



Lab7:创建Log4j自定义节点

目录

目录.....	2
1 介绍.....	3
1.1 实验目的.....	3
1.2 通过本实验.....	3
1.3 实验内容介绍.....	3
1.4 实验所需资料.....	3
2 实验步骤.....	4
2.1 创建队列.....	4
2.2 创建Java工程.....	4
2.3 创建消息流插件节点项目.....	10
2.4 将插件节点部署运行时环境.....	20
2.5 创建消息流工程测试节点.....	21
2.6 部署到Broker.....	24
2.7 测试消息流.....	24

1 介绍

1.1 实验目的

通过本实验，练习在Message Broker中如何创建自定义节点。

1.2 通过本实验

你将能够：

- 创建Websphere Message Broker java 节点
- 创建Websphere Message Broker 节点插件

1.3 实验内容介绍

本实验将会创建Java节点，以及相应的Websphere Message Broker Toolkit插件。最后创建一个简单的测试流程测试该节点。

使用Log4j，可以在运行时改变日志级别，可以定义日志文件大小等...

1.4 实验所需资料

安装完成的Websphere Message Broker V7.0， lab7文件夹中的文件

2 实验步骤

2.1 创建队列

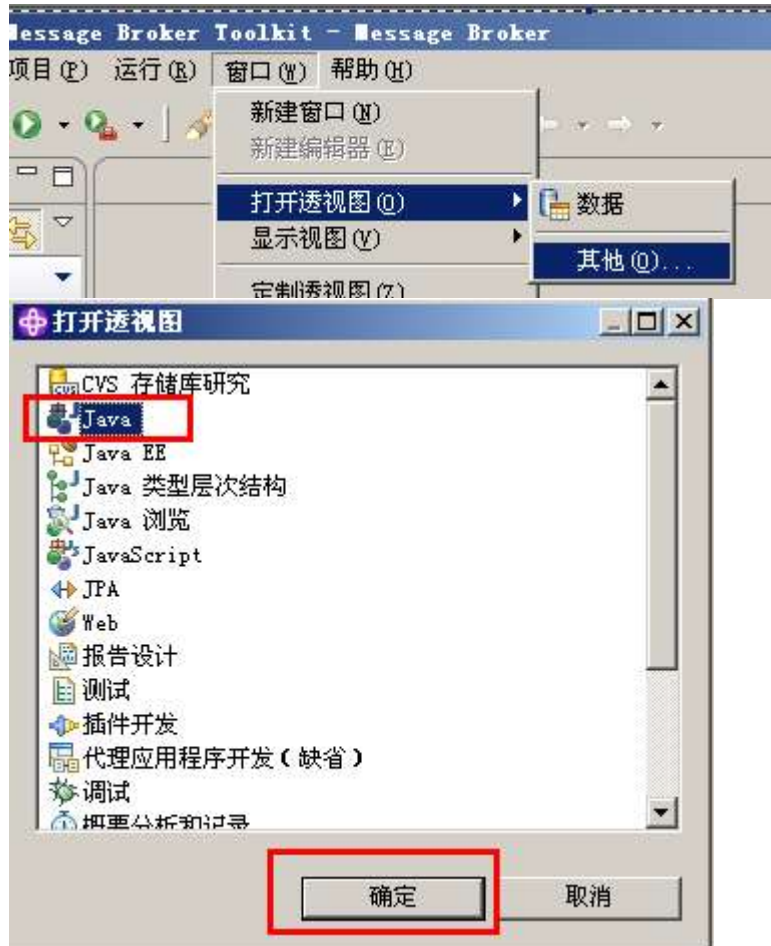
1. 使用下列命令创建本实验所需队列，具体步骤请参考前面实验

- a) `def ql(LAB7_IN) replace`
`def ql(LAB7_OUT) replace`

2.2 创建Java工程

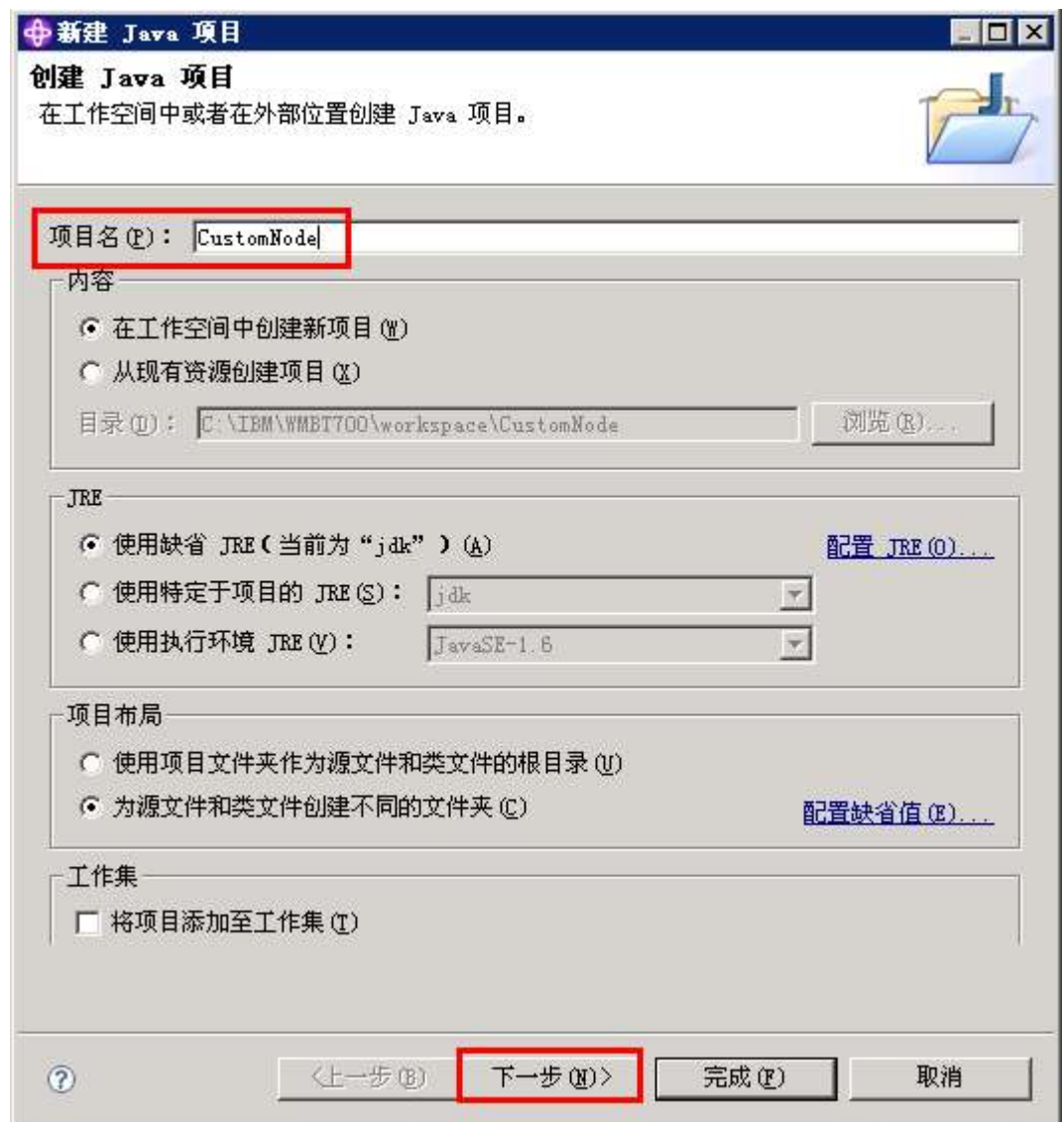
2. 启动Websphere Message Broker Toolkit，并切换至Java开发透视图

- a) 点击“窗口” - “打开透视图” - “其他”，选择“Java”，点击“确定”

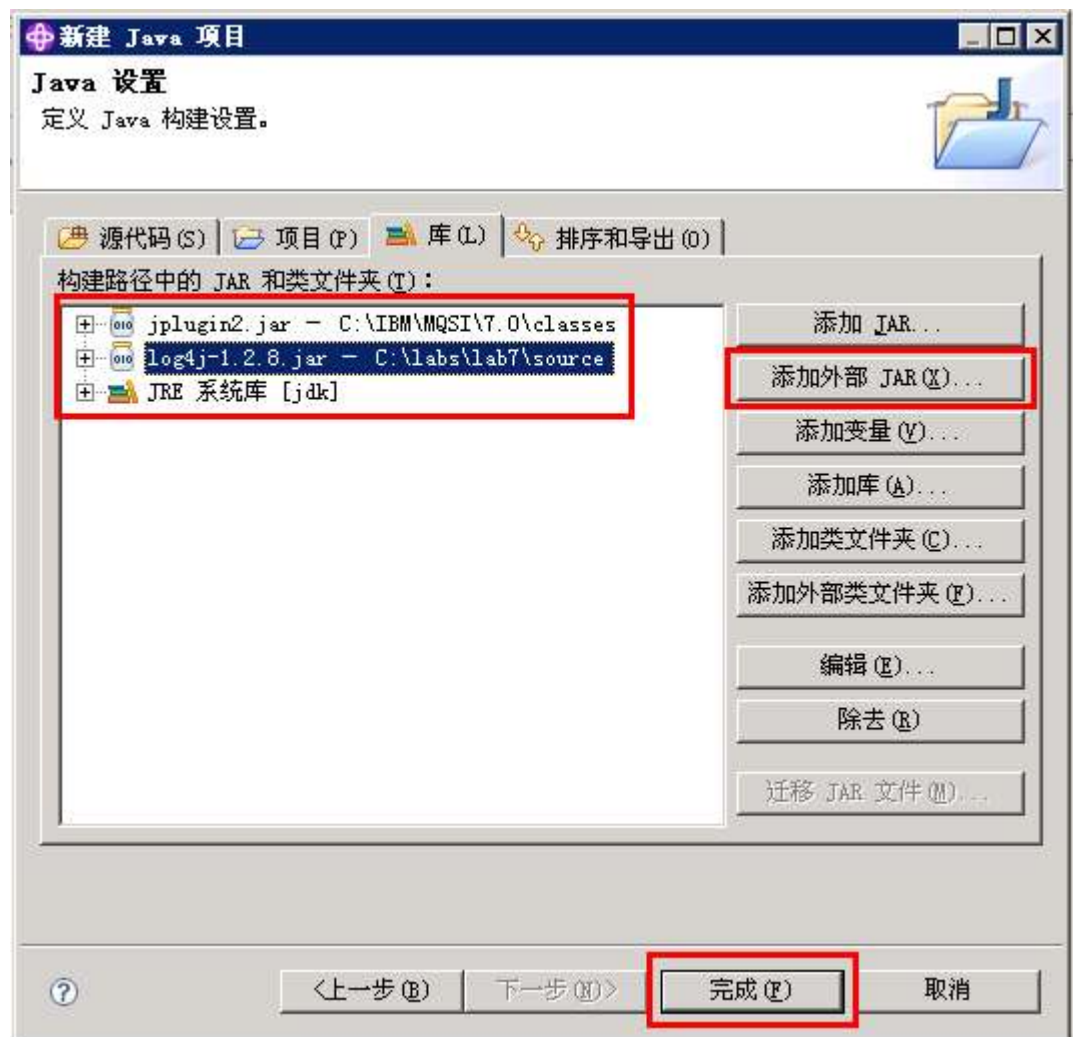


3. 创建Java项目 “CustomNode”

- a) 点击“文件” - “新建” - “Java项目”，输入项目名 “CustomNode”，点击“下一步”。

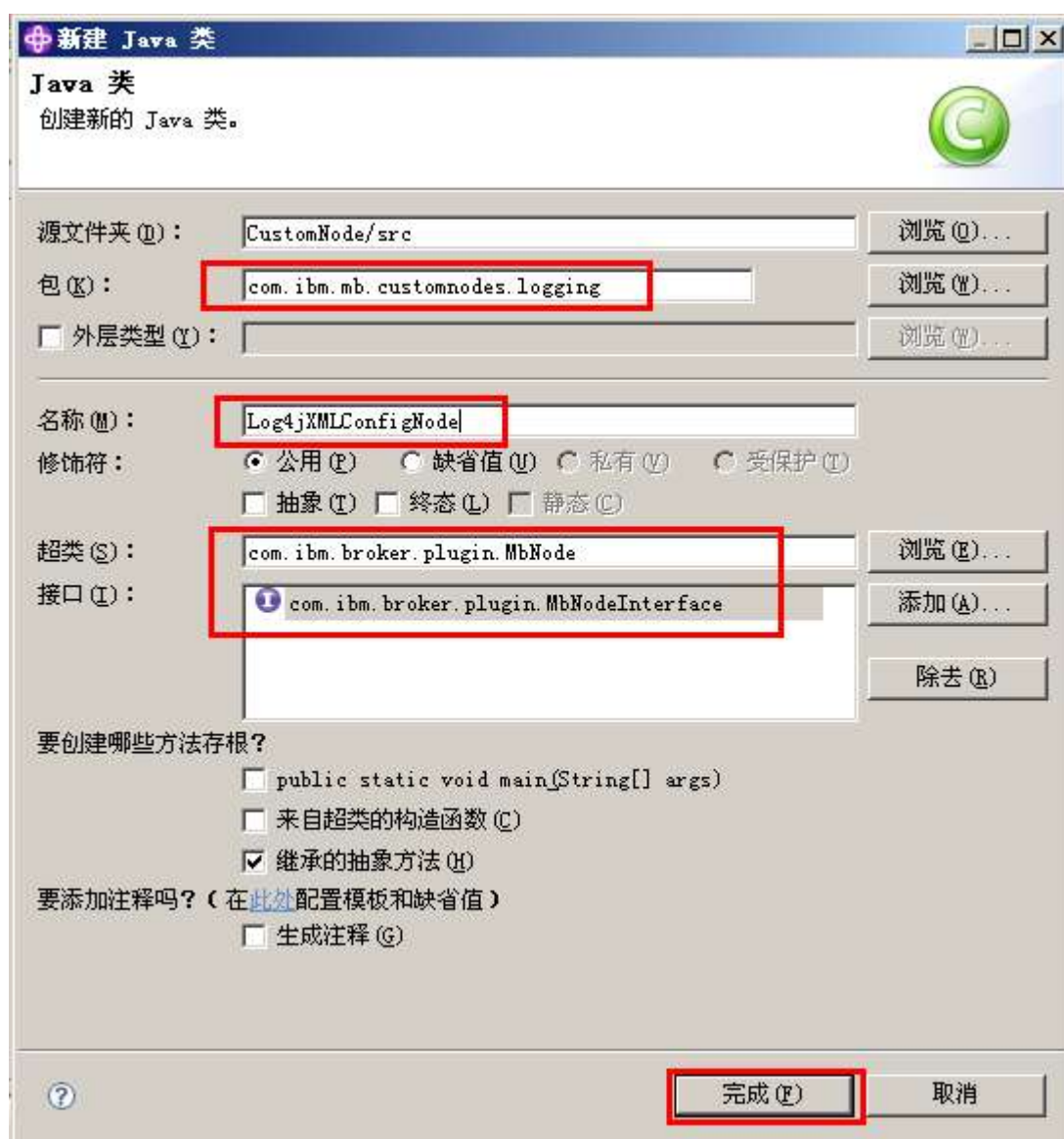


- b) 选择“库 (L)”“标签页，点击”添加外部JAR(X)“，选择添加C:\IBM\MQSI\7.0\classes\jplugin2.jar，点击”打开“，同样添加C:\labs\lab7\source\log4j-1.2.8.jar。点击”完成“完成“创建CustomNode Java项目。

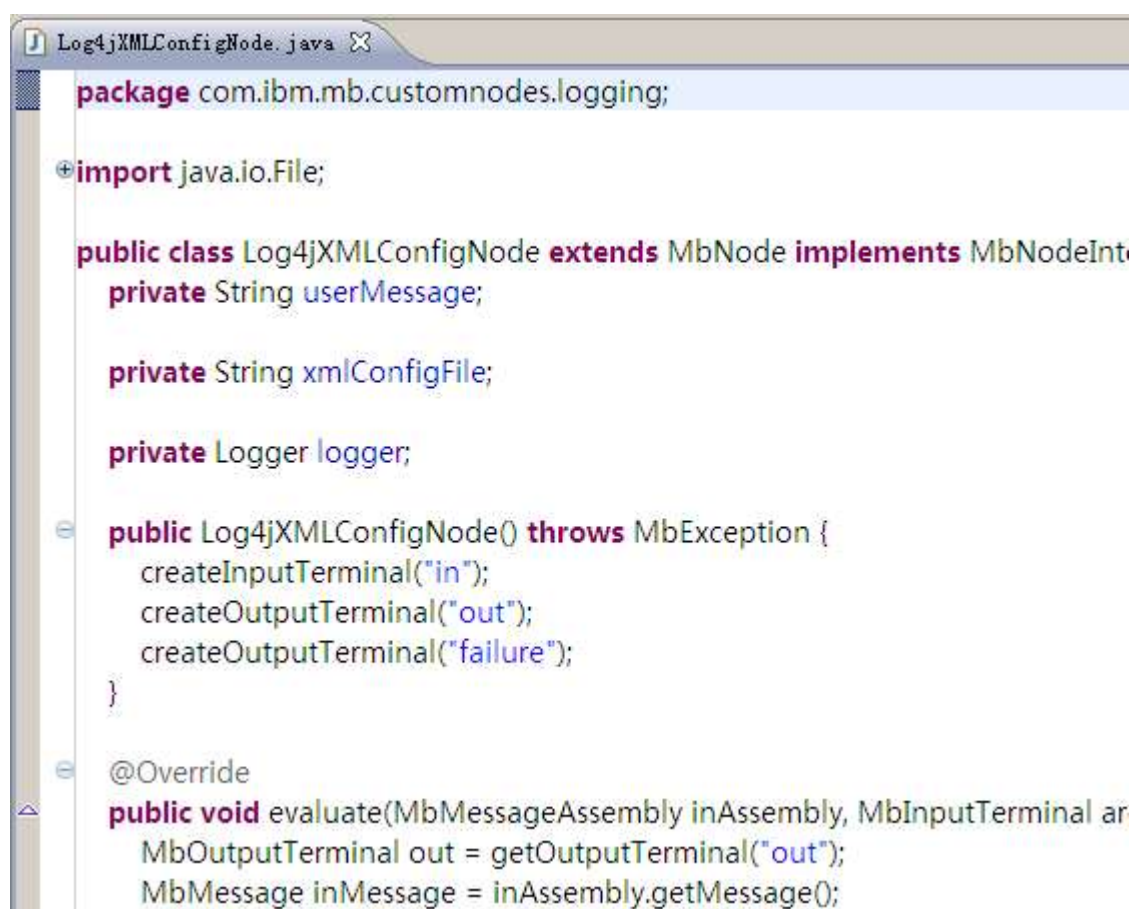


4. 创建类 “Log4jXMLConfigNode”

- a) 右键点击 “CustomNode项目”，选择 “新建” - “类”，输入，包：
com.ibm.mb.customnodes.logging，名称：Log4jXMLConfigNode，超类： ”
com.ibm.broker.plugin.MbNode“，接口： ” com.ibm.broker.plugin.MbNodeInterface
“，点击”完成“。



- b) 将c:\labs\lab7\source\Copy & Paste.txt中代码复制到Log4jXMLConfigNode.java中，并保存。



```
Log4jXMLConfigNode.java
package com.ibm.mb.customnodes.logging;

import java.io.File;
import java.util.*;

public class Log4jXMLConfigNode extends MbNode implements MbNodeInt {
    private String userMessage;

    private String xmlConfigFile;

    private Logger logger;

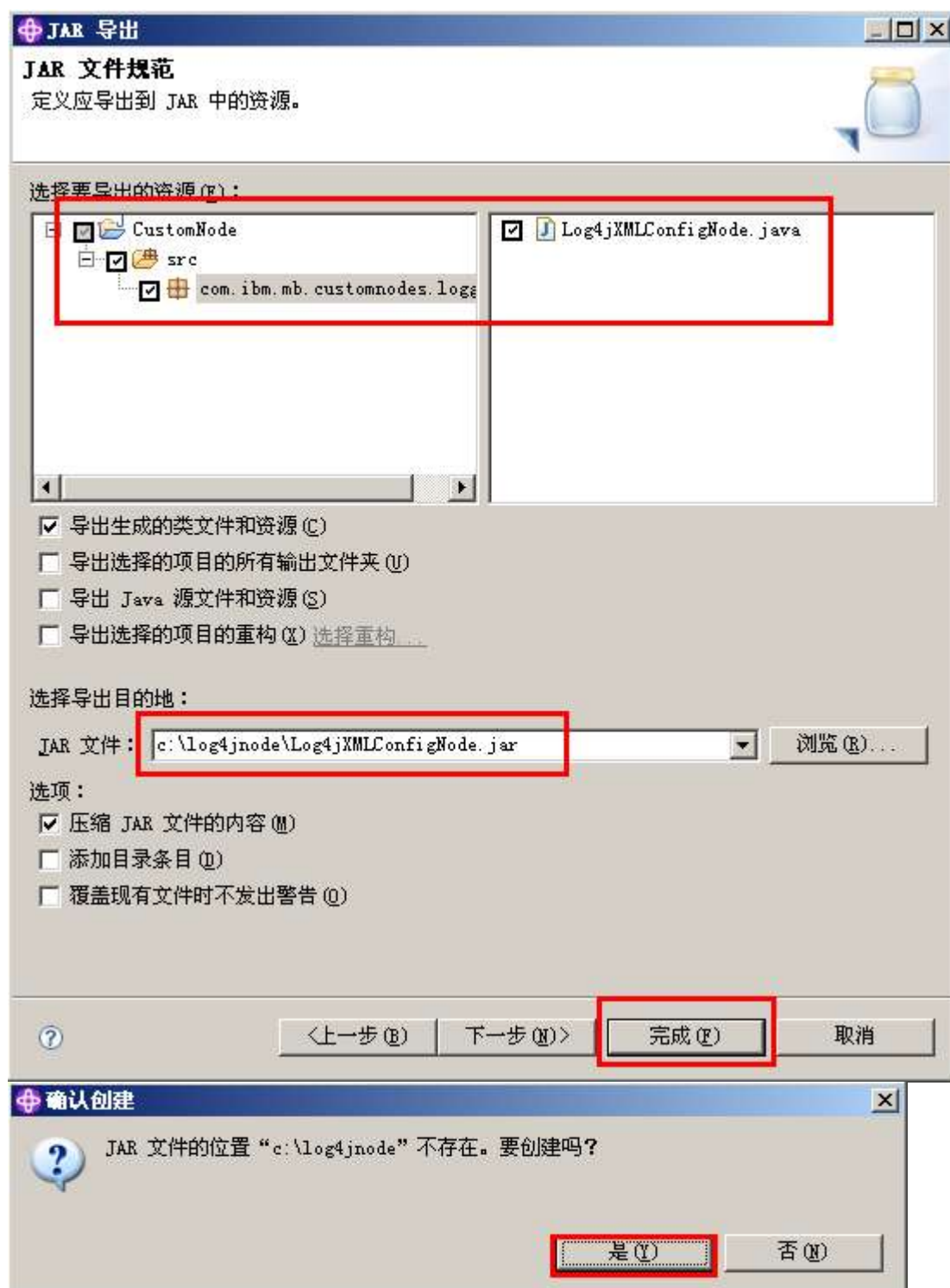
    public Log4jXMLConfigNode() throws MbException {
        createInputTerminal("in");
        createOutputTerminal("out");
        createOutputTerminal("failure");
    }

    @Override
    public void evaluate(MbMessageAssembly inAssembly, MbInputTerminal ar
        MbOutputTerminal out = getOutputTerminal("out");
        MbMessage inMessage = inAssembly.getMessage();
    }
```

5. 从CustomNode工程导出Log4jXMLConfigNode.jar
 - a) 右键点击“CustomNode”工程，选择“导出”，选择“JAR文件”，点击“下一步”



- b) 选择要导出的资源，如下图，导出目的地“c:\log4jnode\Log4jXMLConfigNode.jar”，点击”完成“，在弹出界面，点击”是“是“创建所需文件夹。



2.3 创建消息流插件节点项目

6. 创建 “Log4jXMLConfigVNode” 节点项目

- 点击 “窗口” — “打开透视图” — “代理应用程序开发”，切换至 “代理应用程序开发” 透视图。

- b) 点击“文件” — “新建” — “项目”，选择“消息流插件节点项目”，点击“下一步”，输入类别名“CustomNodes”，点击“下一步”，输入项目名称“Log4jXMLConfigVNode”，点击“下一步”，点击“完成”，点击“否”不切换当前透视图。



新建插件项目

插件项目
创建新的插件项目

项目名 (P): Log4jXMLConfigVNode

☒ 使用缺省位置 (D)

位置 (L): C:\IBM\WMBT700\workspace\Log4jXMLConfigVNode 浏览 (B)...

项目设置

☒ 创建 Java 项目

源文件夹 (S): src

输出文件夹 (O): bin

目标平台

此插件用于运行:

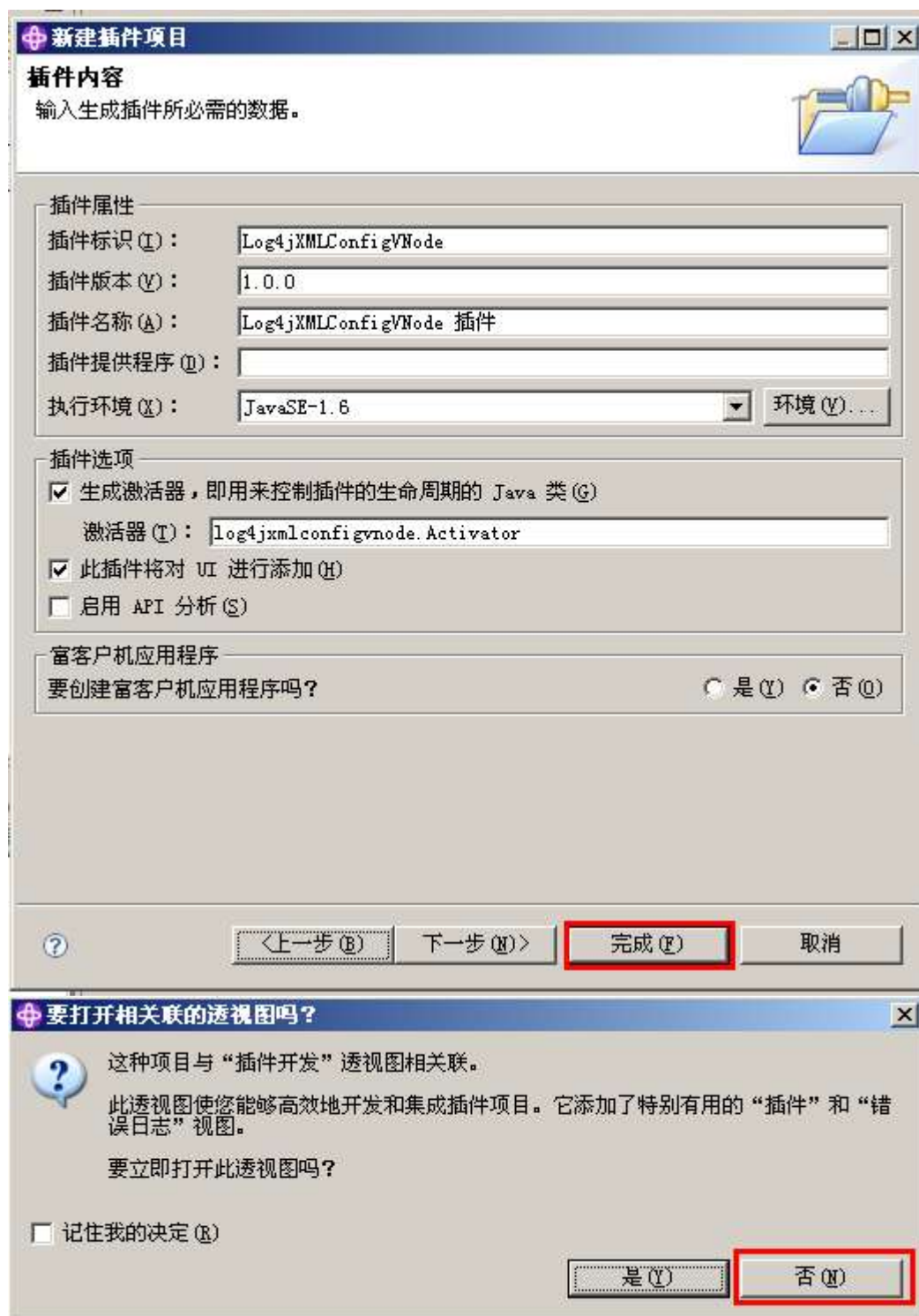
☒ Eclipse 版本: 3.4

☐ OSGi 框架 (A): Equinox

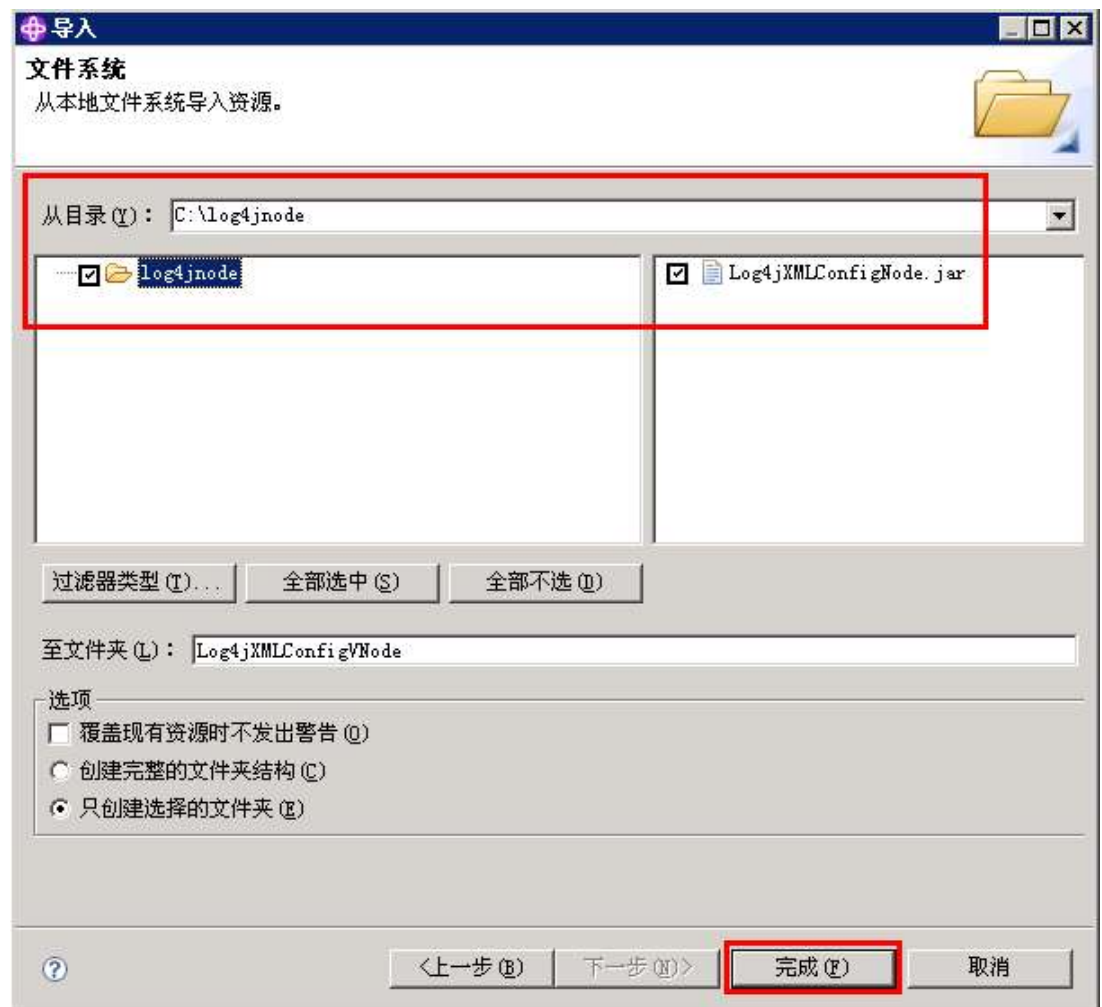
工作集

☐ 将项目添加至工作集 (T)

工作集 (W): 选择 (C)...

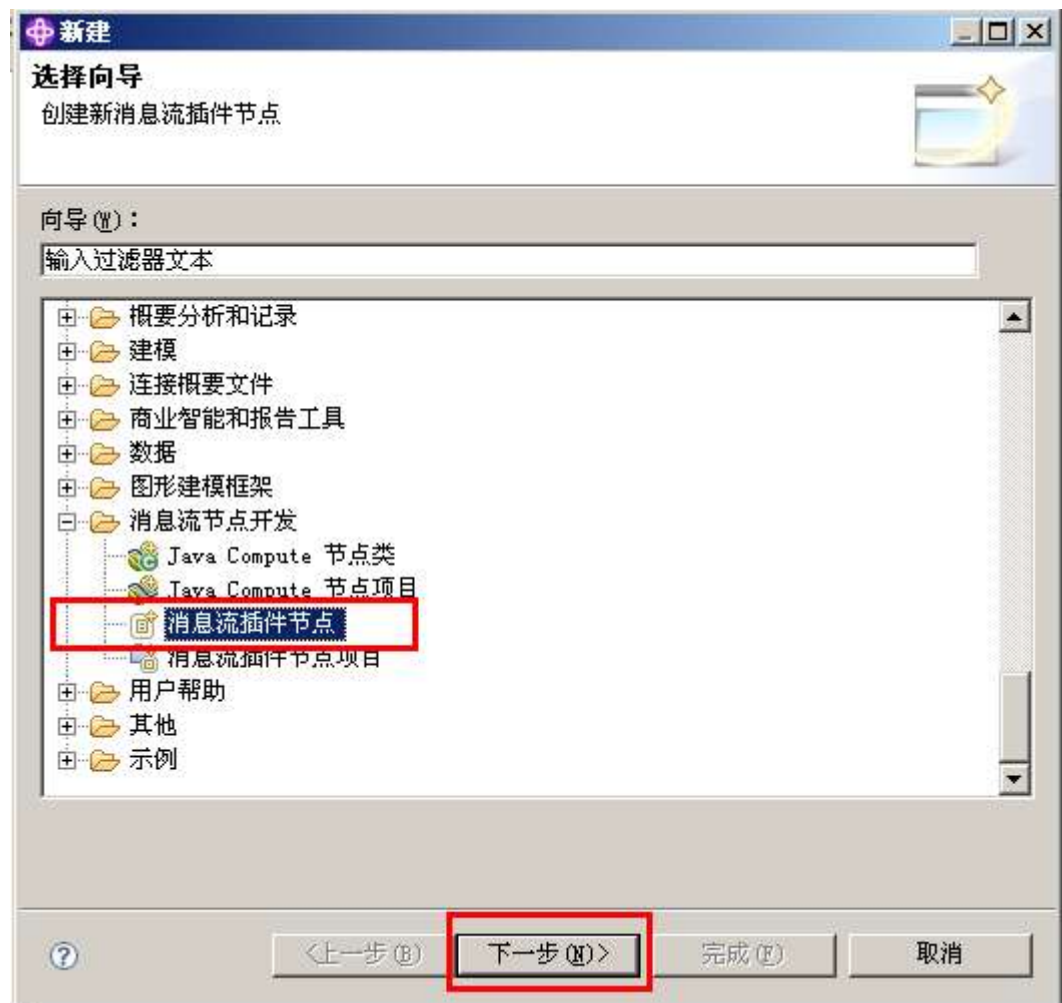


7. 将c:\log4jnode\Log4jXMLConfigNode.jar导入至消息流插件节点项目“Log4jXMLConfigVNode”
 - a) 右键点击“Log4jXMLConfigVNode”,选择“导入” - “文件系统”，点击“下一步”，浏览目录“c:\log4jnode”，选中“Log4jXMLConfigNode.jar”，点击“完成”。



8. 创建消息流插件节点

- a) 右键点击“Log4jXMLConfigVNode”项目，选择“新建” — “其他”，选中“消息流插件节点”，点击“下一步”，输入文件名“Log4jXMLConfig”，点击“完成”。



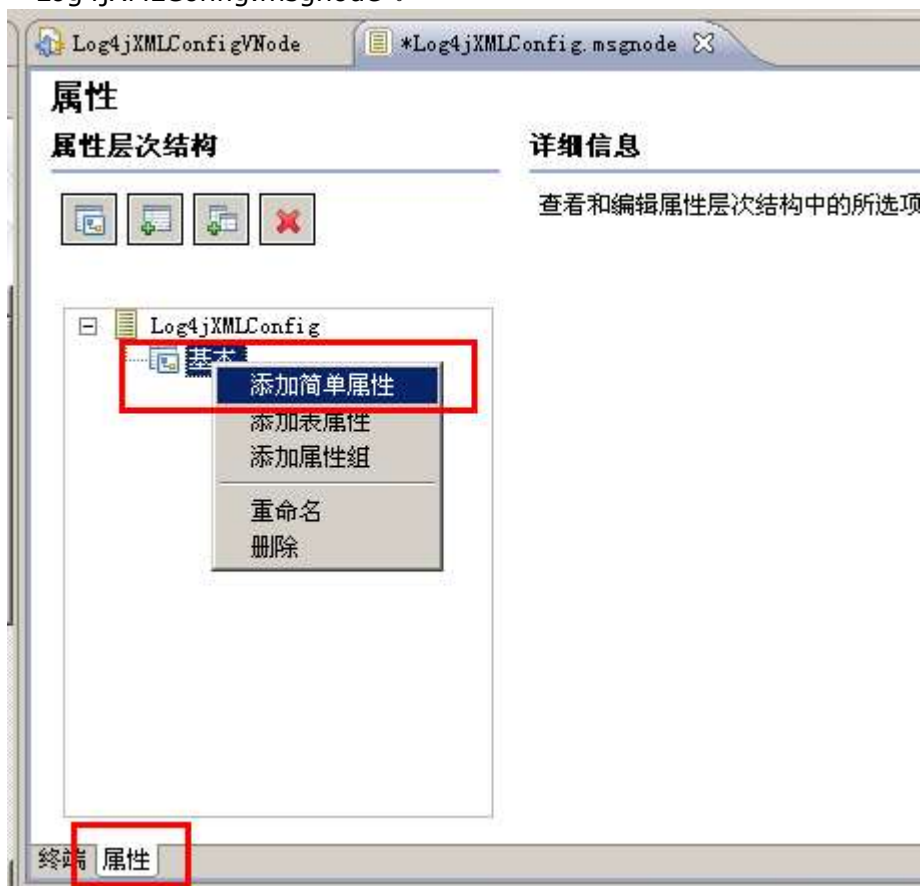


- b) 在Log4jXMLConfig.msgnode编辑界面，点击“添加”，定义输入输出终端“in”、“out”、“failure”，选中“支持动态输入终端”和“支持动态输出终端”



- c) 切换至属性编辑界面，右键点击“基本”，选择“添加简单属性”，添加“userMessage”和“xmlConfigFile”两个属性，保存

“Log4jXMLConfig.msgnode”。



属性

属性层次结构



详细信息

查看和编辑属性层次结构中的所选项。

类型

☒ 内置 字符串型

☐ 枚举

缺省值

定制编译器类

定制编辑器类

☐ 隐藏

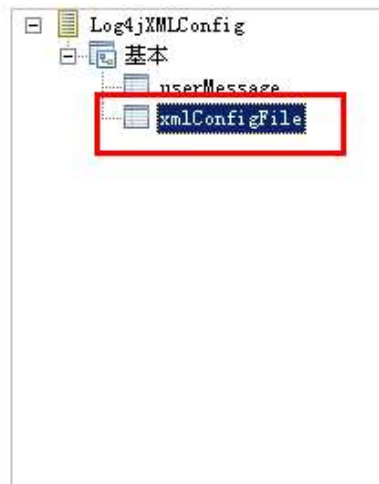
☐ 只读

☐ 必需的

☐ 可配置的

属性

属性层次结构



详细信息

查看和编辑属性层次结构中的所选项。

类型	
<input checked="" type="radio"/> 内置	字符串型
<input type="radio"/> 枚举	
<div>缺省值</div> <div>定制编译器类</div> <div>定制编辑器类</div> <div><input type="checkbox"/> 隐藏</div> <div><input type="checkbox"/> 只读</div> <div><input type="checkbox"/> 必需的</div> <div><input type="checkbox"/> 可配置的</div>	

9. 将消息流插件节点项目导出为jar文件

- 右键点击“Log4jXMLConfigVNode”项目，选择“导出” — “可部署的插件和段”，点击“下一步”，选择目录“C:\IBM\WMBT700”，点击“完成”。



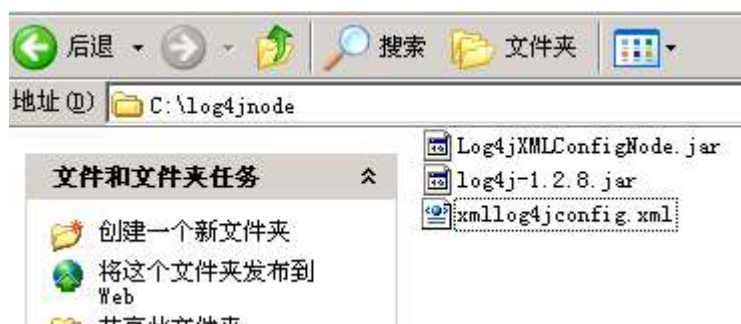


b) 重新启动Websphere Message Broker Toolkit。

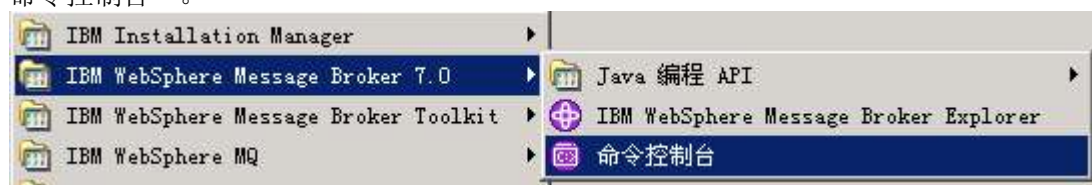
2.4 将插件节点部署运行时环境

10. 打开Websphere Message Broker 命令控制台

- a) 将C:\labs\lab7\source\log4j-1.2.8.jar、C:\labs\lab7\source\xmllog4jconfig.xml拷贝到c:\log4jnode中



- b) 点击操作系统“开始” — “程序” — “IBM Websphere Message Broker 7.0” — “命令控制台”。



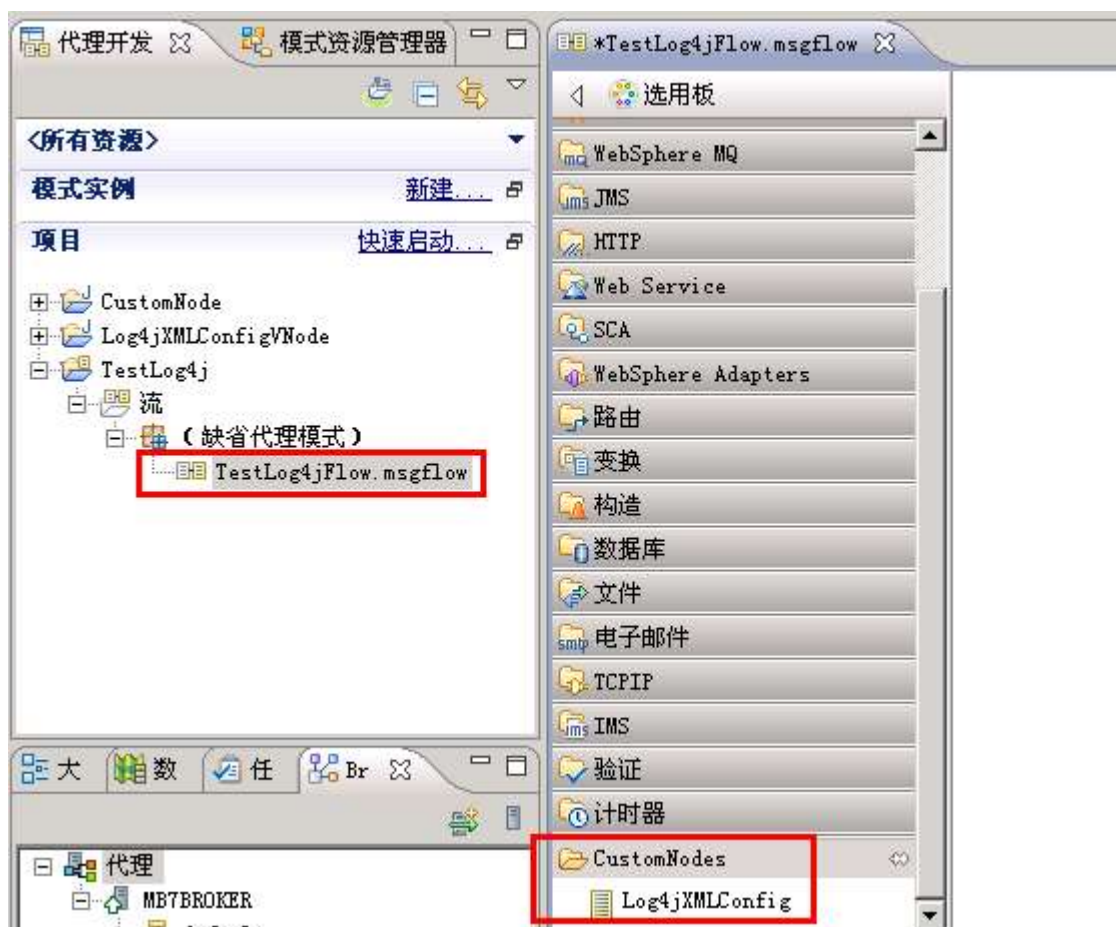
- c) 执行下列命令设置MB7Broker的配置
- ```
mqsistop MB7BROKER
mqsichangebroker MB7BROKER -l c:\log4jnode
mqsistart MB7BROKER
```



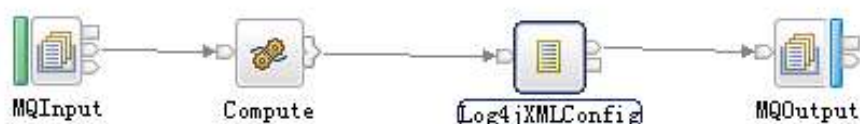
## 2.5 创建消息流工程测试节点

11. 创建消息流工程“TestLog4j”，创建消息流“TestLog4jFlow”，在消息流编辑器视图可看到自定义的节点“Log4jXMLConfig”。





12. 为TestLog4jFlow添加节点，一个MQInput，一个Compute，一个Log4jXMLConfig，一个MQOutput，并进行连接，MQInput 输出—Compute输入，Compute输出—Log4jXMLConfig输入，Log4jXMLConfig输出—MQOutput输入



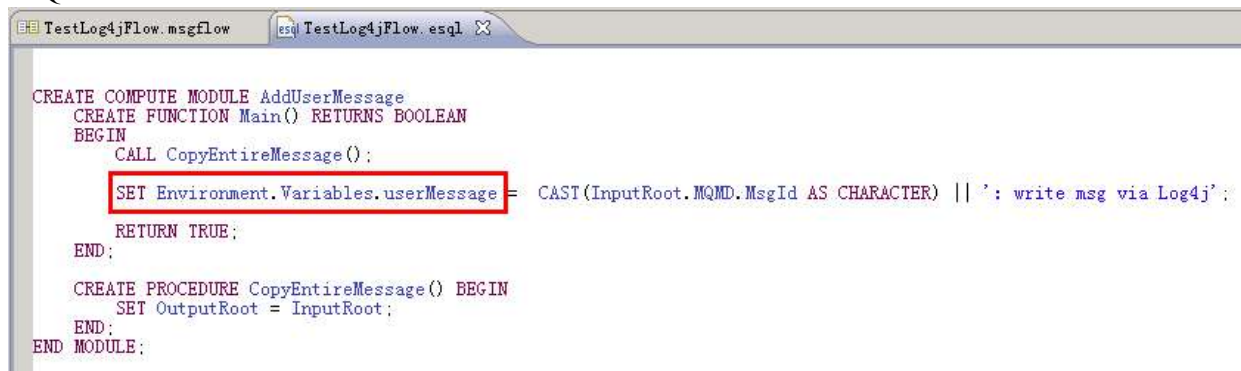
13. 各节点属性配置，如下表，配置完成后保存消息流。

| 节点名称           | 属性                                                         |
|----------------|------------------------------------------------------------|
| MQInput        | 基本 - 队列名称:LAB7_IN                                          |
| Compute        | 基本 -<br>ESQL模块: AddUserMessage                             |
| Log4jXMLConfig | 基本 -<br>xmlConfigFile: C:/log4jnode/<br>xmllog4jconfig.xml |
| MQOutput       | 基本 - 队列名称:LAB7_OUT                                         |

14. 为Compute节点编写ESQL代码



15. 右键点击“Compute”节点，选择“打开ESQL”，ESQL编辑器编辑界面打开，编写ESQL。



```
TestLog4jFlow.msgflow TestLog4jFlow.esql
CREATE COMPUTE MODULE AddUserMessage
CREATE FUNCTION Main() RETURNS BOOLEAN
BEGIN
 CALL CopyEntireMessage();
 SET Environment.Variables.userMessage = CAST(InputRoot.MQMD.MsgId AS CHARACTER) || ': write msg via Log4j';
 RETURN TRUE;
END;
CREATE PROCEDURE CopyEntireMessage() BEGIN
 SET OutputRoot = InputRoot;
END;
END MODULE;
```

## 2.6 部署到Broker

16. 创建消息流归档文件LABS.bar
- 右键单击“TestLog4j”，选择“新建” - “Message Broker归档”
  - 输入名称“LABS”，点击“完成”，工作台会打开LABS.bar编辑器界面
  - 选中TestLog4jFlow.msgflow，点击“构建代理归档”
  - 保存LABS.bar
17. 将LABS.bar拖拽到default执行组进行部署。

## 2.7 测试消息流

18. 使用RFHUTIL工具发送测试数据C:\Labs\Lab7\data\testData.txt.txt到队列LAB7\_IN
- 在Main标签页，队列管理器MB7QMGR，队列LAB7\_IN,点击Read File，选择C:\Labs\Lab7\data\testData.txt.txt
  - 返回Main标签页，点击Write Q发送测试消息
19. 进入c:\log4jnode目录，查看由Log4jXMLConfig节点生成的Debug信息。





20. 使用RFHUTIL,在Main标签页, 队列管理器MB7QMGR, 队列LAB7\_OUT, 点击Read Q, 删除测试消息。

本次实验结束。