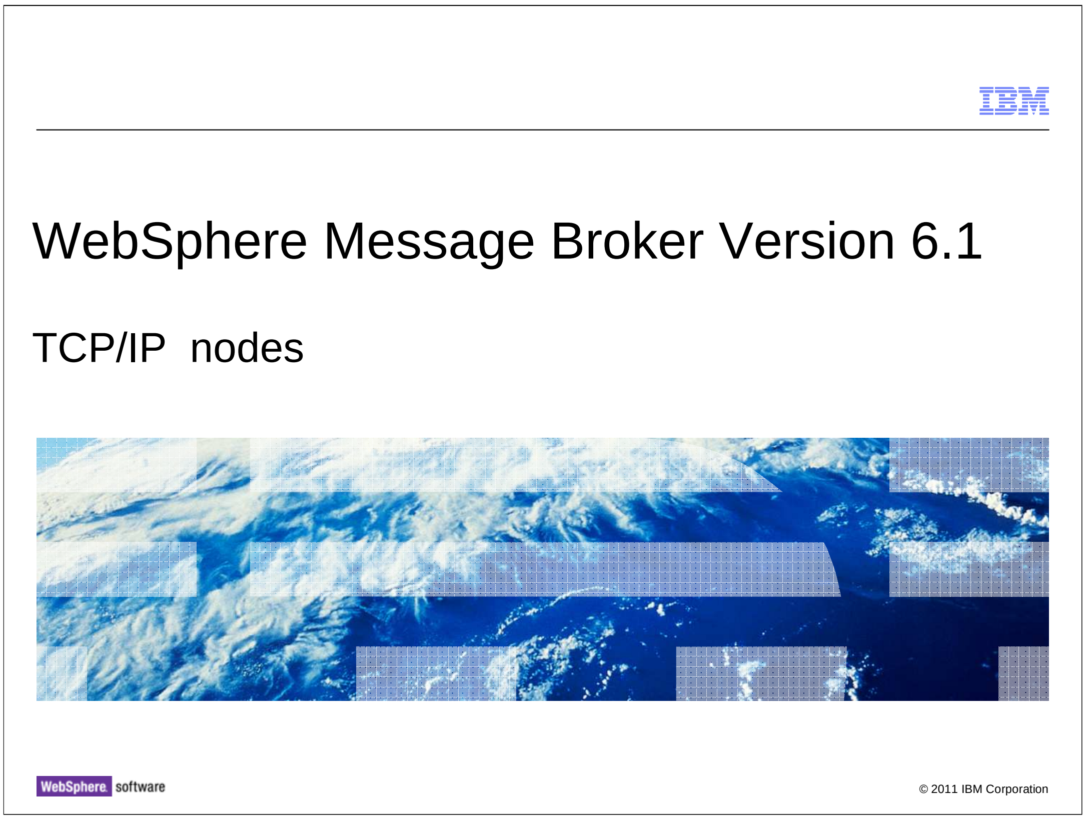
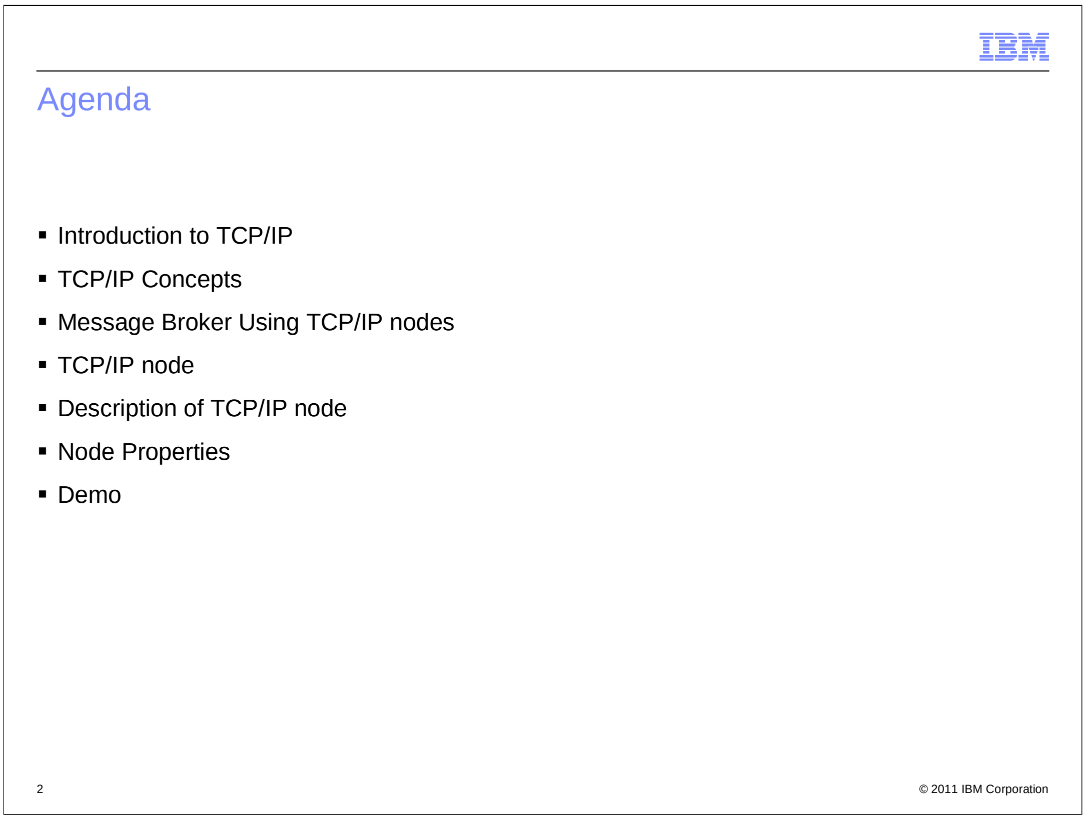


WebSphere Message Broker Version 6.1

TCP/IP nodes

This presentation will discuss the TCP/IP nodes that were introduced to WebSphere Message Broker in version 6.1.

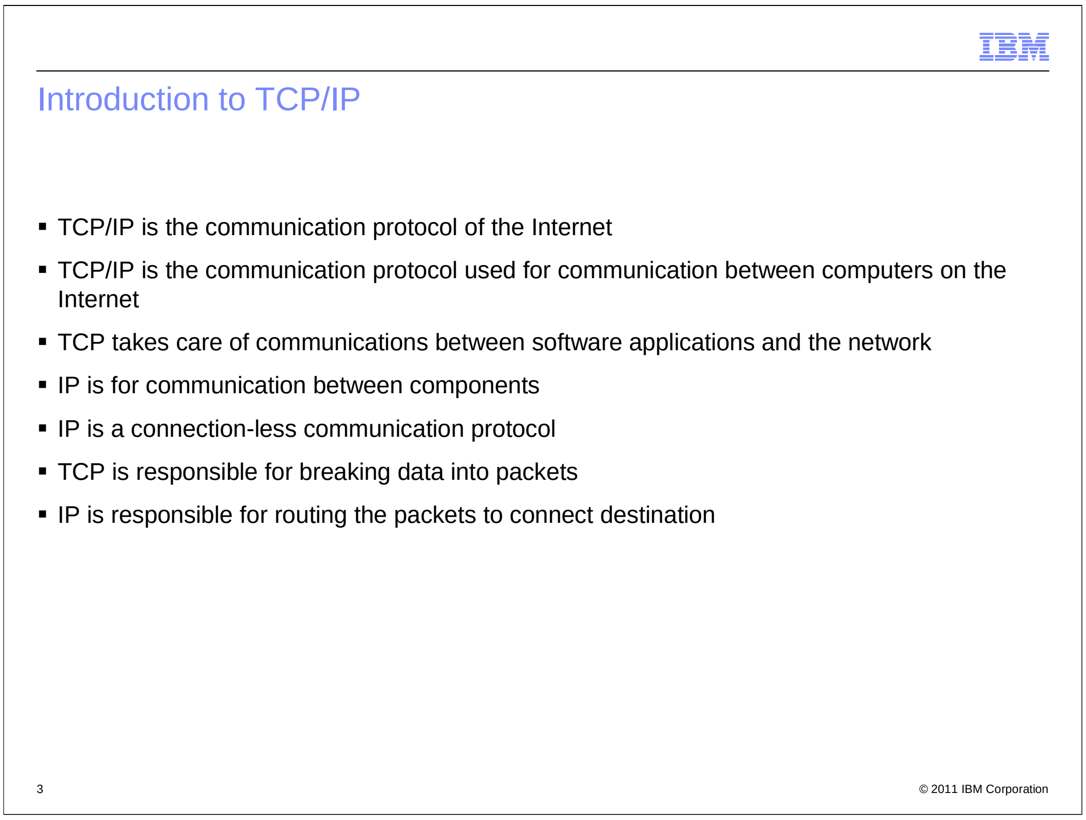


Agenda

- Introduction to TCP/IP
- TCP/IP Concepts
- Message Broker Using TCP/IP nodes
- TCP/IP node
- Description of TCP/IP node
- Node Properties
- Demo

2

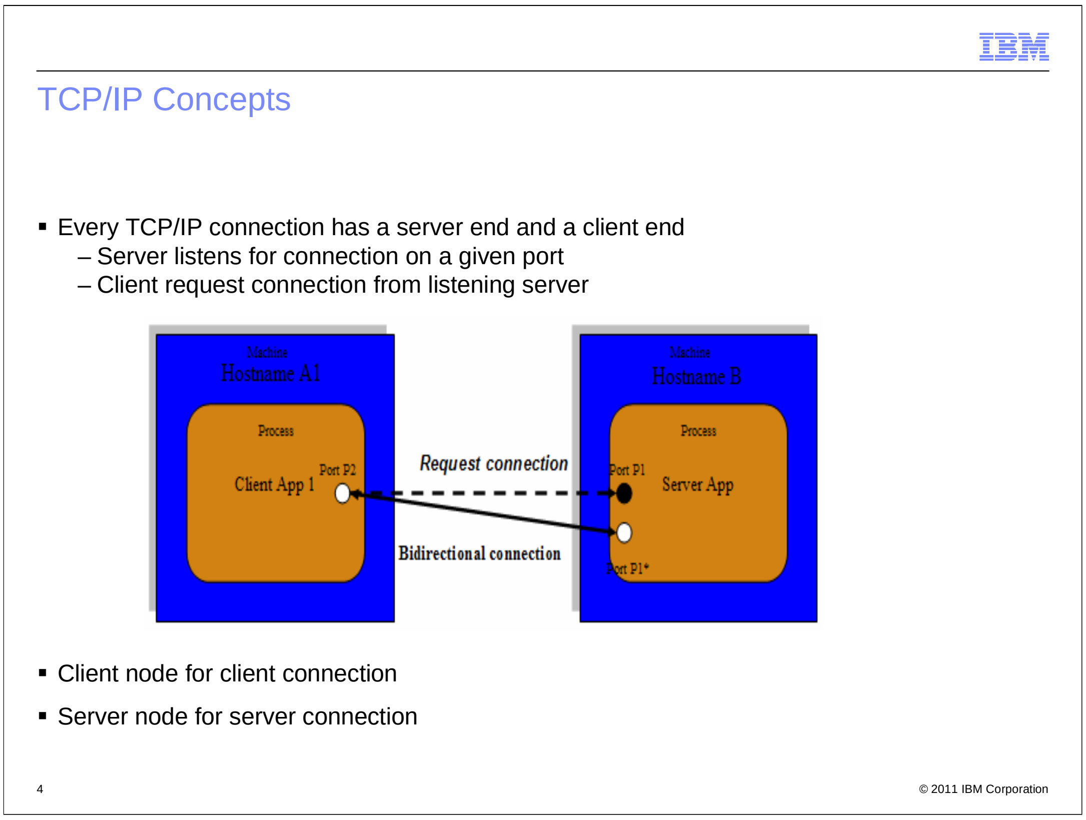
This presentation will provide a brief introduction to the TCP/IP protocol, and will then present some scenarios related to TCPIP protocol. The presentation will also cover some advantages and limitations of the TCP/IP protocol. The session will then describe the WebSphere Message Broker TCPIP nodes and their properties.



Introduction to TCP/IP

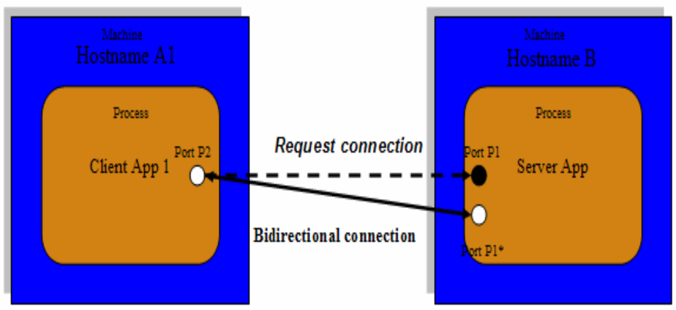
- TCP/IP is the communication protocol of the Internet
- TCP/IP is the communication protocol used for communication between computers on the Internet
- TCP takes care of communications between software applications and the network
- IP is for communication between components
- IP is a connection-less communication protocol
- TCP is responsible for breaking data into packets
- IP is responsible for routing the packets to connect destination

The TCP/IP protocol is used for communication between computers on the Internet. TCP/IP defines how electronic devices, such as computers, should be connected to the Internet, and how data should be transmitted between them. The protocol is most easily understood by breaking it down into constituent layers. The TCP layer takes care of communication between application software, such as a browser, and the network, where as the IP layer is used for communication between computers. TCP is responsible for breaking data up into packets before sending over the network, and assembling the packets when they arrive. IP is responsible for routing the packets to the correct destination.



TCP/IP Concepts

- Every TCP/IP connection has a server end and a client end
  - Server listens for connection on a given port
  - Client request connection from listening server



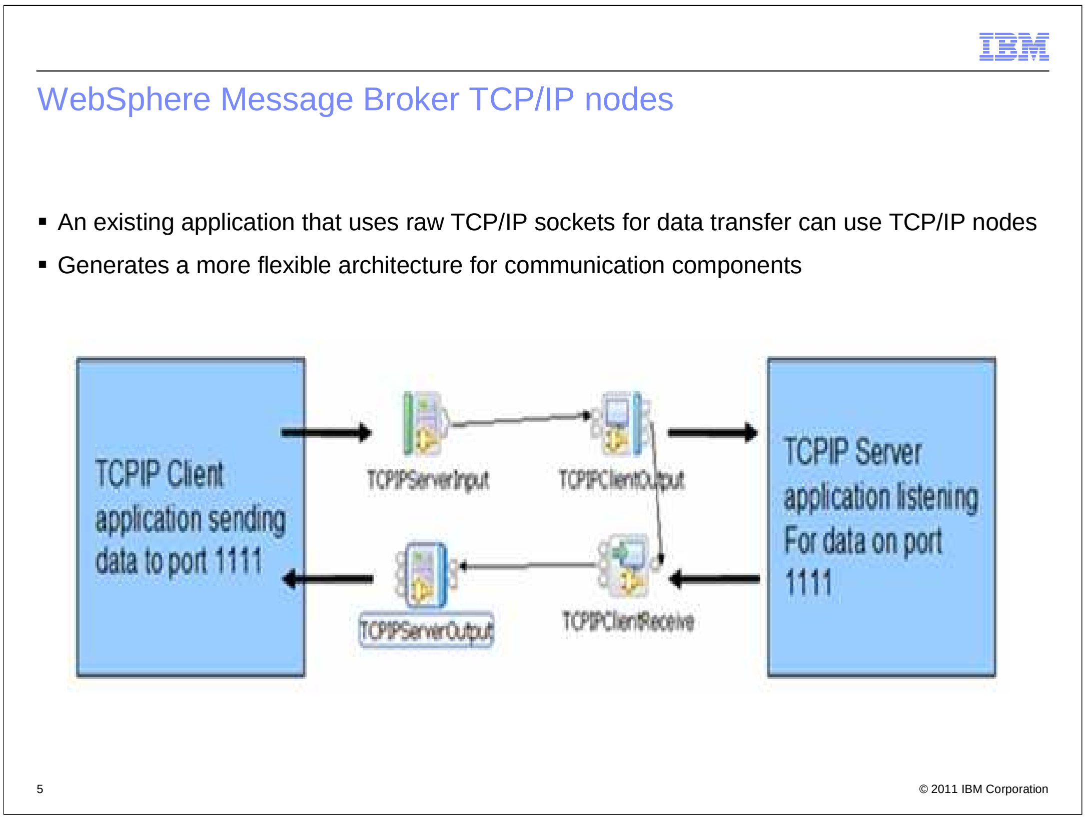
- Client node for client connection
- Server node for server connection

4 © 2011 IBM Corporation

The TCP/IP protocol is one of the simplest protocols used for transferring data between two applications, where the two applications can be running on different computers. The TCP/IP protocol has a client application and a server application, and uses a client socket and a server socket. The client socket is created on the machine where the client application is running and the server socket runs on the server machine. The server socket has a port and listens on that port for the connection from the client application. The client application connects to the server application using a port and the machine's IP address. Once the connection is established, both the client and server application can use the socket to send and receive data.

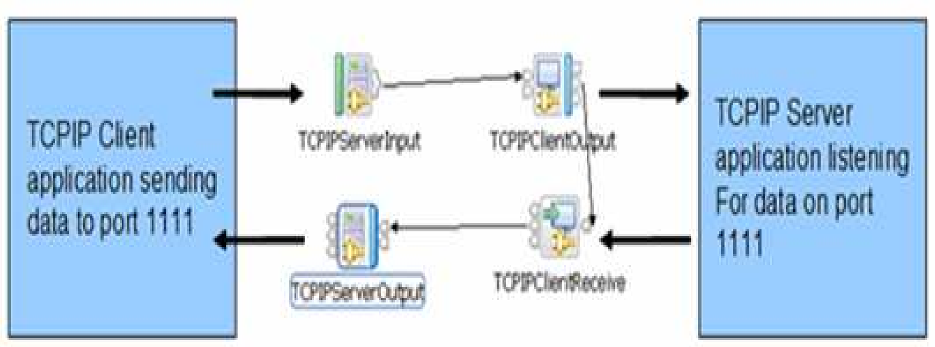
Most modern application communication is built on the TCP/IP protocol. For example, a low-level TCP/IP connection, which has requirements for acknowledgements and retransmissions implemented by the communicating applications, such as MATIP in the airline industry or MLLP in healthcare. WebSphere MQ channels and web browser connections using HTTP is based on TCP/IP.

In many cases, it is easier for an application developer to use protocols which are built on TCP/IP, such as WebSphere MQ or HTTP. However, in some scenarios, these functions might not be available. Many applications cannot communicate using MQ or HTTP, and only offer support for raw TCP/IP connections. In these cases, using the TCP/IP nodes in WebSphere Message Broker is an easy way to achieve the required integration. The TCP/IP protocol itself is generally a fast networking option which requires a low bandwidth. It is ideal for many small devices or circumstances where there is a cost which is dependent on the amount of data that is being transmitted.



WebSphere Message Broker TCP/IP nodes

- An existing application that uses raw TCP/IP sockets for data transfer can use TCP/IP nodes
- Generates a more flexible architecture for communication components

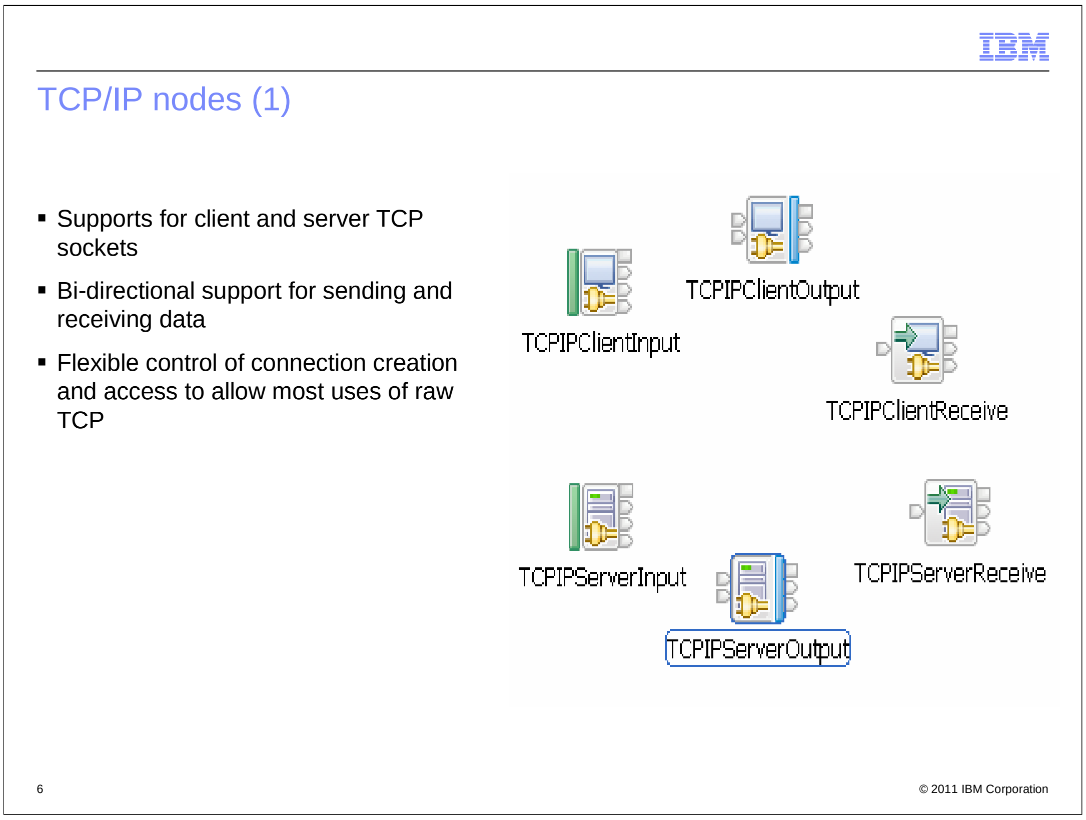

TCPIP Client application sending data to port 1111
TCPIPServerInput
TCPIPClientOutput
TCPIPServerOutput
TCPIPClientReceive
TCPIP Server application listening For data on port 1111

5 © 2011 IBM Corporation

The TCP/IP nodes help to generate a more flexible architecture for communication between the components, where the client and server applications communicate only with raw TCP/IP. For example, you can use WebSphere Message Broker as a router to connect two applications without making any change in their interfaces.

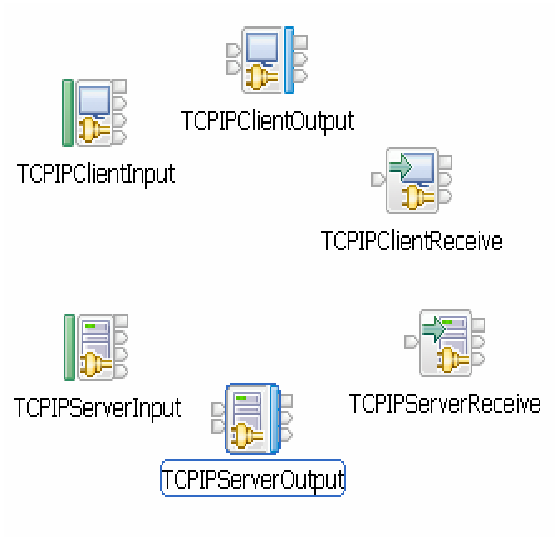
The nodes work best when a simple custom protocol is used on top of TCPIP, where the connection rules are implemented by the rest of the message flow. Typically, this means a simple request/reply or send-and-forget model. If there is a large amount of handshaking and bidirectional communication required by such a protocol, then it might not be appropriate to use the nodes, since the flow might become very complicated. For example, HTTP is based on TCPIP but it is be too complicated to implement using the TCPIP nodes, although in principle it can be done. For this type of application connectivity, consider using the HTTP or SOAP nodes.

The TCPIP protocol has some limitations. It is non-transactional, non-persistent, has no built in security and has no standard way of signaling the start and end of a message. If you have these types of requirements, consider a transport mechanism like WebSphere MQ. However, if you are restricted to raw TCP/IP communications, the WebSphere Message Broker nodes can be used to implement this.



TCP/IP nodes (1)

- Supports for client and server TCP sockets
- Bi-directional support for sending and receiving data
- Flexible control of connection creation and access to allow most uses of raw TCP

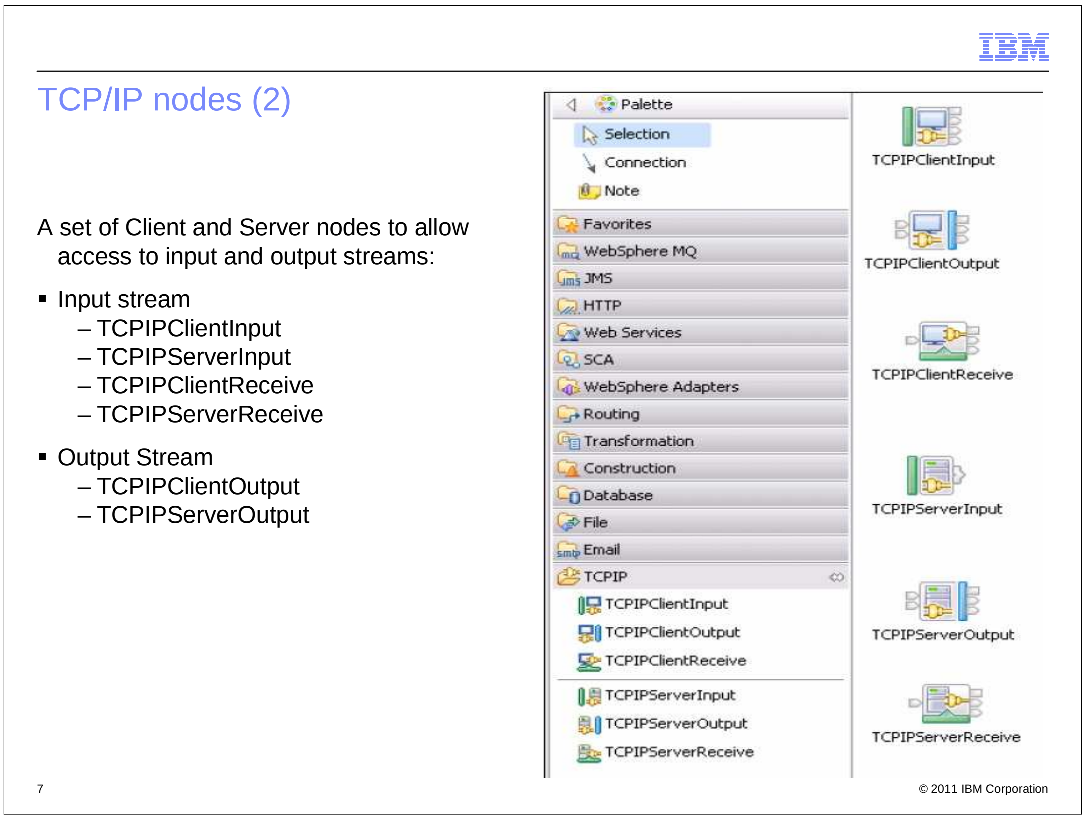


6

WebSphere Message Broker contains six TCP/IP nodes. These are split into two main types: server and client.

The server nodes uses connections which have been made to WebSphere Message Broker from a remote client and the client nodes make connections to remote servers. Each type contains three nodes: input, output and receive. The combination of nodes gives the ability to do bi-directional sending and receiving of data.

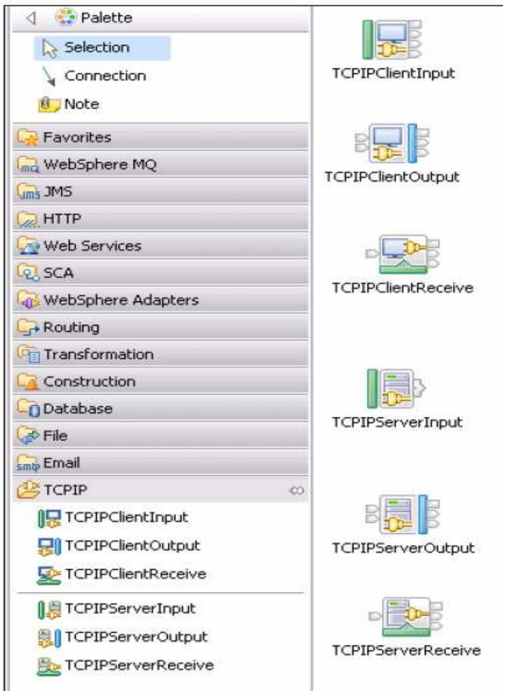
The server and client nodes access the data in the message tree in exactly the same way. The only difference is how the connection is made.



TCP/IP nodes (2)

A set of Client and Server nodes to allow access to input and output streams:

- Input stream
  - TCPIPClientInput
  - TCPIPServerInput
  - TCPIPClientReceive
  - TCPIPServerReceive
- Output Stream
  - TCPIPClientOutput
  - TCPIPServerOutput



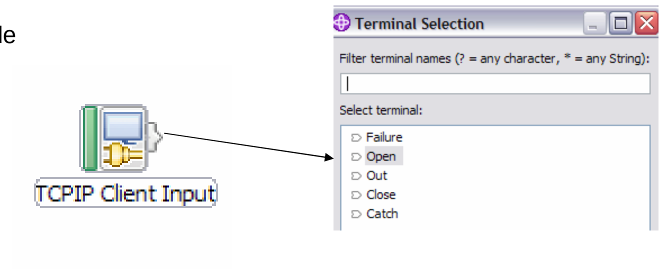
7

The nodes can also be categorized into two different sets in terms of stream.

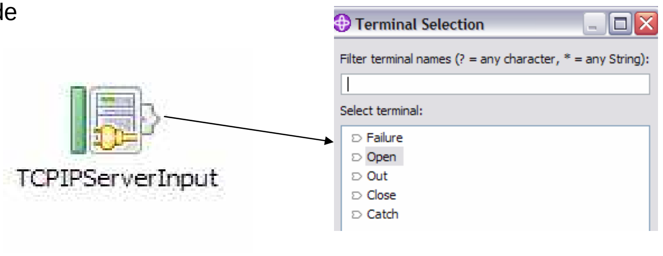
The input stream consists of four nodes, namely TCPIP-Client-Input, TCPIP-Server-Input, TCPIP-Client-Receive and TCPIP-Server-Receive. The output stream consists of TCPIP-Client-Output and TCPIP-Server-Output.

TCP/IP Input nodes

- TCPIPClientInput node



- TCPIPServerInput node



Both the Client Input and Server Input nodes have five terminals.

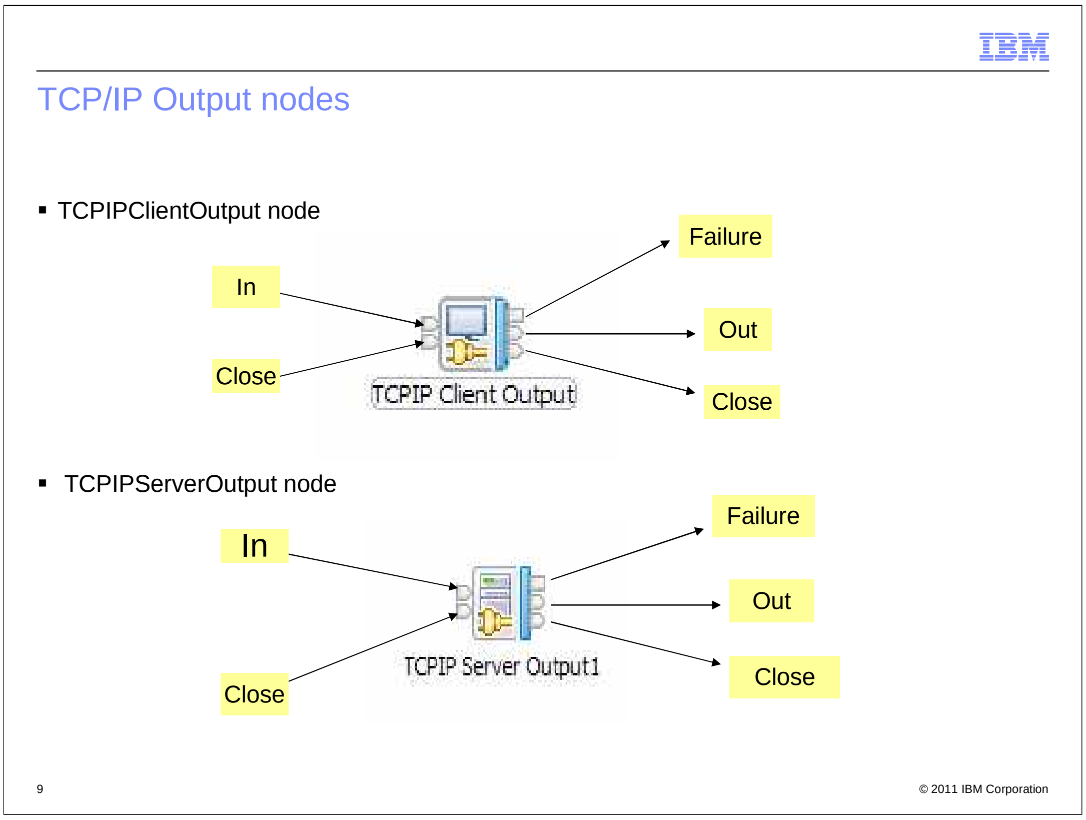
The Open terminals receives an event whenever a new connection is made and gives you a chance to do processing on the connection before it is used.

The Out terminal receives a message whenever a record is received from the connection.

The Failure terminal receives a message if an error occurs receiving a record.

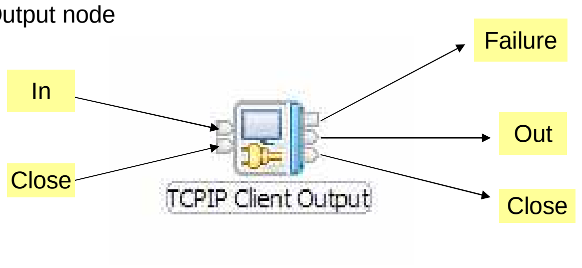
The Close terminals receives an event if a connection is closed.

The Catch terminal is used to catch errors that occurred downstream of the node, in common with most other nodes.

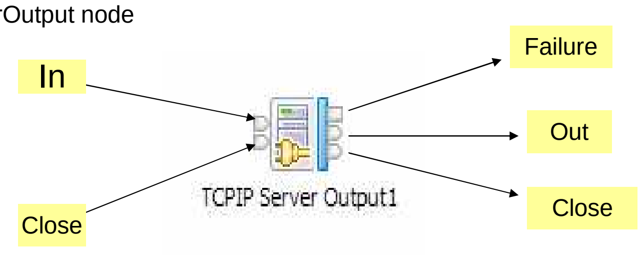


TCP/IP Output nodes

- TCPIPClientOutput node



- TCPIPServerOutput node



9 © 2011 IBM Corporation

Both the TCPIP Client Output node and TCPIP Server Output nodes have two input terminals.

The In terminal is used to accept a message for processing by the node.

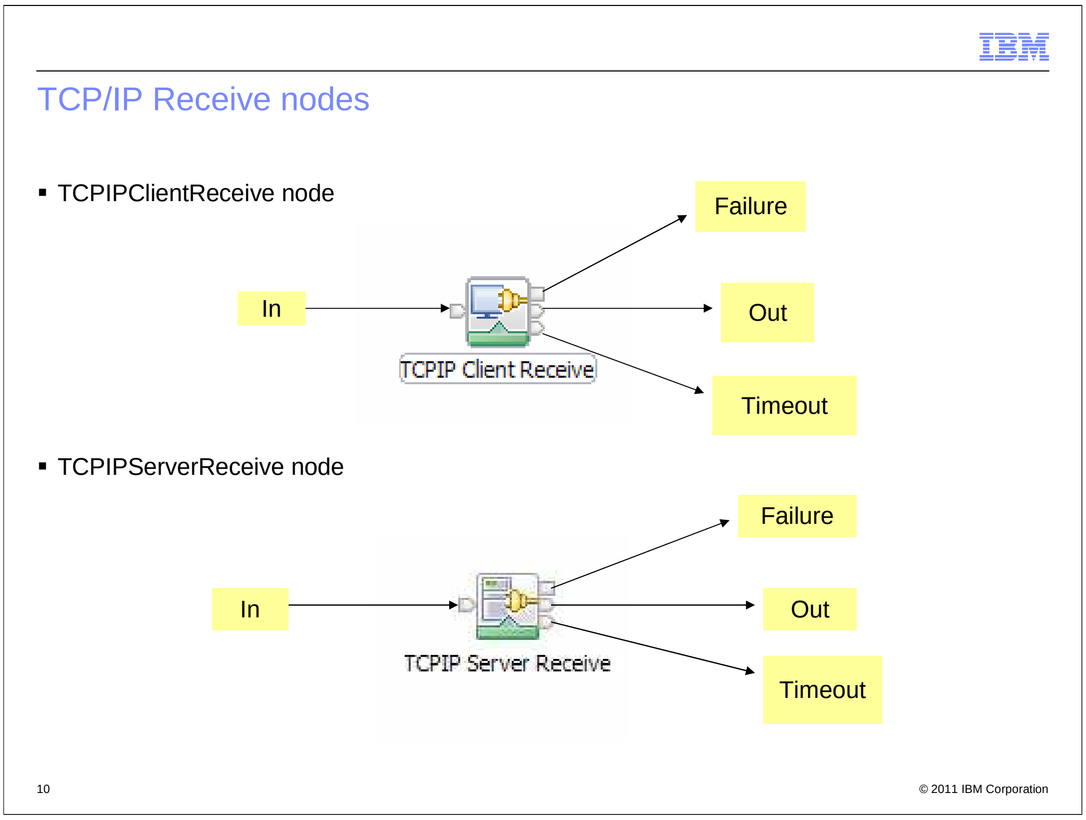
The Close terminal is used to accept requests to close connections.

They also have three output terminals:

The Out terminals is used after successful processing.

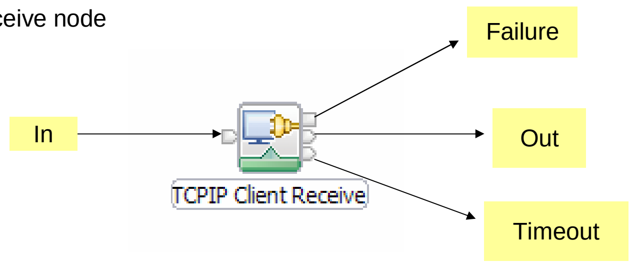
The Failure terminal is used after a processing error.

And the Close terminal is used when a “close connection” request is made using the Close input terminal.

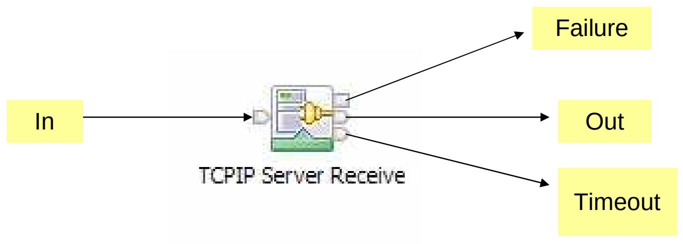


TCP/IP Receive nodes

- TCPIPClientReceive node



- TCPIPServerReceive node

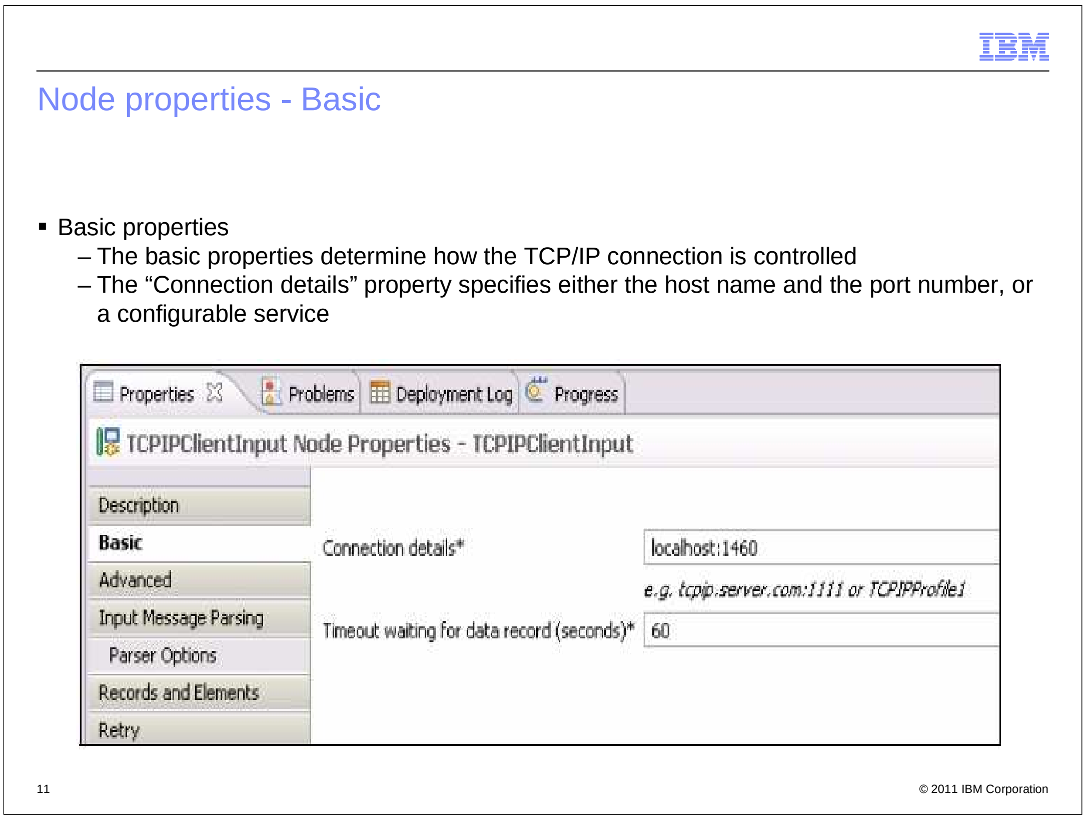


10 © 2011 IBM Corporation

Both the “TCPIP Client Receive” and “TCPIP Server Receive” nodes have an Input terminal, a failure terminal, an Out terminal, and a Timeout terminal. The Timeout terminal is used when the time specified in the Timeout node properties has been exceeded.

The “TCPIP Client Receive” node receives data over a TCPIP client connection. The node waits for the data to be received and to retrieve data.

The “TCPIP Server Receive” node receives data over a server TCPIP connection. The purpose of the “TCPIP Server Receive” node is to wait for the data to be received on a TCP/IP connection and to retrieve the data.



Node properties - Basic

- Basic properties
  - The basic properties determine how the TCP/IP connection is controlled
  - The “Connection details” property specifies either the host name and the port number, or a configurable service

TCPIPClientInput Node Properties - TCPIPClientInput

Connection details* localhost:1460

e.g. tcpip.server.com:1111 or TCPIPProfile1

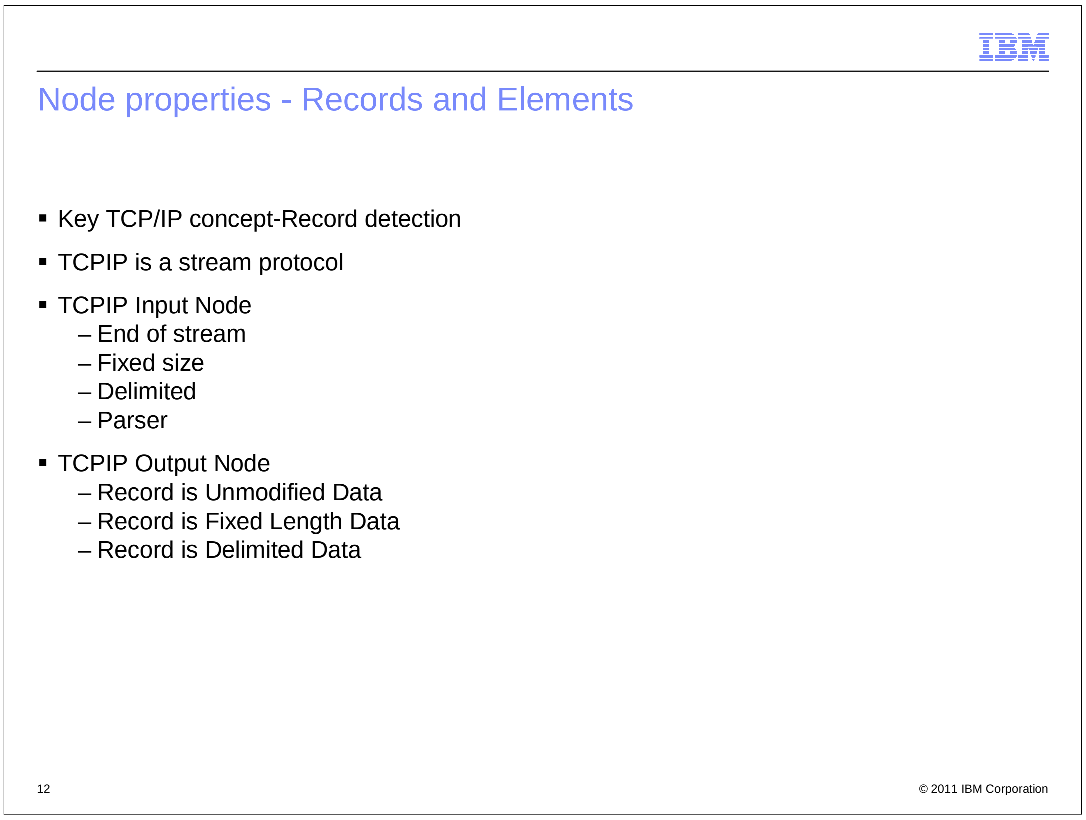
Timeout waiting for data record (seconds)* 60

11 © 2011 IBM Corporation

This slide describes the basic node property which determines how the TCPIP connection is controlled by the node.

The “connection details” property is mandatory and is used to specify either the host name and the port number, or the configurable service.

In the server node case, the property requires only the port, since the host is always the local host. In the client node case, both the host and the port need to be specified.



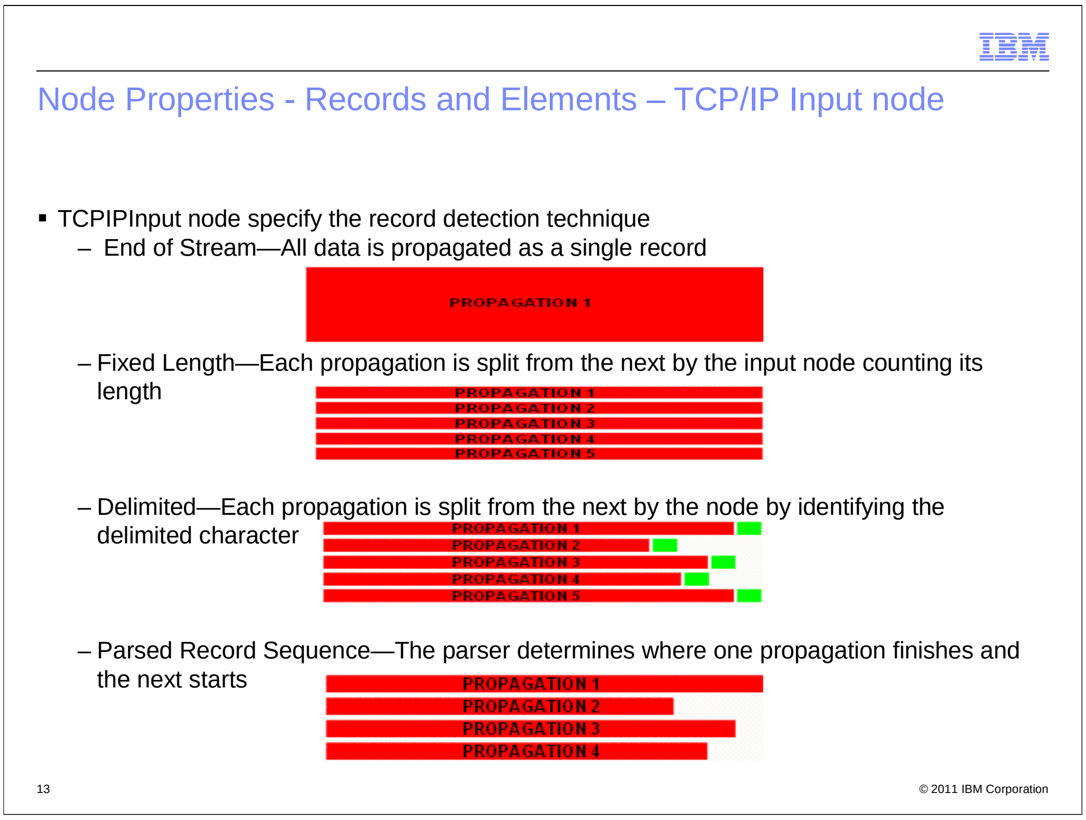
Node properties - Records and Elements

- Key TCP/IP concept-Record detection
- TCPIP is a stream protocol
- TCPIP Input Node
  - End of stream
  - Fixed size
  - Delimited
  - Parser
- TCPIP Output Node
  - Record is Unmodified Data
  - Record is Fixed Length Data
  - Record is Delimited Data

12

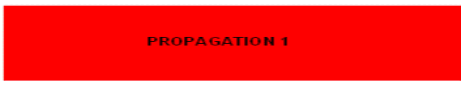
The Records and Elements tab contains the properties which specify how the incoming data stream should be split up before being processed by the message flow.

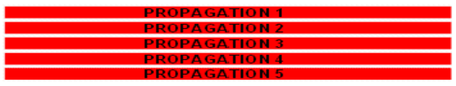
The “Record detection” property is used to determine how the data is split into records, each of which generates a single message. Record detection allows the data to be split into individual parts using several techniques. Alternatively the whole of the input data can be processed in one go. The TCPIP Input node supports End of Stream, Fixed Length, Delimited and Parser. The TCPIP Output node supports “Record is Unmodified Data”, “Record is Fixed Length Data” and “Record is Delimited Data”.

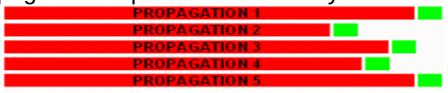


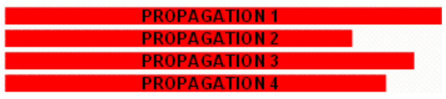
Node Properties - Records and Elements – TCP/IP Input node

- TCPIPInput node specify the record detection technique
  - End of Stream—All data is propagated as a single record
  - Fixed Length—Each propagation is split from the next by the input node counting its length
  - Delimited—Each propagation is split from the next by the node by identifying the delimited character
  - Parsed Record Sequence—The parser determines where one propagation finishes and the next starts









13 © 2011 IBM Corporation

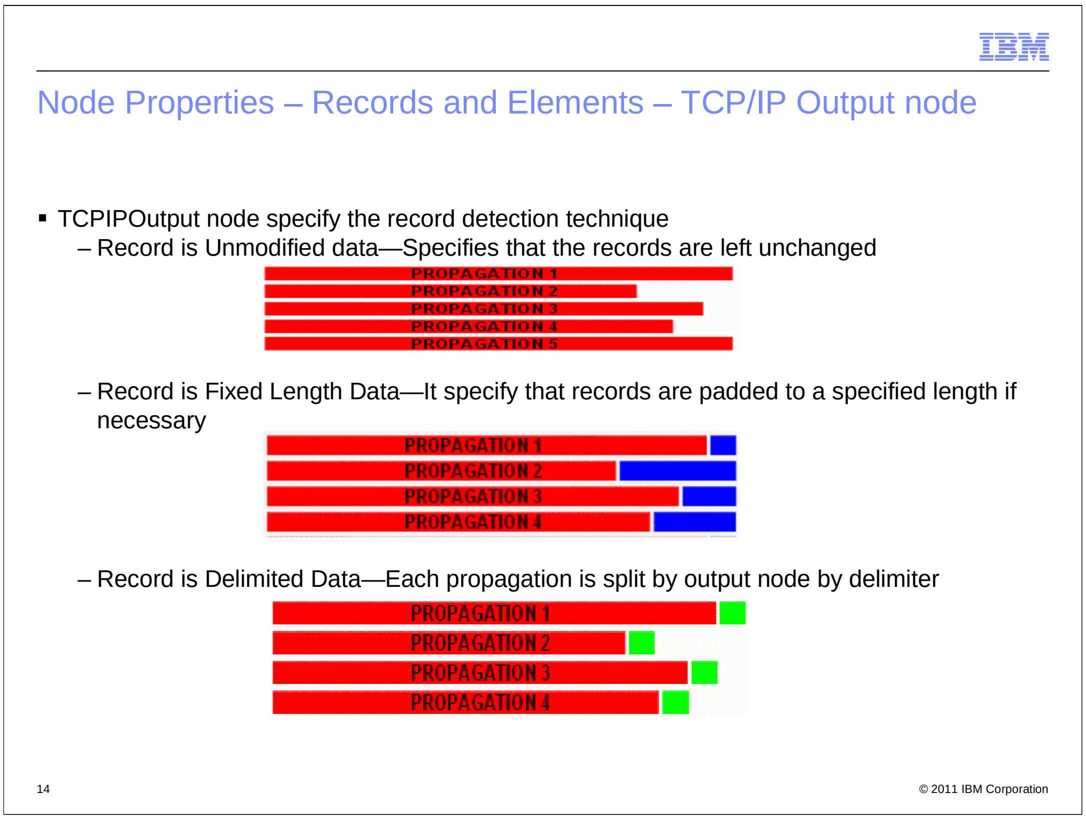
This slide describes the four options available for record detection in the TCPIP Input node. It also summarizes the difference between the options.

“End of stream” specifies that all the data sent in the data stream is a single record.

“Fixed length” breaks the data into a several bytes and propagates each chart as a separate message. The last message might be smaller than the specified length. If you use this option, you should specify the length of record in the Input node property.

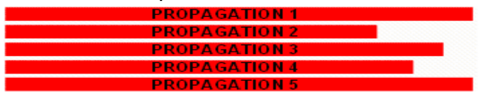
“Delimited” allows you to specify the character that is used to delimit the records. In this case you can specify a standard DOS or UXIX end of line delimiter, or you can specify your own custom delimiter. If you specify “delimiter type” as Infix, and the last part of the data ends with a delimiter, then the input node will propagate a empty record to indicate a delimiter presence. If you specify the “delimiter type” as postfix, the input node will not propagate a non-empty record, regardless of whether the last part of the data ends with a delimiter or not.

The “Parsed Record Sequence” option determines where one propagation finishes and the next starts.

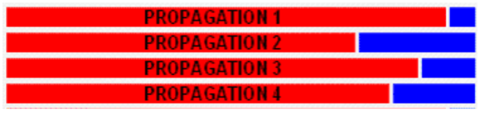


Node Properties – Records and Elements – TCP/IP Output node

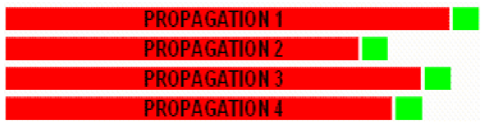
- TCPIPOutput node specify the record detection technique
  - Record is Unmodified data—Specifies that the records are left unchanged



  - Record is Fixed Length Data—It specify that records are padded to a specified length if necessary



  - Record is Delimited Data—Each propagation is split by output node by delimiter



14 © 2011 IBM Corporation

This slide describes the options available for record detection in the TCPIP Output node. The slide also summarizes the differences between the three options.

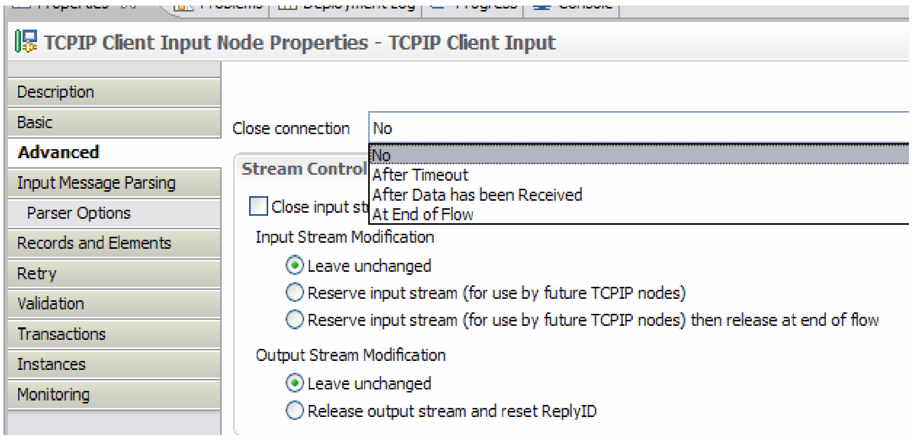
The “Unmodified data” option specifies that the records are left unchanged. No delimiters are written between each propagation. This value is selected as the default.

In “Fixed length” each propagation is split from the next by the Output node padding its length to a required length using the padding bytes. No delimiters are used in the output record.

In “Delimited data” each propagation is split from the next by the TCPIP Output node adding delimiter characters between them.

Node Properties – Advanced – TCP/IP Input node

- This property show how the data stream is controlled



TCPIP Client Input Node Properties - TCPIP Client Input

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

Close connection: No
- [✓ No]
- After Timeout
- After Data has been Received
- At End of Flow

Stream Control

☐ Close input st

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

15

The Advanced property specifies how the node should interact with the stream of TCPIP input socket data and the stream of TCPIP output data. There are two sets of radio button, one referring to the input stream and the second referring to the output stream.

The first section refers to the Input Stream Modification.

Select the first button, “leave unchanged”, to leave the input stream as it was when it entered the node. This value is selected by default.

Select the second button to specify that this input stream is returned to the pool and is available for use by any input or receive node.

Select the third button to specify that this input stream can be used only by this node and by other input or receive nodes that request it by specifying the connection ID. When the connection input stream is reserved, no other nodes can use it without specifying the correct connection ID.

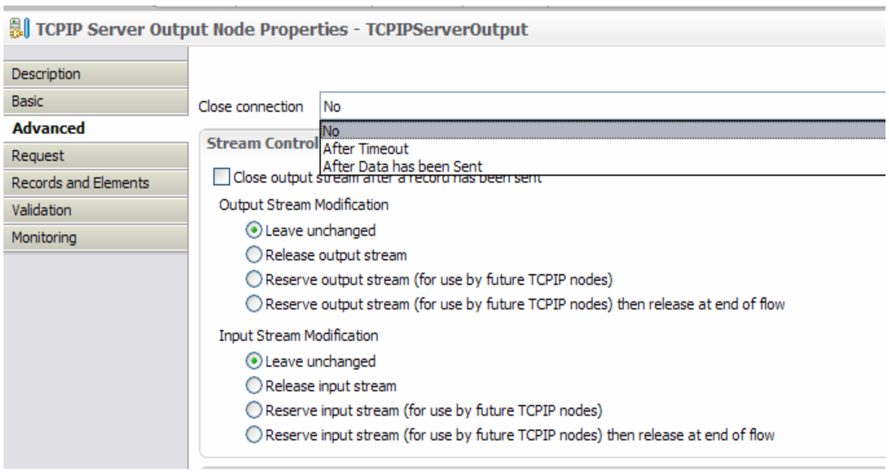
The second section refers to Output Stream Modification.

Select the first button, “leave unchanged”, to leave the output stream as it was when it entered the node. This value is selected by default.

Select the second button to specify that this output stream is returned to the pool and is available for use by other nodes.

Node Properties - Advanced – TCP/IP Output node

- This property shows how the data stream is controlled


TCPIP Server Output Node Properties - TCPIPServerOutput

- Description
- Basic
- **Advanced**
- Request
- Records and Elements
- Validation
- Monitoring

Close connection: No
- No
- After Timeout
- After Data has been Sent

Stream Control

☐ Close output stream after a record has been sent

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

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

For the TCP/IP Output node, the Advanced properties are slightly different.

The Advanced properties specifies how the output node control the output data stream. The first set of radio buttons refers to the output stream and the second set refers to the input stream.

The first section refers to the Output Stream Modification.

Select the first button, “Leave unchanged”, to leave the output stream as it was when it entered the node. This value is selected by default.

Select the second button, “Release output stream”, to specify that this output stream is returned to the pool and is available for use by any output node.

Select the third button to specify that this output stream can be used only by this node and by other output nodes that request it by specifying the connection ID. When the connection input stream is reserved, no other nodes can use it without specifying the correct connection ID.

Select the fourth button to specify that this output stream can be used only by this node and output nodes that request it by specifying the correct connection ID. After the message has been propagated, this output stream is returned to the pool and becomes available for use by any output node.

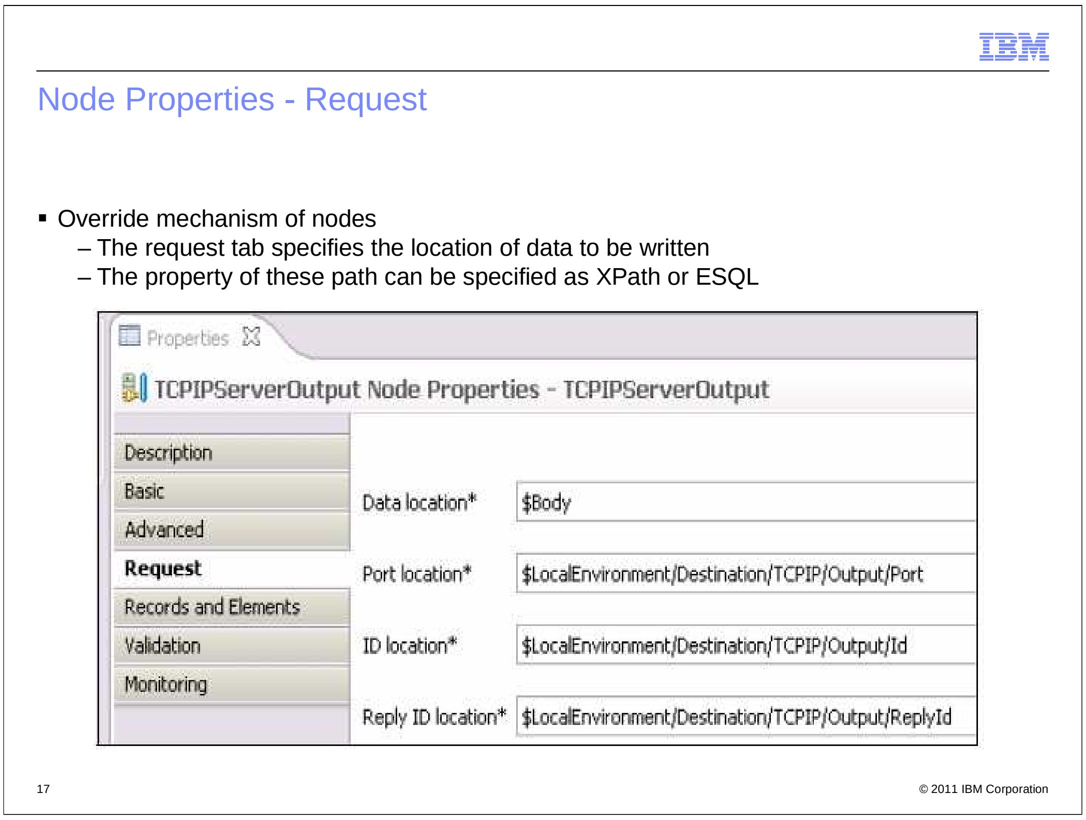
The second section refers to the Input Stream Modification.

Select the first button, “Leave unchanged”, to leave the input stream as it was when it entered the node. This value is selected by default.

Select the second button to specify that this input stream is returned to the pool and is available for use by any input or receive node.

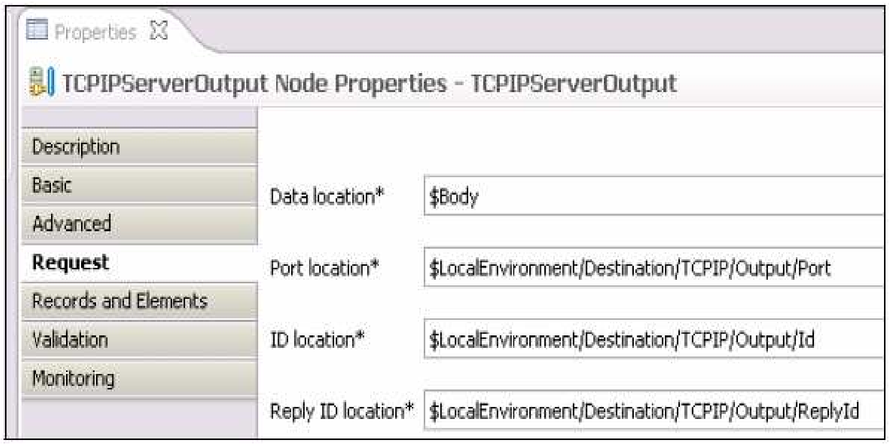
Select the third button to specify that this input stream can be used only by this node and by other input or receive nodes that request it by specifying the connection ID. When the connection input stream is reserved, no other nodes can use it without specifying the correct connection ID.

Select the fourth button to specify that this input stream can be used only by this node and receive nodes that request it by specifying the correct connection ID. After the message has been propagated, this input stream is returned to the pool and becomes available for use by any input or receive node.



Node Properties - Request

- Override mechanism of nodes
  - The request tab specifies the location of data to be written
  - The property of these path can be specified as XPath or ESQL



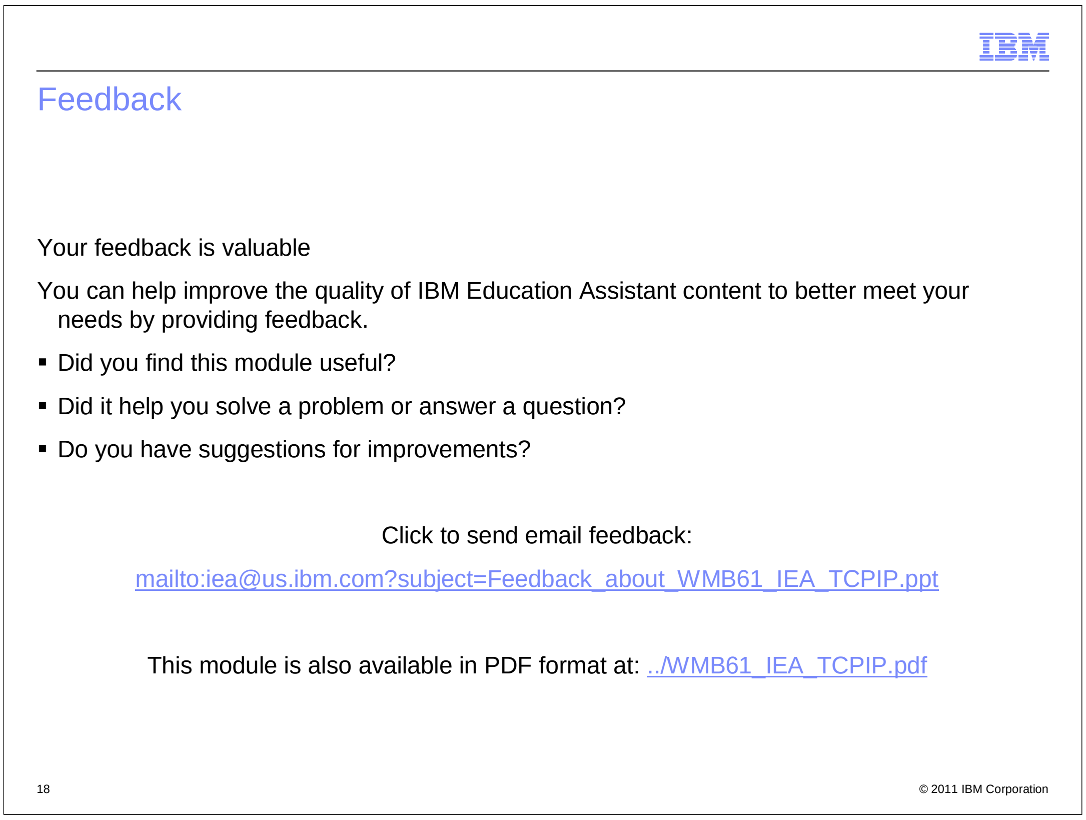
TCPIPServerOutput Node Properties - TCPIPServerOutput

Data location*	$Body
Port location*	$LocalEnvironment/Destination/TCPIP/Output/Port
ID location*	$LocalEnvironment/Destination/TCPIP/Output/Id
Reply ID location*	$LocalEnvironment/Destination/TCPIP/Output/ReplyId

17

The Request properties can be specified for both the TCPIP Output and Receive nodes.

The Request tab specifies the location of data to be written. The value of this path can be specified using XPath or ESQL notation. You can override the port and host name using data in the message or local environment. You can use the connection Id from the local environment to ensure the correct connection is used. You can specify a reply Id field on a connection which is available to any node that uses the same connection later.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

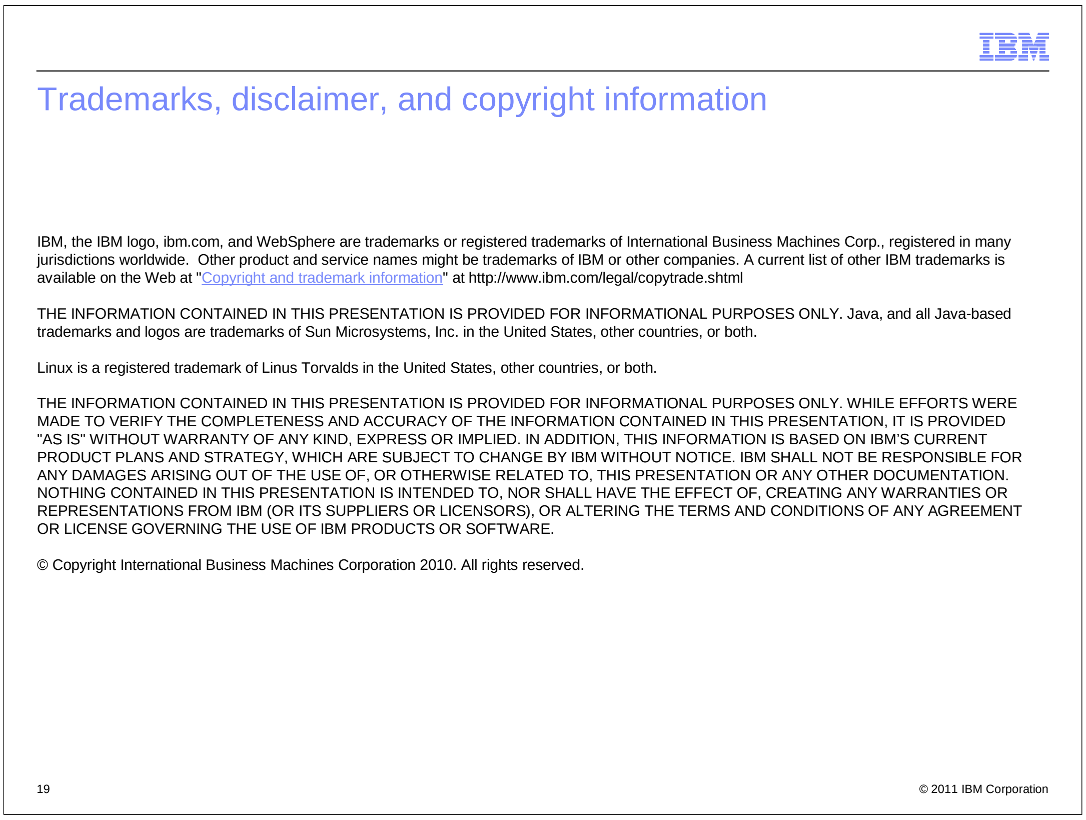
Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WMB61_IEA_TCPIP.ppt

This module is also available in PDF format at: ../WMB61_IEA_TCPIP.pdf

18

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information