WebSphere Message Broker overview

Websphere Message Broker

- WebSphere Message Broker is built upon WebSphere MQ
- An “Advanced ESB” – supports all message formats
- It provides a mechanism for **connecting, routing**, and **transforming** business data from a variety of transports without any need to change the underlying applications generating the data
- Distributes any type of information
- Reduces the number of point-to-point interconnections
- Routes information in real time based on topic and content
- Validates and transforms messages in-flight
- Routes messages based on (evaluated) business rules
- Improves business agility by dynamically reconfiguring information distribution
- Access control to securely deliver personalized information

General Description

Capabilities

- **Message Routing**
  - Message routed from sender to recipient based on the content of the messages
  - business rules based on matching the content of the message
  - capabilities of WebSphere MQ by adding support for other protocols, including real-time Internet, intranet, and multicast endpoints.

- **Message Transformation/Enrichment**
  - between apps to use different formats (e.g. custom format to XML)
  - enriched by integration with multiple sources of data
  - complex manipulation of message data can be performed using the Message Brokers Toolkit (ESQL and Java)
  - Provides adapters nodes for SAP, Siebel and Peoplesoft

- **Publish/Subscribe (Content Based Filtering)**
  - Most Pub/Sub capability is now contained with WMQ v7.0.1

New Broker 6.1 Configurations

#1
WMB Trial

- Free product (valid for 90 days)
- Downloadable from IBM site
- Full WMB palette
- Limited to one Execution Group and 8 message Flows
- Windows or Linux only
- Time bombed to stop working 3 months after install
- Upgradeable to WMB Getting Started Edition or WMB Enterprise Edition
- No re-install necessary – maintains configuration file after application of license file

#4
WMB Enterprise

- Normal full WMB configuration
- Can also be positioned as a shared services ASP model exploiting Broker accounting and statistics

#2
WMB for Remote Adapter Deployment

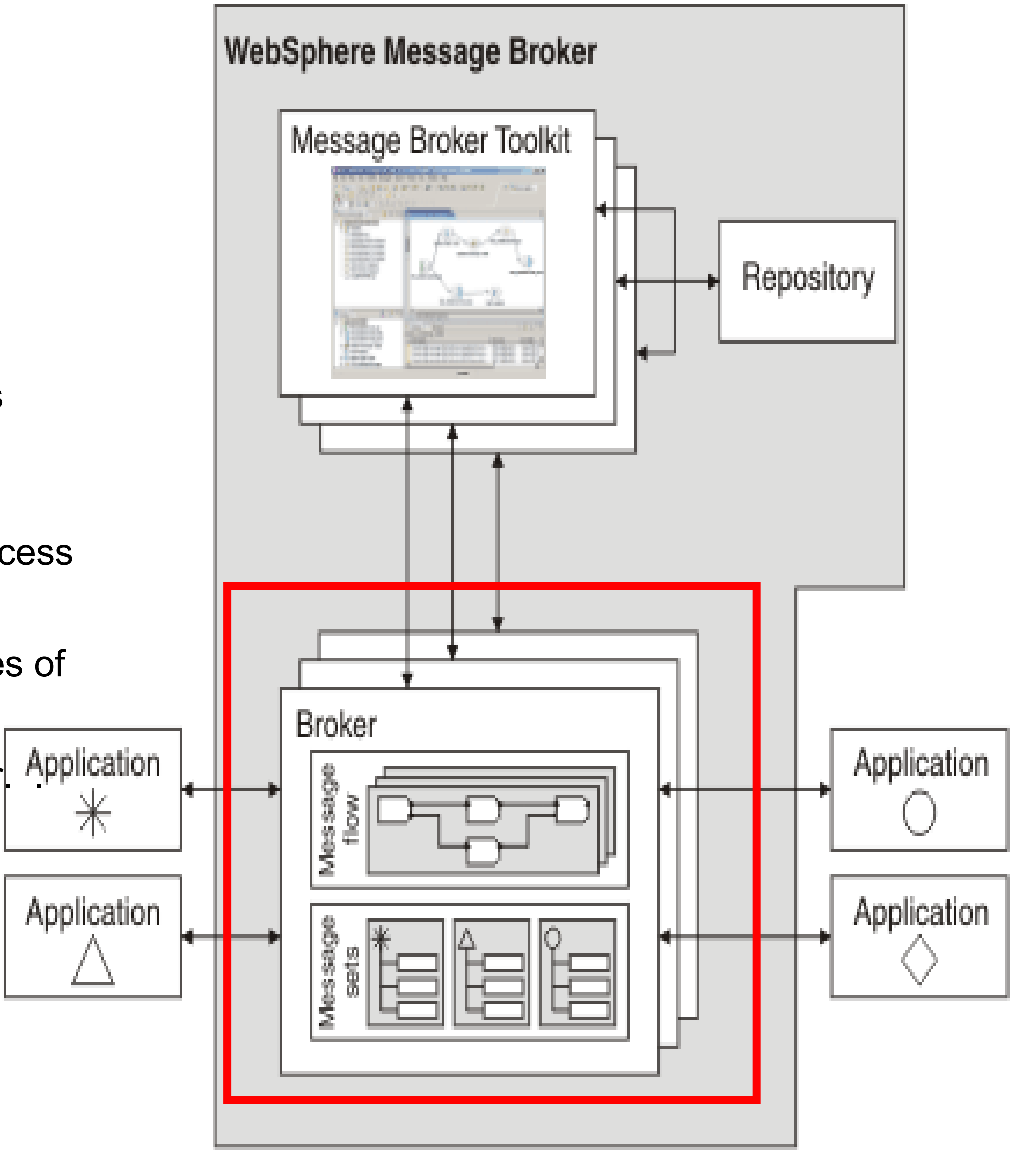
- Physical restrictions
  - Limited WMB palette
  - All input output nodes
  - Java Compute node and JDBC.
  - Must be licensed for JCA adapter
  - 2 execution groups
  - Unlimited flows
- Upgrades
  - No special price discount to upgrade to WMB Getting Started or Full WMB
  - Can be upgrade to WMB Enterprise without re-installing

#3
WMB for Getting Started

- Physical restrictions
  - Full WMB palette
  - 10 message Flows per Execution Group limit
  - 2 exec. group limit
- Upgrades
  - Can be upgraded to a WMB Enterprise license
  - Incentive to get customer to move to unlimited Broker
  - Provides financial model in Enterprise to help Central IT get LOB to adopt a central infrastructure
  - Normal maintenance price for WMB Getting Started = 20% of the entitled license purchase price ($10.5K/year (after first year)).
- Business partners
  - Expect partners/IBM SMB account teams to offer WMB Getting Started as part of a bundle that includes both Education and Services
- Distributed platforms only
  - Getting Started Edition for zOS will be handled by Sub LPAR Licensing.

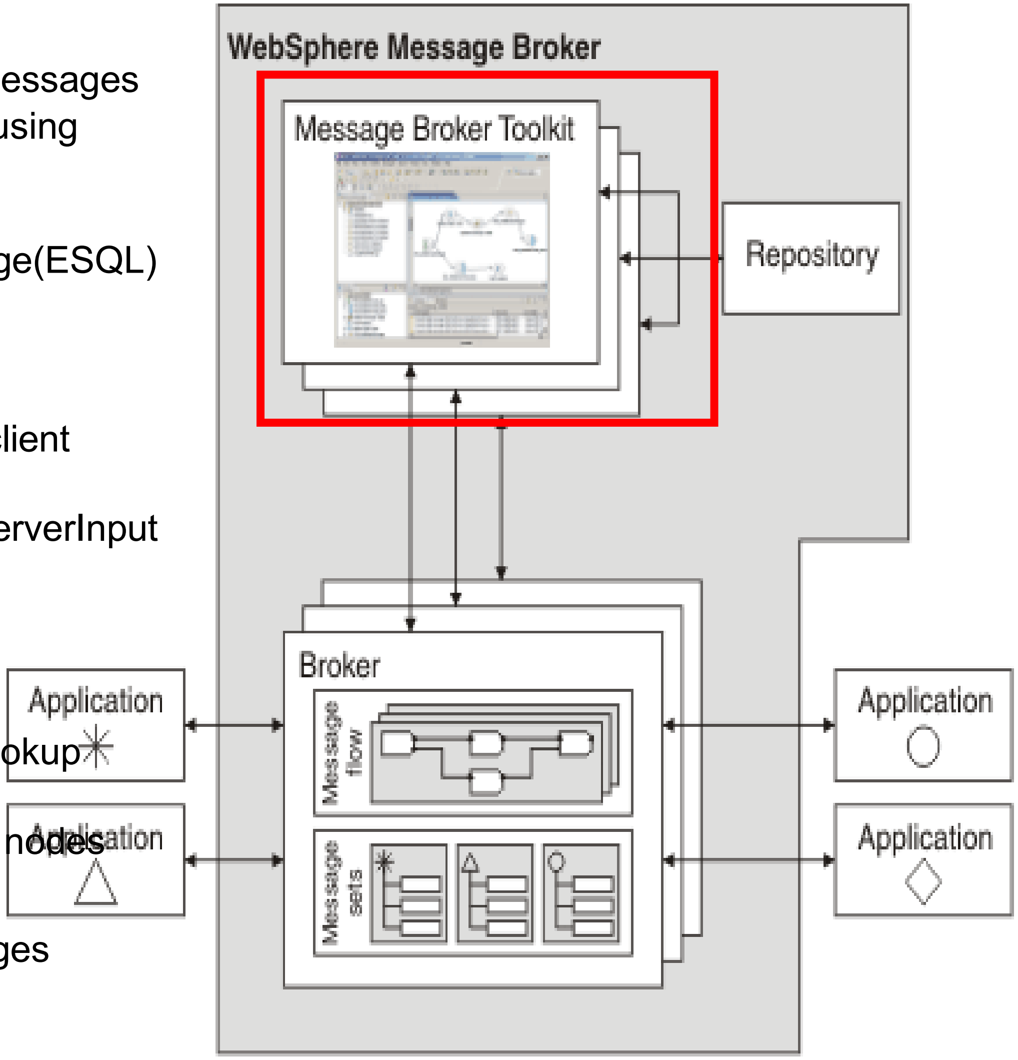
Components - Broker

- The Broker is the runtime component of Websphere Message Broker
- Runs as a system service on Windows
- Controls processes called Execution Groups which run message flows
- Execution Groups run as processes and message flows run as threads within the process
- Administrators decide how many Execution Groups are required and how many instances of a message flow to run.
- The Broker runs on top of a Queue Manager. .



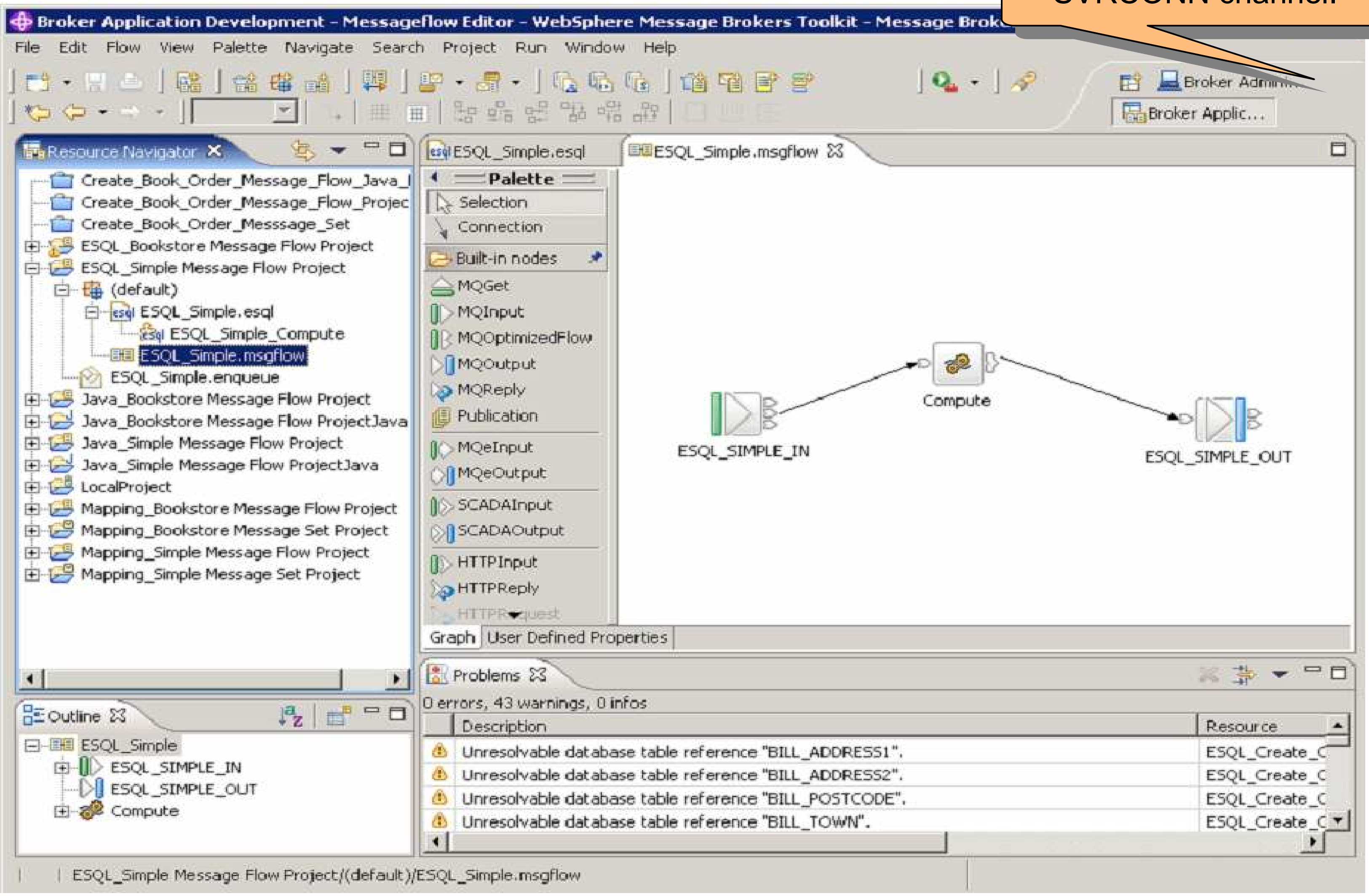
Components – Toolkit (Development environment)

- **Message flows**
  - logic that the broker uses to process messages
  - A predefined sequence of operations (using nodes)
  - methods for defining the logic
    - Extended Structured Query Language(ESQL)
    - Java
  - Input to message flows:
    - MQSeries Queues (MQInput node)
    - HTTP (HTTPInput node) –from ws client
    - Files (FileInput Node)
    - TCP sockets (TCPIPClient/TCPIPServerInput nodes
    - JMS messages (JMSInput node)
    - Adapter nodes
    - SOAP (SOAInput node)
    - WSRR (EndpointLookup/RegistryLookup nodes)
  - Output nodes – similar options to input nodes
- **Message sets**
  - definition of the structure of the messages
- **BAR Files**
  - Broker Archive files are used to package up the artifacts for deployment e.g message flows, message sets.



Components Description – WMB Toolkit

The Toolkit runs as an WMQ Java application and connects into the Broker's Queue Manager via a SVRCONN channel.

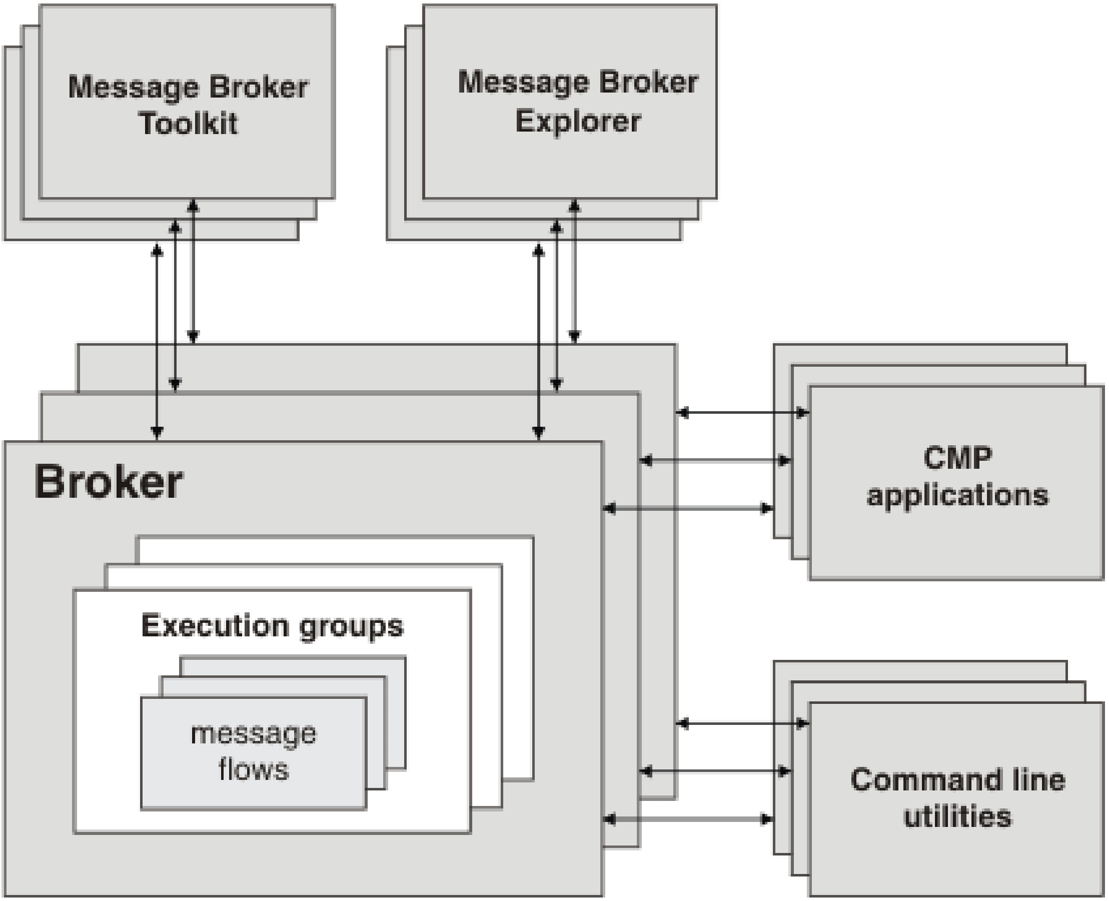

Broker Application Development - Messageflow Editor - WebSphere Message Brokers Toolkit - Message Brok
File Edit Flow View Palette Navigate Search Project Run Window Help
Broker Admin...
Broker Applic...
Resource Navigator
Create_Book_Order_Message_Flow_Java_I
Create_Book_Order_Message_Flow_Projec
Create_Book_Order_Messsage_Set
ESQL_Bookstore Message Flow Project
ESQL_Simple Message Flow Project
(default)
ESQL_Simple.esql
ESQL_Simple_Compute
ESQL_Simple.msgflow
ESQL_Simple.enqueue
Java_Bookstore Message Flow Project
Java_Bookstore Message Flow ProjectJava
Java_Simple Message Flow Project
Java_Simple Message Flow ProjectJava
LocalProject
Mapping_Bookstore Message Flow Project
Mapping_Bookstore Message Set Project
Mapping_Simple Message Flow Project
Mapping_Simple Message Set Project
Outline
ESQL_Simple
ESQL_SIMPLE_IN
ESQL_SIMPLE_OUT
Compute
ESQL_Simple.esql
ESQL_Simple.msgflow
Palette
Selection
Connection
Built-in nodes
MQGet
MQInput
MQOptimizedFlow
MQOutput
MQReply
Publication
MQeInput
MQeOutput
SCADAInput
SCADAOutput
HTTPInput
HTTPReply
HTTPRequest
Compute
ESQL_SIMPLE_IN
ESQL_SIMPLE_OUT
Graph
User Defined Properties
Problems

0 errors, 43 warnings, 0 infos

Description	Resource
Unresolvable database table reference "BILL_ADDRESS1".	ESQL_Create_C
Unresolvable database table reference "BILL_ADDRESS2".	ESQL_Create_C
Unresolvable database table reference "BILL_POSTCODE".	ESQL_Create_C
Unresolvable database table reference "BILL_TOWN".	ESQL_Create_C

ESQL_Simple Message Flow Project/(default)/ESQL_Simple.msgflow

Administration Options



Installation – Pre-Reqs

- **WebSphere MQ v7.0.1**
- **WebSphere MQ Explorer**
- **WebSphere Eclipse Platform v3.3**
- Optional Software
  - Database (User Databases)

    DB2

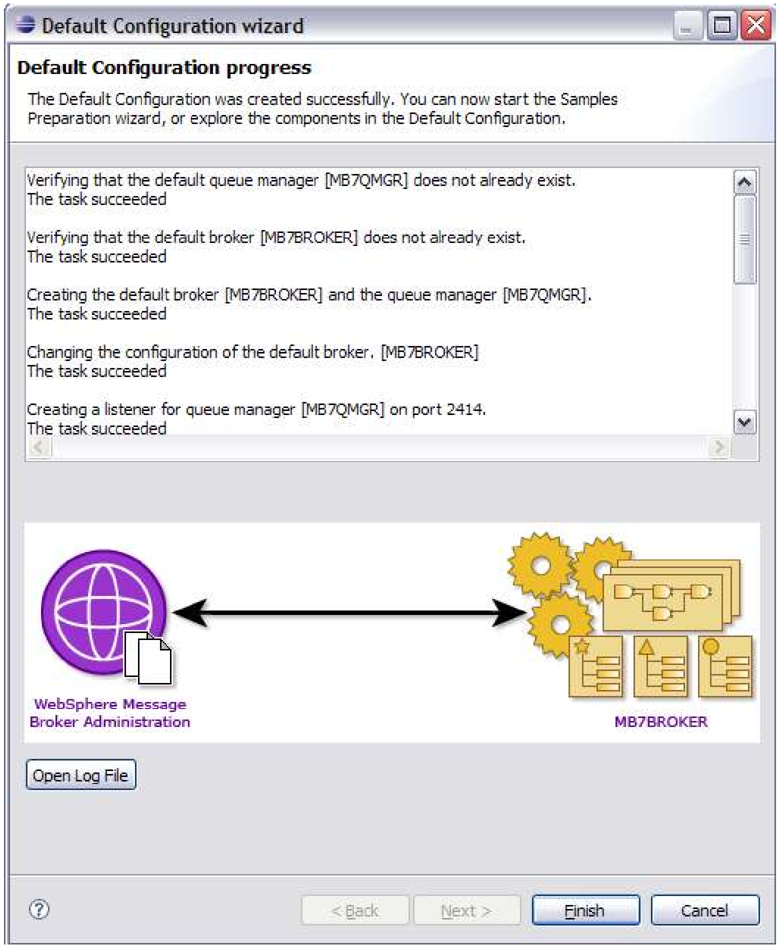
    Microsoft SQL Server

    Oracle

    Sybase

    Informix

Default Configuration Wizard



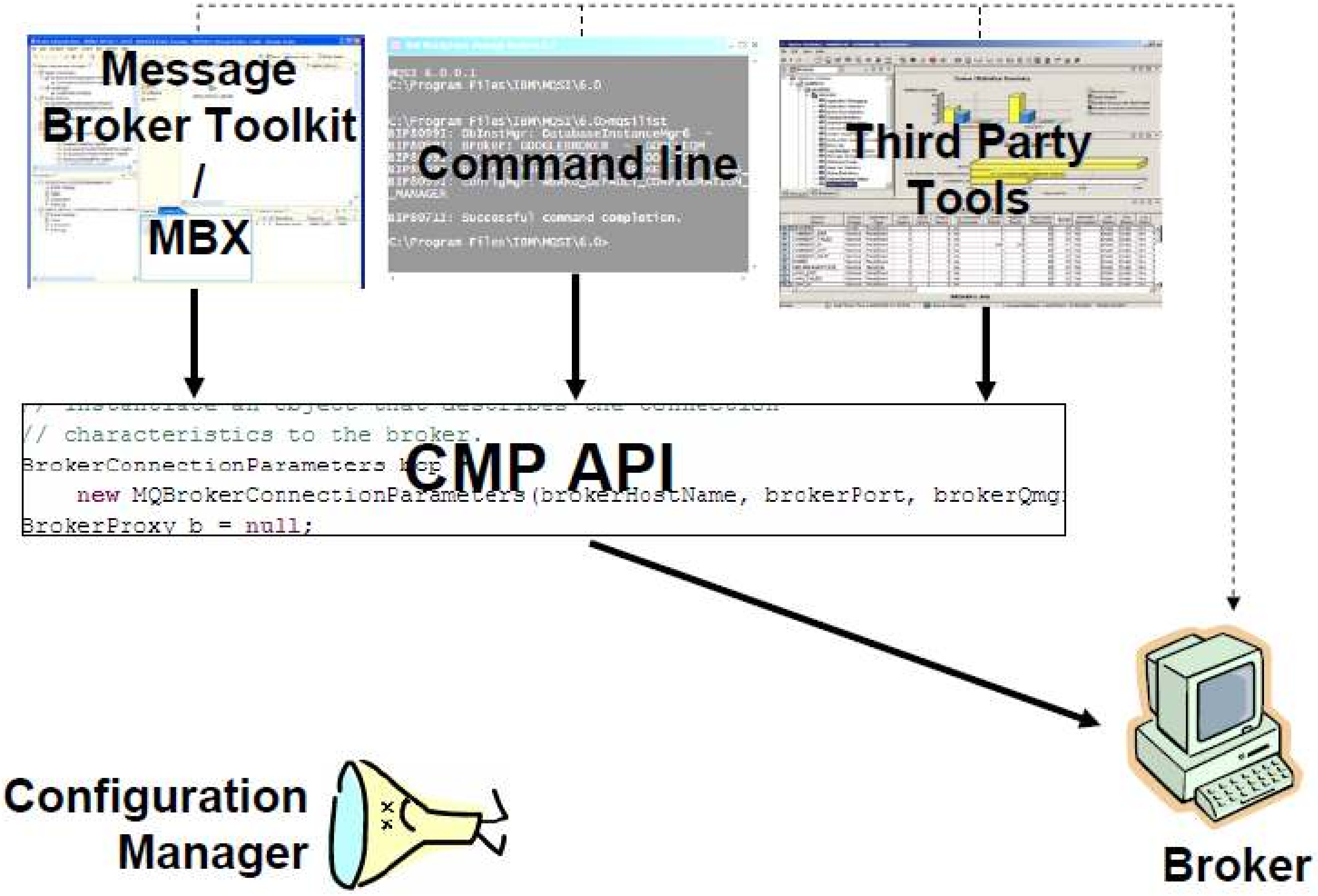
Default Configuration progress

The Default Configuration was created successfully. You can now start the Samples Preparation wizard, or explore the components in the Default Configuration.

```
Verifying that the default queue manager [MB7QMGR] does not already exist.
The task succeeded

Verifying that the default broker [MB7BROKER] does not already exist.
The task succeeded

Creating the default broker [MB7BROKER] and the queue manager [MB7QMGR].
The task succeeded

Changing the configuration of the default broker. [MB7BROKER]
The task succeeded

Creating a listener for queue manager [MB7QMGR] on port 2414.
The task succeeded
```

- Creates a default Broker called “MB7BROKER” and a Queue Manager called “MB7QMGR”
- Used by the samples shipped with the product.
- On Windows the Broker is created with the service id “Local System”

Administration architecture

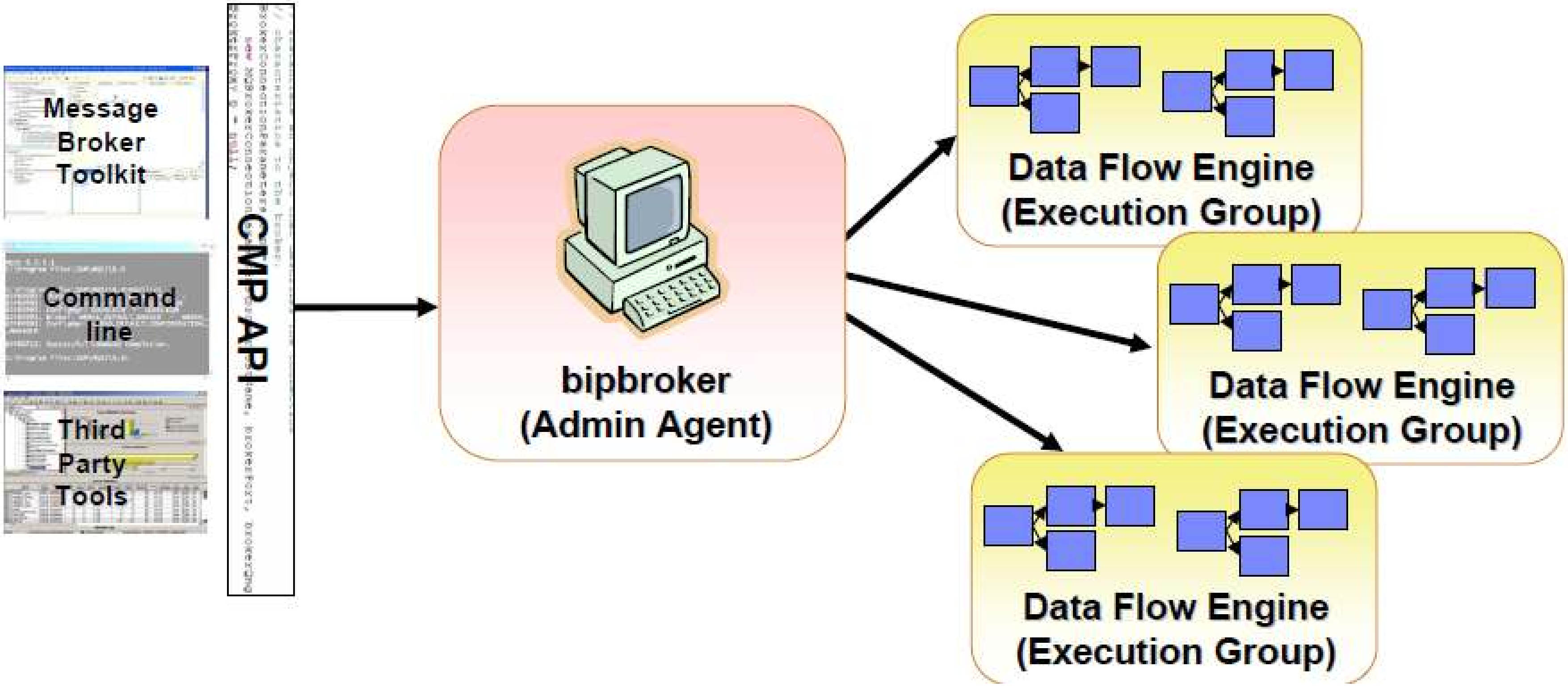


```
MQSI 6.0.0.1
C:\Program Files\IBM\MQSI\6.0

C:\Program Files\IBM\MQSI\6.0>mqsilist
BIP8099I: DbInstMgr: DatabaseInstanceMgr6  -
BIP8099I: Broker: GOODLEBROKER  - [illegible]
BIP[illegible]
BIP[illegible]
BIP8099I: ConfigMgr: [illegible]
_MANAGER

BIP8071I: Successful command completion.

C:\Program Files\IBM\MQSI\6.0>
```

```
// Instantiate an object that describes the connection
// characteristics to the broker.
BrokerConnectionParameters bcp
    new MQBrokerConnectionParameters(brokerHostName, brokerPort, brokerQmgr
BrokerProxy b = null;
```

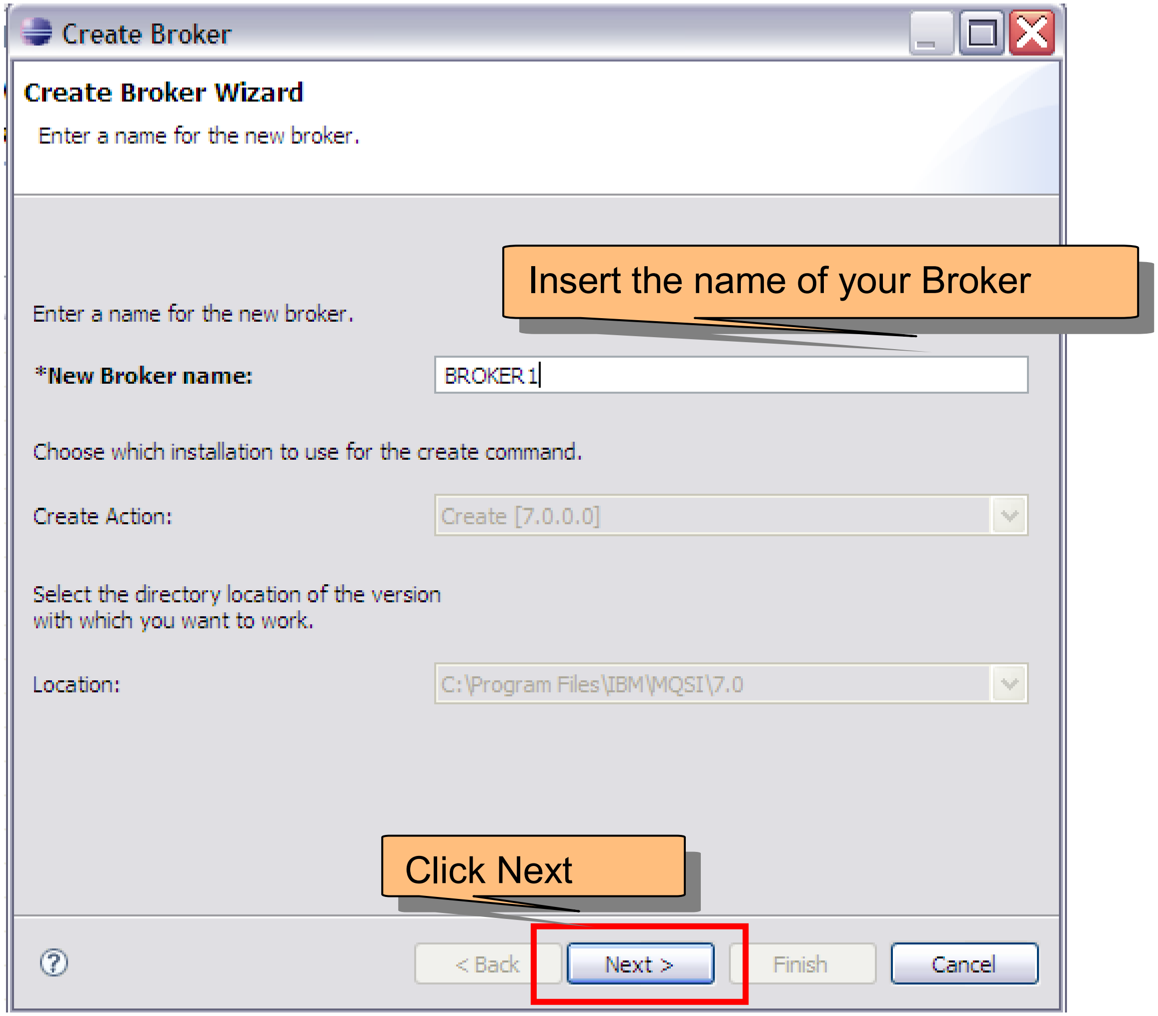
Administration architecture

The admin agent is the main broker process that directs execution groups


Message Broker Toolkit
Command line
Third Party Tools
CMP API
bipbroker (Admin Agent)
Data Flow Engine (Execution Group)
Data Flow Engine (Execution Group)
Data Flow Engine (Execution Group)

Creating a Broker in MBX - Walkthrough

In the Navigator View of MBX go to **Brokers → New → Local Broker**



Create Broker

Create Broker Wizard

Enter a name for the new broker.

Enter a name for the new broker.

***New Broker name:** BROKER1

Choose which installation to use for the create command.

Create Action: Create [7.0.0.0]

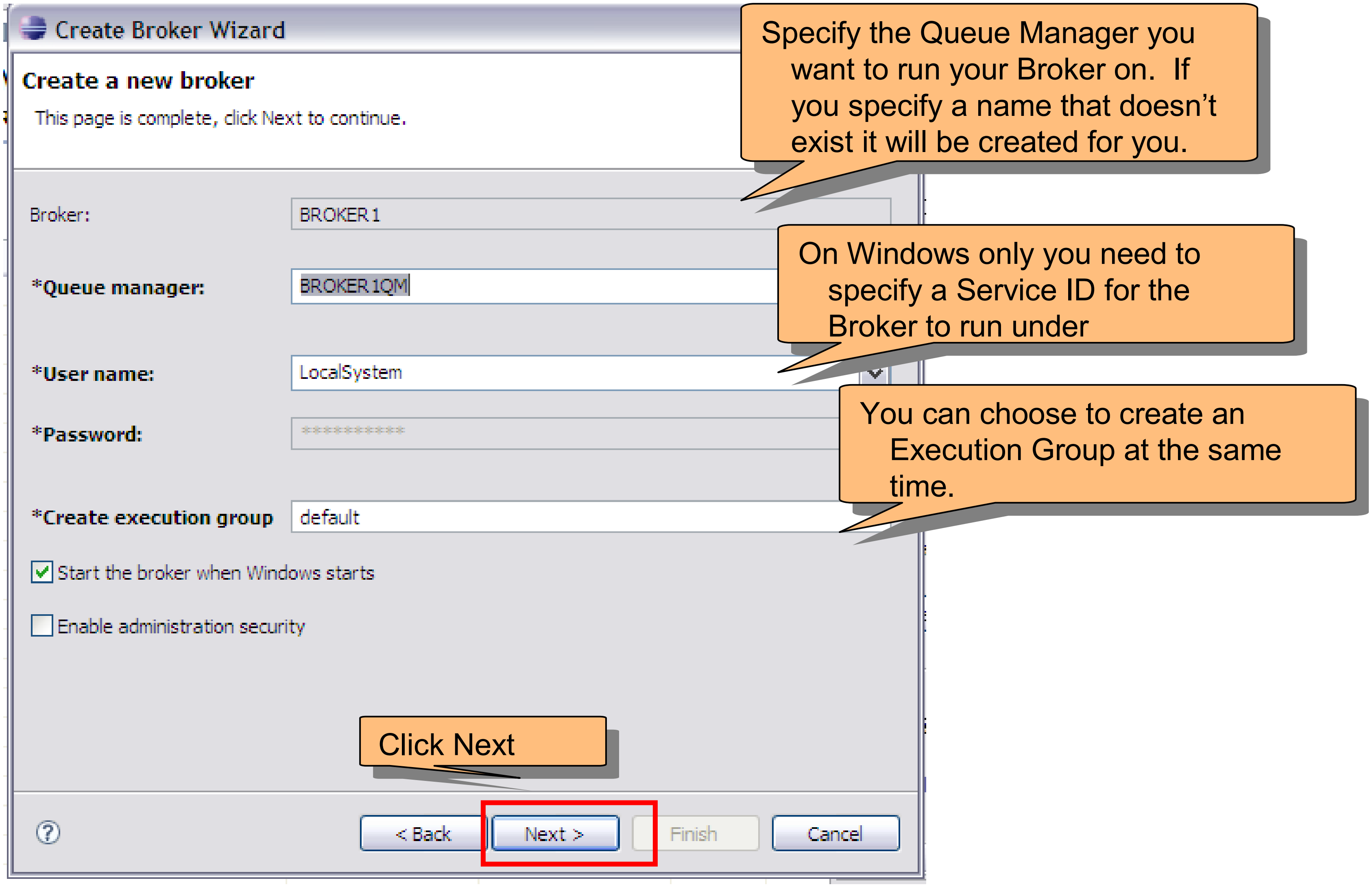
Select the directory location of the version with which you want to work.

Location: C:\Program Files\IBM\MQSI\7.0

Insert the name of your Broker

Click Next

Create a Broker in MBX - Walkthrough



Create Broker Wizard

Create a new broker

This page is complete, click Next to continue.

Broker: BROKER1

***Queue manager:** BROKER1QM

***User name:** LocalSystem

Password:** ********

***Create execution group** default

☑ Start the broker when Windows starts

☐ Enable administration security

Specify the Queue Manager you want to run your Broker on. If you specify a name that doesn't exist it will be created for you.

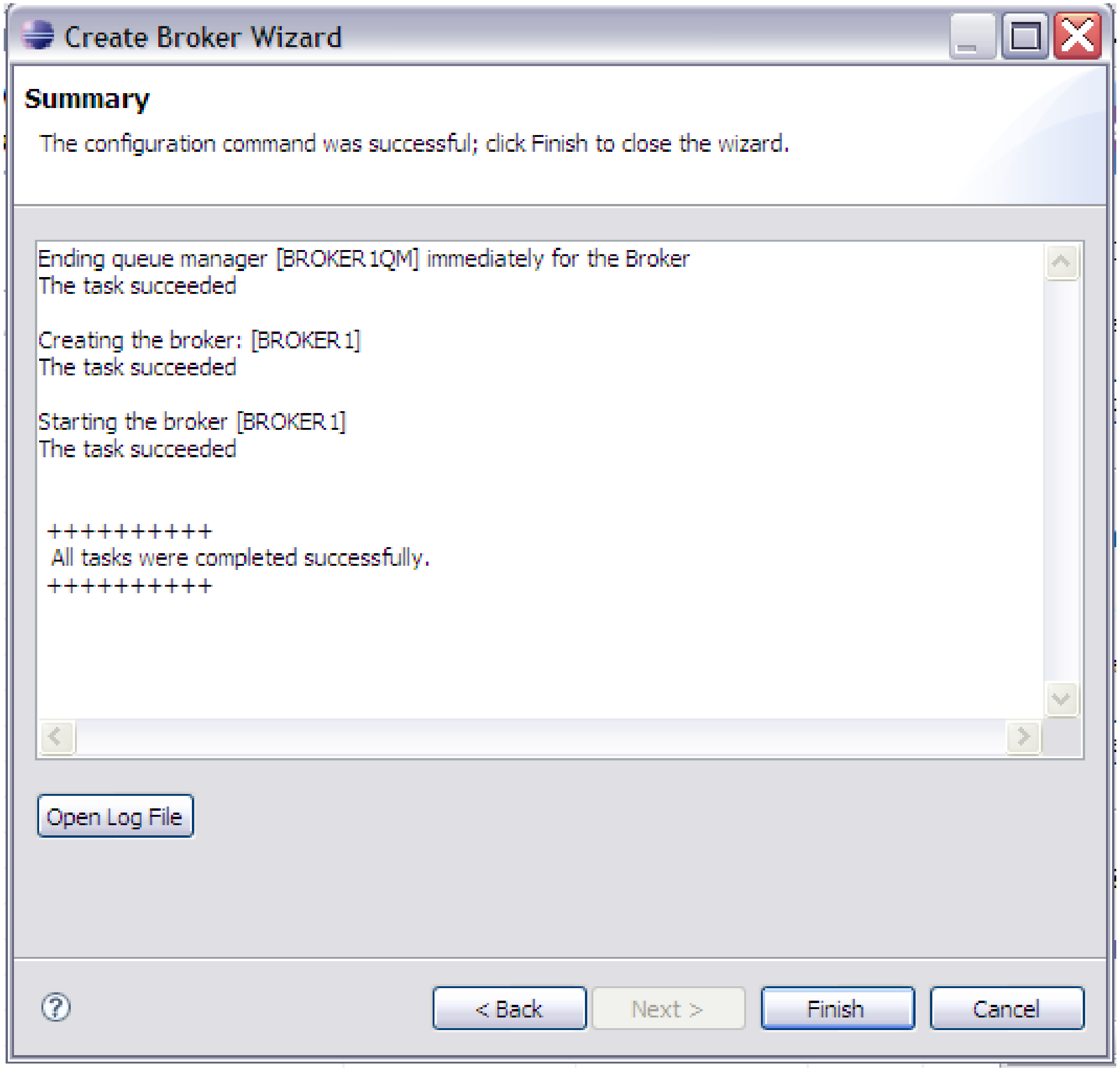
On Windows only you need to specify a Service ID for the Broker to run under

You can choose to create an Execution Group at the same time.

Click Next

© 2009 IBM Corporation

Creating a Broker in MBX - Walkthrough



Summary

The configuration command was successful; click Finish to close the wizard.

```
Ending queue manager [BROKER1QM] immediately for the Broker
The task succeeded

Creating the broker: [BROKER1]
The task succeeded

Starting the broker [BROKER1]
The task succeeded


 ++++++++++
 All tasks were completed successfully.
 ++++++++++
```

Start, stop and display components

- **MB Explorer**
- **Command Environment:**

1. Display components

- Mqsilist
- Mqsilist <brokername>
- Mqsilist <brokername> -e <executiongroupname>
- Mqsiservice <brokername>
- Mqsireportbroker <brokername>

2. Start / stop Broker

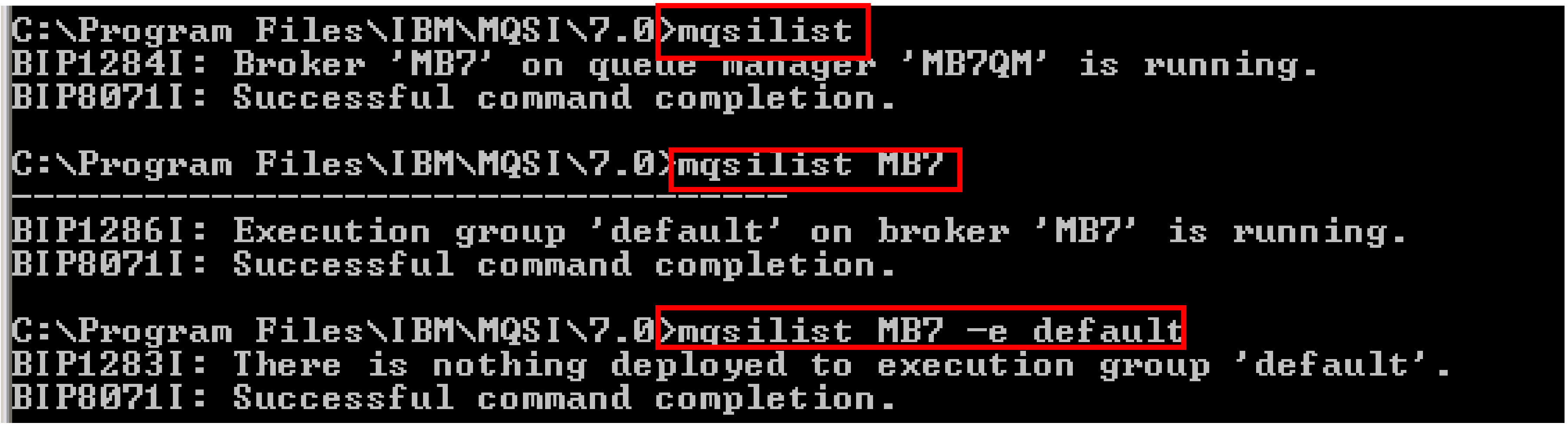
- Mqsistart <Brokername>
- Mqsistop <brokername>

Nb use –i for immediate, force stop a controlled stop fails to complete
Tips: Always check the event log after start/stop to ensure it has completed successfully

- **Windows Services**
  - The Broker and it’s Queue Manager can be configured to start automatically as a service.

mqsilist

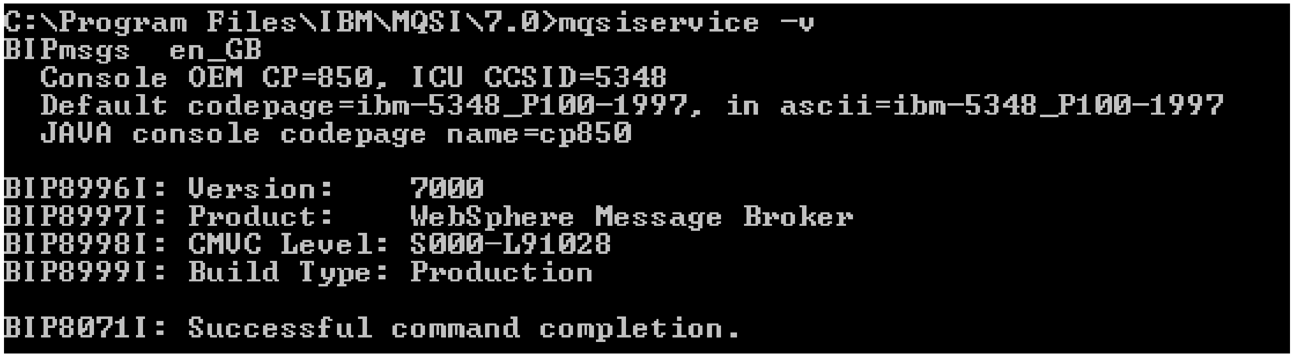
- Displays the status of Brokers and Execution Groups



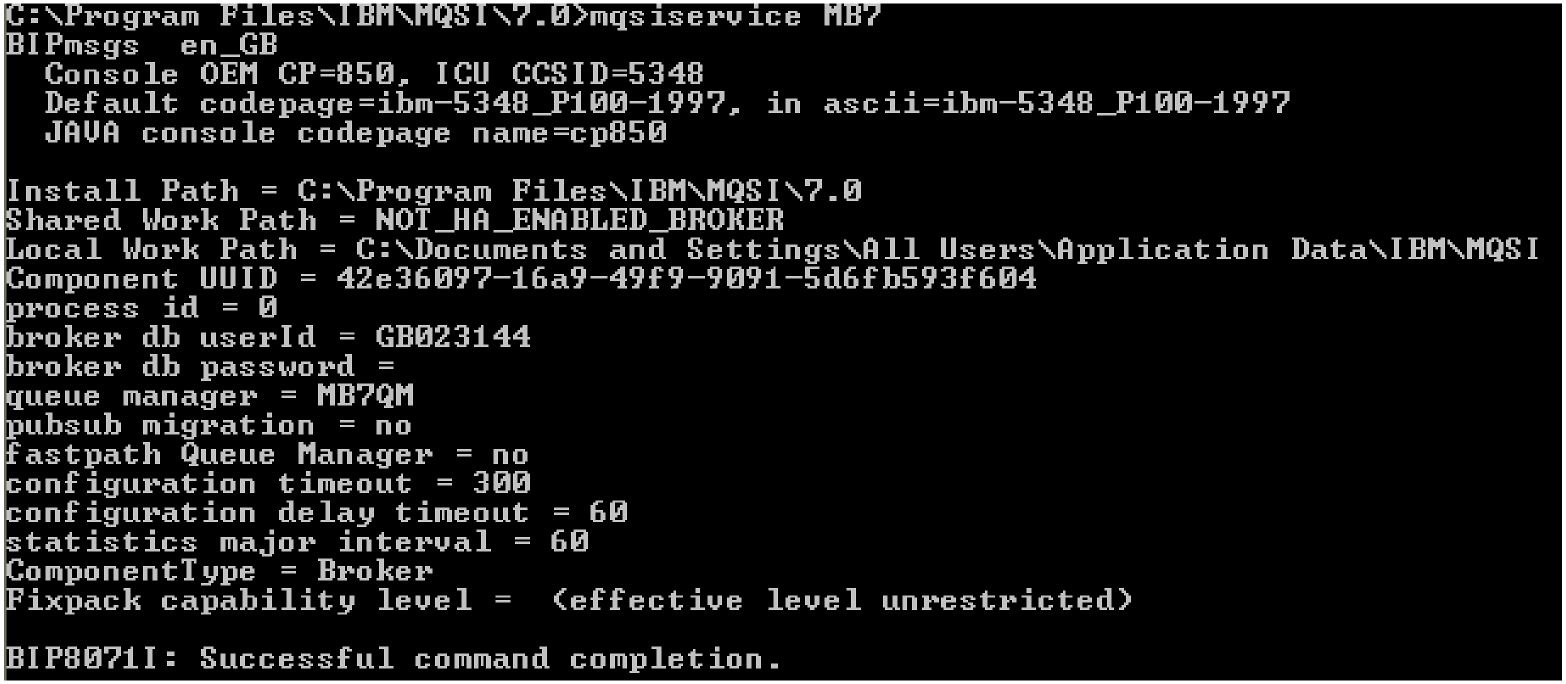
```
C:\Program Files\IBM\MQSI\7.0>mqsilist
BIP1284I: Broker 'MB7' on queue manager 'MB7QM' is running.
BIP8071I: Successful command completion.

C:\Program Files\IBM\MQSI\7.0>mqsilist MB7
-----------------------------------
BIP1286I: Execution group 'default' on broker 'MB7' is running.
BIP8071I: Successful command completion.

C:\Program Files\IBM\MQSI\7.0>mqsilist MB7 -e default
BIP1283I: There is nothing deployed to execution group 'default'.
BIP8071I: Successful command completion.
```

mqsiservice

- Undocumented command (liable to change)
- Displays information about your installation and defined Brokers



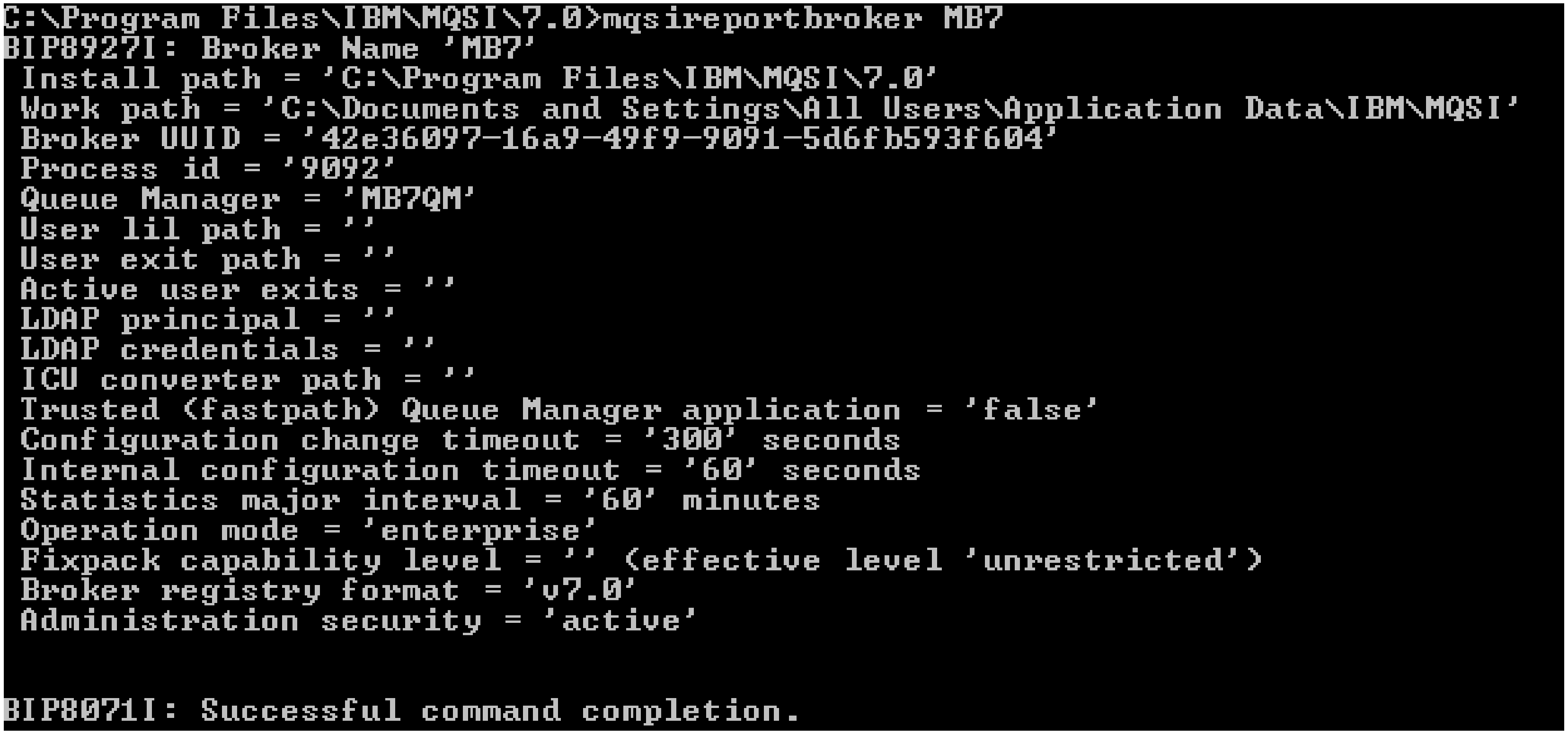
```
C:\Program Files\IBM\MQSI\7.0>mqsiservice -v
BIPmsgs  en_GB
  Console OEM CP=850, ICU CCSID=5348
  Default codepage=ibm-5348_P100-1997, in ascii=ibm-5348_P100-1997
  JAVA console codepage name=cp850

BIP8996I: Version:    7000
BIP8997I: Product:    WebSphere Message Broker
BIP8998I: CMVC Level: S000-L91028
BIP8999I: Build Type: Production

BIP8071I: Successful command completion.
```



```
C:\Program Files\IBM\MQSI\7.0>mqsiservice MB7
BIPmsgs  en_GB
  Console OEM CP=850, ICU CCSID=5348
  Default codepage=ibm-5348_P100-1997, in ascii=ibm-5348_P100-1997
  JAVA console codepage name=cp850

Install Path = C:\Program Files\IBM\MQSI\7.0
Shared Work Path = NOT_HA_ENABLED_BROKER
Local Work Path = C:\Documents and Settings\All Users\Application Data\IBM\MQSI
Component UUID = 42e36097-16a9-49f9-9091-5d6fb593f604
process id = 0
broker db userId = GB023144
broker db password =
queue manager = MB7QM
pubsub migration = no
fastpath Queue Manager = no
configuration timeout = 300
configuration delay timeout = 60
statistics major interval = 60
ComponentType = Broker
Fixpack capability level =  (effective level unrestricted)

BIP8071I: Successful command completion.
```

mqsireportbroker

- Displays the Brokers registry entries – similar output to mqsiservice <brokername>
- Documented interface

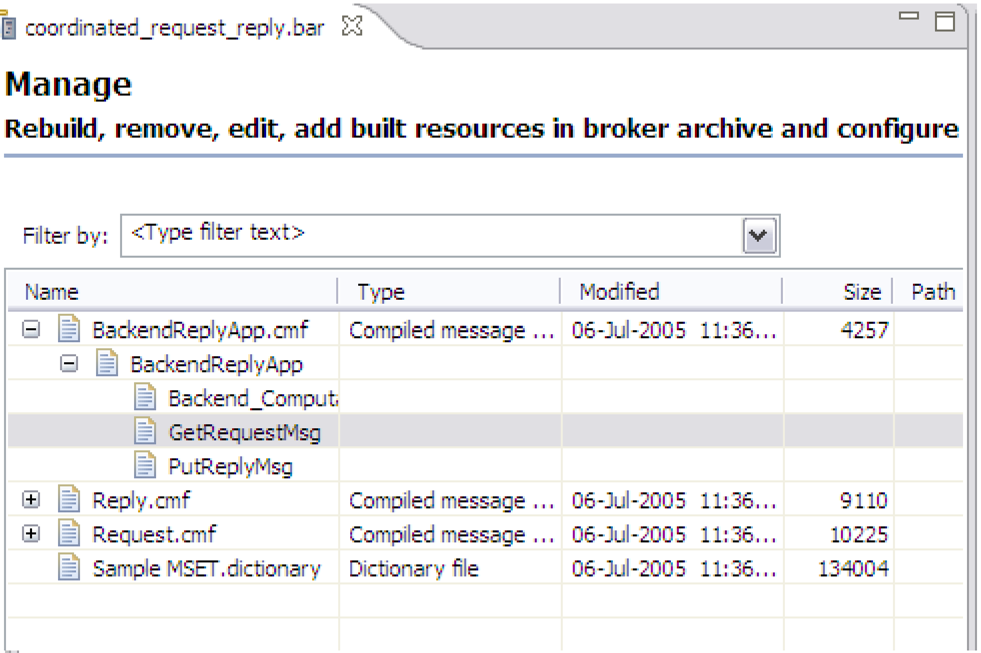


```
C:\Program Files\IBM\MQSI\7.0>mqsireportbroker MB7
BIP8927I: Broker Name 'MB7'
 Install path = 'C:\Program Files\IBM\MQSI\7.0'
 Work path = 'C:\Documents and Settings\All Users\Application Data\IBM\MQSI'
 Broker UUID = '42e36097-16a9-49f9-9091-5d6fb593f604'
 Process id = '9092'
 Queue Manager = 'MB7QM'
 User lil path = ''
 User exit path = ''
 Active user exits = ''
 LDAP principal = ''
 LDAP credentials = ''
 ICU converter path = ''
 Trusted (fastpath) Queue Manager application = 'false'
 Configuration change timeout = '300' seconds
 Internal configuration timeout = '60' seconds
 Statistics major interval = '60' minutes
 Operation mode = 'enterprise'
 Fixpack capability level = '' (effective level 'unrestricted')
 Broker registry format = 'v7.0'
 Administration security = 'active'


BIP8071I: Successful command completion.
```

Customizing BAR files

- Configurable properties of a BAR file
  - Enables administrators to change environment specific properties prior to deployment without having to edit the message flows and rebuild the bar file.
  - Can be done within the broker archive editor in MBX or using the **mqsiapplybaroverride** command
  - Properties are contained within the deployment descriptor for the bar file (broker.xml)

- Example configurable properties
  - Queue names
  - Queue Manager names
  - Database connections



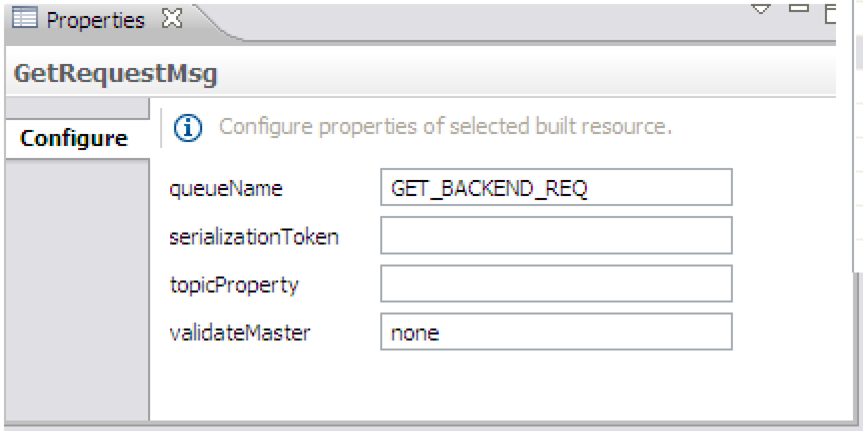
coordinated_request_reply.bar

Manage

Rebuild, remove, edit, add built resources in broker archive and configure

Filter by: <Type filter text>

Name	Type	Modified	Size	Path
⊟ BackendReplyApp.cmf	Compiled message ...	06-Jul-2005 11:36...	4257	
⊟ BackendReplyApp				
Backend_Comput				
GetRequestMsg				
PutReplyMsg				
⊞ Reply.cmf	Compiled message ...	06-Jul-2005 11:36...	9110	
⊞ Request.cmf	Compiled message ...	06-Jul-2005 11:36...	10225	
Sample MSET.dictionary	Dictionary file	06-Jul-2005 11:36...	134004	



Properties

GetRequestMsg

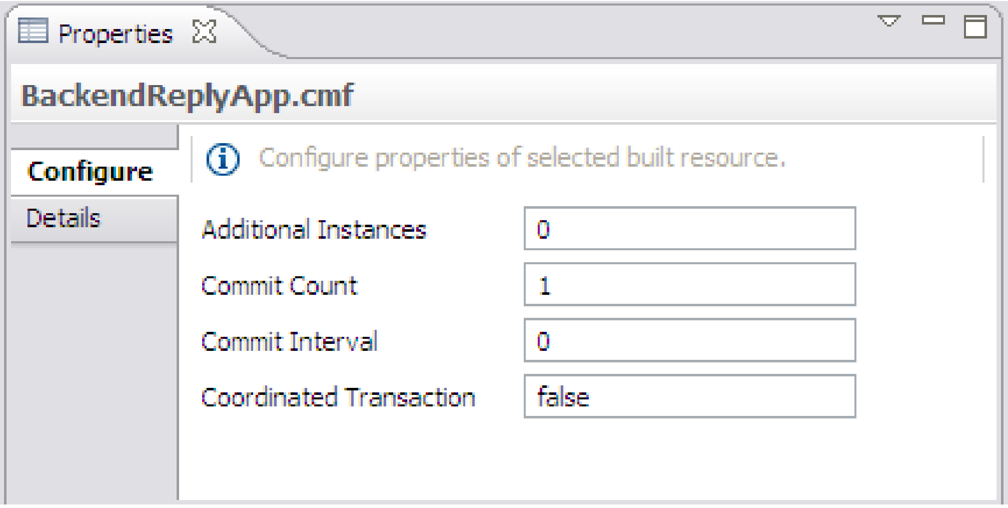
Configure

Configure properties of selected built resource.

queueName	GET_BACKEND_REQ
serializationToken	
topicProperty	
validateMaster	none

Customising BAR files

- Configure BAR files to deploy multiple instances of message flows to an Execution Group
- To deploy multiple instances of the flow with different configurable properties rename the message flow file (.cmf) in the Bar File Editor. Ensure you clear the “remove contents of the Broker archive before building” to prevent the renamed flow from being removed from the bar when building it again. Add the message flow to the BAR file again and edit it as required.
- Other options for scaling throughput are to add additional execution groups or Brokers.



Configure properties of selected built resource.

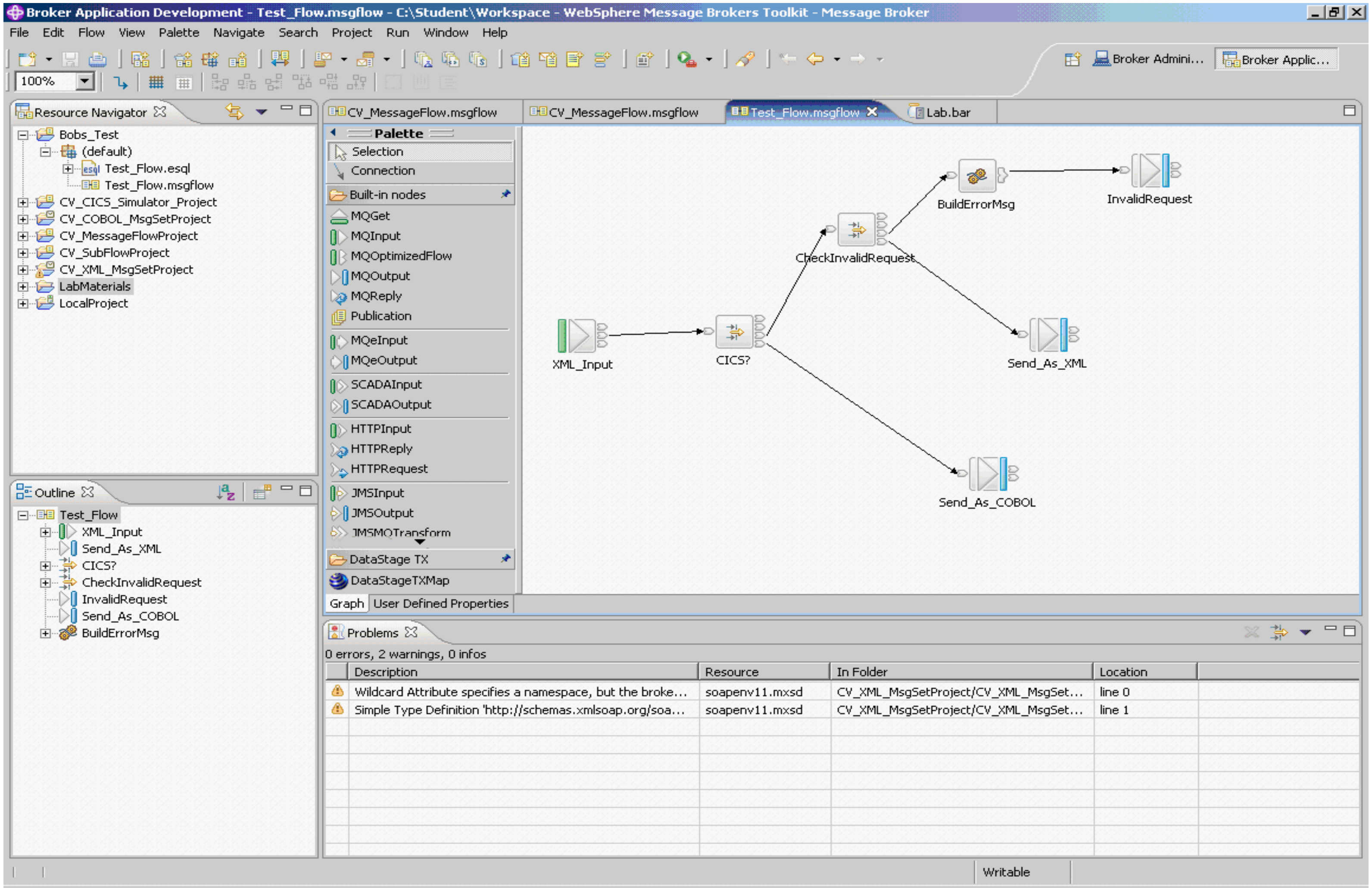
Additional Instances	0
Commit Count	1
Commit Interval	0
Coordinated Transaction	false

Deploying BAR files

- BAR files are deployed to Execution Groups within the Message Broker
- Deployment methods
  - MBX
  - MB Toolkit
  - Mqsideploy command
  - Adminstration API

- If you use MB Toolkit, the MBX of the Admin API the request is **asynchronous** and control is returned to the application immediately
- If you use **mqsideploy** the deployment command is **synchronous** and the command waits for a response
- Check for successful deployment messages in the Event log.
- Preferred methods for deploying/configuring bar files is to use scripts. There are lots of articles on available methods on developerworks e.g:
  - WMB deployment scripting using ANT
    http://www.ibm.com/developerworks/websphere/library/techarticles/0706_spriet/0706_spriet.html
  - Coordinated deployment in WMB v6 using the CMP API:
    http://www.ibm.com/developerworks/websphere/library/techarticles/0611_lucas/0611_lucas.html

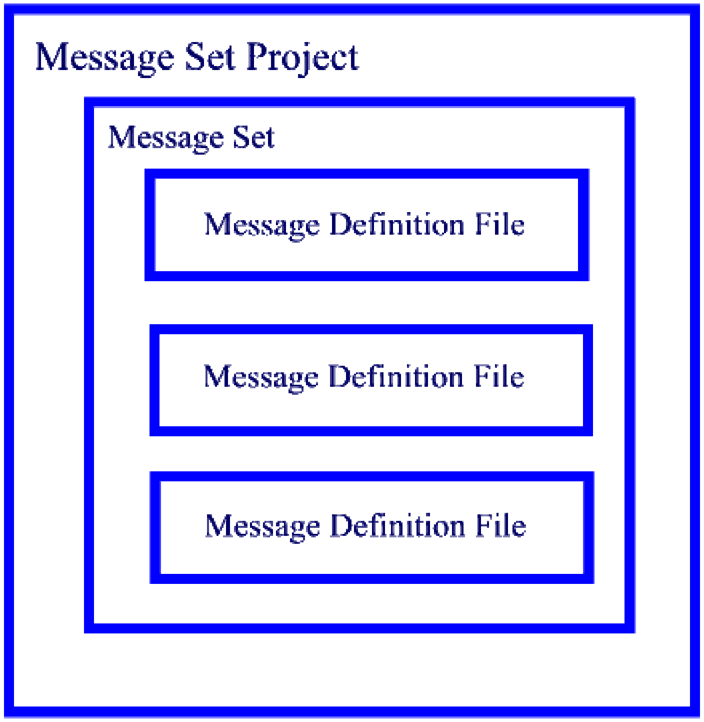
The Message Brokers Toolkit



0 errors, 2 warnings, 0 infos

	Description	Resource	In Folder	Location
⚠	Wildcard Attribute specifies a namespace, but the broke...	soapenv11.mxsd	CV_XML_MsgSetProject/CV_XML_MsgSet...	line 0
⚠	Simple Type Definition 'http://schemas.xmlsoap.org/soa...	soapenv11.mxsd	CV_XML_MsgSetProject/CV_XML_MsgSet...	line 1

Relationship of Message Set Projects/ Sets

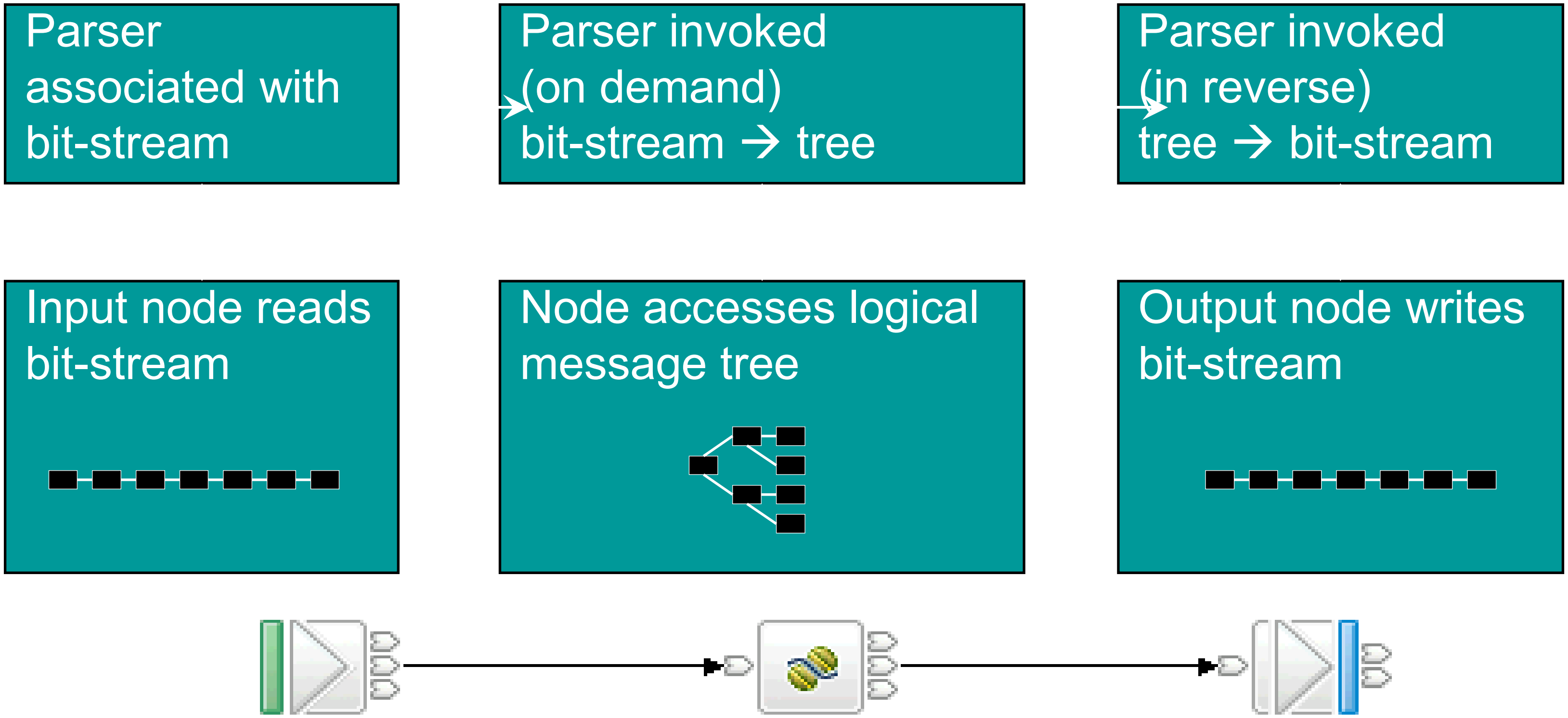


A Message Set Project contains one Message Set

A Message Set can contain multiple Message Definition Files

A Message Definition File contains the information required to parse a particular message

Message Broker Overview – flows, nodes, message tree



Parser associated with bit-stream

Parser invoked (on demand) bit-stream → tree

Parser invoked (in reverse) tree → bit-stream

Input node reads bit-stream

Node accesses logical message tree

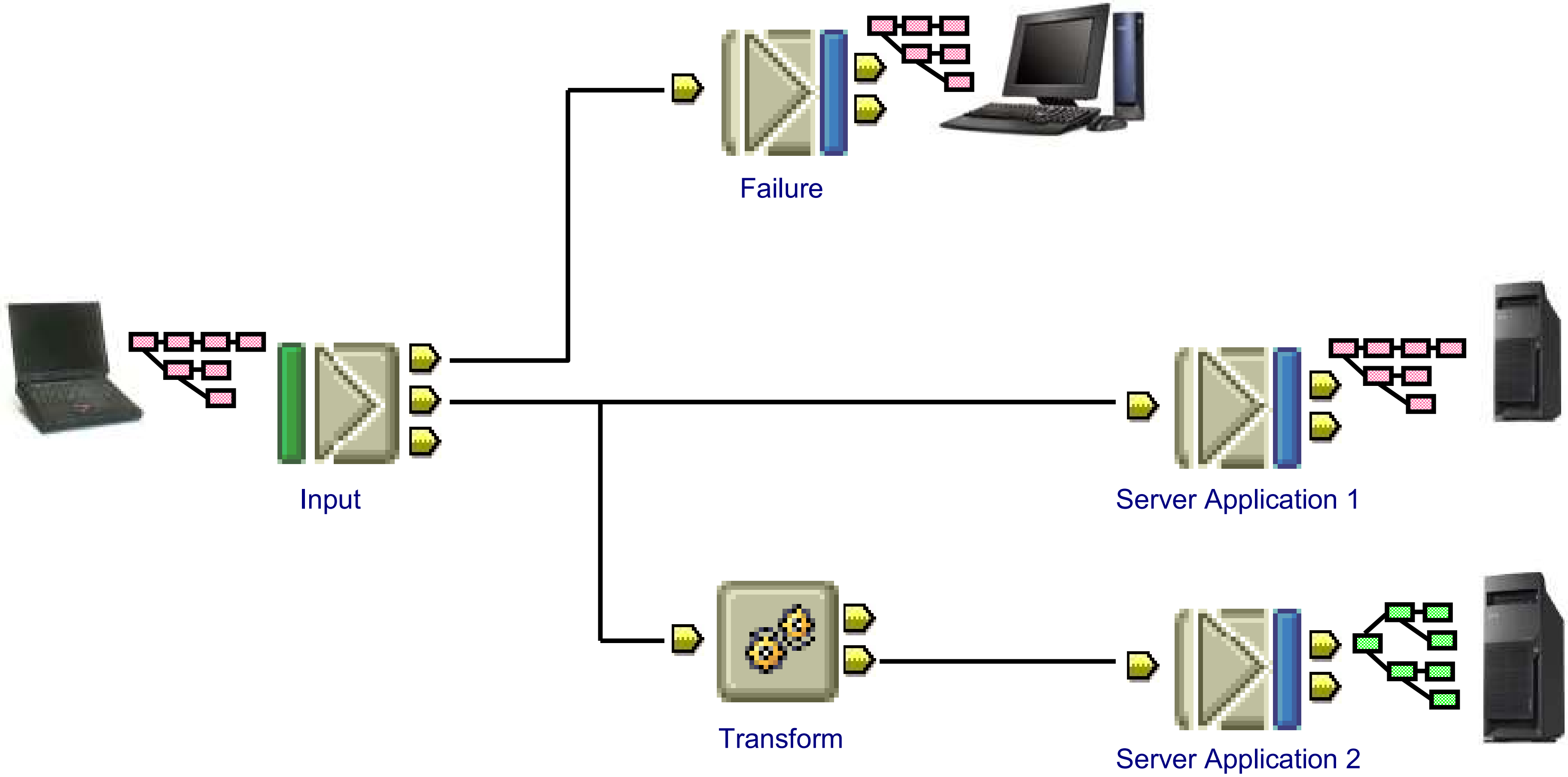
Output node writes bit-stream

Message Routing

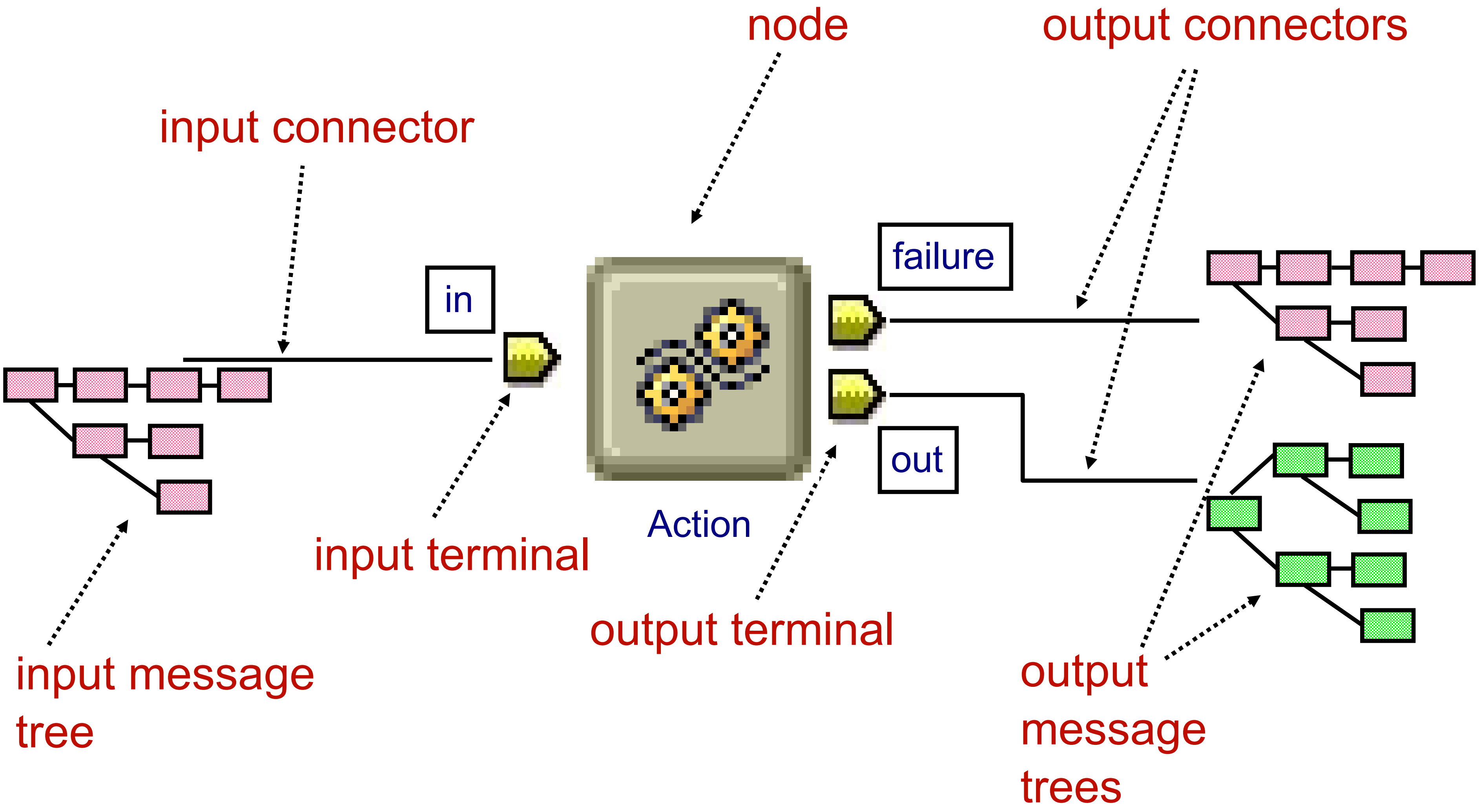
- One of Message Broker’s core strengths is content based routing – the ability to dynamically make a routing decision based on the actual data in a particular message passing through the Flow at that instant

- There are many ways to do Routing in a Flow
  - Filter node
  - Compute node
  - Java Compute node
  - Destination List
  - Route To Label

Message Flows



Message Processing Nodes



WMB- Message Flows

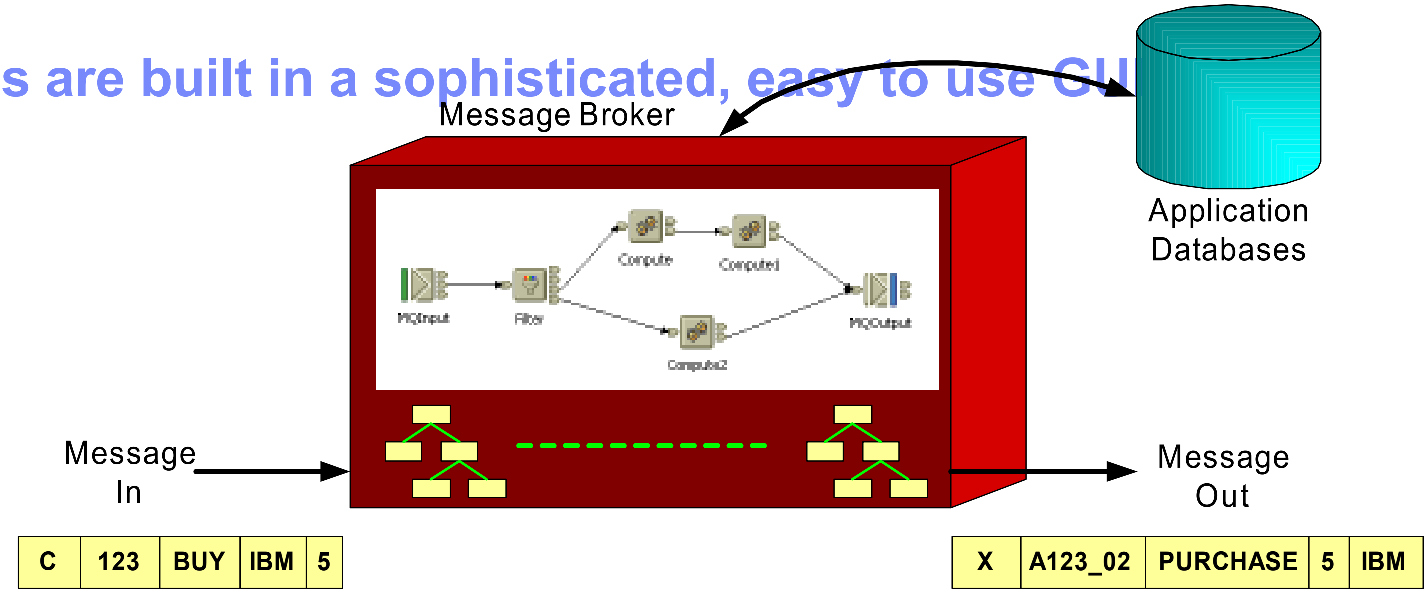
- A sequence of processing steps to transform/route messages as they pass through the Broker

- Created using a palette of built-in functions (“nodes”) that can be “wired” together to form a “message flow”

- Flows can be transactional or non-transactional

- Flows are reusable (can be nested)

- Flows are built in a sophisticated, easy to use GUI



C	123	BUY	IBM	5

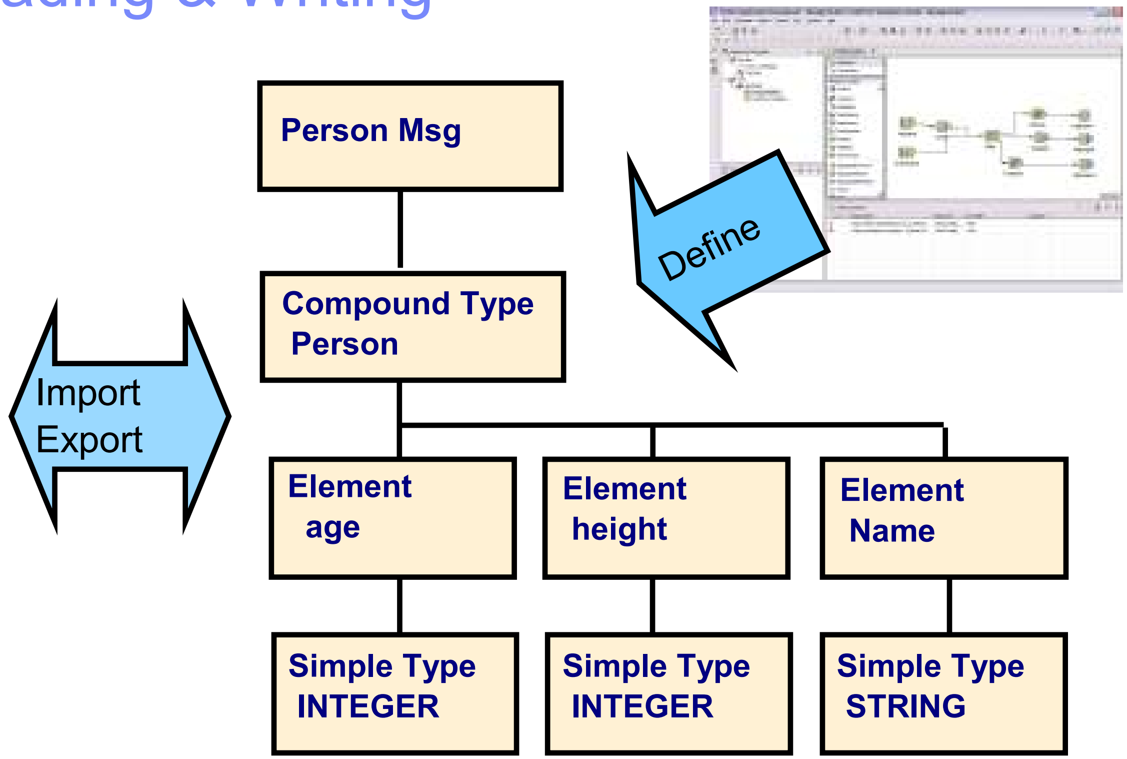
X	A123_02	PURCHASE	5	IBM

The Message Model - Reading & Writing

```
<Person age='32' height='172'>
<Name>Joe Bloggs</Name>
</Person>
```

```
    struct {
      int    height;
      int    age;
      char   Name[48];
    } Person;
 +
 172 32 Joe Bloggs
```

```
PER + 172 + 32 + Joe Bloggs
```



Physical Layers

🟡 XMLAttribute 'height'	🟡 XMLElement 'Name'
🟢 4 byte signed	🟢 48 characters space padded
🔴 All significant digits, rjustify	🔴 left justify space padded

The Message Model - ESQL Processing

DataInsert

```
IF Body.Person.height > 183 THEN

 INSERT INTO Database.TallPeople
        (Name,Height,Age)
 VALUES (Body.Person.Name,
         Body.Person.height,
         Body.Person.age);
ENDIF;
```

Compute

```
IF (XML format required) THEN
 OutputRoot.Properties.MessageFormat = 'XML';
ELSE IF (custom format)
 OutputRoot.Properties.MessageFormat = 'CWF';
ELSE IF (SWIFT format)
 OutputRoot.Properties.MessageFormat = 'TDS';
ENDIF;
```

Data types

- INTEGER
- FLOAT
- DECIMAL
- STRING
- DATETIME
- BOOLEAN
- REFERENCE
- NULL
- ...

Operators

- - + * /
- ||
- AND OR NOT
- = <> > >= < <=
- IN BETWEEN
- LIKE
- IS EXISTS
- ...

Statements

Basic
- DECLARE
- SET
- IF ENDIF
- WHILE

Tree
- MOVE
- CREATE
- DETACH
- ATTACH

Database
- INSERT
- DELETE
- UPDATE
- PASSTHRU
- EVAL

Node
- PROPAGATE
- RETURN
- THROW
- ...

Functions

String
- LENGTH
- TRIM LTRIM RTRIM
- OVERLAY
- POSITION
- SUBSTRING
- UCASE LCASE

Numeric
- ABS
- BITAND NOT (X)OR
- MOD ROUND
- SQRT
- TRUNCATE

Datetime
- EXTRACT
- CURRENTDATE
- CURRENTTIME

Field

CARDINALITY
- FIELDTYPE
- SAMEFIELD

Complex
- CAST
- SELECT
- ...

WMB- Reusable Nodes

- MQInput
- MQeInput
- SCADAInput
- HTTPInput
- Real-timeInput
- Real-timeOptimizedFlow

- Aggregate Request
- Aggregate Reply
- Aggregate Control

- MQOutput
- MQReply
- Publication
- MQeOutput
- SCADAOutput
- HTTPReply
- HTTPRequest

- RouteToLabel
- Label
- Throw
- TryCatch
- FlowOrder
- Filter
- Trace

- Compute
- Extract
- ResetConent Descriptors
- XMLTransformation
- Mapping
- Check

- Database
- DataInsert
- DataDelete
- DataUpdate
- Warehouse

Thank You