IBM Software Group

Using TCP/IP Based Applications with WebSphere Message Broker

David Brickell (dmbricke@us.ibm.com)
Minsung Byun (mbyun@us.ibm.com)
Vivek Grover (vgrover@us.ibm.com)
WebSphere Message Broker Level 2 Support
16 November 2011

WebSphere® Support Technical Exchange

ON DEMAND BUSINESS

Agenda

- TCP/IP basics
- WMB TCP/IP transport
- TCP/IP nodes
- Common TCP/IP scenarios
- Problem diagnosis
- Common problems

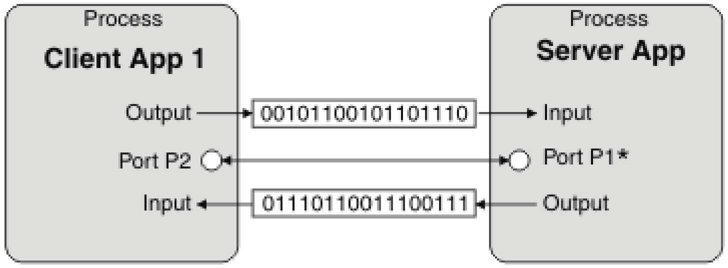
TCP/IP basics

- Transmission Control Protocol / Internet Protocol
- Internet Protocol Suite consisting of a set of protocols
- Contains different layers where each layer is responsible for certain operations (example: application layer, transport layer, internet layer, link layer)
- Each layer builds upon the layer below it, adding new functionality
- Works with data as IP packets
- Works on client – server principle
- Stream based protocol

TCP/IP basics

Basic working:

- If an application wants to communicate with another via TCP, it sends a communication request
- This request must be sent to an exact address
- When the receiving application accepts the communication request, a handshake is established
- After the "handshake" between the two applications, TCP will set up a "full-duplex" communication between the two applications



TCP/IP basics

- Connects applications using raw TCP/IP sockets
- Transfer of data is bidirectional between client and server

Advantages:

- Quick and easy to configure
- Maintains sequence of data
- Prevents loss of data

Limitations:

- Non-transactional and non-persistent
- No built-in security
- No standard way of signaling the start and end of message

WMB TCP/IP Transport

- By using MQ as its transport mechanism, the mentioned limitations can be avoided
- Applications using raw TCP/IP sockets can be easily integrated to use WMB TCP/IP nodes
- Available simple port to port communication between client and server
- A server can accept multiple connections from other client applications and these connections can be in the same DataFlowEngine or different DataFlowEngines
- Connection requested by client application processes (single or multiple) can connect to the same server application
- The client and server ends of the connection are identical and both can perform the same operations

WMB TCP/IP Transport

Basic working:

- The server application listens on a local port (on the computer that is running the application) for requests for connections to be made by a client application.
- The client application requests a connection from the server port, which the server then accepts.
- When the server accepts the request, a port is created on the client computer and is connected to the server port.
- A socket is created on both ends of the connection, and the details of the connection are encapsulated by the socket.
- The server port remains available to listen for further connection requests

WMB TCP/IP Transport

- An execution group process contains the connection manager, which makes the connections and so has minimum and maximum number of connections
- An execution group can have only one server node flow as it uses a specific port
- Multiple client node flows can be deployed to an execution group

WMB TCP/IP Transport

- Nodes require hostname and port to initiate the connection :
  - Configuration on the node
  - Configurable service
- The connection manager is created when the first TCP/IP node flow is deployed The connection manager is deleted when the last TCP/IP node flow is removed
- Each connection has an input stream and an output stream, both of which have two main states within the connection manager: available and reserved

TCP/IP nodes

- TCPIPClientInput node

TCPIP Client Input

- TCPIPClientOutput node

TCPIP Client Output

- TCPIPClientReceive node

TCPIP Client Receive

- TCPIPServerInput node

TCPIP Server Input

- TCPIPServerOutput node

TCPIP Server Output

- TCPIPServerReceive node

TCPIP Server Receive

TCP/IP nodes

- TCPIPClientInput: The TCPIPClientInput node is used to create a client connection to a raw TCPIP socket, and receive data over that connection.

- TCPIPClientOutput: The TCPIPClientOutput node is used to create a client connection to a raw TCPIP socket, and send data over that connection to an external application.

- TCPIPClientReceive: The TCPIPClientReceive node is used to receive data over a client TCP/IP connection

TCP/IP nodes

- TCPIPServerInput: The TCPIPServerInput node is used to create a server connection to a raw TCPIP socket, and to receive data over that connection

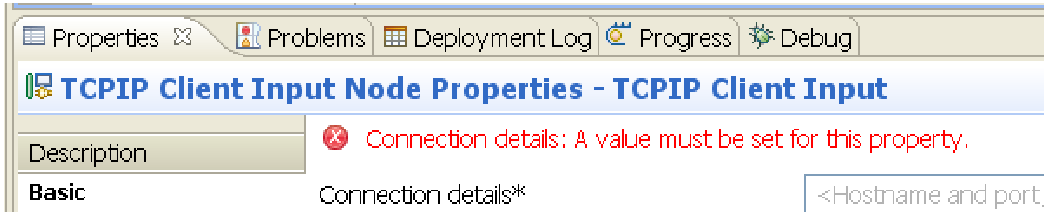
- TCPIPServerOutput: The TCPIPServerOutput node is used to create a server connection to a raw TCPIP socket, and send data over that connection to an external application

- TCPIPServerReceive: The TCPIPServerReceive node is used to receive data over a server TCP/IP connection

TCP/IP nodes

TCP/IP Node Configuration

- Basic: Connection details
  - Client: Hostname and port or configurable service



Properties | Problems | Deployment Log | Progress | Debug

TCPIP Client Input Node Properties - TCPIP Client Input

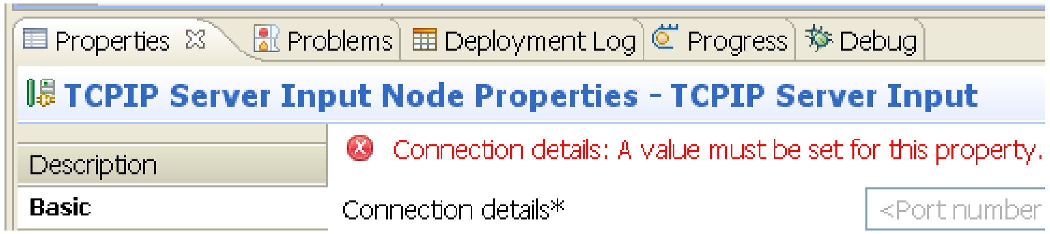
Description

Basic

Connection details: A value must be set for this property.

Connection details* <Hostname and port

  - Server: Port number or configurable service



Properties | Problems | Deployment Log | Progress | Debug

TCPIP Server Input Node Properties - TCPIP Server Input

Description

Basic

Connection details: A value must be set for this property.

Connection details* <Port number

- Other options: Parsing, Close connection, Request properties, and instances

Common TCP/IP scenarios

- Scenarios may be chosen based on user's requirements. Some examples include:
  - Connect existing TCP/IP client applications with MQ using the TCPIPServerInput,TCPIPServerReceive, and TCPIPServerOutput nodes
  - Connect existing applications to TCP/IP server programs, using the TCPIPClientInput,TCPIPClientReceive, and TCPIPClientOutput nodes

Common TCP/IP scenarios

- Scenarios may be chosen based on reservation/availability of connections. Some examples include:
  - Configuring a server socket so that connections expire after a specified time
  - Configuring a client socket to make 100 connections at deployment or startup time
  - Configuring a TCPIP client node to dynamically call a port
  - Configuring a TCPIP server receive node to wait for data on a specified port
  - Synchronous/Asynchronous request/reply scenarios
  - TCP/IP nodes and SSL

Common TCP/IP scenarios

The reserve mechanism provides the following options:

Leave unchanged: Default value indicating stream is left available. For example, when you are moving data from an input stream to a file

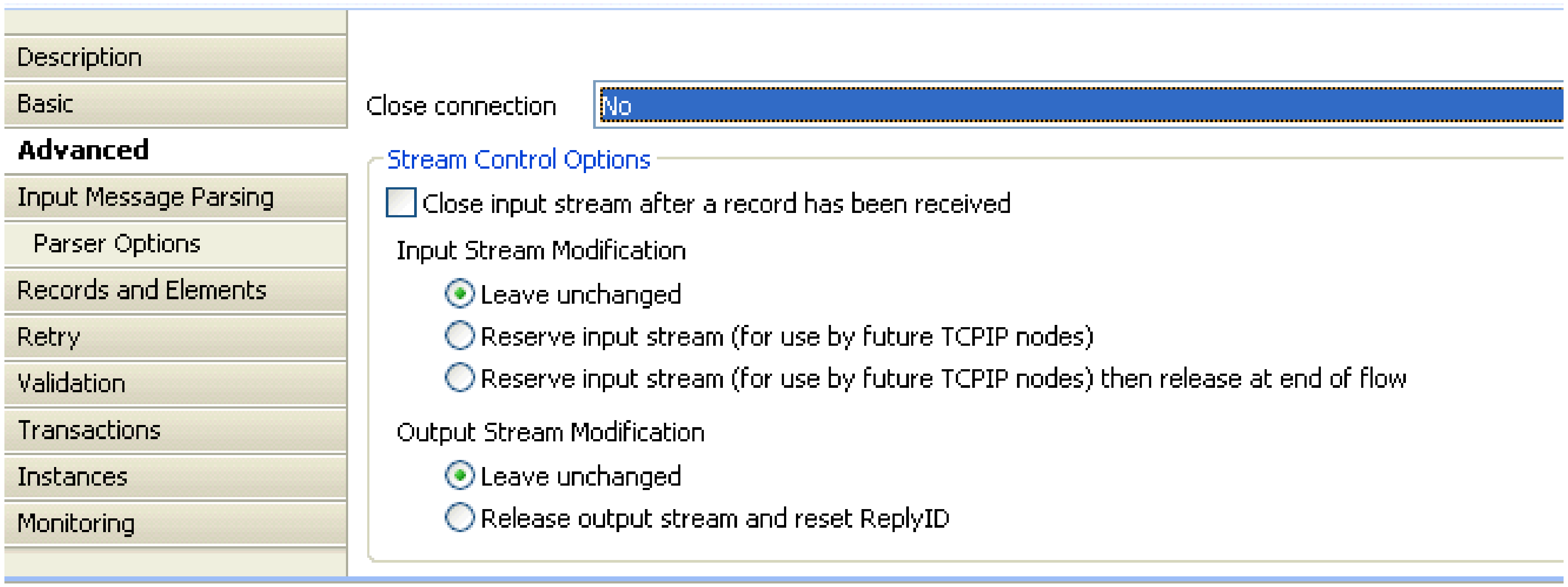
Reserve: To connect a series of nodes to give complex processing on a stream in an ordered, controlled, synchronous sequence. For example, asynchronous request and reply

Release: For all nodes, the connection for the streams is released

Reserve and release at the end of the flow: to reserve a connection and to ensure that the connection's stream is released when the message flow has finished processing (including any error conditions that might occur)

Common TCP/IP scenarios

Stream control options configurable on the nodes



Description
Basic
Advanced
Input Message Parsing
Parser Options
Records and Elements
Retry
Validation
Transactions
Instances
Monitoring

Close connection: No

Stream Control Options

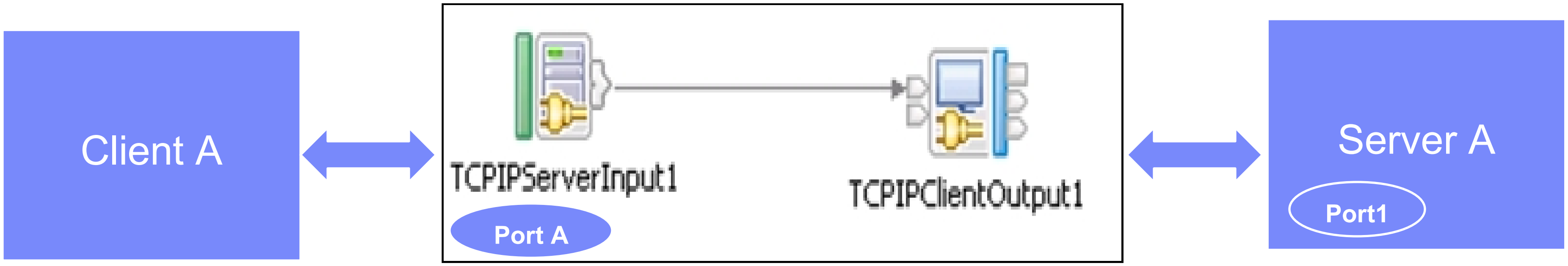
☐ Close input stream after a record has been received

Input Stream Modification
- [✓ Leave unchanged]
- Reserve input stream (for use by future TCPIP nodes)
- Reserve input stream (for use by future TCPIP nodes) then release at end of flow

Output Stream Modification
- [✓ Leave unchanged]
- Release output stream and reset ReplyID

Common TCP/IP scenarios

Add broker as an intermediary router:

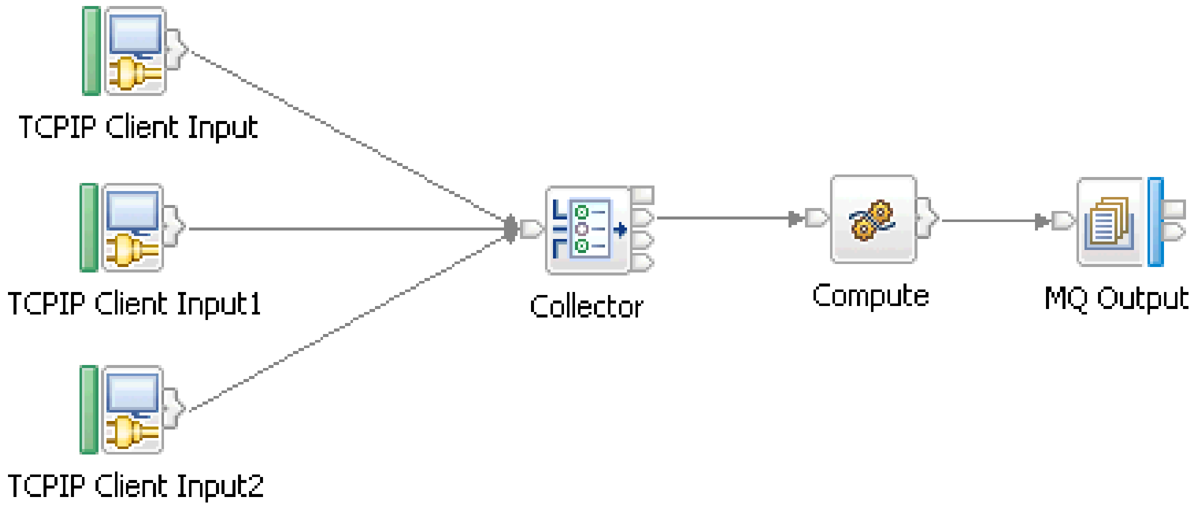


Client A connects to Broker TCPIPServerInput1 and Broker TCPIPClientOutput1 connects to Server A

Expense submission and Price change notification scenarios:
http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/topic/com.ibm.etools.mft.doc/ac67392_.htm

Common TCP/IP scenarios

Different Client input nodes taking data from different TCP servers and after further processing send the message to an output queue

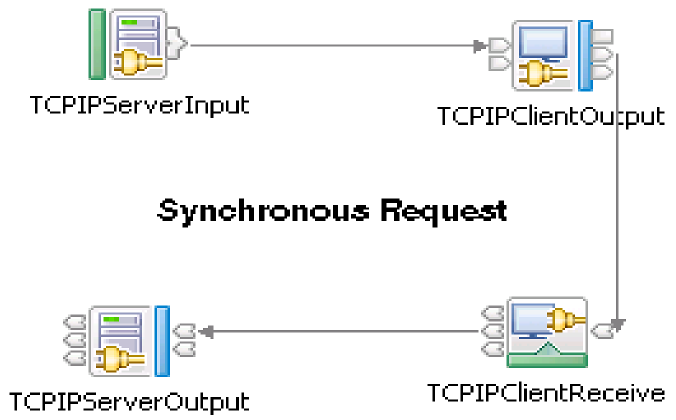


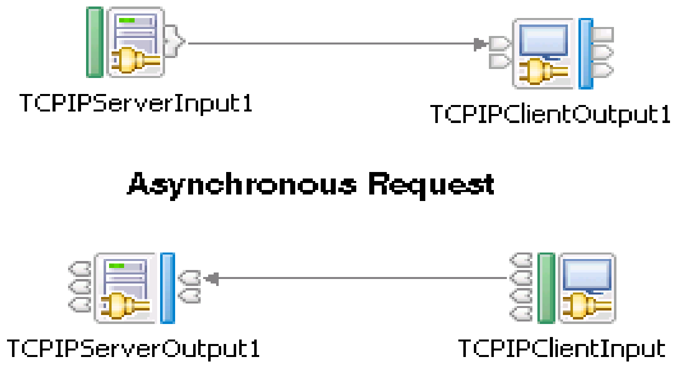
Common TCP/IP scenarios

Synchronous and Asynchronous scenario:

- Synchronous routing is the combination of the incoming server request and outgoing client request from the Request/Reply pattern

- Asynchronous routing is the combination of the incoming server request and outgoing client request but with the client reply received asynchronously via a client input node



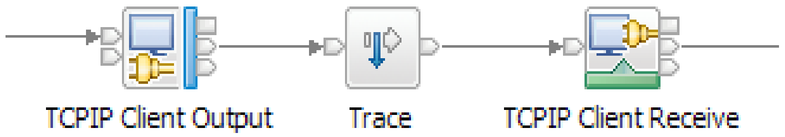


Problem Diagnosis

- Common Problems
  - Connectivity
  - Flow Design
- Tool
  - netstat, telnet, or driver flow
  - server and user trace

http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/topic/com.ibm.etools.mft.doc/au14270_.htm

  - trace node with “${LocalEnvironment}” pattern



Problem Diagnosis – Connectivity

- netstat -anp tcp (Windows®)

```
Proto     Local Address         Foreign Address                          State
TCP       0.0.0.0:7777          0.0.0.0:0                                LISTENING
TCP       9.65.224.170:139      0.0.0.0:0                                LISTENING
TCP       9.65.224.170:7777     9.17.195.139:1352                        ESTABLISHED
TCP       9.65.224.170:1382     9.17.186.253:80                          CLOSE_WAIT
TCP       9.65.224.170:7777     9.17.195.139:1352                        TIME_WAIT
```

- LISTEN or LISTENING - Server is ready to accept connection
- ESTABLISHED - Client received server's SYN and session is established
- CLOSE_WAIT - passive close
- TIMED_WAIT - active close

Problem Diagnosis – Connectivity

- telnet <hostname> <port number>

  The server input node is available

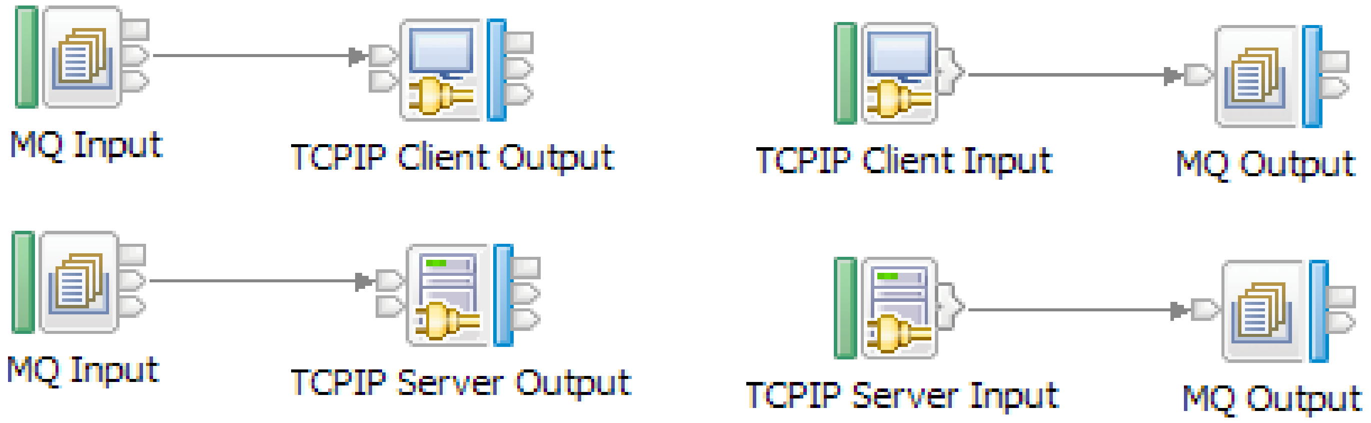
UserTrace   BIP3560I: A record has been received from Hostname "localhost" on Port "7777" in TCPIP input node "TCPIPServerInput" in message flow "TCPIPServerSimulation".

The server input node is not available or not able to reach to the server side

```
C:\Documents and Settings\Administrator>telnet localhost 7777
Connecting To localhost...Could not open connection to the host, on port
7777: Connect failed
```

Problem Diagnosis – Connectivity

Driver flow



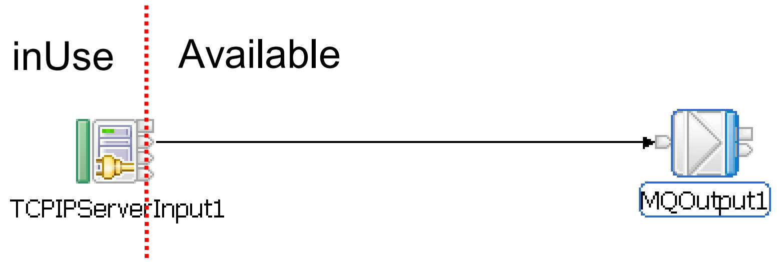
Problem Diagnosis – Flow Design

- Incorrectly define the message resulting in blocking waits followed by exception.
- Not reserving connections correctly
- Incorrectly configuring the nodes to use the correct connection id.

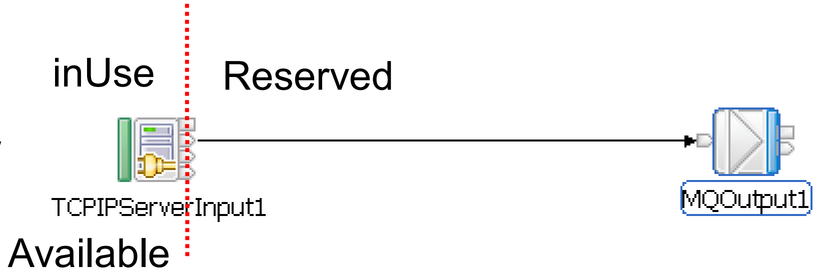
Problem Diagnosis – Flow Design

- Example of reserving connection

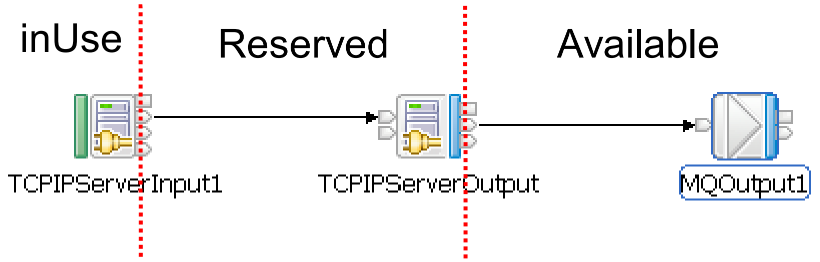
Default



Reserved to end of flow



Reserve and release in another node



Common Problem Example

Problem:

On Sparc Solaris 10 the following error in the syslogs is seen:
Failed to create a client connection using hostname: '', port: ''. Reason: 'Invalid argument'

Solution:

You can try the following two methods to resolve the error:
a. Change the *MQSIJVERBOSE* environment variable,

for example:

```
export MQSIJVERBOSE=-java.nio.channels.spi.SelectorProvider
=sun.nio.ch.PollSelectorProvider
```

b. Change the limit of maximum file handles to *value* instead of *RLIM64_INFINITY*

Common Problem Example

Problem:

The message BIP3559E showing maximum number of connections has been reached. Or number of TCPIP connection keep growing cause BIP3559E

Solution:

a) Increase the 'MaximumConnections' parameter in the TCPIPServer node.

1. To display all TCPIPServer configurable services:

```
mqsireportproperties <brokerName> -c TCPIPServer -o AllReportableEntityNames -r
```

2. To change the MaximumConnections property, please issue:

```
mqsichangeproperties <brokerName> -c TCPIPServer -o MaximumConnections -v <integer_value>
```

Common Problem Example

Problem:

The message BIP3559E showing maximum number of connections has been reached. Or number of TCPIP connection keep growing cause BIP3559E

Solution:

b) The 'ExpireConnectionSec' so that inactive connections will be released after a specific time.

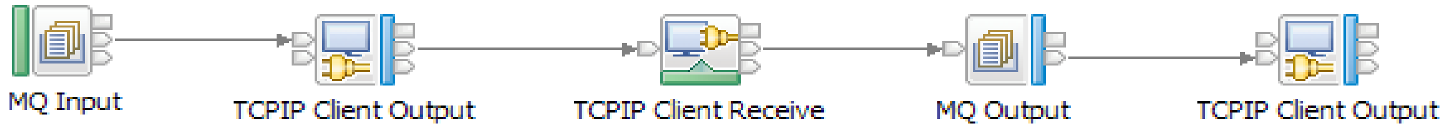
1. To change the ExpireConnectionSec property, please issue:

mqsichangeproperties <brokerName> -c TCPIPServer -o MaximumConnections -v <integer_value>

Common Problem Example

Problem:

The message BIP3587E showing that no connections are available on Host.



close output stream After record has been sent

close the connection after the data has been received.

$LocalEnvironment/ Destination/ TCPIP/ Output/Id

Solution:

a) On the TCPIPClientOutput node, change to 'Reserve output stream and release at end of flow option'.

b) On the TCPIPClientReceive node, change to 'Close After Timeout'.

c) On the final TCPIPClientOutput node, ensure to use correct connection id.
$LocalEnvironment/TCPIP/Receive/ConnectionDetails/Id

References

WebSphere Message Broker V7 Information Center

Integrating with TCP/IP using WebSphere Message Broker

TCP/IP Tutorial and Technical Overview

Connecting Your Business Using IBM® WebSphere Message Broker V7 as an ESB

WebSphere Message Broker support page

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at: http://www.ibm.com/software/websphere/support/supp_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/
- Join the Global WebSphere Community: http://www.websphereusergroup.org
- Access key product show-me demos and tutorials by visiting IBM® Education Assistant: http://www.ibm.com/software/info/education/assistant
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically: http://www.ibm.com/software/websphere/support/d2w.html
- Sign up to receive weekly technical My Notifications emails: http://www.ibm.com/software/support/einfo.html

Connect with us!

1. **Get notified on upcoming webcasts**
   Send an e-mail to wsehelp@us.ibm.com with subject line "wste subscribe" to get a list of mailing lists and to subscribe
2. **Tell us what you want to learn**
   Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com
3. **Be connected!**
   Connect with us on Facebook
   Connect with us on Twitter

Questions and Answers