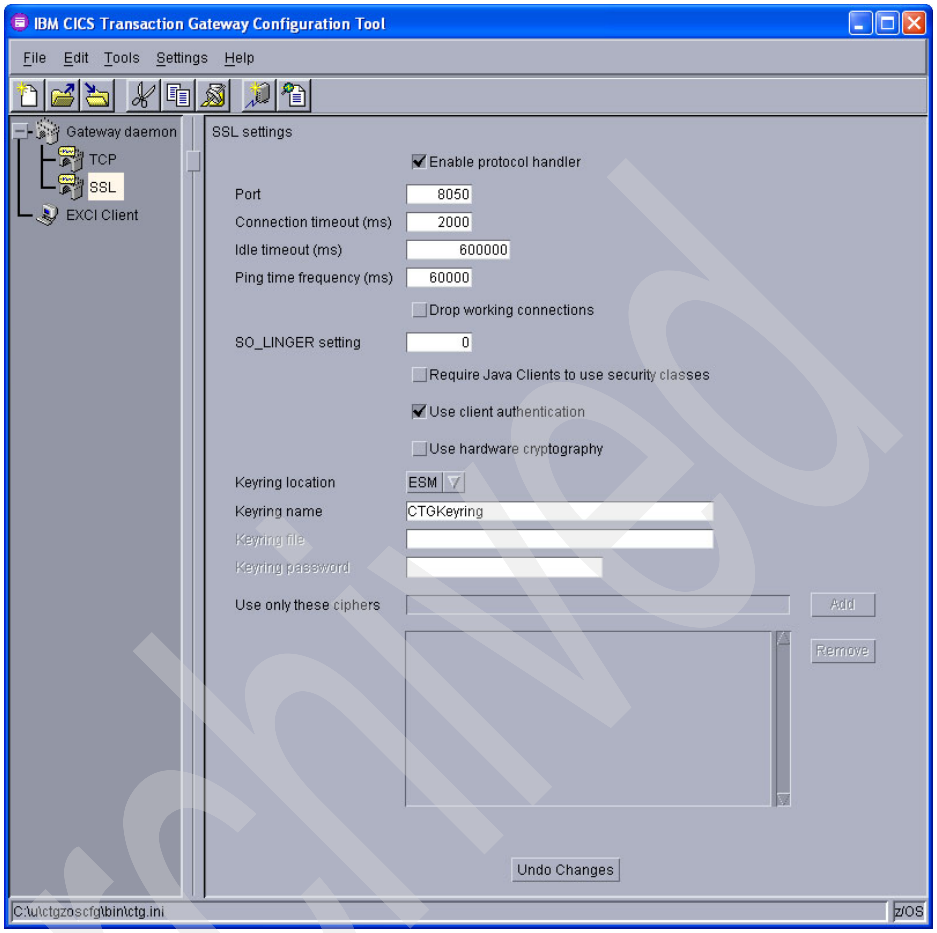
☐ Use hardware cryptography

Keyring location: HFS
Keyring name:
Keyring file: /u/ssl/keystore.jks
Keyring password: *******
Use only these ciphers:

C:\u\ctgzoscfg\bin\ctg.ini

Figure 6 CTG - JSSE SSL with mutual authentication

Figure 7 on page 18 shows how we configured SSL mutual authentication by checking the "Use client authentication" box and specifying a RACF Keyring as the Keyring name and indicating ESM as the Keyring location.


IBM CICS Transaction Gateway Configuration Tool
File Edit Tools Settings Help
Gateway daemon
TCP
SSL
EXCI Client
Add
Remove
Undo Changes
z/OS

SSL settings

☑ Enable protocol handler

Port: 8050

Connection timeout (ms): 2000

Idle timeout (ms): 600000

Ping time frequency (ms): 60000

□ Drop working connections

SO_LINGER setting: 0

□ Require Java Clients to use security classes

☑ Use client authentication

□ Use hardware cryptography

Keyring location: ESM

Keyring name: CTGKeyring

Keyring file:

Keyring password:

Use only these ciphers:

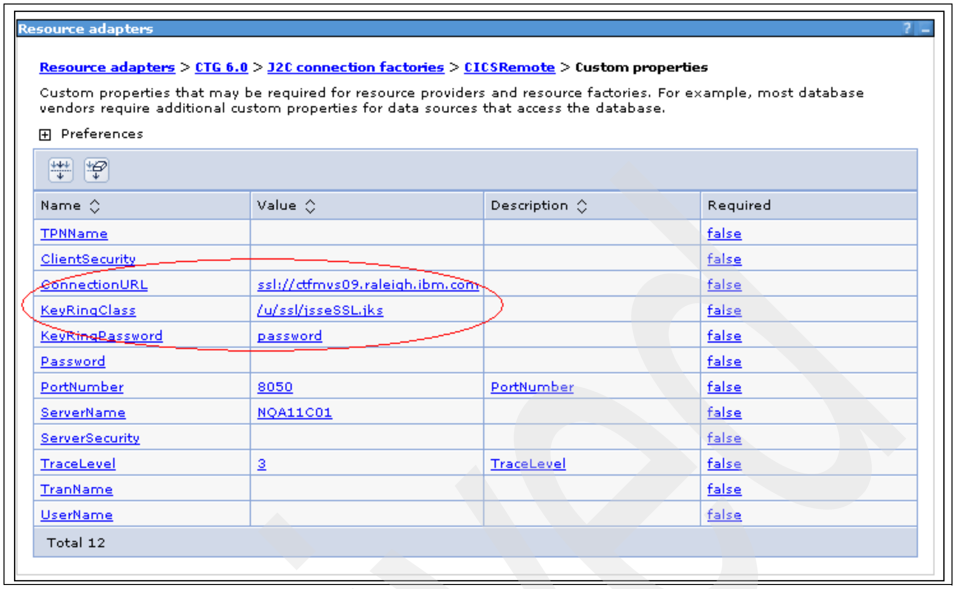
C:\u\ctgzoscfg\bin\ctg.ini

Figure 7 CTG on Z/OS V6 system SSL with mutual authentication

> **Note:** The Keyring specified (CTGKeyring) must be defined for the user ID associated with the CTG task. Also note that a password is not required since access to a RACF Keyring is implicit to the owner of the Keyring.

CICS J2C Connection Factory - SSL custom properties

Configuring a WebSphere CICS J2C Connection Factory requires providing the appropriate ConnectionURL (SSL as the protocol), a JSSE keystore file and password, and the port on which CTG listens for SSL requests (see Figure 8 on page 19).



Resource adapters > CTG 6.0 > J2C connection factories > CICSRemote > Custom properties

Custom properties that may be required for resource providers and resource factories. For example, most database vendors require additional custom properties for data sources that access the database.

⊞ Preferences

Name	Value	Description	Required
TPNName			false
ClientSecurity			false
ConnectionURL	ssl://ctfmvs09.raleigh.ibm.com		false
KeyRingClass	/u/ssl/jsseSSL.jks		false
KeyRingPassword	password		false
Password			false
PortNumber	8050	PortNumber	false
ServerName	NQA11C01		false
ServerSecurity			false
TraceLevel	3	TraceLevel	false
TranName			false
UserName			false

Total 12

Figure 8   CICS J2C Connection Factory custom properties

Application- versus container-managed security and CICS

During our testing we observed the behavior demonstrated in the following flowcharts for determining the connection identity during application- and container-managed security scenarios.

CICS Connection Identity when ResAuth=Application

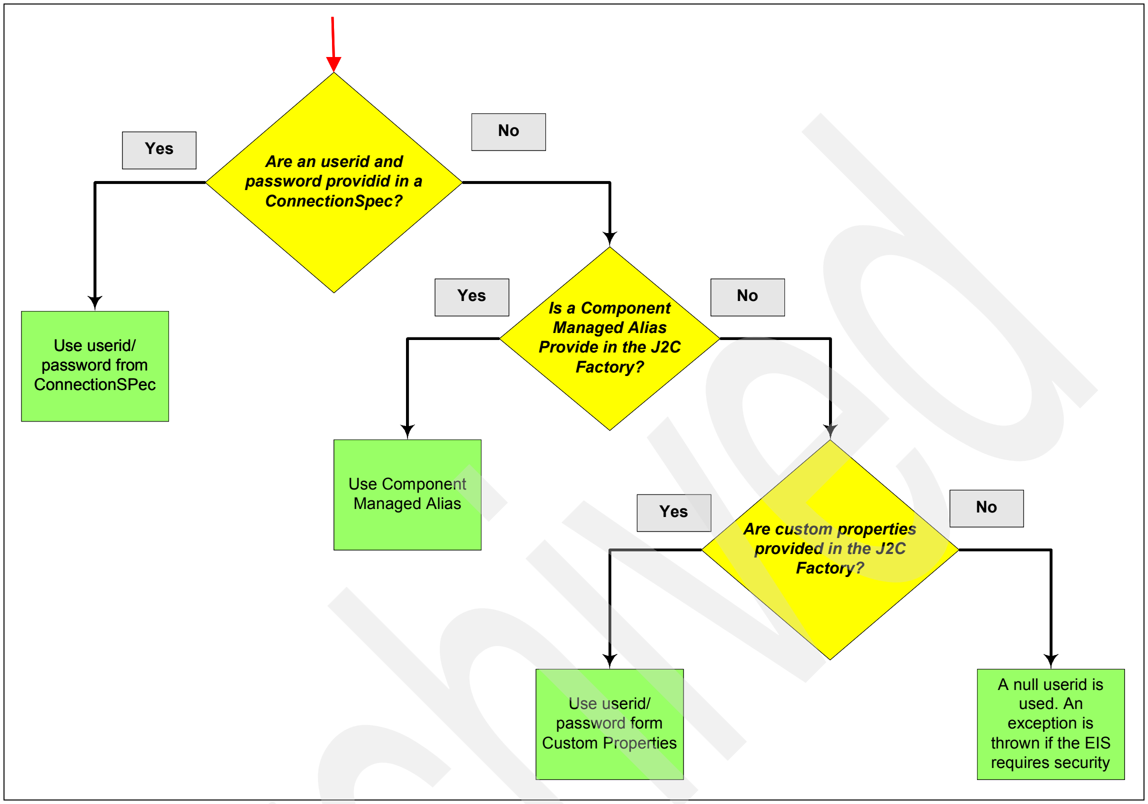

Are an userid and password providid in a ConnectionSpec?
Yes
No
Use userid/ password from ConnectionSPec
Is a Component Managed Alias Provide in the J2C Factory?
Yes
No
Use Component Managed Alias
Are custom properties provided in the J2C Factory?
Yes
No
Use userid/ password form Custom Properties
A null userid is used. An exception is thrown if the EIS requires security

Figure 9 CICS Connection® Identity with application-managed security

CICS ResAuth=Container

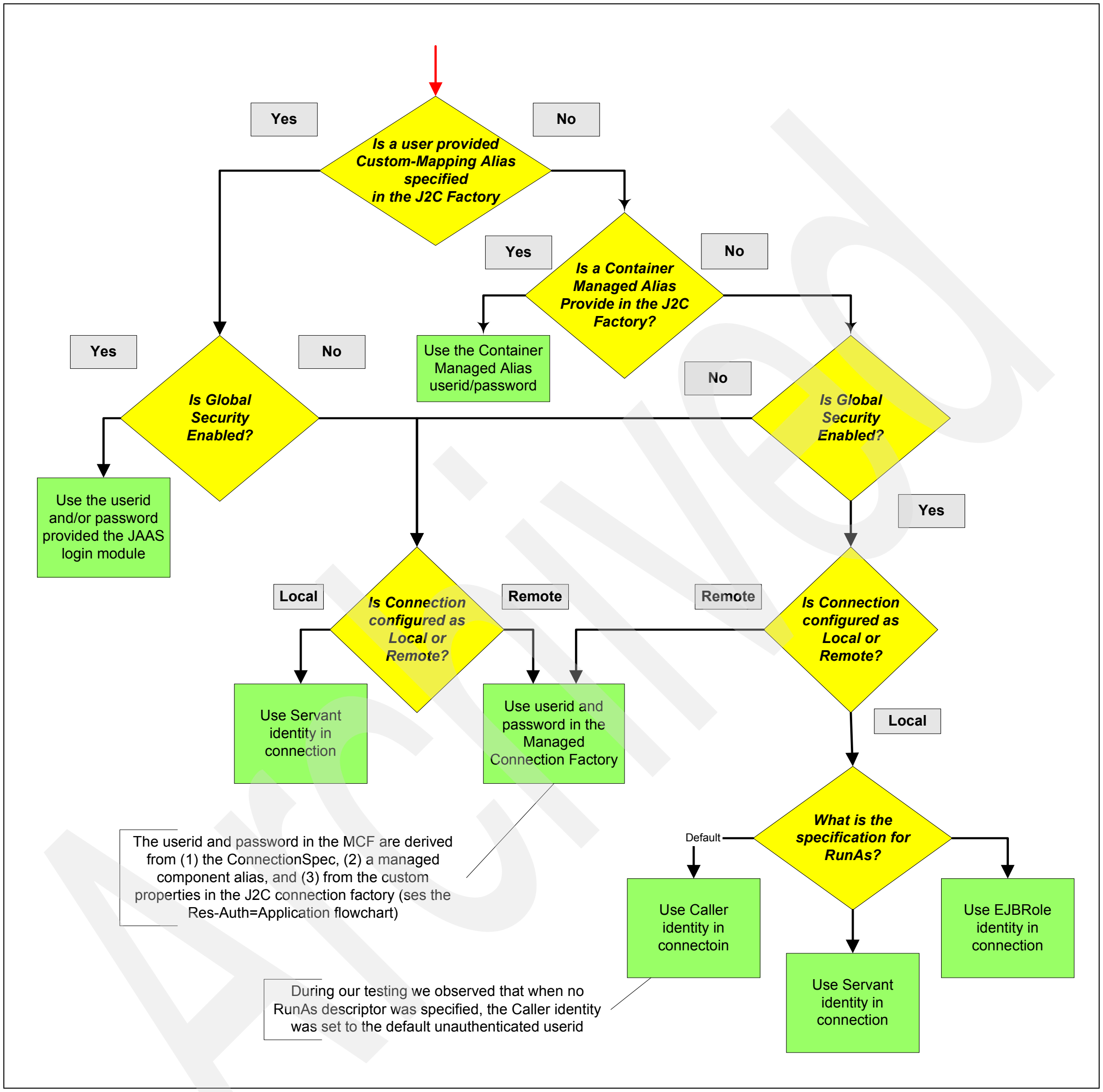

Yes
Is a user provided Custom-Mapping Alias specified in the J2C Factory
No
Yes
Is a Container Managed Alias Provide in the J2C Factory?
No
Use the Container Managed Alias userid/password
Yes
Is Global Security Enabled?
No
Use the userid and/or password provided the JAAS login module
No
Is Global Security Enabled?
Yes
Local
Is Connection configured as Local or Remote?
Remote
Use Servant identity in connection
Use userid and password in the Managed Connection Factory
Remote
Is Connection configured as Local or Remote?
Local
The userid and password in the MCF are derived from (1) the ConnectionSpec, (2) a managed component alias, and (3) from the custom properties in the J2C connection factory (ses the Res-Auth=Application flowchart)
Default
What is the specification for RunAs?
Use Caller identity in connectoin
Use Servant identity in connection
Use EJBRole identity in connection
During our testing we observed that when no RunAs descriptor was specified, the Caller identity was set to the default unauthenticated userid

Figure 10   CICS Connection Identity with container-managed security

Identity Mapping and Certificate Name Filtering with CICS

ServerSecurity and ClientSecurity are optional custom properties in a CICS J2C Connection Factory that can specify exit points. A ServerSecurity exit on CTG on z/OS can be used to extend the functionality of SSL security between

WebSphere and CICS to assert an identity between WebSphere and CICS using RACF Identity Mapping and Certificate Name Filtering.

For example, the ServerSecurity exit in CTG on z/OS can be used to obtain the RACF identity mapped to the client certificate and subsequently use this identity as the connection identity. This allows the asserting of an identity without even sending a user ID with the connection request. Only an encrypted digital certificate is exchanged between WebSphere and CTG.

Example 4   ServerSecurity exit code snippet

```
// get the certificate distinguished names
Principal issuer = clientCert[0].getIssuerDN();
Principal subject = clientCert[0].getSubjectDN();
// validate issuer and subject’s Distinguished Name
RACFUserid racf = new RACFUserid(clientCert[0].getEncoded());
racfUserid = racf.getRACFUserid();
ECIRequest eci;
if (gatewayRequest instanceof ECIRequest) {
   eci = (ECIRequest) gatewayRequest;
   eci.Userid = racfUserid;
```

Note: Even though the ServerSecurity and ClientSecurity exits are specified in a WebSphere CICS J2C Connection Factory, CTG on the server can be configured to require these exists be specified for each connection request and prevent attempts to circumvent the security provided by these exits.

WebSphere and IMS connectivity

J2C connectivity from WebSphere to IMS requires IMS Connect to act as a protocol gateway between WebSphere and IMS.

- The client component of IMS Connect is IMS Connector for Java (IC4J), which is a Java application that runs within WebSphere on any platform and is installed as a resource adapter.
- IMS Connect converts a request from IC4J and forwards the request to IMS using IMS’s Open Transaction Management Architecture (OTMA) format.
- IMS Connect is a native z/OS application that listens on TCP/IP ports for remote requests over TCP/IP, or uses XCF for local z/OS requests.
- IMS Connect can authenticate a user ID/password with the local SAF database.
- IMS Connect can authorize a local z/OS client’s access to IMS Connect using the SAF’s FACILITY resource class.

The following list summarizes the characteristics of the security interactions between WebSphere and IMS Connect.

- Thread Identity support for connection identity for local connections uses the default principal mapping module.
- Thread Identity supported for remote connections requires a custom principal mapping module.
- J2C reauthentication is supported by IMS Connector for Java.
- Identity assertion of a WebSphere-provided identity to IMS is possible when:
  - IMS Connect authentication checking (RACF=N) is disabled, or
  - An IMS Connect trusted user exit is used to bypass authentication checks for trusted clients.

There is a security challenge if you disable IMS Connect authentication either totally or selectively. How can you ensure some level of trust between WebSphere and IMS Connect?

There are options available to address this challenge:

- Mutual authentication of client certificates between WebSphere and a remote IMS Connect.
- Enabling of SAF FACILITY resource checking of access between WebSphere on z/OS and a local IMS Connect.
- Include specific information in the connection request, which an IMS Connect trusted user exit can verify and use to bypass further authentication of the request.

Note: If authentication has not been disabled or bypassed, IMS Connect will still try to authenticate a request with an SAF regardless of the level of trust between WebSphere and IMS Connect.

WebSphere, IMS Connect, and IMS topologies

Now let us summarize WebSphere, IMS Connect, and IMS configuration topologies and security considerations for each.

Distributed WebSphere accessing a remote IMS Connect

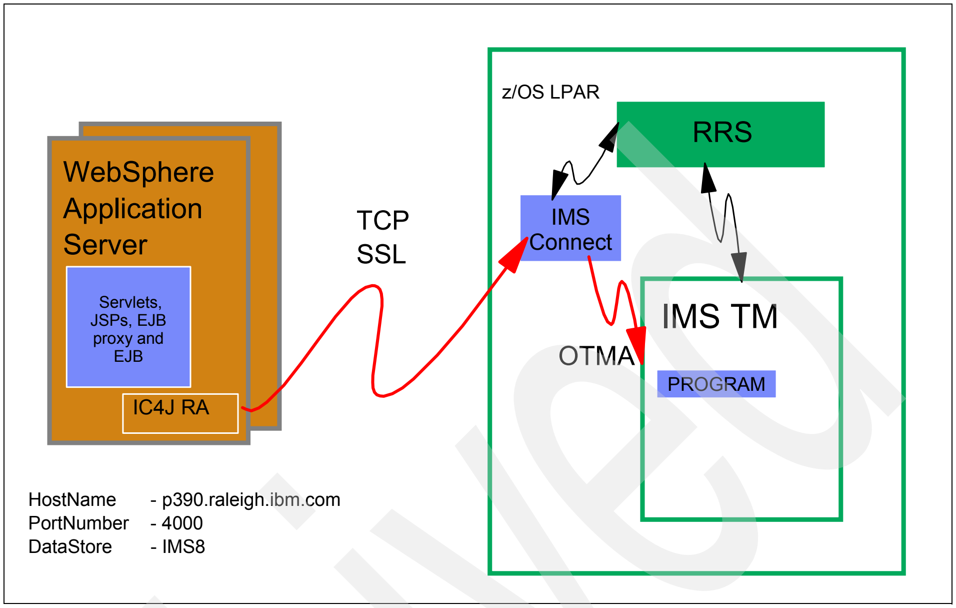

z/OS LPAR
RRS
WebSphere Application Server
Servlets, JSPs, EJB proxy and EJB
IC4J RA
TCP
SSL
IMS Connect
OTMA
IMS TM
PROGRAM
HostName - p390.raleigh.ibm.com
PortNumber - 4000
DataStore - IMS8

Figure 11 Distributed WebSphere accessing IMS Connect

Distributed WebSphere accessing IMS Connect has the following security properties:

- TCP and SSL protocols are supported for communications from a distributed WebSphere to IMS Connect.
- Access to IMS Connect can be controlled by mutual authentication of client certificates when using SSL communications.
- Identity assertion from WebSphere is possible if IMS Connect security is disabled or a trusted user exit is being used.
- The are no SAF resource controls for accessing IMS Connect remotely.
- Access to an IMS TM from IMS Connect is controlled by the RACF FACILITY resources class.

Note: Full 2PC is supported for IMS in a WebSphere global transaction.

WebSphere on z/OS accessing a local IMS Connect

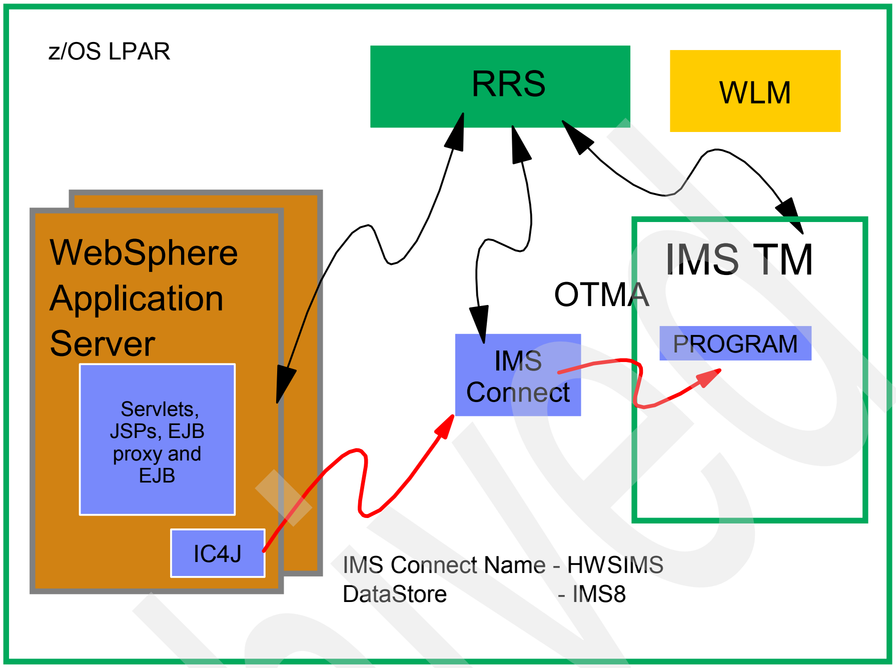

z/OS LPAR
RRS
WLM
WebSphere Application Server
Servlets, JSPs, EJB proxy and EJB
IC4J
OTMA
IMS TM
IMS Connect
PROGRAM
IMS Connect Name - HWSIMS
DataStore - IMS8

Figure 12 WebSphere on z/OS accessing a local IMS Connect

WebSphere on z/OS accessing IMS Connect “locally” has the following security properties:

- IMS Connector for Java (IC4J) accesses IMS Connect using Cross Coupling Facilities (XCF).
- Identity assertion from WebSphere is possible if IMS Connect security is disabled or a trusted user exit is being used.
- Access to IMS Connect from WebSphere on z/OS can be controlled by an SAF FACILITY resource.
- Access to IMS TM from IMS Connect can also be controlled by an SAF FACILITY resource.

Note: Full 2PC is supported for IMS in a WebSphere global transaction.

WebSphere on z/OS accessing a remote IMS Connect

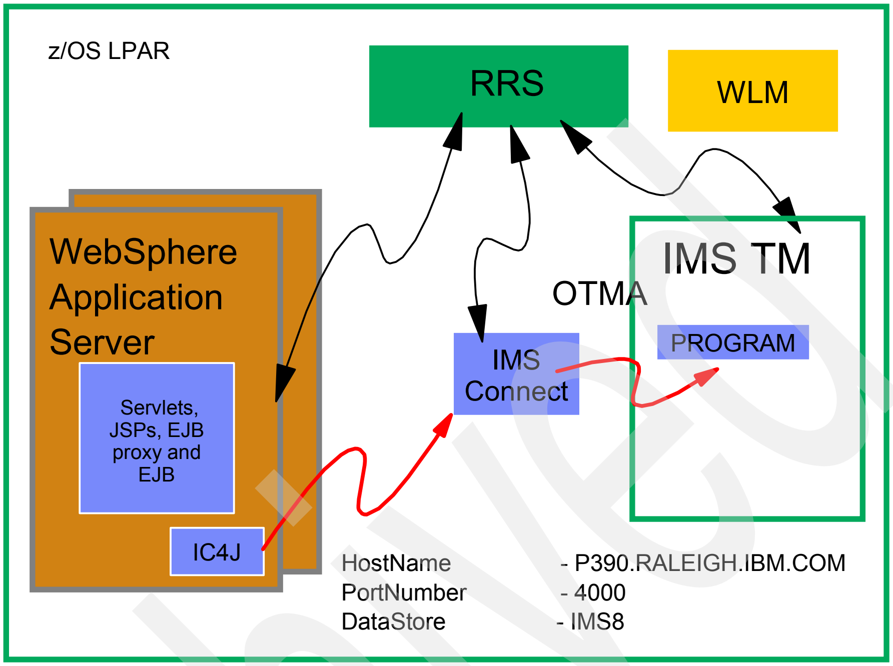

z/OS LPAR
RRS
WLM
WebSphere Application Server
Servlets, JSPs, EJB proxy and EJB
IC4J
IMS Connect
OTMA
IMS TM
PROGRAM
HostName - P390.RALEIGH.IBM.COM
PortNumber - 4000
DataStore - IMS8

Figure 13 WebSphere on z/OS accessing a remote IMS Connect

WebSphere on z/OS accessing IMS Connect remotely has the following security properties:

- TCP/IP or SSL protocols are supported for communications from IMS Connector for Java (IC4J) to IMS Connect.
- Access to IMS Connect can be controlled by mutual authentication of client certificates when using SSL communications.
- Identity assertion from WebSphere is possible if IMS Connect security is disabled or a trusted user exit is being used, and a custom principal mapping module is being used.
- Access to IMS TM from IMS Connect can also be controlled by an SAF FACILITY resource.

Note: Full 2PC is supported for IMS in a WebSphere global transaction.

Configuring IMS Connect for Security

Now we will explore IMS Connect and WebSphere IMS J2C Connection Factories configuration options and exits which can be used to enable SSL security between WebSphere and IMS Connect.

> **Note:** IMS Connect is a native MVS™ application with configuration parameters specified in PDS members and exits written in Assembler.

IMS Connect SSL options summary

IMS Connector for Java in the form of J2C Connection Factories on all platforms supports JSSE Keystores created with **`ikeyman`** and **`keytool`** commands. IMS Connector for Java V2.2.3, when installed in WebSphere on z/OS, can also support RACF Keyrings created with the RACDCERT command.

IMS Connect as a native MVS application can only support System SSL Keyrings created with the **`gskkyman`** command or RACF Keyrings created with the RACDCERT command.

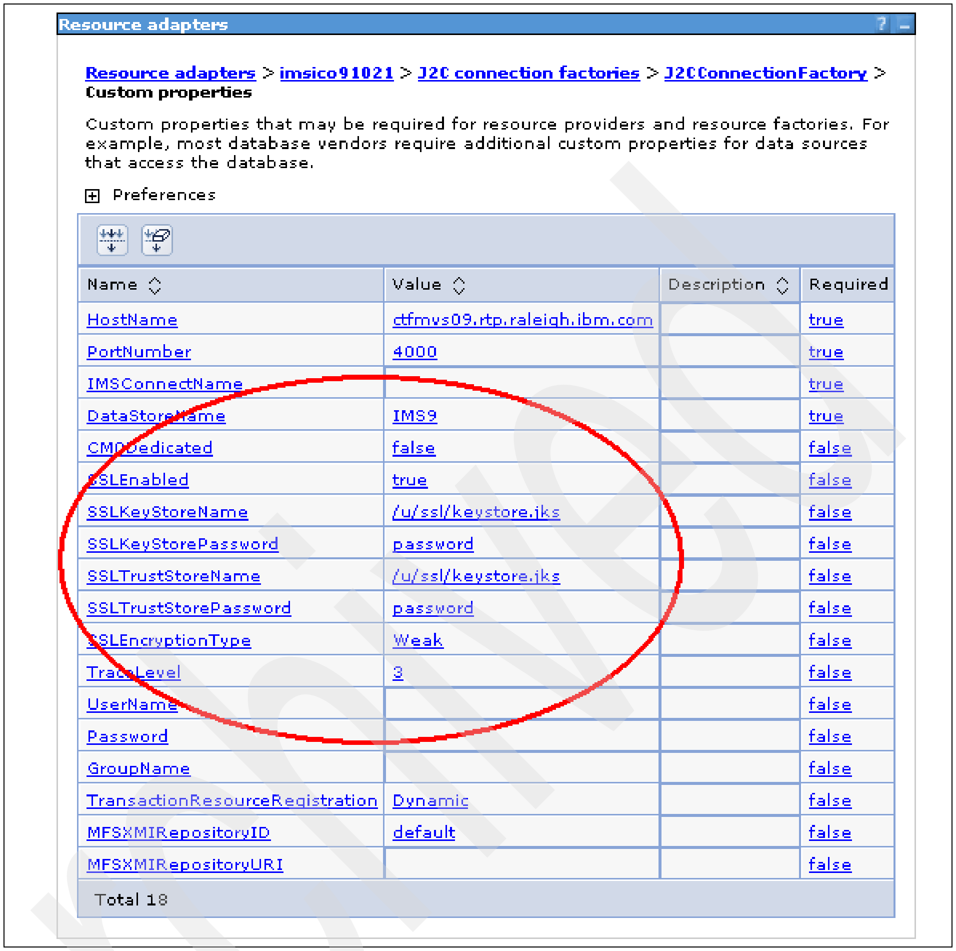
Example 5 shows the IMS Connect SSL configuration parameters we used during our testing. GSK_KEYRING_FILE specified a RACF Keyring and GSK_KEYRING_LABEL specified the Keyring label. Note that a password is not required since Keyring owners have implicit access to their Keyrings.

Example 5 IMS Connect SSL parameters

```
DEBUG_SSL=ON
GSK_CLIENT_AUTH_TYPE=GSK_CLIENT_AUTH_FULL_TYPE
GSK_SESSION_TYPE=GSK_SERVER_SESSION_WITH_CL_AUTH
GSK_PROTOCOL_SSLV2=GSK_PROTOCOL_SSLV2_ON
GSK_PROTOCOL_SSLV3=GSK_PROTOCOL_SSLV3_ON
GSK_PROTOCOL_TLSV1=GSK_PROTOCOL_TLSV1_ON
GSK_KEYRING_FILE=IMSKeyring
GSK_KEYRING_PW=
GSK_KEYRING_STASH_FILE=
GSK_KEYRING_LABEL=IMSConnect
GSK_V2_CIPHER_SPECS=6321
GSK_V3_CIPHER_SPECS=0906030201
```

IMS J2C Connection Factory - SSL custom properties

Configuring a WebSphere IMS J2C Connection Factory requires setting the SSLEnabled custom property to True and providing a JSSE keystore and truststore file name and password, and the encryption type; see Figure 14 on page 28.



Resource adapters > imsico91021 > J2C connection factories > J2CConnectionFactory > **Custom properties**

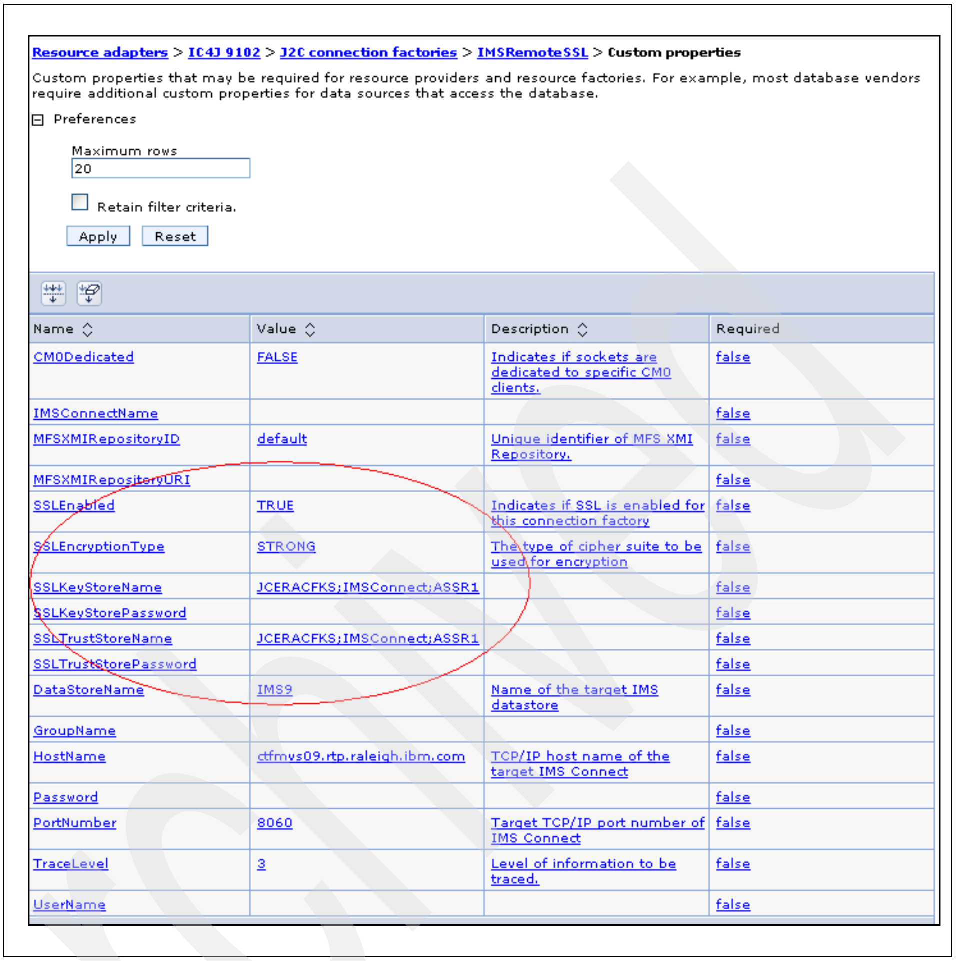
Custom properties that may be required for resource providers and resource factories. For example, most database vendors require additional custom properties for data sources that access the database.

Name	Value	Description	Required
HostName	ctfmvs09.rtp.raleigh.ibm.com		true
PortNumber	4000		true
IMSConnectName			true
DataStoreName	IMS9		true
CM0Dedicated	false		false
SSLEnabled	true		false
SSLKeyStoreName	/u/ssl/keystore.jks		false
SSLKeyStorePassword	password		false
SSLTrustStoreName	/u/ssl/keystore.jks		false
SSLTrustStorePassword	password		false
SSLEncryptionType	Weak		false
TraceLevel	3		false
UserName			false
Password			false
GroupName			false
TransactionResourceRegistration	Dynamic		false
MFSXMIRepositoryID	default		false
MFSXMIRepositoryURI			false

Total 18

Figure 14 IMS J2C Connection Factory JSSE custom properties

Figure 15 on page 29 shows the custom properties used with RACF Keyrings on z/OS. Note that the user ID specified in the truststore and keystore strings must be the user ID of the Servant region. Also note that a password is not required since access to a Keyring is implicit to its owner.



Resource adapters > **IC4J 9102** > **J2C connection factories** > **IMSRemoteSSL** > **Custom properties**

Custom properties that may be required for resource providers and resource factories. For example, most database vendors require additional custom properties for data sources that access the database.

⊟ Preferences

Maximum rows: 20

□ Retain filter criteria.

Name	Value	Description	Required
CM0Dedicated	FALSE	Indicates if sockets are dedicated to specific CM0 clients.	false
IMSConnectName			false
MFSXMIRepositoryID	default	Unique identifier of MFS XMI Repository.	false
MFSXMIRepositoryURI			false
SSLEnabled	TRUE	Indicates if SSL is enabled for this connection factory	false
SSLEncryptionType	STRONG	The type of cipher suite to be used for encryption	false
SSLKeyStoreName	JCERACFKS;IMSConnect;ASSR1		false
SSLKeyStorePassword			false
SSLTrustStoreName	JCERACFKS;IMSConnect;ASSR1		false
SSLTrustStorePassword			false
DataStoreName	IMS9	Name of the target IMS datastore	false
GroupName			false
HostName	ctfmvs09.rtp.raleigh.ibm.com	TCP/IP host name of the target IMS Connect	false
Password			false
PortNumber	8060	Target TCP/IP port number of IMS Connect	false
TraceLevel	3	Level of information to be traced.	false
UserName			false

Figure 15 IMS J2C Connection Factory SSL custom properties

IMS Connect trusted user support

IMS Connect has an exit for Java clients (HWSJAVA0) that can be configured to bypass authentication for trusted Java clients (for example, WebSphere).

When the exit determines that the request is from a trusted client (any request for datastore IMS8 by user ASSR1), further RACF checks by IMS Connect are bypassed; see Example 6.

Example 6   HWSJAVA0 sample for trusted user support

```
         CLC   OMUSR_DESTID(4),=C‘IMS8'  IS THE REQUEST FOR  IMS8
         BNE   NTRTUSER               CONTINUE, IF NOT
         CLC   OMSECUID(5),=C'ASSR1'     IS THE REQUEST FROM USER ASSR
         BNE   NTRTUSER                 CONTINUE, IF NOT
         WTO   'Trusted user flag set: userid ASSR1 and datastore HWS8'
         MVI   OMUSR_FLAG2,OMUSR_TRSTUSR INDICATE REQUEST FROM TRUSTED USER
NTRTUSER EQU   *
```

Application- versus container-managed security and IMS

During our testing we observed the behavior demonstrated in the following flowcharts for determining the connection identity during application- and container-managed security scenarios.

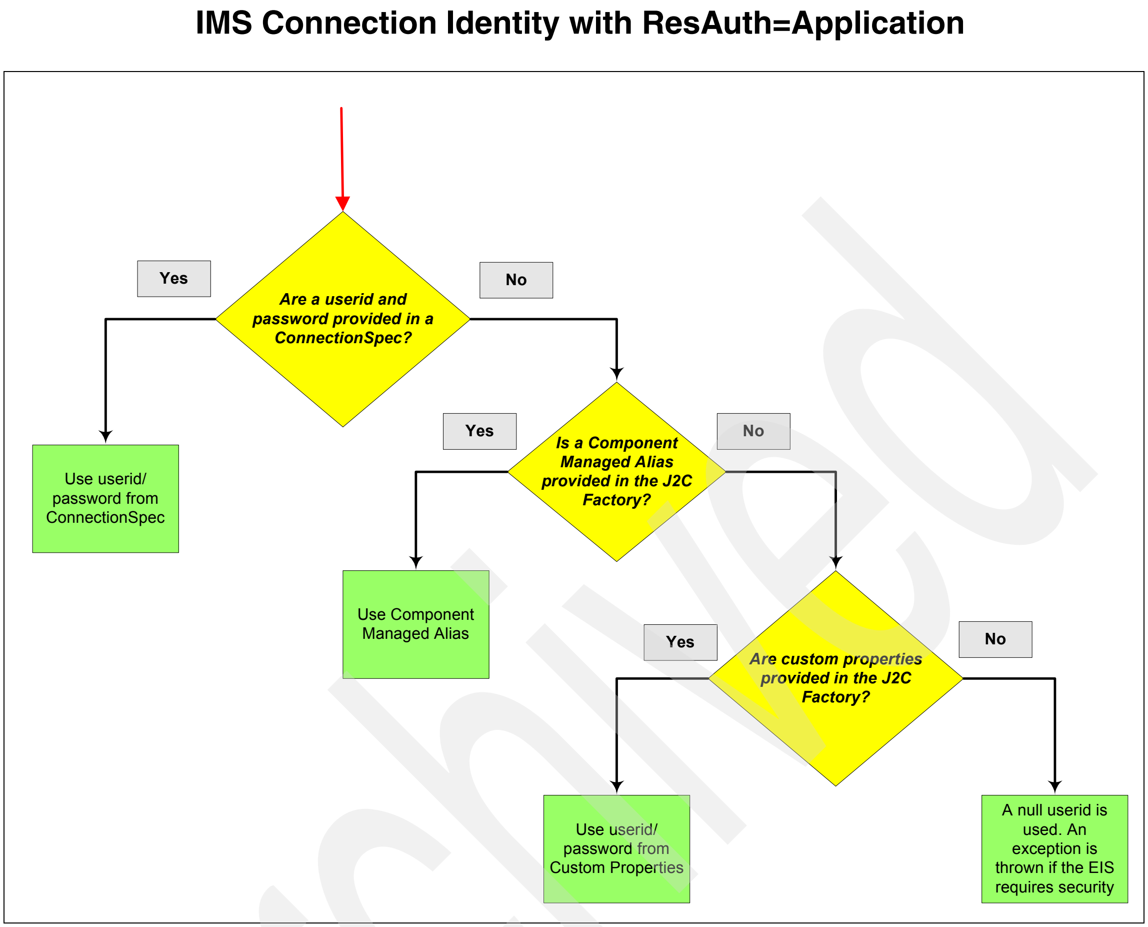
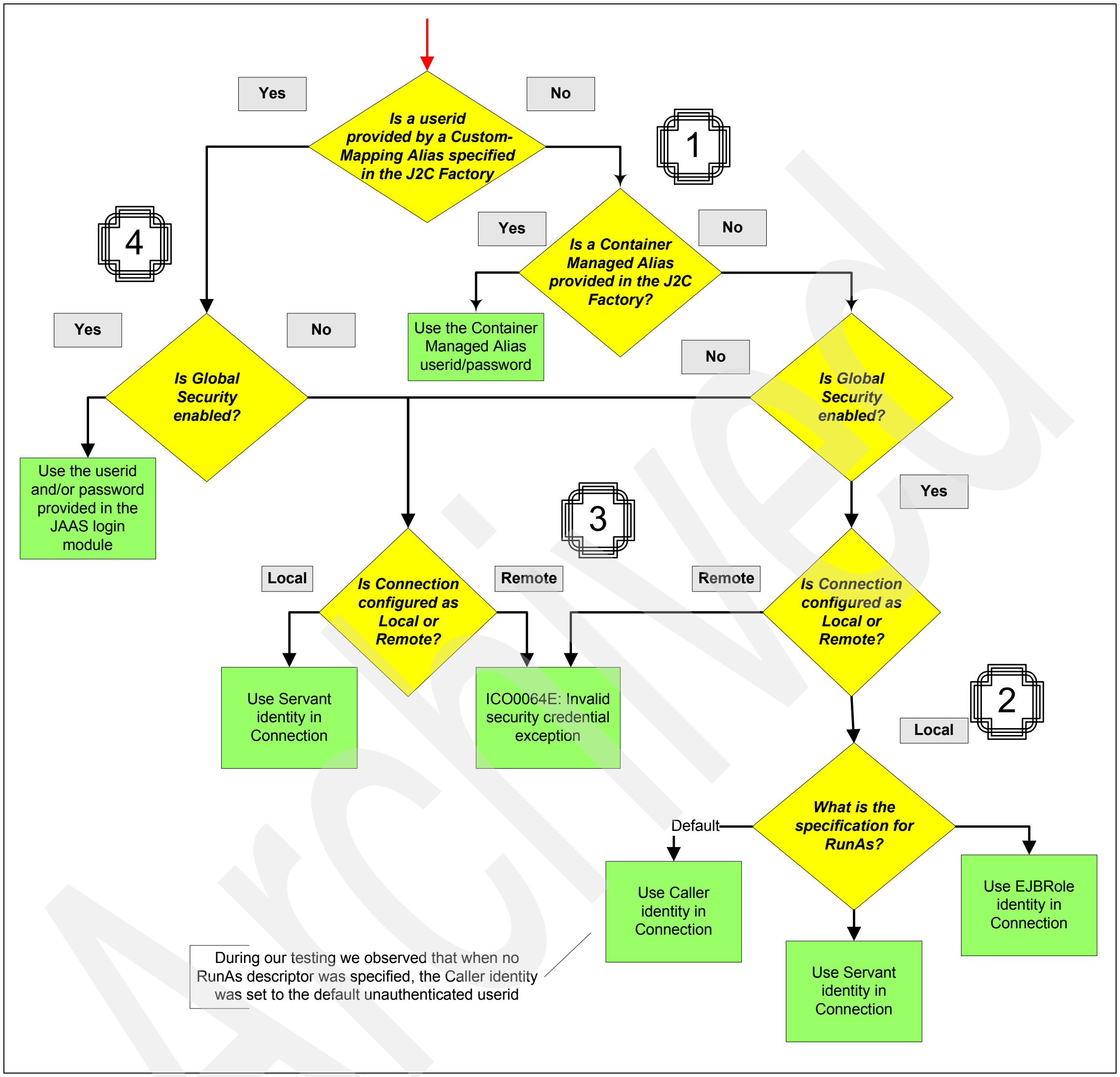

IMS Connection Identity with ResAuth=Application
Yes
Are a userid and password provided in a ConnectionSpec?
No
Use userid/ password from ConnectionSpec
Yes
Is a Component Managed Alias provided in the J2C Factory?
No
Use Component Managed Alias
Yes
Are custom properties provided in the J2C Factory?
No
Use userid/ password from Custom Properties
A null userid is used. An exception is thrown if the EIS requires security

Figure 16 IMS Connection Identity with application-managed security

IMS Connection Identity with ResAuth=Container


Is a userid provided by a Custom-Mapping Alias specified in the J2C Factory
Yes
No
1
Is a Container Managed Alias provided in the J2C Factory?
Yes
No
Use the Container Managed Alias userid/password
4
Yes
No
Is Global Security enabled?
Use the userid and/or password provided in the JAAS login module
No
Is Global Security enabled?
Yes
3
Local
Is Connection configured as Local or Remote?
Remote
Use Servant identity in Connection
ICO0064E: Invalid security credential exception
Remote
Is Connection configured as Local or Remote?
2
Local
What is the specification for RunAs?
Default
Use Caller identity in Connection
Use Servant identity in Connection
Use EJBRole identity in Connection

During our testing we observed that when no RunAs descriptor was specified, the Caller identity was set to the default unauthenticated userid

Figure 17 IMS Connectin Identity when Container Managed Security

Identity mapping and certificate name filtering with IMS

We wanted to assert an identity using only a digital certificate. We modified the trusted user exit (HWAJAVA0) (see Example 7 on page 33) to extend it to obtain the RACF identity mapped to the client certificate, and subsequently use this identity as the connection identity in requests to IMS.

Example 7   Identity Mapping Version of the HWSJAVA0 exit

```
         CALL IRRSIA00,(SAF_work_area,   Call RACF initACEE Service           X
               ALET, SAF_return_code,  SAF Return Code (output)               X
               ALET, RACF_return_code,RACF Return Code (output)               X
               ALET, RACF_reason_code, RACF Reason Code (output)              X
               Function_code,          Certificate Inquiry (x'08')            X
               Attributes,             Return X500 Name Pair (x'00400000')    X
               RACF_userid ,           RACF Userid (output)                   X
               ACEE_ptr,               n/a                                    X
               APPL_id,                n/a                                    X
               Password,               n/a                                    X
               Logstring,              n/a                                    X
               EXPREA_CERT,            Certificate (input)                    X
               ENVR_in,                n/a                                    X
               ENVR_out,               n/a                                    X
               OUTAREA,                n/a                                    X
               X500NamePair ,          X500 Name Pair (output)                X
               VarList),VL  Variable List
*     Validate issuer's and subject's Distinguished Names
         MVC  OMSECUID,RACF_userid
```

The team that wrote this Redpaper

This IBM Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Alex Louwe Kooijmans is a project leader at the International Technical Support Organization, Poughkeepsie Center. He joined IBM in 1986, and currently leads residencies to create Redbooks™ in the area of Java and WebSphere on z/OS. He also teaches workshops in this area. Alex is on his second assignment to the ITSO. Before joining the ITSO, he worked in various other roles, such as IT Specialist supporting customers in Europe getting started with WebSphere on various platforms, and Client IT Architect in the Financial Services Sector in The Netherlands. He has a lot of experience in Java and WebSphere on z/OS, and integrating WebSphere with DB2, MQ, CICS, and IMS. He has been working for IBM since 1986.

Mitch Johnson is a Senior Software Engineer in IBM Software Services for WebSphere (ISSW) at IBM Research Triangle Laboratory. His areas of expertise include enterprise connectivity and installation, configuration, and administration of WebSphere for z/OS and IMS, as well as CICS, TXSeries®, and DB2 on various platforms. He holds a Bachelor of Science degree in computer science from North Carolina State University in Raleigh, North Carolina.

Thanks to Richard M Conway of the International Technical Support Organization, Poughkeepsie Center for his contributions to this project.

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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.

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

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates 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. 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 IBM's application programming interfaces.

Send us your comments in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099, 2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

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

CICS Connection®
CICS®
DB2®
IBM®
IMS™
MVS™
Redbooks™
Redbooks (logo) ™
RACF®
TXSeries®
WebSphere®
z/OS®

The following terms are trademarks of other companies:

EJB, Java, JDBC, J2EE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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