

IBM WebSphere Application Server V4.0

J2EE Connector (J2C)



J2EE Connector: Topics

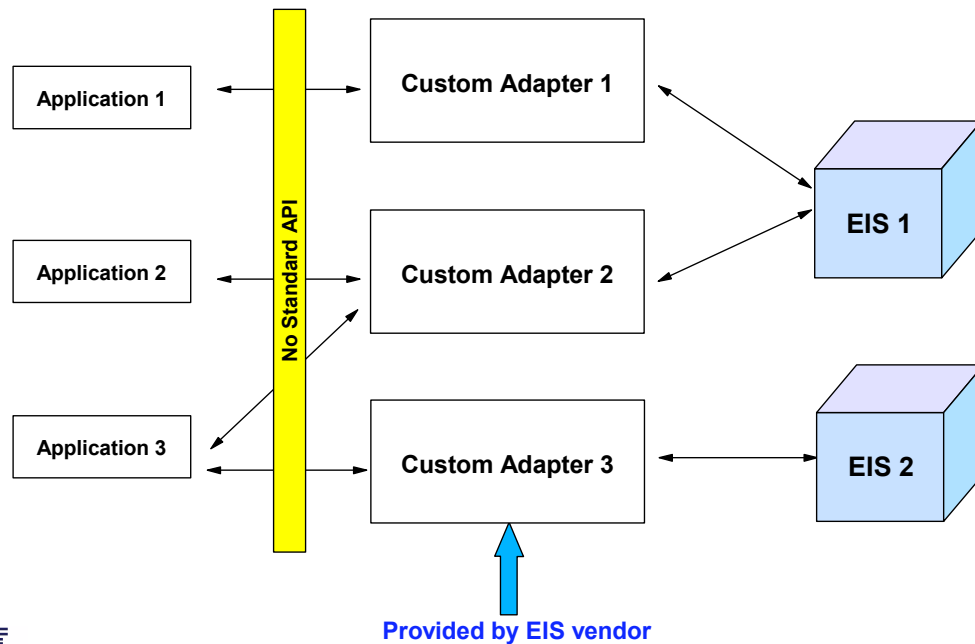


- Interaction with Enterprise Information Systems (EIS)
- J2C Architecture
- J2C Components
- WebSphere 4.0 support for J2C
- Sample code



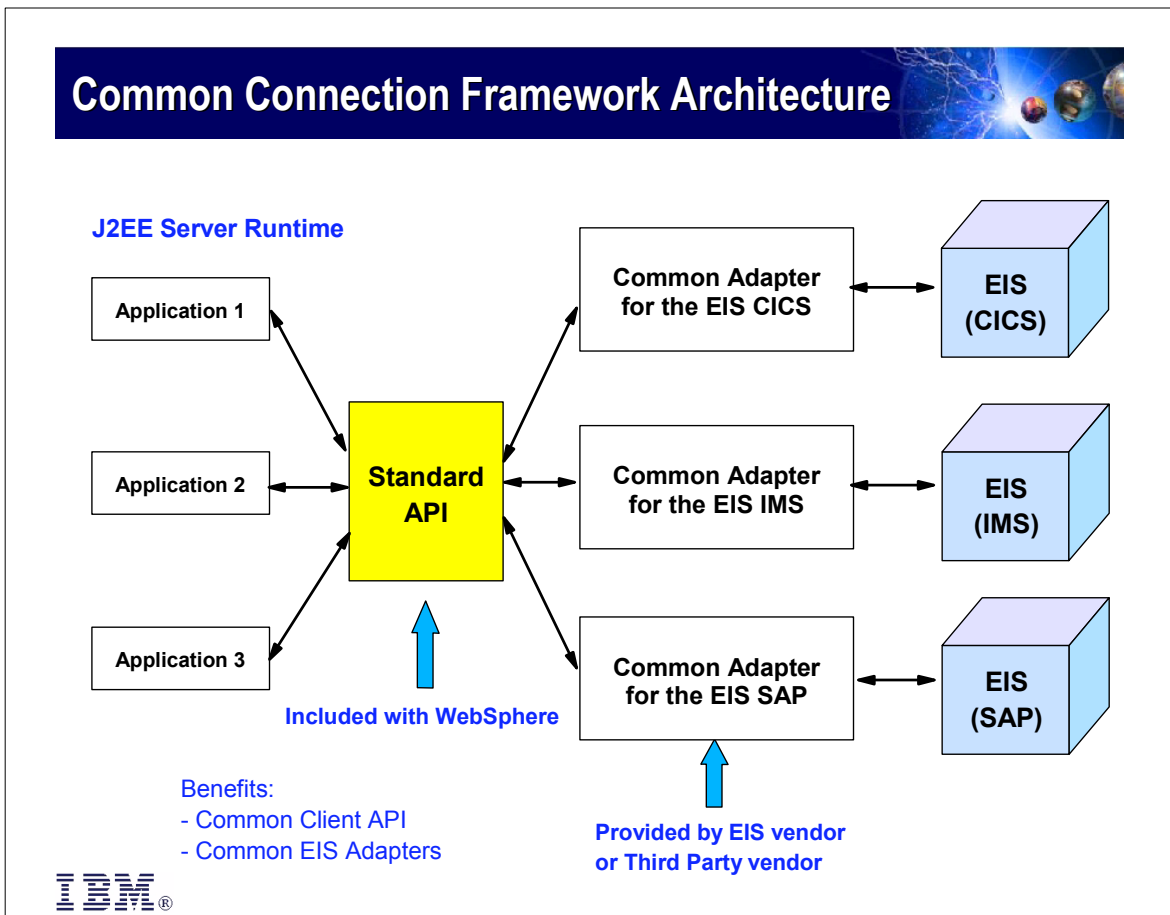
Connecting to EIS - Past

Examples of EIS:
CICS, IMS, SAP, Oracle ERP



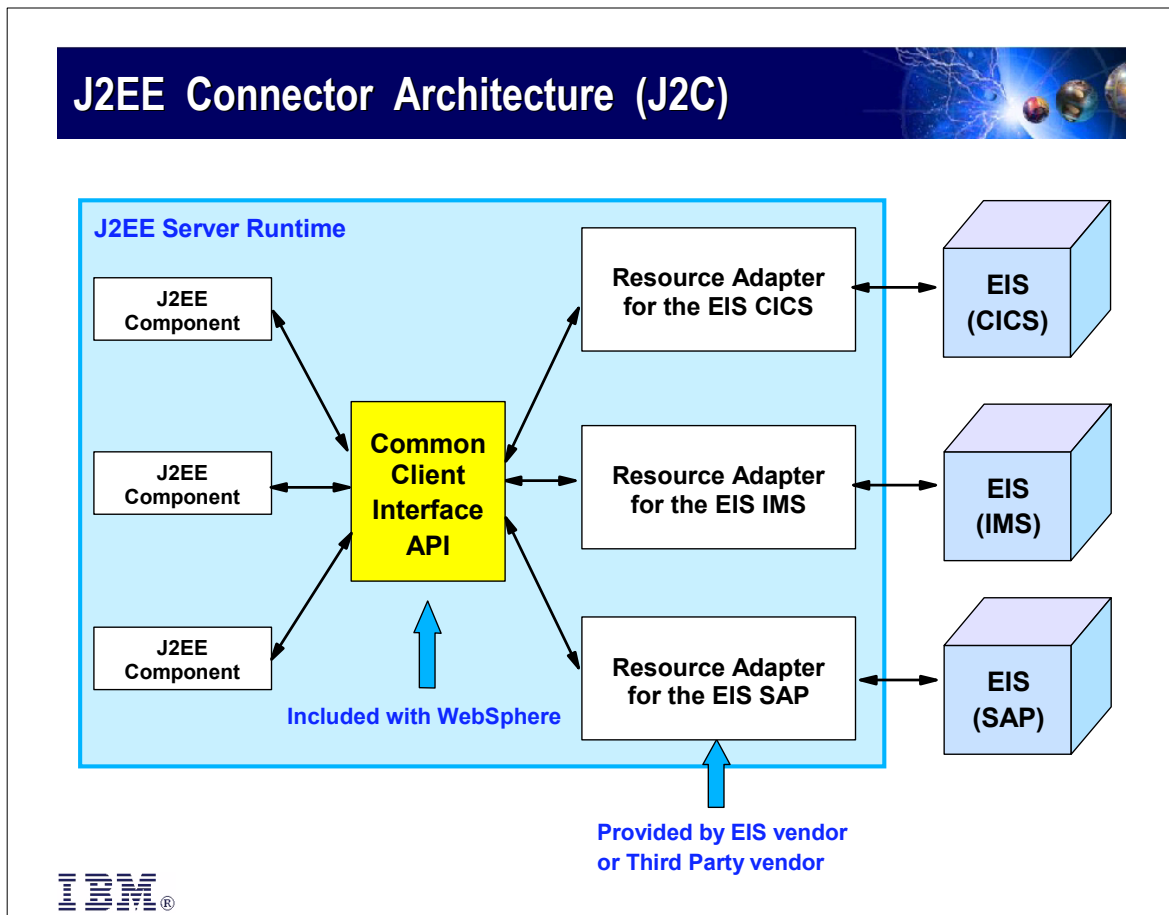
- ▶ Complexity of integrating heterogeneous EIS
 - ▶ No standard
 - ▶ Vendors use vendor-specific architecture to provide connectivity between application servers and EIS
- ▶ Acronyms Demystified:
 - ▶ EIS - Enterprise Information System
 - ▶ CICS - "Customer Information Control System"
 - ▶ IMS - "Information Management System" - IBM's first database system.
 - ▶ SAP - "Systems, Applications and Products in Data Processing"
 - ▶ ERP - "Enterprise Resource Planning"

Common Connection Framework Architecture



- ▶ "The original CCF was an IBM technology developed by the Visual Age for Java group, subsequently incorporated into the Java standards" - as J2C.
- ▶ www-4.ibm.com/software/ts/cics/library.ts390/vol1iss11/artical7.html

J2EE Connector Architecture (J2C)



- ▶ The J2EE Connector Architecture, or J2C is not part of the J2EE version 1.2 specification. J2C is part of the J2EE version 1.3 draft.
 - ▶ This is one of the added value features of WebSphere 4.0.
- ▶ J2C is similar to the Common Connector Framework, implemented for the Java Platform.
- ▶ J2C provides specialized access to Enterprise Resource Planning (ERP) and mainframe systems such as IBM's CICS and IMS.
 - ▶ J2C simplifies access to diverse back end Enterprise Information Systems, or EIS, by providing a common client interface API.
 - ▶ Applications can be coded to this single interface, rather than having unique interfaces for each back end system.
- ▶ The link from the common client interface is a Resource Adapter provided by the back end system vendor, or a third party.

Java 2 Connector



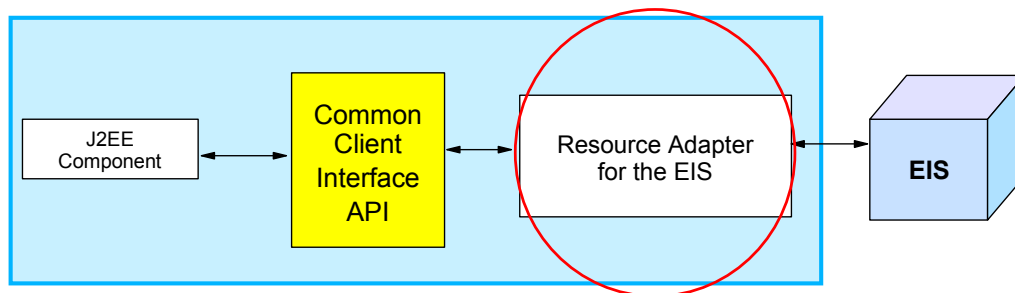
- J2C 1.0 is part of J2EE 1.3 spec
 - ▶ Implemented in WebSphere V4.0 AE as Technology Preview
- A standard architecture for connecting J2EE components to heterogeneous EIS
- Benefits:
 - ▶ Simplifies the integration of diverse EIS
 - ▶ Each EIS requires just one implementation of the J2EE Connector Technology.
- J2C Components:
 - ▶ Resource Adapter
 - Supplied by EIS or third party vendor
 - ▶ Common Client Interface
 - Packaged with WebSphere V4.0



- ▶ Same implementation is portable across all compliant J2EE servers.
- ▶ EIS vendors no longer need to customize for each application server

J2C: Resource Adapter

- Provides connectivity between J2EE Component and EIS
- Plugs into an application server
- Collaborates with the Application server to provide important services
 - ▶ connection, transactions, security, etc.
- Stored in a .rar (Resource Adapter archive) file

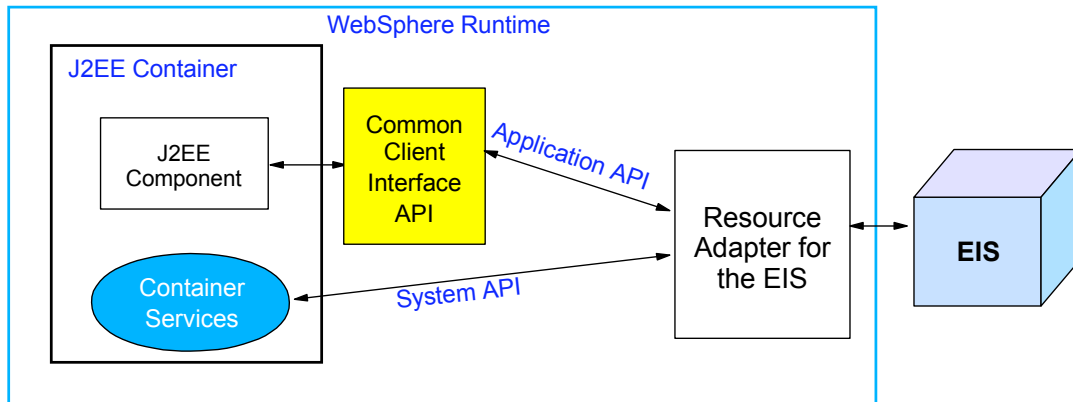


IBM®

- ▶ A resource adapter is similar to a JDBC driver.
 - ▶ Through a standard API, the adapter provides access to a resource outside of the J2EE server.
 - ▶ It is written not by business application developers, but by middleware software developers.
- ▶ Example of EISs include ERP (Enterprise Resource Planning), mainframe transaction processing, database systems, and legacy application not written in the Java.
- ▶ Each EIS requires just one implementation of the J2EE Connector Technology.
 - ▶ Same implementation is portable across all compliant J2EE servers.
 - ▶ EIS vendors no longer need to customize for each application server

J2C: Resource Adapter Details

- Two types of API implemented by a Resource Adapter
 - ▶ Application API
 - defines the API through which a J2EE component accesses the EIS.
 - ▶ System API links the Resource Adapter to J2EE server services
 - Connection, transaction, security, etc.

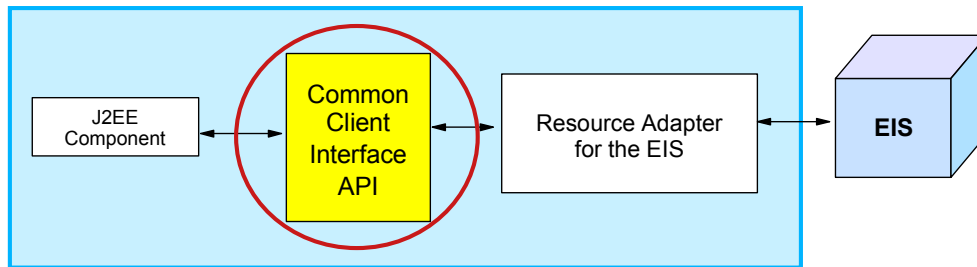


IBM®

- ▶ A resource adapter is similar to a JDBC driver.
- ▶ Through a standard API, the adapter provides access to a resource outside of the J2EE server.
 - ▶ It is written not by business application developers, but by middleware software developers.
- ▶ System Contract for Resource Adapter
 - ▶ Connection Management Contract
 - ▶ Allows App server to do connection pool to an underlying EIS
 - ▶ Transaction Management Contract
 - ▶ Allows App Server to use a transaction manager to manage transactions across multiple resource managers
 - ▶ Security Contract
 - ▶ Supports for a secure application environment, which reduces security threats to EIS

J2C: Common Client Interface (CCI)

- CCI API provides access from J2EE components to an EIS through a Resource Adapter
 - ▶ Simplifies writing code to connect to an EIS's data store
 - ▶ J2EE components interact with CCI only - not directly with Resource Adapter



IBM®

- ▶ Supported by most J2EE compliant EIS vendors
 - ▶ It will become the standard for J2EE enterprise application connectivity.

IBM WebSphere Application Server V4.0

WebSphere V4.0 J2EE Connector



WebSphere V4.0 J2C: Support



- Basic runtime support based on Proposed Final Draft of the J2EE Connector Architecture spec
- Installation, configuration, management of Resource Adapters
 - ▶ Through Admin Console GUI
- Installation of J2C CCI layer runtime
 - ▶ Separately installable entity
 - <http://www7b.boulder.ibm.com/wsdd/downloads/jca.html>
- Support use of IBM one phase commit (1 PC) Resource Adapters
 - ▶ HOD, CICS, IMS - available in separate packages
- Support use of 3rd party 1PC Resource Adapters
 - ▶ SAP, JDEdwards, PeopleSoft - available in separate packages



WebSphere V4.0 J2C: Support ..



- Connection Management
 - ▶ ConnectionManager implementation
 - ▶ ConnectionEventListener implementation
 - ▶ Pool Manager implementation

- Transaction Management
 - ▶ Satisfied by JTA implementation
 - ▶ Connection sharing will be assumed within global transactions
 - ▶ CCI local transactions

- Security Management

- Logging and tracing support for Resource Adapters



WebSphere V4.0 J2C: Configuration



- Two configurable objects in the Admin Server
 - ▶ Resource Adapter
 - Represents the library that supplies implementation code for connecting applications to some EIS
 - ▶ Connection Factory
 - Represents the configuration of a specific connection to the EIS supported by the Resource Adapter
- WebSphere Administrator role:
 - ▶ Defines the Resource Adapter
 - ▶ One or more Connection Factories associated with the Resource Adapter
- Application code:
 - ▶ Requests a connection from the Connection Factory through JNDI lookup
 - ConnectionFactory connects the Application to the Resource Adapter



WebSphere V4.0 J2C: Admin Console GUI

The screenshot displays the WebSphere Advanced Administrative Console interface. On the left, a tree view shows the hierarchy of the administrative domain, with the 'CICS Resource Adapter' under 'J2C Resource Adapters' highlighted. A red circle highlights the 'JDBC Providers' folder, which contains 'Sample DB Driver' and 'Data Sources'. The right pane shows the configuration details for the selected resource adapter, with the 'Connections' tab active. The configuration includes the name '*CICS Resource Adapter', a description field, and an archive file name of '*cicseci.rar'.

WebSphere Advanced Administrative Console

Console View Tools Help

WebSphere Administrative Domain

- Virtual Hosts
- ServerGroups
- Nodes
 - NodeName
 - Application Servers
 - Default Server
 - MyBank Server
 - Installed EJB Mod
 - Installed Web Mo
 - Generic Servers
 - Enterprise Applications
 - sampleApp
 - EJB Modules
 - Web Modules
 - MyBank Application
 - EJB Modules
 - Web Modules
 - Resources
 - JDBC Providers
 - Sample DB Driver
 - Data Sources
 - JavaMail Sessions
 - URL Providers
 - J2C Resource Adapters
 - CICS Resource Adapter

J2C Connection Factories

Name

General | Connections | Advanced | Nodes

Name: *CICS Resource Adapter

Description:

Archive file name: *cicseci.rar



WebSphere V4.0 J2C: RA Properties GUI

Create a new
Resource
Adapter

Resource
Adapter
Archive File

J2C Resource Adapter Properties

General | Connections | Advanced | Nodes

Name: *CICS Resource Adapter

Description:

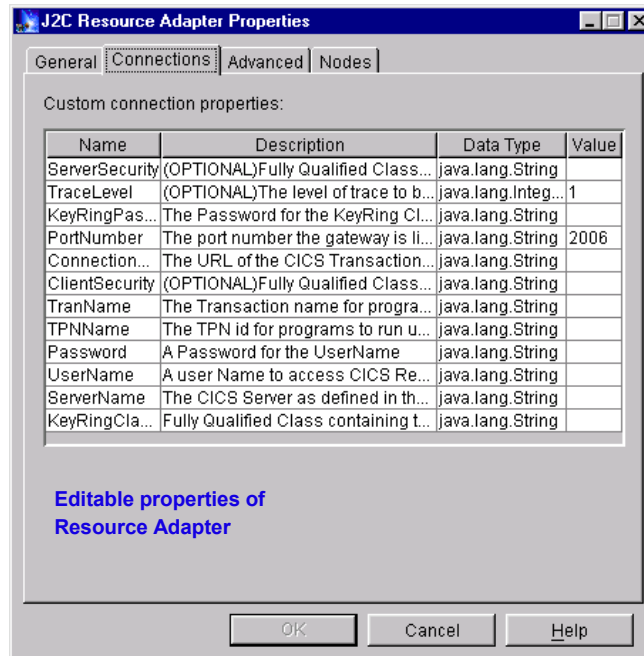
Archive file name: cicsecl.rar

OK Cancel Help

cicsecl.rar (Resource Archive file for cics remote gateway connector)



WebSphere V4.0 J2C: RA Properties GUI...

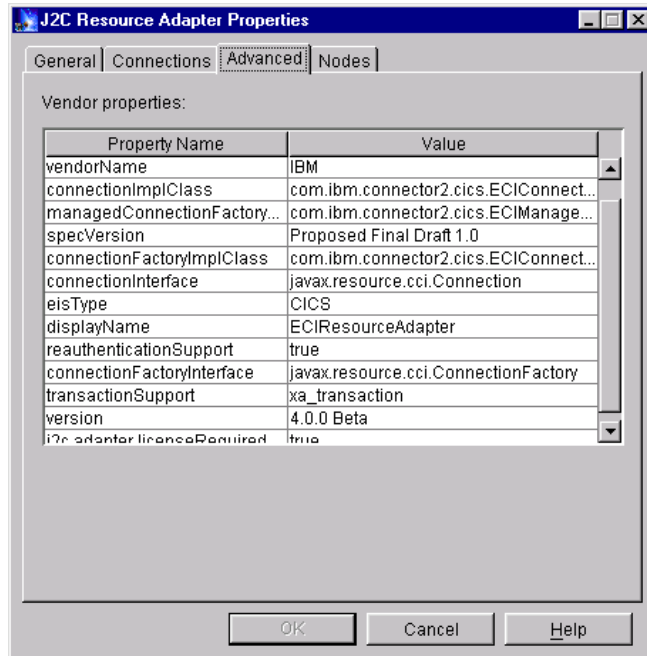


- ▶ These properties will be propagated to the Connection Factory objects as they are created. They are not editable at this point, but they will be in the Connection Factory.

WebSphere V4.0 J2C: RA Properties GUI...



**Read only attributes.
Indicates the
capabilities of
Resource Adapter**



WebSphere V4.0 J2C: ConnectionFactory GUI



J2C Connection Factory Properties

General | Advanced | Connections

Name: *CICS Connection Factory

JNDI binding path: ejbs/myECICConnection

Description:

J2C resource adapter: *CICS Resource Adapter

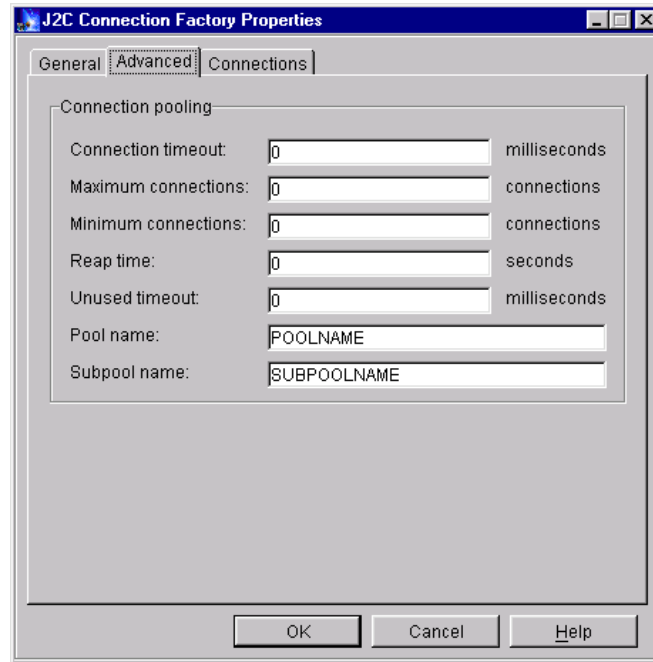
OK Cancel Help

JNDI lookup used by
Application
code to access EIS resource



WebSphere V4.0 J2C: ConnectionFactory GUI ...

Connection pooling attributes



The screenshot shows the 'J2C Connection Factory Properties' dialog box with the 'Advanced' tab selected. The 'Connection pooling' section contains the following fields:

Connection timeout:	0	milliseconds
Maximum connections:	0	connections
Minimum connections:	0	connections
Reap time:	0	seconds
Unused timeout:	0	milliseconds
Pool name:	POOLNAME	
Subpool name:	SUBPOOLNAME	

Buttons: OK, Cancel, Help



► Connection Timeout (J2C Connection Factory)

Number of milliseconds after which a connection request is determined to have timed out and a `ResourceAllocationException` is thrown. The wait might be necessary if the maximum value of connections has been reached (`MaxConnections`). This value has no meaning if the maximum connections property has not been set.

If the connection timeout is set to `DEFAULT_CONNECTION_TIMEOUT`, the `ResourceAllocationException` is thrown immediately after the pool manager determines that the maximum number of connections has been used. If the connection timeout is set to `DISABLE_CONNECTION_TIMEOUT`, the pool manager waits until a connection can be allocated. (In other words, it waits until the number of connections falls below the maximum connections).

► Maximum Connections (J2C Connection Factory)

The maximum number of managed connections that can be created by a particular `ManagedConnectionFactory`. After this number is reached, no new connections are created and either the requester waits or the `ResourceAllocationException` is thrown. If maximum connections is set to `DEFAULT_MAX_CONNECTIONS`, the number of connections can grow indefinitely.

► Minimum Connections (J2C Connection Factory)

The minimum number of managed connections to maintain. If this number is reached, the garbage collector will not discard any managed connections. Note, if the actual number of connections is lower than the value specified by the minimum connections settings, no attempt will be made to increase the number of connections to the minimum.

► Reap Time (J2C Connection Factory)

Number of seconds between runs of the garbage collector. The garbage collector discards all connections that have been unused for the value specified by the unused timeout.

To disable the garbage collector, set the reap time to `DEFAULT_REAP_TIME`. Another way to disable the garbage collector is to set the unused timeout to `DEFAULT_UNUSED_TIMEOUT`.

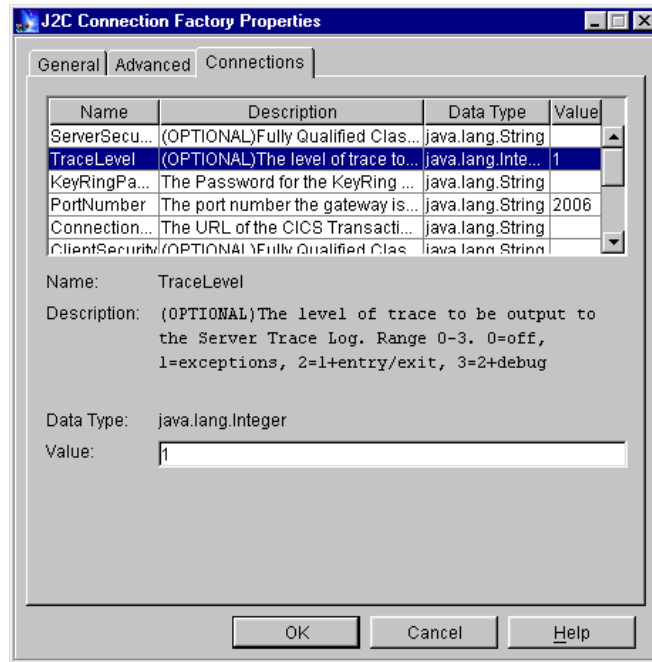
► Unused Timeout (J2C Connection Factory)

Number of milliseconds after which an unused connection is discarded. Setting this value to `DEFAULT_UNUSED_TIMEOUT` disables the garbage collector.

► Pool Name (J2C Connection Factory)

The name used by the pool manager to group managed connections created by different `ManagedConnectionFactory`s. The pool name, together with the subpool name, is used to provide usage statistics for billing, licensing, and other such activities.

WebSphere V4.0 J2C: ConnectionFactory GUI ...



WebSphere V4.0: J2C Future Direction



■ Future Directions

- ▶ Final Implementation of J2EE/CA 1.0 spec
- ▶ Two phase commit connectors
- ▶ J2EE 1.3 and EJB 2.0 enhancements



Appendix: CCI Code Sample



J2C: Interaction with CCI

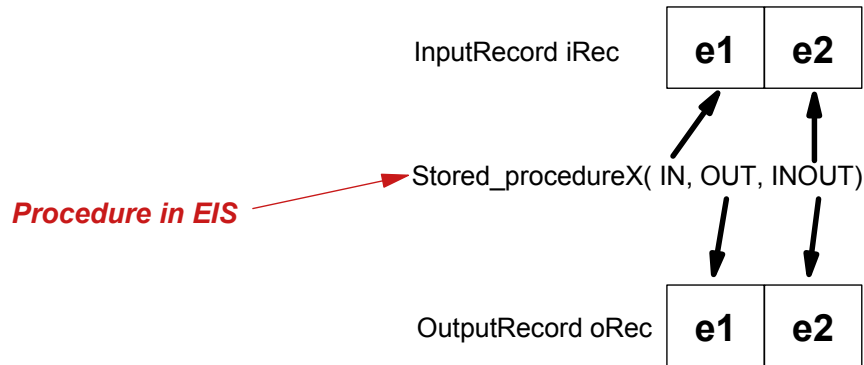


- To access an EIS from a J2EE component:
 - ▶ Lookup the **ConnectionFactory** for the resource adapter
 - ▶ Create a **Connection** object to the EIS using **ConnectionFactory** object.
 - Connection object represents the actual connection to the EIS
 - Used for subsequent interactions with the EIS
 - Connection properties (like user name, password, etc.) are passes via **ConnectionSpec** object
 - ▶ For each operation needed on EIS, create an **Interaction**, defined by an **InteractionSpec**
 - InteractionSpec specifies EIS data and operations
 - ▶ Create **Record** instances for transferring data into or out of the EIS function
 - ▶ Perform the desired function (**Interaction.execute(...)**)
 - ▶ Process output and close the connection



J2C: CCI Input/Output Record

- Steps to pass Input and Output parameters to EIS function:
 - ▶ Create InputRecord to map the IN and INOUT parameters
 - ▶ OutputRecord maps the OUT and INOUT parameters.
 - ▶ Each element of an input or output record corresponds to a parameter of the EIS function in the same order



J2C: CCI Sample Code - get Connection



```
import javax.resource.cci.*;
import javax.resource.*;
```

// JNDI lookup on ConnectionFactory

```
InitialContext ic = new InitialContext();
cf = (ConnectionFactory) ic.lookup("java:comp/env/eis/EISConn");
```

// Create Connection object

// Used for subsequent interactions with the EIS

```
ConnectionSpec spec = new CciConnectionSpec(user, password);
con = cf.getConnection(spec);
```

Properties passed to Connection via Spec

Connection Object - connection to EIS



J2C: CCI Sample Code -Interact with EIS



```
// Create Interaction and InteractionSpec objects For each operation on EIS, create Interaction
Interaction ix = con.createInteraction();
CciInteractionSpec iSpec = new CciInteractionSpec();
iSpec.setSchema(user);
iSpec.setCatalog(null);
iSpec.setFunctionName("COUNTCOFFEE");

// Create Input record object Specifies all Input and InputOutput data
RecordFactory rf = cf.getRecordFactory();
IndexedRecord iRec = rf.createIndexedRecord("InputRecord");

// Execute (will execute the proc COUNTCOFFEE in the EIS)
Record oRec = ix.execute(iSpec, iRec);
Output Execute

// process Output and Close connection
Iterator iterator = ((IndexedRecord)oRec).iterator();
...
Iterate thru' output
```



J2C: Transaction options



- Container Managed Transaction
 - ▶ Global and can span multiple resource managers

- Bean Managed Transaction
 - ▶ Must handle its own transaction
 - ▶ Use LocalTransaction interface methods to handle local transactions
 - ▶ Managed by the EIS Resource Manager
 - ▶ Within your code:
 - Obtain a reference to a local transaction context
 - ◆ `LocalTransaction transaction = con.getLocalTransaction();`
 - Use the Transaction API: `begin()`, `end()`, `commit()`, `rollback()`...

