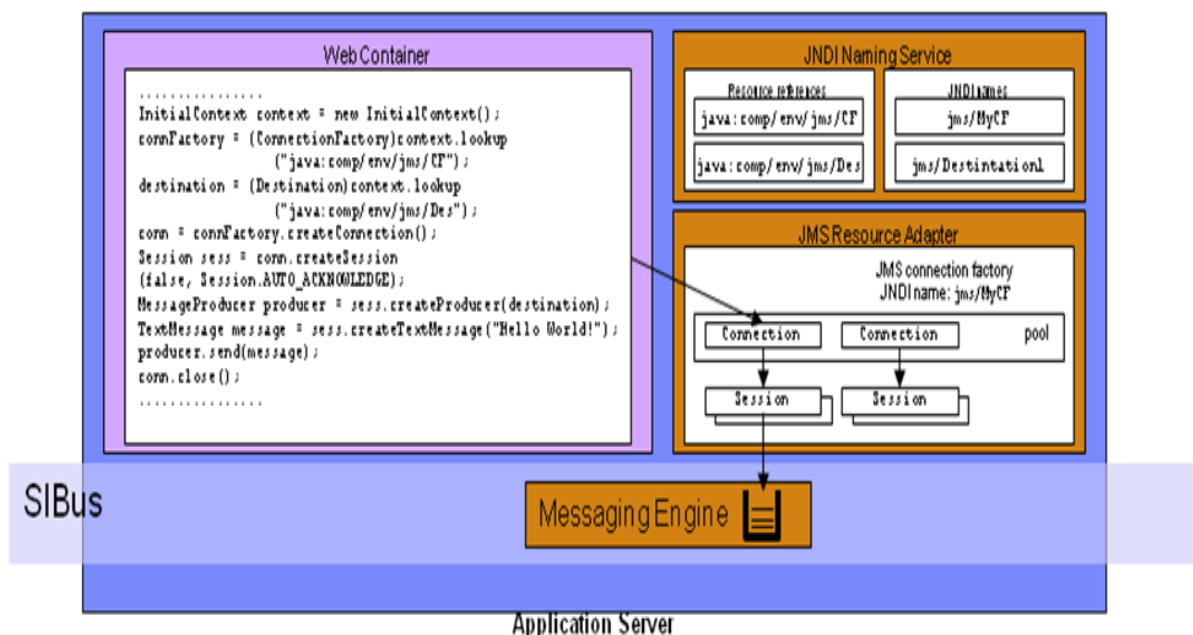


하나씩 쉽게 따라 해보는 IBM WebSphere Application Server(WAS) v7 – 7

이정운 (juwlee@kr.ibm.com)

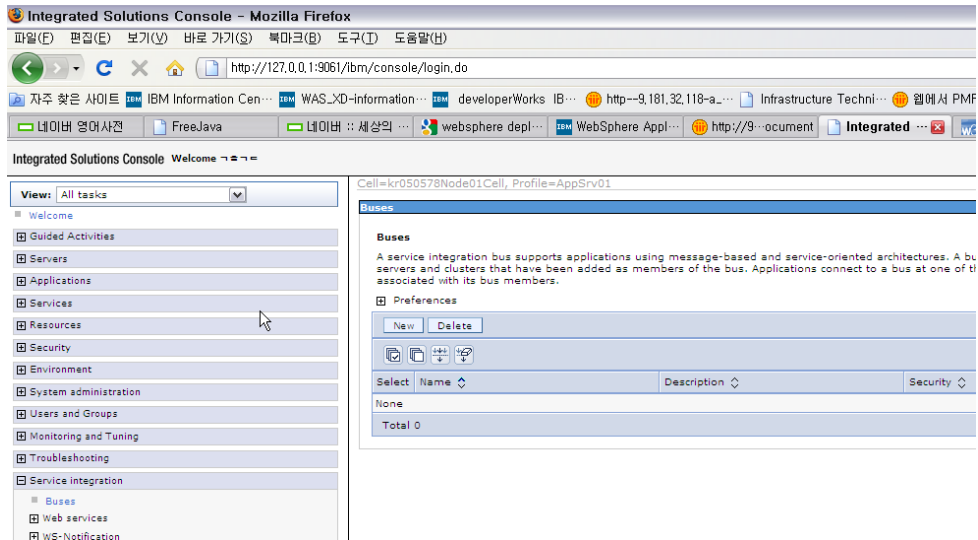
하나씩 쉽게 따라 해보는 IBM WAS v7 시리즈 그 일곱번째 이야기 JMS 입니다. JMS 란 Java Message Service 의 약자로서 자바를 이용해서 메시지를 만들고 이를 이용한 통신을 지원하는 기능입니다. 또한, 기업 메시징 시스템 액세스를 위한 API로서, 컴퓨터간 데이터 교환, 이벤트 안내, 서비스 요구 사항 등과 같은 정보를 포함하는 메시지를 비동기적으로 주고받는 비즈니스 애플리케이션을 작성하고 다른 시스템이나 다른 언어로 쓰여진 프로그램을 일원화하는 데 주로 사용되며, 자바 객체를 포함한 메시지와 확장성 생성 언어(XML) 페이지를 포함한 메시지를 지원합니다. IBM WAS v7 는 기본적으로 JMS 를 위한 메시징 엔진을 제공하고 해당 Spec 을 준수하여 개발된 어플리케이션에 이를 이용한 메세징 서비스를 가능하게 합니다.

그러기 위해서는 SIB 라는 Service Integration Bus 를 먼저 생성해야 합니다. SIB 는 WebSphere 가 J2EE 애플리케이션을 실행하는, 웹 기반 소프트웨어 실행 엔진을 넘어서 비즈니스의 중추가 될 수 있는 ESB 라는 개념까지 확장하여 동기/비동기 메시징을 통해 서비스들간의 통합을 지원하는 통신 관리 컴포넌트입니다. 다시말하면, 메시지 전송을 위한 가장 기본적인 메커니즘으로서 다양한 메세징 프로토콜을 지원하며 그에 수반하는 다양한 비즈니스 패턴을 구현한 수 있는 환경을 제공해줍니다. 이 때, SIB가 제공할 수 있는 가장 기본 환경중의 하나가 바로 JMS 입니다. 이를 이용해서 메세징 서비스를 하기위해 Connection 을 생성해주는 Connection Factory 를 만들어야 하며 이렇게 가지고 온 Connection 의 목적지가 되는 Queue 나 Topic 을 생성해야 합니다. 어떠신가요? 지금까지 설명으로는 너무 어렵고 잘 이해가 안가실 것입니다. “백문이 불여일견” 한번 보시고 따라하시면 보다 더 이해가 쉬우실 테니 하나씩 진행을 해볼 까요.

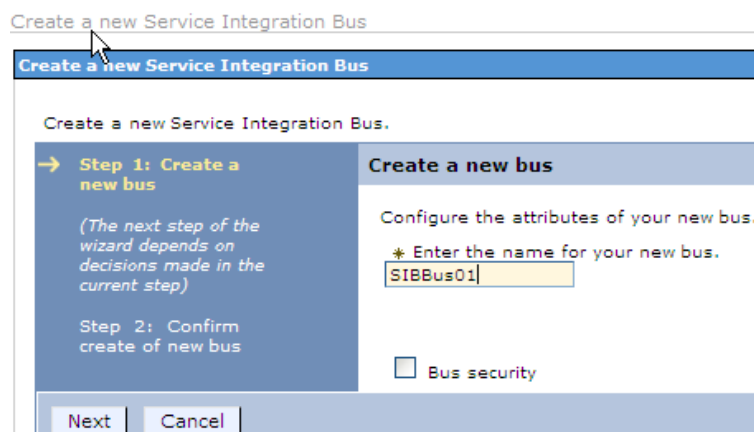


Part 1. SIB 생성

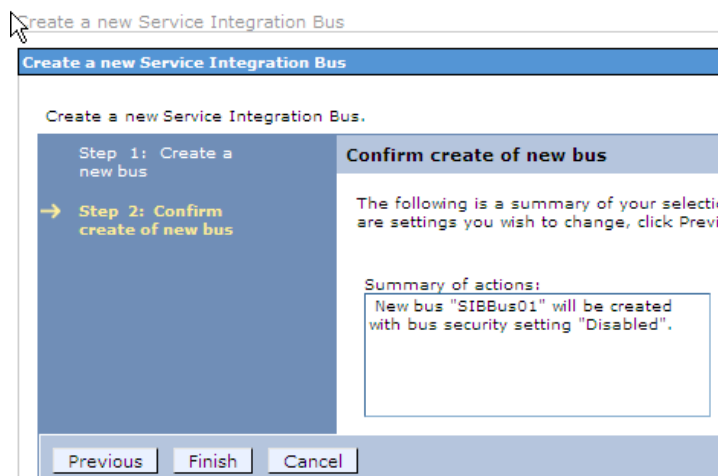
위에서 설명했지만 메세징 서비스를 위해서는 SIB 를 먼저 생성해야 합니다. SIB 는 메세징서비스를 위한 기반 메커니즘을 제공해줍니다. SIB 를 만들기 위해서 우선 관리콘솔로 들어가서 Service integration > Buses 메뉴를 선택하고 New 버튼을 클릭합니다.



SIB 마법사가 나오면 사용할 SIB 의 이름을 입력한 후 다음을 클릭합니다.



요약정보를 확인한 후 종료를 눌러서 SIB 생성을 완료합니다.



위의 단계를 완료하면 주어진 이름으로 SIB 가 생성된 것을 메뉴에서 확인하실 수 있습니다. 생성된 SIB 를 확인하고 세부 설정을 위하여 이름을 클릭하고 세부메뉴로 들어갑니다.

Buses

A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.

Preferences

Select	Name	Description	Security
<input type="checkbox"/>	SIBBus01		Disabled

Total 1

SIB 세부 메뉴에서 Bus members 를 선택합니다.

(이 메뉴를 이용해서 SIB 에 들어가게 되는 Member 를 선택하는 것으로 Member 는 WAS Server 나 MQ Server 가 됩니다.)

Buses > SIBBus01

A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.

Configuration Local Topology

General Properties

Name
SIBBus01

UUID
7F49677215058327

Description

Inter-engine transport chain

☐ Discard messages

☒ Configuration reload enabled

Default messaging engine high message threshold
50000 messages

Limit the range of available bootstrap members to:

- ☒ All members of the cell with the Service Integration Bus Service enabled
- ☐ Bus members and nominated bootstrap members
- ☐ Bus members only

Apply OK Reset Cancel

Topology

- ☒ Bus members
- ☐ Messaging engines
- ☐ Foreign bus connections
- ☐ Bootstrap members

Destination resources

- ☐ Destinations
- ☐ Mediations

Services

- ☐ Inbound services
- ☐ Outbound services
- ☐ WS-Notification services
- ☐ Reliable messaging state

Additional Properties

- ☐ Custom properties
- ☐ Security

Bus members 메뉴에 들어오면 SIB에 새롭게 member 를 추가하기 위하여 add 버튼을 클릭합니다.

Buses > SIBBus01 > Bus members

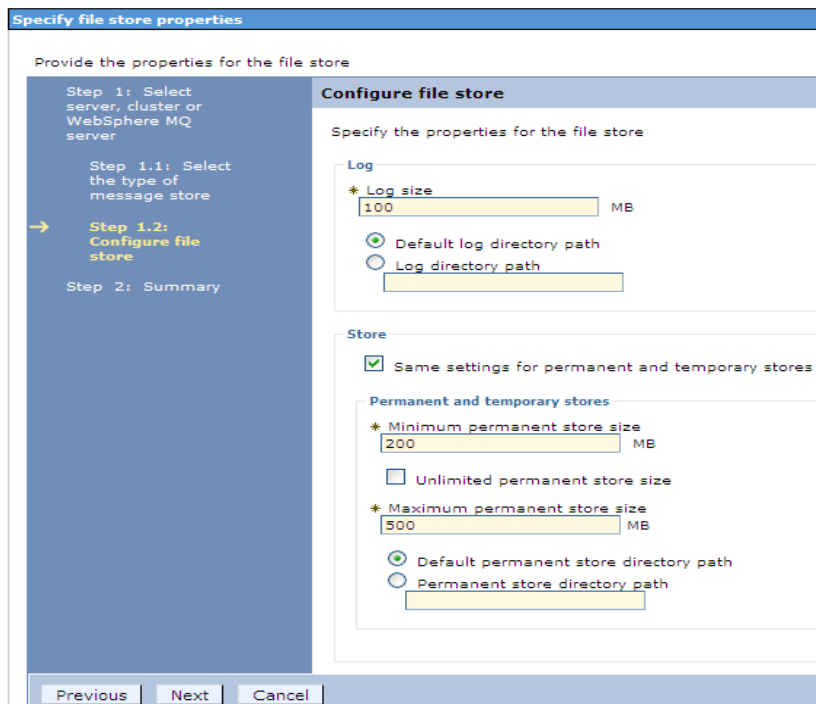
Bus members are the servers, WebSphere MQ servers and clusters that have been added to the bus.

Preferences

Bus member 마법사 화면에서 member 로 추가할 서버 타입을 선택합니다.
(여기서는 지금 구동중인 WAS server 를 선택합니다.)

메시지가 저장될 저장장치의 타입을 선택합니다.
(Data store 는 DB를 지칭하는 것이며 여기서는 단순 구성을 위해 File store 를 선택합니다.)

Log 사이즈와 같은 File 저장소의 각설정을 확인하고 다음을 클릭합니다.



Specify file store properties

Provide the properties for the file store

Step 1: Select server, cluster or WebSphere MQ server

Step 1.1: Select the type of message store

→ Step 1.2: Configure file store

Step 2: Summary

Configure file store

Specify the properties for the file store

Log

* Log size: 100 MB

☒ Default log directory path

☐ Log directory path:

Store

☒ Same settings for permanent and temporary stores

Permanent and temporary stores

* Minimum permanent store size: 200 MB

☐ Unlimited permanent store size

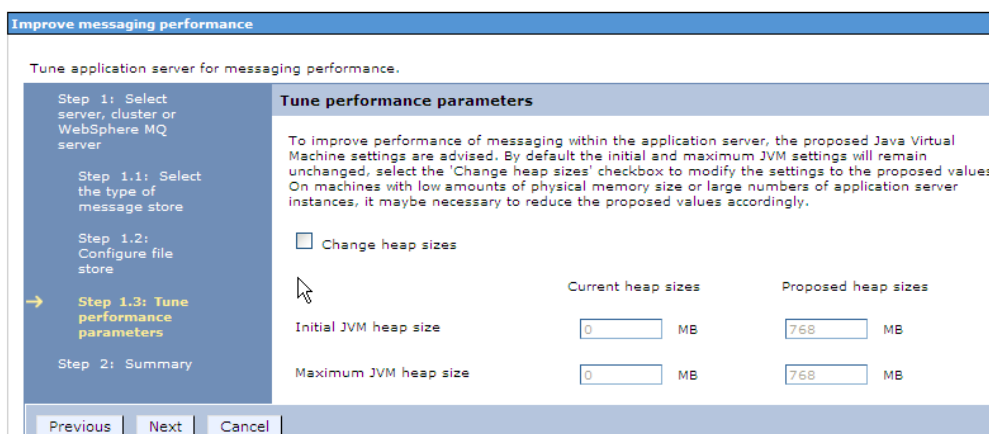
* Maximum permanent store size: 500 MB

☒ Default permanent store directory path

☐ Permanent store directory path:

Previous Next Cancel

성능 파라미터 조정 메뉴를 확인하고 다음을 클릭합니다.



Improve messaging performance

Tune application server for messaging performance.

Step 1: Select server, cluster or WebSphere MQ server

Step 1.1: Select the type of message store

Step 1.2: Configure file store

→ Step 1.3: Tune performance parameters

Step 2: Summary

Tune performance parameters

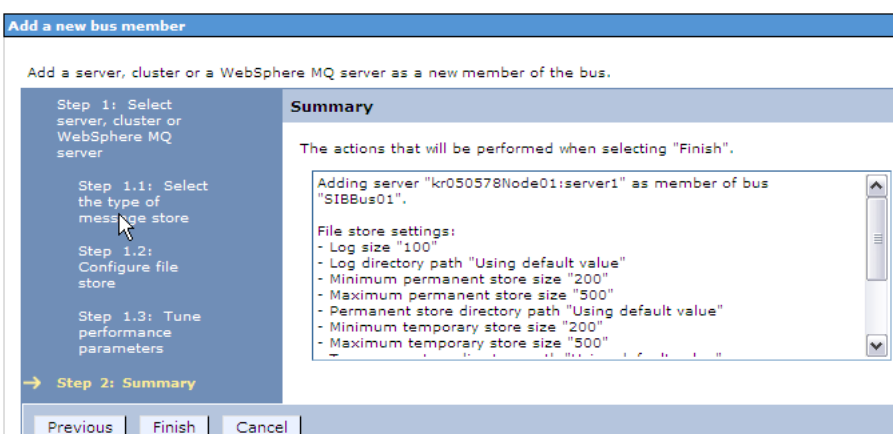
To improve performance of messaging within the application server, the proposed Java Virtual Machine settings are advised. By default the initial and maximum JVM settings will remain unchanged. select the 'Change heap sizes' checkbox to modify the settings to the proposed values. On machines with low amounts of physical memory size or large numbers of application server instances, it may be necessary to reduce the proposed values accordingly.

☐ Change heap sizes

	Current heap sizes	Proposed heap sizes
Initial JVM heap size	0 MB	768 MB
Maximum JVM heap size	0 MB	768 MB

Previous Next Cancel

마지막으로 요약 정보를 확인하고 Bus member 추가를 종료합니다.



Add a new bus member

Add a server, cluster or a WebSphere MQ server as a new member of the bus.

Step 1: Select server, cluster or WebSphere MQ server

Step 1.1: Select the type of message store

Step 1.2: Configure file store

Step 1.3: Tune performance parameters

→ Step 2: Summary

Summary

The actions that will be performed when selecting "Finish".

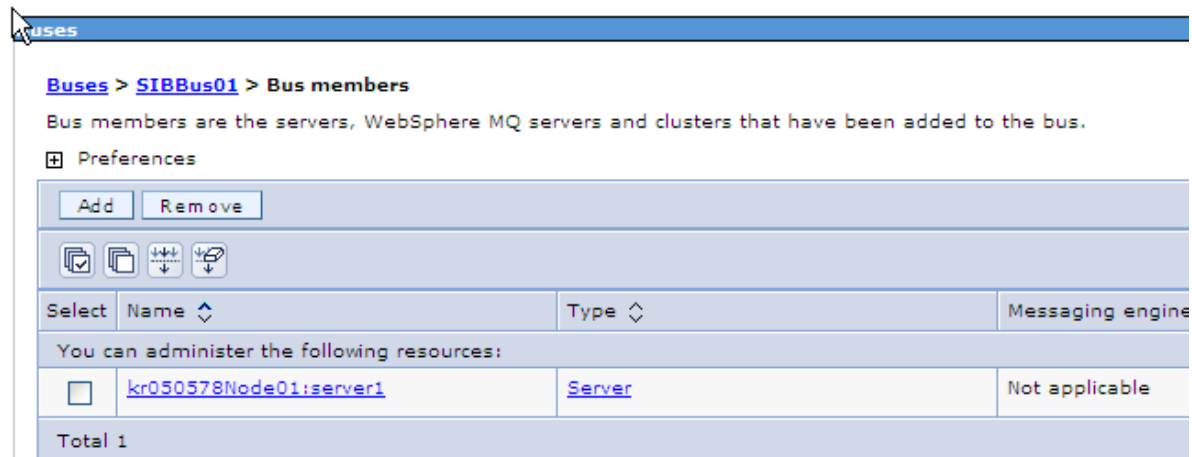
Adding server "kr050578Node01:server1" as member of bus "SIBBus01".

File store settings:

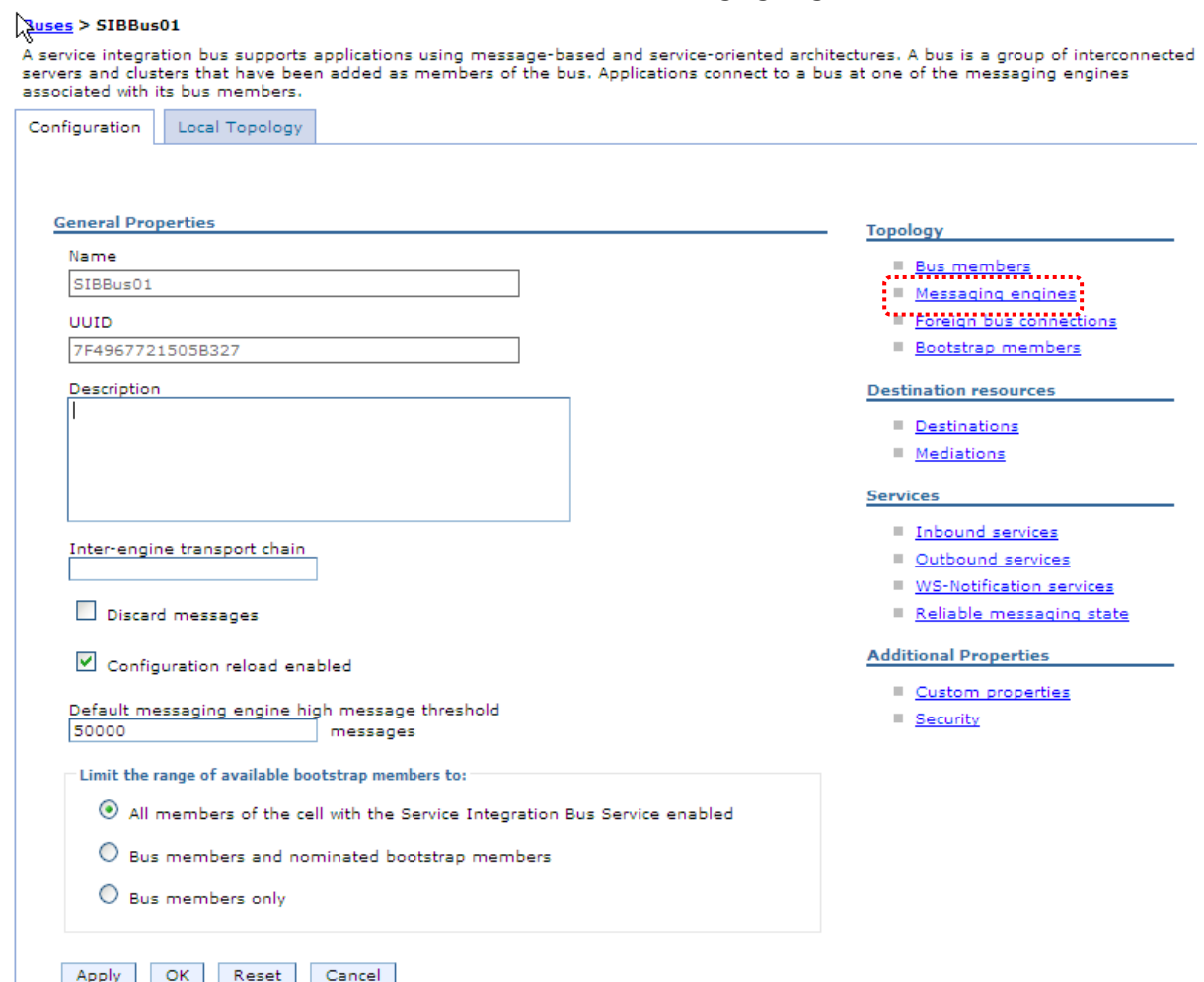
- Log size "100"
- Log directory path "Using default value"
- Minimum permanent store size "200"
- Maximum permanent store size "500"
- Permanent store directory path "Using default value"
- Minimum temporary store size "200"
- Maximum temporary store size "500"

Previous Finish Cancel

Bus members 메뉴에서 정상적으로 member 가 추가되었는지 확인합니다.



정상적으로 member 가 추가되었는지 확인이 되었으면 해당 서버를 재시작합니다. (강좌 1 참조)
서버를 재시작하고 만들어진 Bus 의 세부 메뉴에서 Messaging engines 메뉴를 클릭합니다.



만들었던 Bus 의 구성원인 server1 에서 정상적으로 Messaging engine 이 시작된 걸 확인하실 수 있습니다.

[Buses](#) > [SIBBus01](#) > Messaging engines

A messaging engine is a component, running inside a server, that manages messaging resources for a bus connected to a messaging engine when they access a service integration bus.

Preferences

<input type="button" value="Start"/> <input type="button" value="Stop"/>			
Select	Name	Description	Status
You can administer the following resources:			
<input type="checkbox"/>	kr050578Node01.server1-SIBBus01		
Total 1			

여기까지 따라오셨다면 정상적으로 SIB 를 만들고 Member 를 추가하여 해당 Member 에서 Messaging Engine이 정상적으로 수행된 것을 확인하신 것 입니다. 이 Messaging engine 이 SIB 내에서 실제로 Message 를 전달하는 역할을 하는 엔진입니다. 그러면 실제 JMS Message 를 주고 받기 위하여 JMS provider 의 Connection factory 를 만들어서 connection 을 받아오는 단계를 진행해 보도록 하겠습니다.

Part 2. Connection Factory, Queue생성

이전 단계까지 SIB 를 생성하고 Message engine 을 구동했다면 이제 실제 JMS provider 가 제공하는 메세징 기능을 활용하기 위하여 Connection Factory 를 생성하고 목적지인 Queue 를 만드는 것을 해보도록 하겠습니다. 관리콘솔에서 Resources > JMS > JMS providers 를 선택한 후 나타나는 리스트에서 Default message provider 를 선택합니다.

(Default message provider 는 WAS 에서 기본적으로 제공되는 내장 JMS provider 입니다.)

The screenshot shows the Integrated Solutions Console interface. On the left is a navigation tree with 'Resources' expanded and 'JMS providers' selected. The main panel displays the 'JMS providers' configuration page. It includes a description of JMS providers and a table listing available providers. The first two rows, 'Default messaging provider' and 'Default messaging provider', are highlighted with a red dashed box. The table has columns for 'Select', 'Name', 'Description', and 'Scope'.

Select	Name	Description	Scope
<input type="checkbox"/>	Default messaging provider	Default messaging provider	Cell=kr050578Node01Cell
<input type="checkbox"/>	Default messaging provider	Default messaging provider	Node=kr050578Node01,Server=server1
<input type="checkbox"/>	Default messaging provider	Default messaging provider	Node=kr050578Node01
<input type="checkbox"/>	V5 default messaging provider	V5 Default Messaging Provider	Node=kr050578Node01,Server=server1
<input type="checkbox"/>	V5 default messaging provider	V5 Default Messaging Provider	Node=kr050578Node01
<input type="checkbox"/>	V5 default messaging provider	V5 Default Messaging Provider	Cell=kr050578Node01Cell
<input type="checkbox"/>	WebSphere MQ messaging provider	WebSphere MQ Messaging Provider	Node=kr050578Node01,Server=server1
<input type="checkbox"/>	WebSphere MQ messaging provider	WebSphere MQ Messaging Provider	Node=kr050578Node01
<input type="checkbox"/>	WebSphere MQ messaging provider	WebSphere MQ Messaging Provider	Cell=kr050578Node01Cell

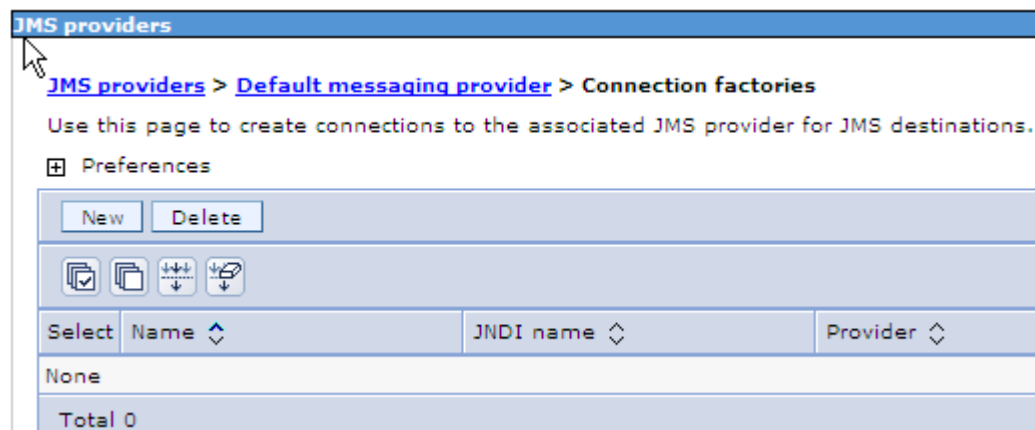
JMS Provider 세부 메뉴가 나오면 Connection factories 를 선택합니다.

[JMS providers](#) > [Default messaging provider](#)

A JMS provider enables messaging based on the Java Message Service (JMS). It provides J2EE connection JMS destinations.

The screenshot shows the 'Configuration' dialog box for the 'Default messaging provider'. It has two tabs: 'General Properties' and 'Additional Properties'. The 'General Properties' tab is active, showing fields for 'Scope' (Cell=kr050578Node01Cell), 'Name' (Default messaging provider), and 'Description' (Default messaging provider). The 'Additional Properties' tab is also visible, showing a list of sub-properties: 'Connection factories', 'Queue connection factories', 'Topic connection factories', 'Queues', 'Topics', and 'Activation specifications'. The 'Connection factories' sub-property is highlighted with a red dashed box.

Connection factories 메뉴가 나오면 New 버튼을 선택합니다.



New 버튼을 누르면 Connection factory 를 설정할 수 있는 화면이 하단 처럼 나옵니다.

[JMS providers](#) > [Default messaging provider](#) > [Connection factories](#) > [New](#)
A JMS connection factory is used to create connections to the associated JMS provider of JMS destinations, for both point-to-point and publish/subscribe messaging. Use connection factory administrative objects to manage JMS connection factories for the default messaging provider.

Configuration

General Properties

Administration

Scope

Provider

* Name

* JNDI name

Description

Category

Connection

* Bus name

Target

Target type

Target significance

The additional properties will not be available until the general properties for this item are applied or saved.

Additional Properties

- Connection pool properties

Related Items

- JAAS - J2C authentication data
- Buses

이름과 JNDI name 을 입력하고 이전에 만들어 놓은 SIB 를 선택하고 확인을 선택합니다.

[JMS providers](#) > [Default messaging provider](#) > [Connection factories](#) > [New](#)

A JMS connection factory is used to create connections to the associated JMS provider of JMS destinations, for both point-to-point and publish/subscribe messaging. Use connection factory administrative objects to manage JMS connection factories for the default messaging provider.

Configuration

General Properties

Administration

Scope

Cell=kr050578Node01Cell

Provider

Default messaging provider

* Name

JMSMsgCF

* JNDI name

jms/JMSMsgCF

Description

Category

The additional properties will not be available until the general properties for this item are applied or saved.

Additional Properties

- Connection pool properties

Related Items

- JAAS - J2C authentication data
- Buses

Connection

* Bus name

SIBBus01

Target

Target type

Bus member name

Target significance

Preferred

Target inbound transport chain

Connection factory 의 설정이 완료되면 이제 목적지인 Queue 설정을 하도록 하겠습니다. 관리콘솔에서 Resources > JMS > Queues 메뉴를 선택합니다.

Integrated Solutions Console Welcome gg5

View: All tasks

- Welcome
- Guided Activities
- Servers
- Applications
- Services
- Resources
 - Schedulers
 - Object pool managers
 - JMS
 - JMS providers
 - Connection factories
 - Queue connection factories
 - Topic connection factories
 - Queues
 - Topics
 - Activation specifications

Cell=kr050578Node01Cell, Profile=AppSrv01

Queues

Queues

A JMS queue is used as a destination for point-to-point messaging.

Scope: Cell=kr050578Node01Cell

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

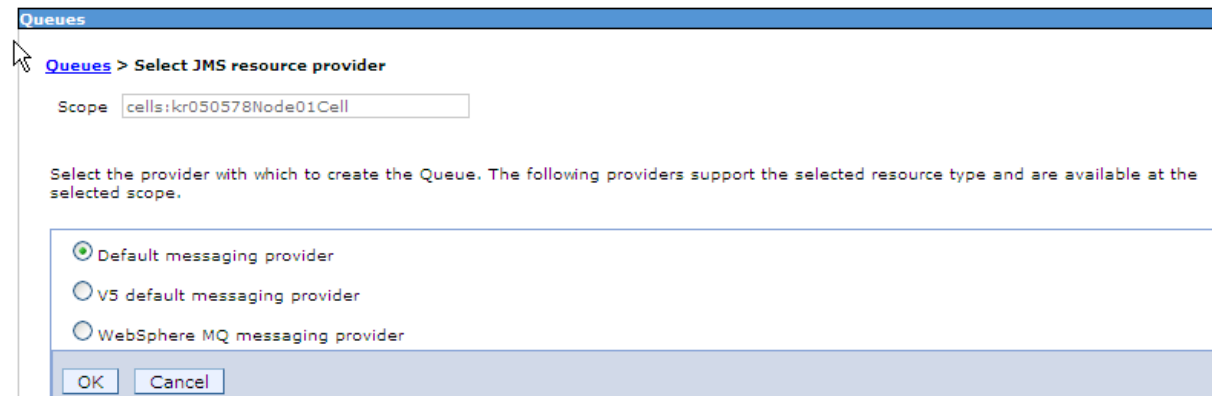
Cell=kr050578Node01Cell

Preferences

New Delete

Select	Name	JNDI name	Provider	Dest
None				
Total 0				

Queue 생성 마법사가 나타나면 provider 타입을 Default messaging provider 로 선택하고 다음을 클릭합니다.



Queues

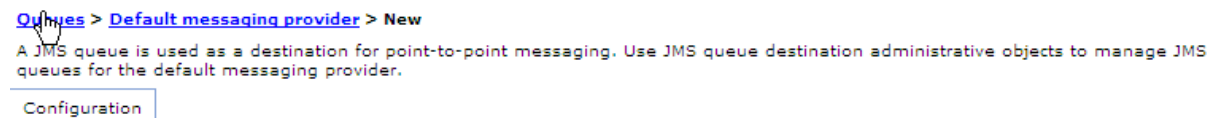
Queues > Select JMS resource provider

Scope:

Select the provider with which to create the Queue. The following providers support the selected resource type and are available at the selected scope.

☒ Default messaging provider
☐ V5 default messaging provider
☐ WebSphere MQ messaging provider

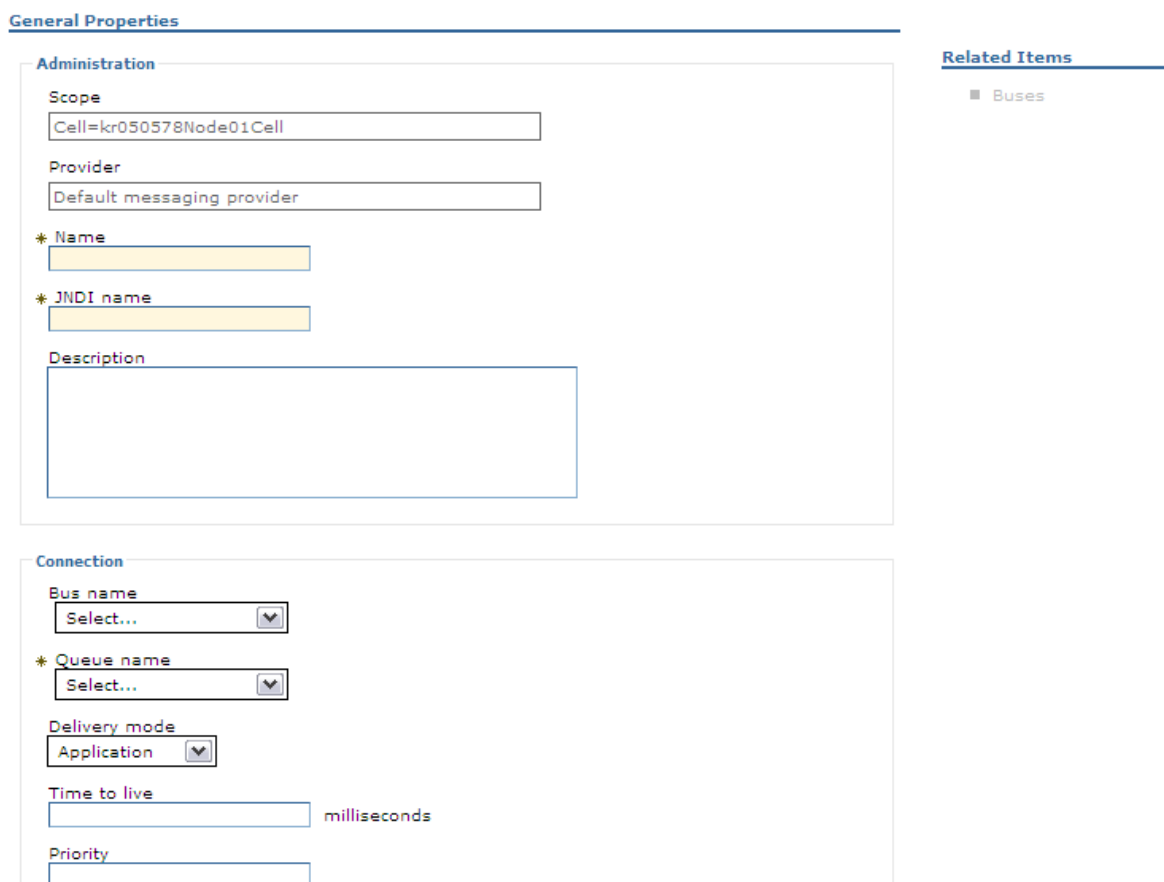
다음을 클릭하면 Queue 세부 설정을 할 수 있는 화면을 확인할 수 있습니다.



Queues > Default messaging provider > New

A JMS queue is used as a destination for point-to-point messaging. Use JMS queue destination administrative objects to manage JMS queues for the default messaging provider.

Configuration



General Properties

Administration

Scope:

Provider:

* Name:

* JNDI name:

Description:

Connection

Bus name:

* Queue name:

Delivery mode:

Time to live: milliseconds

Priority:

Related Items

- Buses

이름과 JNDI 명을 넣어준 후에 이전에 만든 Bus 를 선택합니다. 그리고 Queue name 을 입력하는 화면에서 create Service Integration Bus destination(SIB 목적지 생성) 항목을 선택합니다.

General Properties

Administration

Scope
Cell=kr050578Node01Cell

Provider
Default messaging provider

* Name
JMMsg

* JNDI name
jms/JMMsg

Description

Connection

Bus name
SIBBus01

* Queue name
Select...
Select...
SYSTEM.Exception.Destination.kr050578Node01.server1-SIBBus01
Create Service Integration Bus destination
other, please specify

Queue 목적지 생성 마법사가 나오면 식별자에 원하는 이름을 넣어주고 다음을 클릭합니다.

Create a new queue for point-to-point messaging.

→ **Step 1: Set queue attributes**

Step 2: Assign the queue to a bus member

Step 3: Confirm queue creation

Set queue attributes

Configure the attributes of your new queue

* Identifier
MsgQueue

Description

Next Cancel

해당 Queue 를 할당할 Bus member 를 선택하고 다음을 클릭합니다.

Create a new queue for point-to-point messaging.

Step 1: Set queue attributes	Assign the queue to a bus member
→ Step 2: Assign the queue to a bus member	Assign the queue to a bus member that will store and process the messages for the queue.
Step 3: Confirm queue creation	Bus member Node=kr050578Node01;Server=server1
<input type="button" value="Previous"/> <input type="button" value="Next"/> <input type="button" value="Cancel"/>	

마지막으로 요약정보를 확인하고 종료를 누르면 Queue 목적지 생성을 완료할 수 있습니다.

Create a new queue for point-to-point messaging.

Step 1: Set queue attributes	Confirm queue creation
Step 2: Assign the queue to a bus member	To complete creation of the queue, click Finish. If you want to change any selections, click Previous.
→ Step 3: Confirm queue creation	Summary of actions: New queue "MsgQueue" will be created. A Queue point for "MsgQueue" will be created for bus member "kr050578Node01;server1" of bus "SIBBus01".
<input type="button" value="Previous"/> <input type="button" value="Finish"/> <input type="button" value="Cancel"/>	

Queue 세부 설정으로 화면이 돌아오면 방금 만든 Queue 이름을 선택하고 확인을 클릭합니다.

General Properties

Administration	
Scope	Cell=kr050578Node01Cell
Provider	Default messaging provider
* Name	JMSMsg
* JNDI name	jms/JMSMsg
Description	
Connection	
Bus name	SIBBus01
* Queue name	MsgQueue

확인이 되면 지정된 이름으로 Queue 가 만들어진 것을 확인하실 수 있습니다.

Queues

A JMS queue is used as a destination for point-to-point messaging.

☐ Scope: Cell=kr050578Node01Cell

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Cell=kr050578Node01Cell

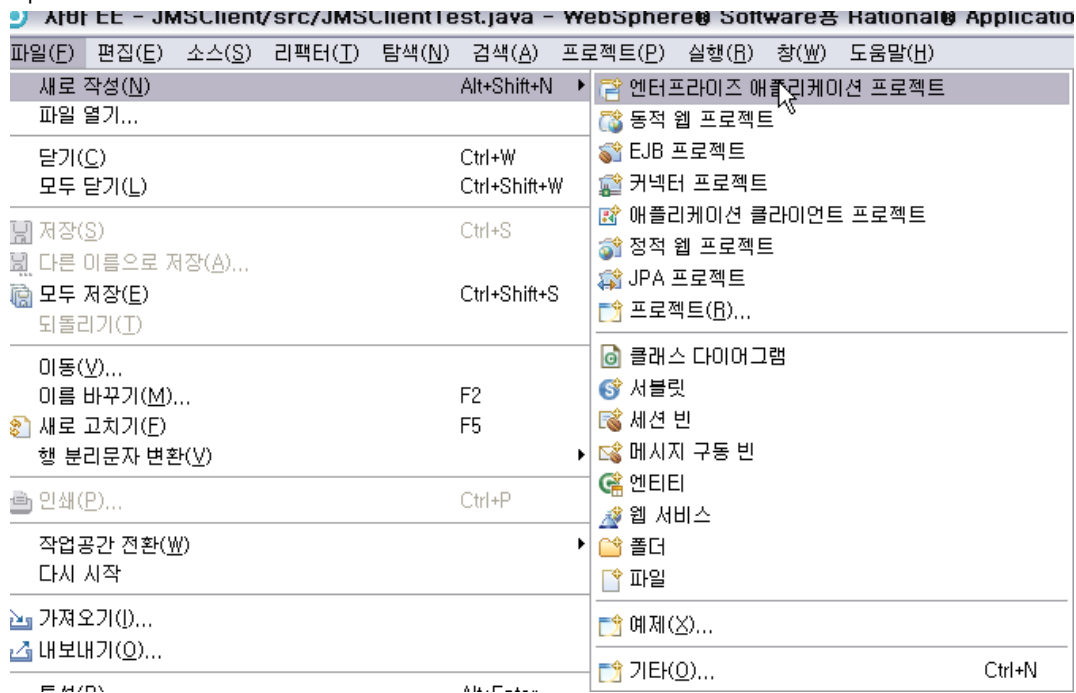
☐ Preferences

New Delete					
Select	Name ↕	JNDI name ↕	Provider ↕	Description ↕	Scope ↕
You can administer the following resources:					
<input type="checkbox"/>	JMMsg	jms/JMMsg	Default messaging provider		Cell=kr050578Node01Cell
Total 1					

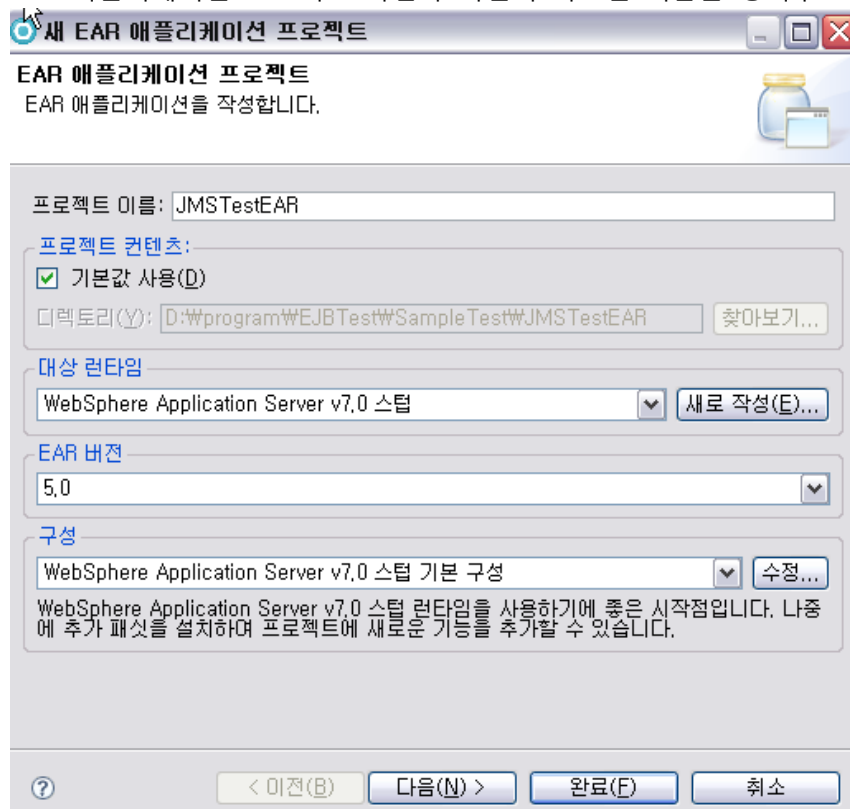
여기까지 잘 따라오셨다면 기본 JMS provider 를 사용하기 위해서 Connection factory 를 생성하는 것과 해당 Connection 의 목적지인 Queue 생성을 완료하신 것입니다. 이제 WAS 콘솔에서 진행할 준비는 끝났으니 실제 어플리케이션을 개발하여 Messaging 서비스를 해보도록 하겠습니다.

Part 3. JMS Client 개발

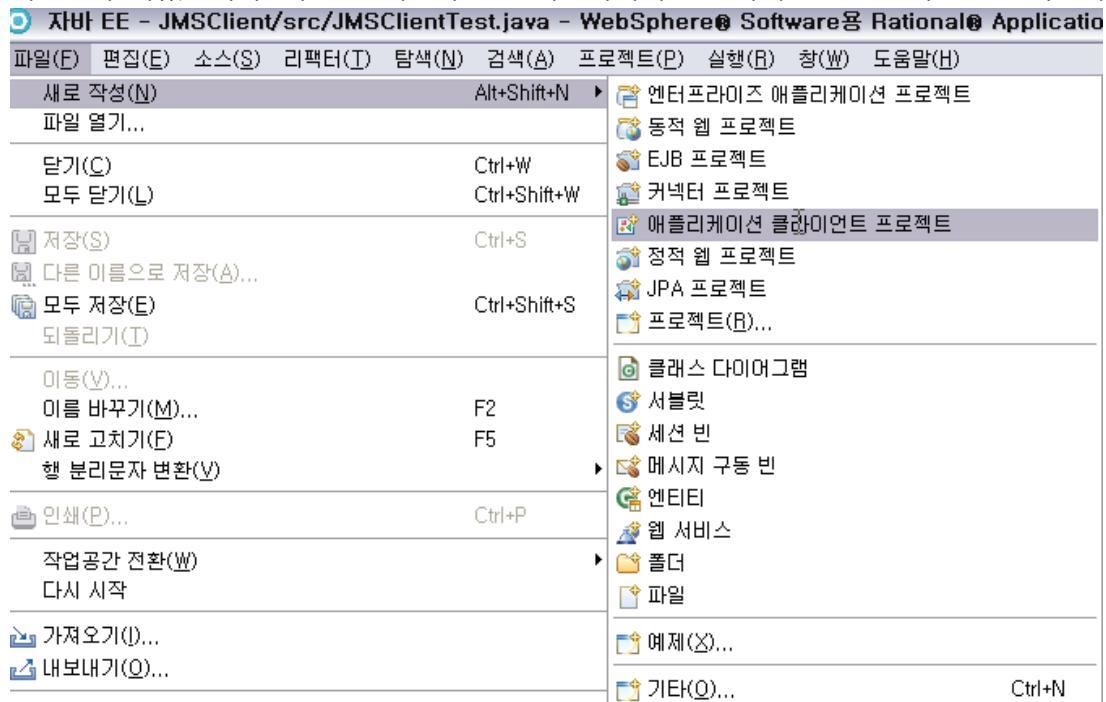
이전 단계까지는 JMS 서비스를 하기 위해서 WAS 설정을 하였습니다. 이제 실제 JMS Service 를 하기 위하여 JMS Client 를 개발하고 이를 이용해서 Message 를 전달을 해보도록 하겠습니다. RAD v7.5 를 시작한 후에 파일 > 새로 작성 > 엔터프라이즈 어플리케이션 프로젝트를 선택합니다.



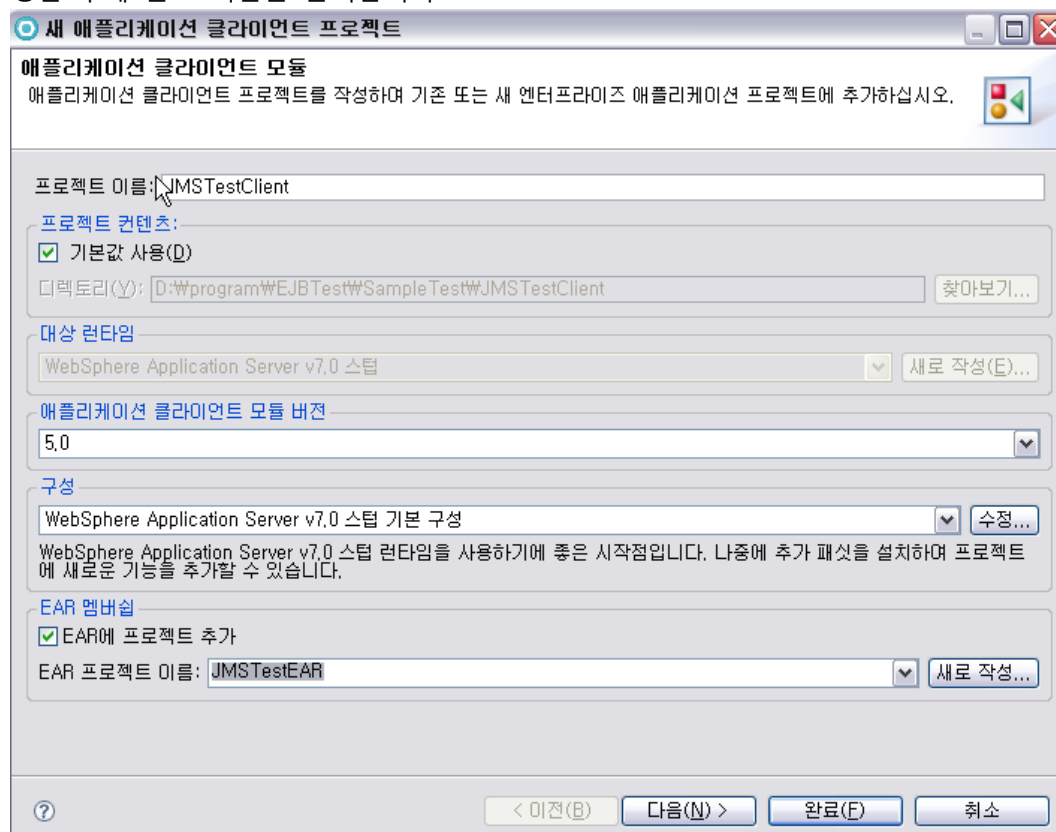
EAR 어플리케이션 프로젝트 마법사 화면이 나오면 이름을 넣어주고 완료버튼을 클릭합니다.



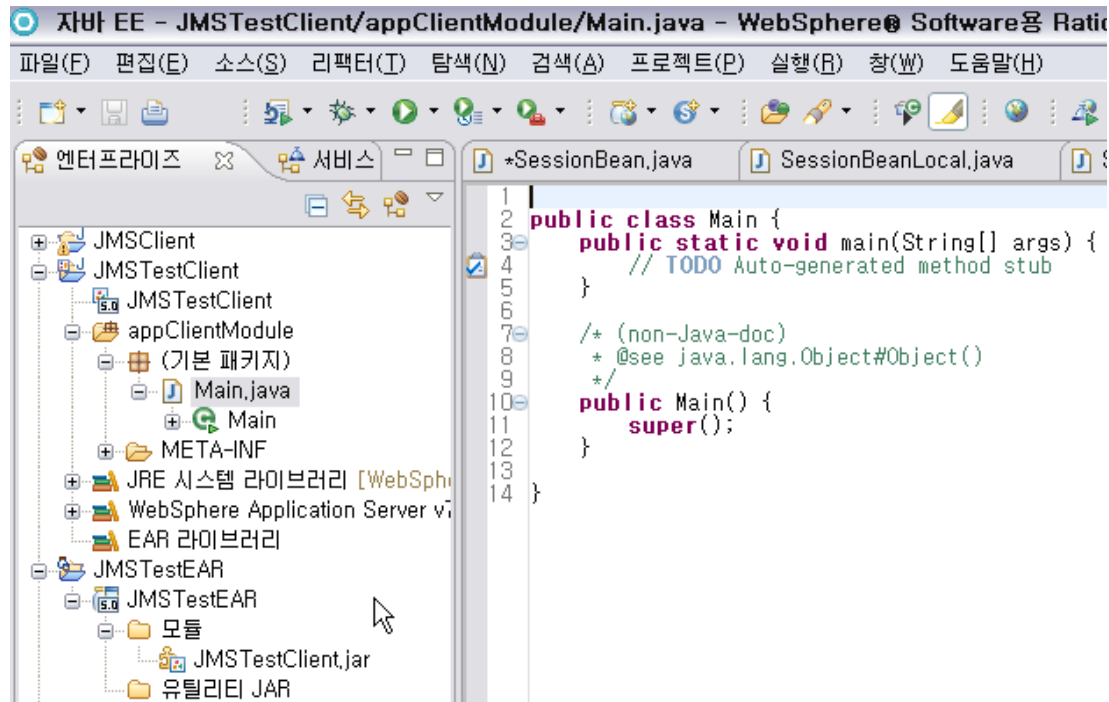
EAR 프로젝트를 만들었으면 이제 실질적으로 사용될 어플리케이션 클라이언트 프로젝트를 만들 어보도록 하겠습니다. 파일 > 새로작성 > 어플리케이션 클라이언트 프로젝트를 선택합니다.



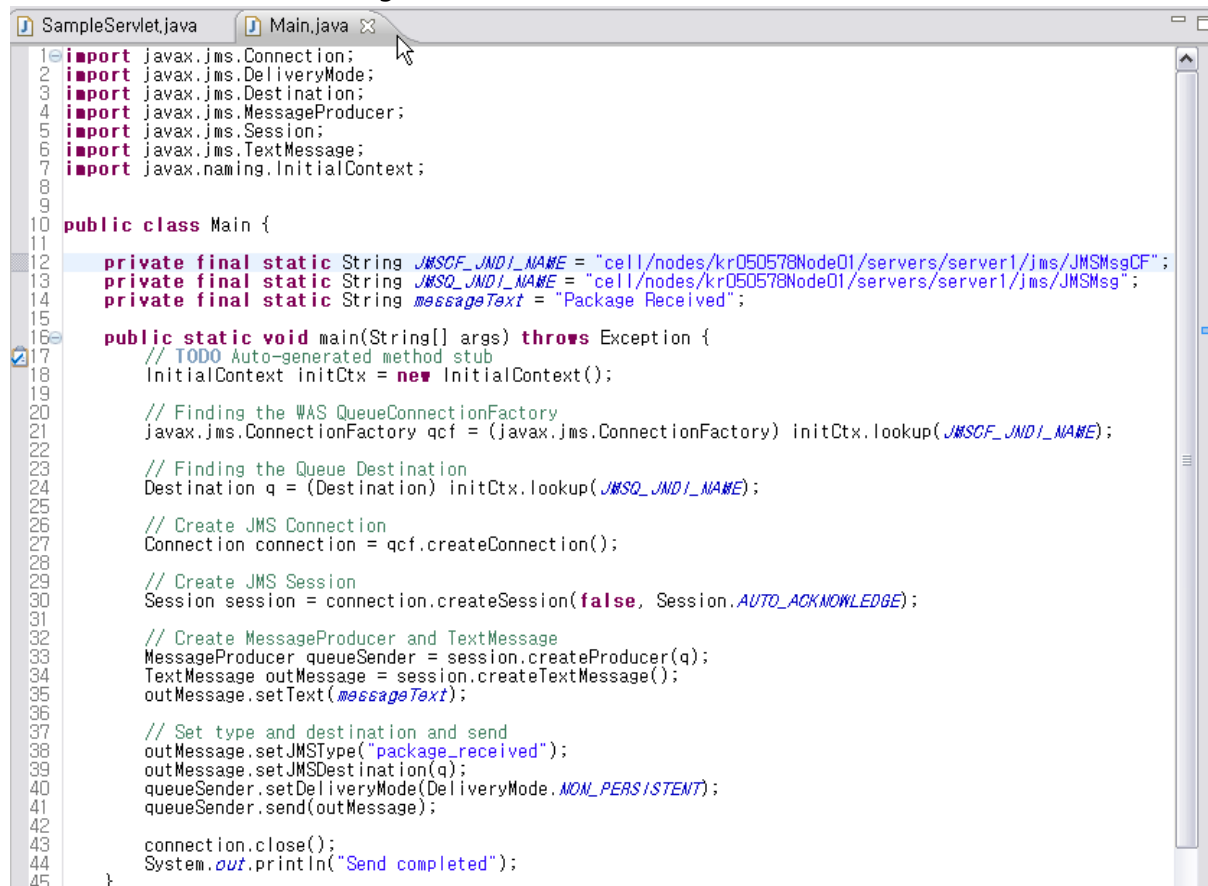
어플리케이션 클라이언트 프로젝트 마법사에서 이름을 넣어주고 이전에 만든 EAR 에 추가를 설정한 후에 완료 버튼을 클릭합니다.



완료를 하게 되면 하단 처럼 기본적으로 생성된 Main 클래스를 확인하실 수 있습니다.



위의 Main 클래스를 이용해서 JMS Message 를 보낼 수 있는 Code 를 작성합니다. 본 Code 는 Connection Factory 를 받아와서 Connection 을 만들고 Session을 만든 후에 목적지로 해당 Queue 를 연결한 후에 Message 를 만들어 Send 하는 단순한 어플리케이션입니다.



위의 소스 중에 가장 중요한 것은 Connection factory 와 Queue 를 받아오기 위한 JNDI 이름입니다. 이를 받아오기 위해서 단순히 jms/JMSMsgCF 같은 이름만 쓰면 안되고 실제 WAS에 등록된 JNDI 이름을 사용해서 받아 와야 합니다. WAS에 등록된 JNDI 이름을 확인하기 위해서는 하단 처럼 CMD 창으로 들어갑니다.

WAS product 가 설치된 폴더로 이동해서 bin 폴더에서 dumpNameSpace.bat 을 입력하면 WAS 에 등록된 JNDI 이름을 모두 확인할 수 있습니다.

```

D:\IBM\WebSphere7\AppServer\profiles\AppSrv01\bin>dumpNameSpace.bat

초기 컨텍스트 가져오는 중
시작 컨텍스트 가져오는 중

=====
네임 스페이스 덤프
컨텍스트 팩토리: com.ibm.websphere.naming.WsnInitialContextFactory
프로바이더 URL: corbaloc:iiop:localhost:2809
요청된 루트 컨텍스트: cell
시작 컨텍스트: <top>=kr050578Node01Cell
형식화 규칙: jndi
덤프 시간: Fri Jun 26 18:34:17 KST 2009
=====

네임 스페이스 덤프 시작
=====

1 <top>
2 <top>/persistent                                javax.naming.Context
  
```

하단에 보시면 이번에 등록된 JMS 관련 JNDI 이름을 compound name 으로 확인할 수 있습니다. 해당 이름 앞에 cell 만 붙이시고 사용하시면 됩니다.

(만약, 등록한 JMS 의 JNDI 이름이 나타나지 않으면 서버를 재시작한 후 시도해 보기 바랍니다.)

```

5 <top>/nodes/kr050578Node01                        javax.naming.Context
6 <top>/nodes/kr050578Node01/node                    javax.naming.Context
6   kr050578Node01Cell/nodes/kr050578Node01 컨텍스트에 링크
7 <top>/nodes/kr050578Node01/cell                    javax.naming.Context
7   kr050578Node01Cell 컨텍스트에 링크
8 <top>/nodes/kr050578Node01/nodename                 java.lang.String
9 <top>/nodes/kr050578Node01/servers                  javax.naming.Context
10 <top>/nodes/kr050578Node01/servers/server1         javax.naming.Context
11 <top>/nodes/kr050578Node01/servers/server1/jms     javax.naming.Context
12 <top>/nodes/kr050578Node01/servers/server1/jms/JMSMsg
12                                                    com.ibm.ws.sib.api.jms.
impl.JmsQueueImpl
13 <top>/nodes/kr050578Node01/servers/server1/jms/JMSMsgCF
13                                                    javax.jms.ConnectionFac
tory
14 <top>/nodes/kr050578Node01/servers/server1/jms/JMSMsgAS
14                                                    com.ibm.ejs.j2c.Activat
ionSpecBindingInfo
  
```

위에서 찾은 이름을 실제 호출할 JNDI 이름으로 넣어주면 JMS Client 개발은 완료됩니다.

```
public class Main {  
    private final static String JMSCF_JNDI_NAME = "cell/nodes/kr050578Node01/servers/server1/jms/JMSMsgCF";  
    private final static String JMSQ_JNDI_NAME = "cell/nodes/kr050578Node01/servers/server1/jms/JMSMsg";  
    private final static String messageText = "Package Received";  
}
```

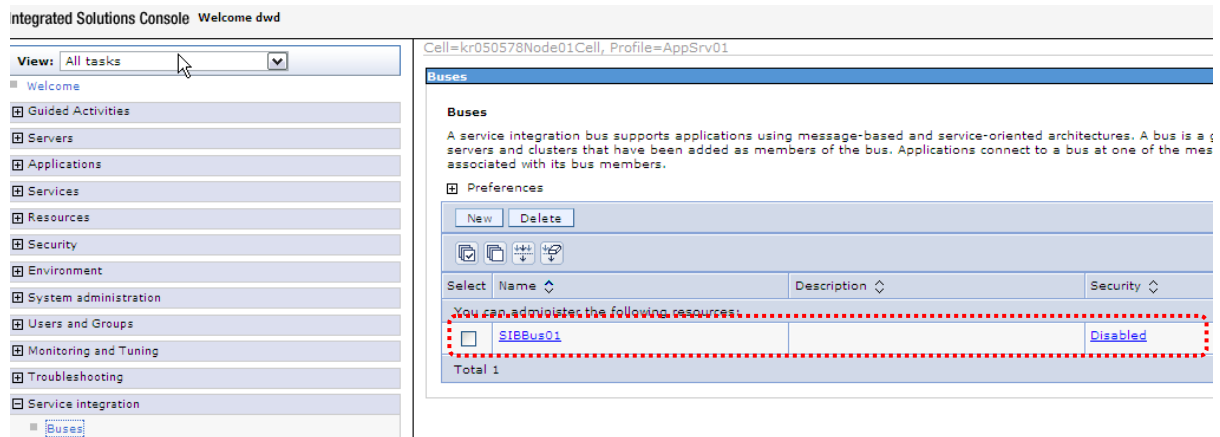
JMS Client 를 완료했다면 해당 소스를 EAR 로 반출해야 사용하실 수 있습니다. EAR로 반출한 후에 Client 를 실행할 수 있는 launchClient.bat 스크립트를 이용해서 해당 EAR 파일을 수행하면 됩니다. (launchClient.bat 파일은 서버 시작 스크립트가 위치한 동일 bin 폴더에 있습니다.)

```
D:\wbiz_work\01.SWG\00.WAS\WAS_Sample\EAR>D:\IBM\WebSphere7\AppServer\profiles\App  
pSvc01\bin\launchClient.bat JMSTestEAR.ear  
IBM WebSphere Application Server, 릴리스 7.0  
Java EE Application Client Tool  
Copyright IBM Corp., 1997-2008  
WSCL0012I: 명령행 인수 처리 중.  
WSCL0013I: Java EE 응용프로그램 클라이언트 환경 초기화 중  
[09. 6. 27 0:40:44:187 KST] 00000000 W UOW=null source=com.ibm.ws.ssl.config.  
SSLConfig org=IBM prod=WebSphere component=Application Server thread=[P=843312:0  
=0:CTI]  
CWPKI0041W: 하나 이상의 키 스토어에서 기본 암호를 사용하고 있습니다.  
WSCL0035I: Java EE Application Client Environment의 초기화가 완료되었습니다.  
WSCL0014I: 응용프로그램 클라이언트 class Main 호출 중  
Send completed  
D:\wbiz_work\01.SWG\00.WAS\WAS_Sample\EAR>
```

위의 그림 처럼 "Send completed" 라는 메시지가 나오면 성공적으로 message 전송을 완료한 것입니다.

Part 4. Queue 확인

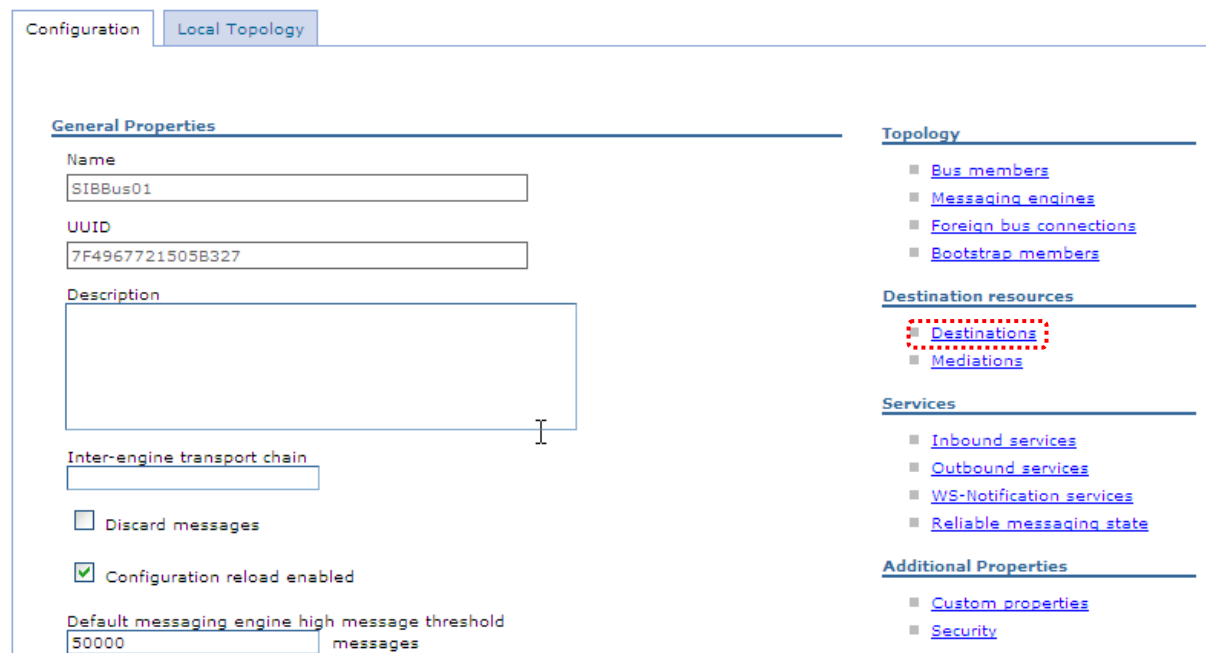
JMS 클라이언트를 이용해서 Message 를 전송했으므로 WAS 관리콘솔에서 해당 Queue 에 들어가 실제로 데이터가 전송되었는지 확인해보도록 하겠습니다. 관리콘솔에서 System integration > Buses 메뉴에 들어가 해당 SIB 를 선택하여 세부메뉴로 들어갑니다.



SIB 세부메뉴에서 Destination 을 선택합니다.

[Buses](#) > [SIBBus01](#)


A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.







Destination 리스트가 나오면 이번에 새로 생성하고 Message 를 보냈던 Queue 이름을 선택합니다.

[Buses](#) > [SIBBus01](#) > [Destinations](#)

A bus destination is defined on a service integration bus, and is hosted by one or more locations within the bus. The destination can be used as producers, consumers, or both to exchange messages.

 Preferences

<div>NewDeleteMediateUnmediate</div>				
<div></div>				
Select	Identifier	Bus	Type	Description
You can administer the following resources:				
<input type="checkbox"/>	Default.Topic.Space	SIBBus01	Topic space	
<input type="checkbox"/>	MsgQueue	SIBBus01	Queue	
<input type="checkbox"/>	SYSTEM.Exception.Destination.kr050578Node01.server1-SIBBus01	SIBBus01	Queue	
Total 3				

해당 Queue 의 세부메뉴가 나오면 Queue point 를 선택합니다.

[Buses](#) > [SIBBus01](#) > [Destinations](#) > [MsgQueue](#)

A queue for point-to-point messaging.

Configuration

General Properties

Identifier
MsgQueue

UUID
0F5BEC22740CE026E64A2FAF

Type
Queue

Description

Mediation

Quality of Service

☒ Enable producers to override default reliability

Default reliability
Assured persistent

Maximum reliability
Assured persistent

Default priority
0

Exception destination

☐ None

☐ Override messaging engine blocked retry timeout default

Blocked retry timeout in milliseconds

Message points

☒ Queue points

☐ Mediation points

Additional Properties

Context properties

Mediation execution points

Related Items

Application resources topology

Queue point 에 들어가시면 하단의 그림처럼 Queue point 이름과 함께 현재 큐의 깊이(depth)가 나옵니다. 지금 JMS Client 로 Message 를 하나 전송하였기 때문에 1로 되어 있는 것을 확인할 수 있습니다.

[Buses](#) > [SIBBus01](#) > [Destinations](#) > [MsgQueue](#) > Queue points

The message point for a queue, for point-to-point messaging.

Preferences

Refresh	
Identifier	Queue Depth
You can administer the following resources:	
MsgQueue@kr050578Node01.server1-SIBBus01	1
Total 1	

좀 더 명확한 확인을 위하여 JMS client 를 다시 실행시켜 Message 를 하나 더 전송합니다.

```
D:\Whiz_work\W01.SVGW00.WAS\WAS_Sample\WEAR>D:\IBM\WebSphere7\AppServer\profiles\WAp
pSrv01\bin\launchClient.bat JMSTestEAR.ear
IBM WebSphere Application Server, 릴리스 7.0
Java EE Application Client Tool
Copyright IBM Corp., 1997-2008
WSCL0012I: 명령행 인수 처리 중.
WSCL0013I: Java EE 응용프로그램 클라이언트 환경 초기화 중
[09. 6. 27 0:49:53:437 KST] 00000000 W UOW=null source=com.ibm.ws.ssl.config.
SSLConfig org=IBM prod=WebSphere component=Application Server thread=IP=392593:0
=0:CTI
CWPKI0041W: 하나 이상의 키 스토어에서 기본 암호를 사용하고 있습니다.
WSCL0035I: Java EE Application Client Environment의 초기화가 완료되었습니다.
WSCL0014I: 응용프로그램 클라이언트 class Main 호출 중
Send completed
```

다시 Queue point 를 선택해서 보면 Queue Depth 가 2로 증가된 것을 확인하실 수 있습니다. 그러면 이제 좀 더 자세히 해당 Message 를 확인하기 위해서 Queue point 이름을 클릭하여 세부 정보로 들어갑니다.

[Buses](#) > [SIBBus01](#) > [Destinations](#) > [MsgQueue](#) > Queue points

The message point for a queue, for point-to-point messaging.

Preferences

Refresh	
Identifier	Queue Depth
You can administer the following resources:	
MsgQueue@kr050578Node01.server1-SIBBus01	2
Total 1	

Queue point 세부 정보 화면에서 Runtime 탭을 선택합니다.

[Bus](#) > [SIBBus01](#) > [Destinations](#) > [MsgQueue](#) > [Queue points](#) > [MsgQueue@kr050578Node01.server1-SIBBus01](#)
The message point for a queue, for point-to-point messaging.

Configuration **Runtime**

General Properties

Identifier

UUID

Destination type

High message threshold
 messages

☒ Send allowed

Target UUID

Runtime 탭을 선택하면 현재의 Message depth를 확인할 수 있습니다. 이를 확인한 후에 Messages 메뉴를 선택합니다.

[Bus](#) > [SIBBus01](#) > [Destinations](#) > [MsgQueue](#) > [Queue points](#) > [MsgQueue@kr050578Node01.server1-SIBBus01](#)
The message point for a queue, for point-to-point messaging.

Configuration **Runtime**

General Properties	Additional Properties
Identifier <input type="text" value="MsgQueue"/>	Messages Known remote queue points
High message threshold <input type="text" value="50000"/>	
<input checked="" type="checkbox"/> Send allowed	
Current message depth <input type="text" value="2"/>	
<input type="button" value="OK"/>	

1

The messages on the message point.

☒
☐
☐
☐

Select	Position	System message ID	State	Transaction ID
You can administer the following resources:				
<input type="checkbox"/>	1	CD4CEC2CAA23F134_500001	Available	
<input type="checkbox"/>	2	CD4CEC2CAA23F134_500002	Available	
Total 2				

1

The properties for a message on the message point.

Message Properties	Additional Properties
System message ID <input type="text" value="CD4CEC2CAA23F134_500001"/>	<div>Message body</div>
State <input type="text" value="Available"/>	
Transaction ID <input type="text"/>	
Message type <input type="text" value="JMS"/>	
Approximate length <input type="text" value="537"/> Bytes	
Time stamp <input type="text" value="Jun 27, 2009 12:40:56 AM"/>	
Message wait time <input type="text" value="11 minutes"/>	
Current messaging engine arrival time <input type="text" value="Jun 27, 2009 12:40:56 AM"/>	
Redelivered count <input type="text" value="0"/>	
Security user ID <input type="text"/>	
Producer type <input type="text" value="API"/>	
Exception destination timestamp <input type="text"/>	
Exception destination reason <input type="text"/>	

API Message properties

Message ID

Message body 메뉴를 선택하면 해당 Message 의 body 를 확인하실 수 있습니다. 하단의 화면에서도 볼 수 있지만 JMS Client 어플리케이션에서 설정한 'Package Received' 하는 Text 를 확인하실 수 있습니다.

[Buses](#) > [SIBBus01](#) > [Destinations](#) > [MsgQueue](#) > [Queue points](#) > [MsgQueue@kr050578Node01.server1-SIBBus01](#) > [Messages](#) > [CD4CEC2CAA23F134_500001](#) > Message body

The contents of the message body.

Approximate total message size (including headers)

537 Bytes

* Displayable message body size

1024 Bytes

0000000016: 5061636b 61676520 52656365 69766564 'Package Received'

Apply Cancel

여기까지 잘 따라오셨다면 WAS 에서 SIB 설정을 하고 기본 JMS provider 를 이용해서 설정한 후에 JMS Client 로 Message 전송까지 완료한 것입니다. 어때요? 처음 이야기는 어려웠을지 모르지만 실제로 따라 해보면서 진행해보니 쉽다는 것을 이해할 수 있나요? 이제 어느 정도 JMS 에 대해서 이해가 되었다면 이번 강의는 여기서 마무리 하고 다음 강의로 넘어가겠습니다. 다음 강의는 이번 강좌에서 JMS client 이용해서 보냈던 Message 를 비동기적으로 서버 어플리케이션에서 활용하기 위한 MDB(Message Driven Bean) 관련입니다. 그럼 다음강좌로 휘리릭~~~

참고 1) IBM Information Center for WebSphere Application Server v7

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.multipplatform.doc/info/welcome_nd.html

참고 2) InforCenter – Developing a JMS client

(http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.doc/info/aes/aes/tmj_devcl.html?resultof=%22%6a%6d%73%22%20%22%6a%6d%22%20)

※이 자료의 저작권은 작성자에게 있으며 유포는 자유로이 허용되나 상업적으로 이용은 금합니다.