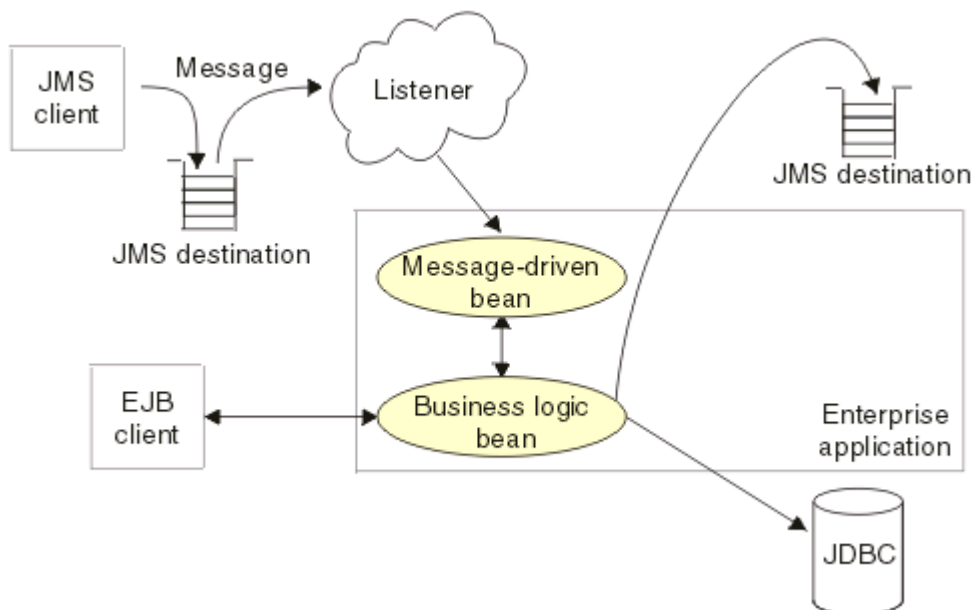


하나씩 쉽게 따라 해보는 IBM WebSphere Application Server(WAS) v7 – 8

이정운 (juwlee@kr.ibm.com)

하나씩 쉽게 따라 해보는 IBM WAS v7 시리즈 그 여덟번째 이야기 MDB 입니다.(벌써 여덟번째라니! ^^&) MDB 란 Message Driven Bean 의 약자로서 EJB 의 여러가지 Bean 중의 한 종류 입니다.(Session Bean, Entity Bean 그리고 Message Driven Bean) MDB는 비동기적인 메시지 처리를 위해 사용되는 Bean 클래스로서 동시성을 가지고 어플리케이션이 구동되거나 Message 가 처리되는 것이 아니라 JMS Client 같은 것들이 WAS 내의 해당 Queue 로 Message 를 전송하게 되면 그 Action 이 발생된 시점에 Event 가 발생하며 MDB 에서 정의된 기능이 수행되는 클래스 입니다. 즉 MDB 는 해당 Queue 나 Topic 을 지속적으로 모니터링 하다가 Message 가 전송되는 Event 를 통해서 수행되는 Bean 클래스입니다. 이처럼 보통 MDB 는 해당 Queue 나 Topic 에 등록하여 리스너하다가 Message 가 전송되면 비동기적으로 처리하는 서버사이드 어플리케이션에서 많이 사용됩니다. 자,그럼 실제 MDB 어플리케이션 작성과 수행을 하나씩 진행해 볼까요.

(이 강좌를 수행하시려면 반드시 7강의 SIB 설정과 Queue 설정이 필요합니다. 그러므로 이전 강좌를 완료하신 후에 수행해야만 합니다.)

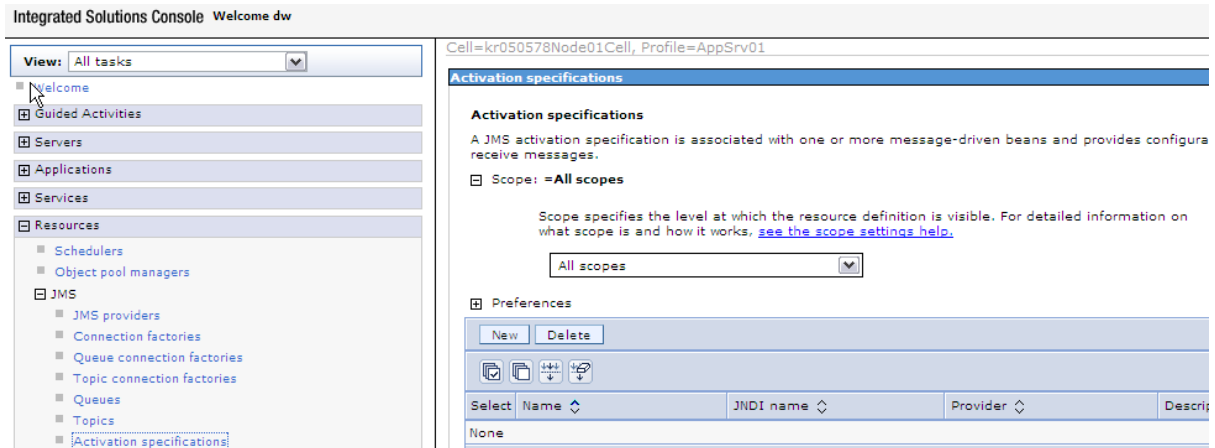


(참조:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/cmb_ovrvw.html)

Part 1. Activation Spec 설정

위에 설명한 것 처럼 MDB를 작성하기 전에 SIB 설정과 목적지인 Queue 설정이 완료되어야 합니다. 두 설정이 완료되면 이제 Activation Specification 설정을 해야 합니다. Activation specification 설정은 MDB를 해당 목적지인 Queue 에 등록하며 Message 를 받기위한 설정입니다. 해당 설정을 하기 위하여 관리콘솔에서 Resources > JMS > Activation specifications 메뉴를 선택합니다.



Scope 를 Cell 로 변경하고 Activation specification 을 새로 추가하기 위하여 New 버튼을 클릭합니다.

Activation specifications

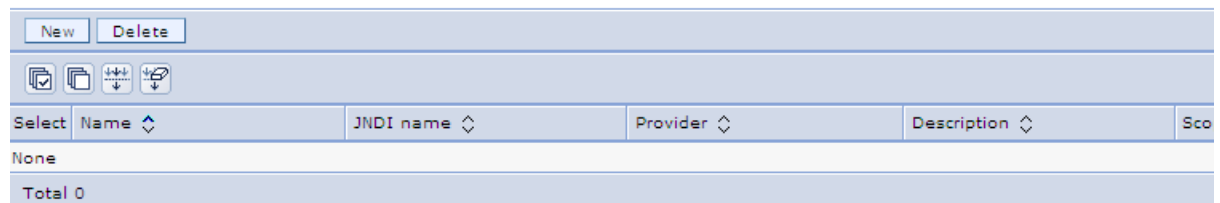
A JMS activation specification is associated with one or more message-driven beans and provides configuration necessary for them to receive messages.

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



연결할 Messaging provider 타입을 선택하는 화면이 나오면 기본 Messaging provider 를 선택하고 OK 를 클릭합니다.

Activation specifications > Select JMS resource provider

Scope: cells:kr050578Node01Cell

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

☒ Default messaging provider
☐ WebSphere MQ messaging provider

OK Cancel

다음으로 Activation specification 의 세부 설정 메뉴를 볼 수 있습니다.

[Activation specifications](#) > [Default messaging provider](#) > [New](#)

A JMS activation specification is associated with one or more message-driven beans and provides the configuration necessary for them to receive messages.

Configuration

General Properties

Administration

Scope
Cell=kr050578Node01Cell

Provider
Default messaging provider

* Name
[]

* JNDI name
[]

Description
[]

Destination

* Destination type
Queue [v]

* Destination JNDI name
[]

Message selector
[]

Related Items

- JAAS - J2C authentication data
- Buses

세부 설정 메뉴가 나오면 하단과 같이 이름과 JNDI 이름을 등록하고 리슨하고자 하는 목적지의 타입과 JNDI 이름, SIB bus 이름을 입력합니다.

[Activation specifications](#) > [Default messaging provider](#) > [New](#)

A JMS activation specification is associated with one or more message-driven beans and provides the configuration necessary for them to receive messages.

Configuration

General Properties

Administration

Scope
Cell=kr050578Node01Cell

Provider
Default messaging provider

* Name
JMSMsgAS

* JNDI name
jms/JMSMsgAS

Description
[]

Destination

* Destination type
Queue [v]

* Destination JNDI name
jms/JMSMsg

Message selector
[]

* Bus name
SIBBus01 [v]

Acknowledge mode
Auto-acknowledge [v]

Related Items

- JAAS - J2C authentication data
- Buses

이후 확인 버튼을 누르면 하단처럼 지정된 이름으로 Activation specification 이 설정되어 있는 것을 확인할 수 있습니다.

Activation specifications



A JMS activation specification is associated with one or more message-driven beans and provides configuration necessary for them to receive messages.

☐ 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 ▼

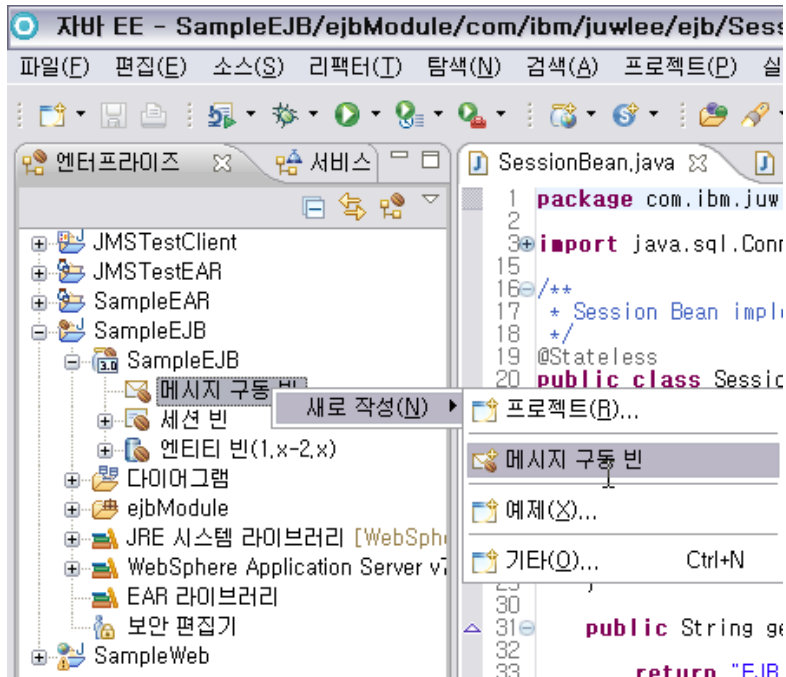
☐ Preferences

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

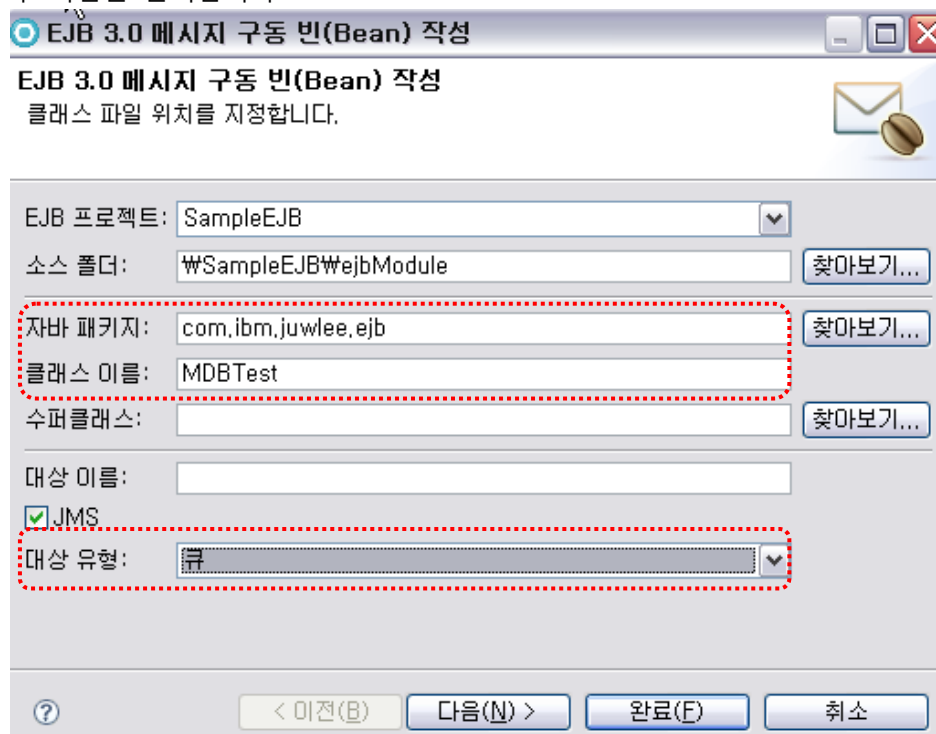
여기까지 잘 따라오셨다면 MDB를 연결하기 위한 Activation specification 설정을 마무리 하신 것입니다. MDB 어플리케이션을 사용하실 때는 이것만 잘 설정하시면 되므로 그리 어렵지 않으실 것입니다. 그럼 실제 MDB 어플리케이션을 간단하게 작성해보도록 하겠습니다.

Part 2. MDB 어플리케이션 개발

Activation specification 설정이 완료되었다면 MDB 어플리케이션을 개발하기 위하여 RAD v7.5 를 실행시킵니다. 새롭게 어플리케이션을 작성하는 것이 아니라 이전 강좌까지 사용되었던 SampleEAR.ear 어플리케이션을 활용해서 추가하는 작업을 하도록 하겠습니다. SampleEAR 어플리케이션을 열고 메시지 구동 빈 위치에서 마우스 우 클릭을 합니다. 팝업 메뉴에서 새로 작성 > 메시지 구동 빈을 누릅니다.



메시지 구동 빈 작성 마법사 화면이 나오면 자바 패키지 명과 클래스 이름, 대상 유형을 지정한 후 다음을 클릭합니다.



마법사의 다음 단계를 확인하고 완료를 클릭합니다.

EJB 3.0 메시지 구동 빈(Bean) 작성

EJB 3.0 메시지 구동 빈(Bean) 작성
메시지 구동 빈(Bean) 특정 정보를 입력합니다.

빈 이름:

트랜잭션 유형:

인터페이스: 추가... 제거

메시지 리스너 인터페이스: [javax.jms.MessageListener](#)

작성하려는 메소드 스탭 선택

- ☒ 상속된 abstract 메소드
- ☒ 수퍼클래스로부터의 생성자

? < 이전(B) 다음(N) > 완료(F) 취소

완료를 하게 되면 MDB 를 위한 클래스가 동적으로 생성되며 필요한 요건에 맞는 소스 틀이 하 단처럼 만들어 집니다.

The screenshot shows an IDE with a project structure on the left and the source code of `MDBTest.java` on the right. The project structure includes `JMSTestClient`, `JMSTestEAR`, `SampleEAR`, `SampleEJB`, `다이머그램`, `ejbModule`, `com.ibm.juwlee.ejb` (containing `MDBTest.java` and `MDBTest`), `SessionBean.java`, `SessionBeanLocal.java`, `META-INF`, `JRE 시스템 라이브러리 [WebSphere]`, `WebSphere Application Server v7`, `EAR 라이브러리`, `보안 편집기`, and `SampleWeb`. The `MDBTest.java` file is selected, and its content is displayed in the editor.

```
1 package com.ibm.juwlee.ejb;
2
3 import javax.ejb.ActivationConfigProperty;
4
5 /**
6  * Message-Driven Bean implementation class for: MDBTest
7  */
8
9 @MessageDriven(
10     activationConfig = { @ActivationConfigProperty(
11         propertyName = "destinationType", propertyValue = "javax.jms.Queue"
12     ) })
13 public class MDBTest implements MessageListener {
14
15     /**
16      * Default constructor.
17      */
18     public MDBTest() {
19         // TODO Auto-generated constructor stub
20     }
21
22     /**
23      * @see MessageListener#onMessage(Message)
24      */
25     public void onMessage(Message message) {
26         // TODO Auto-generated method stub
27     }
28 }
29
30
31
32
33 }
```

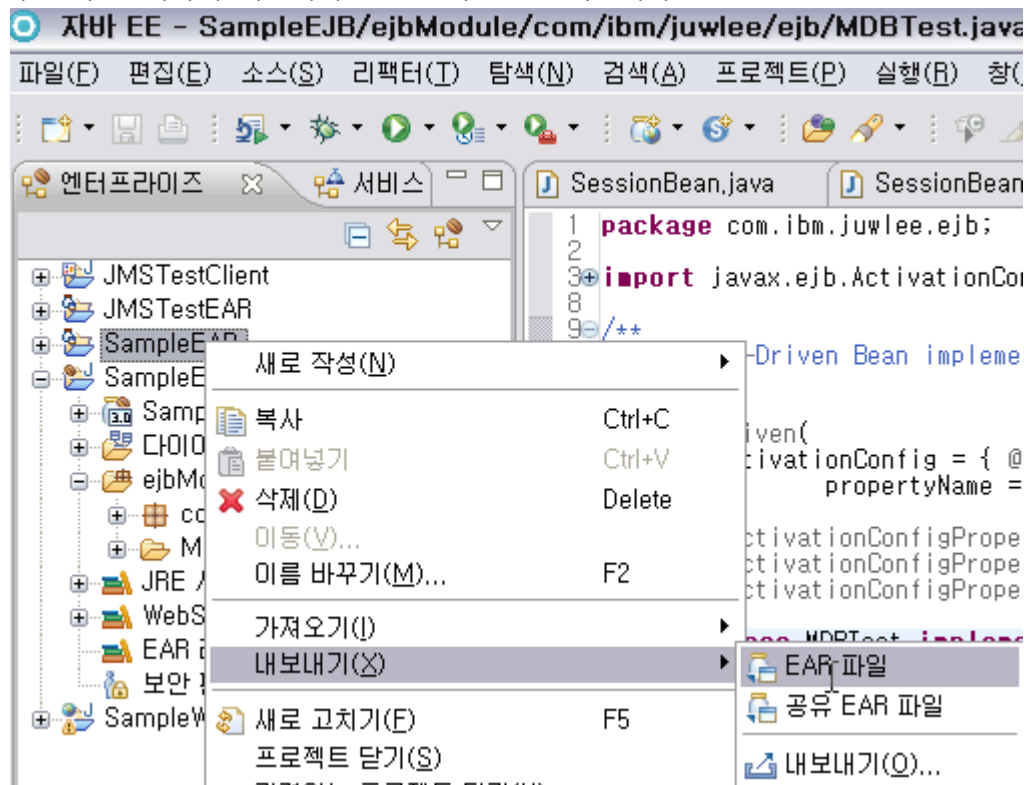
해당 소스에서 onMessage() 메소드 부분이 event 가 발생되었을 때 실제로 실행되는 소스부분입니다. 따라서 간단하게 Message 를 출력하는 소스를 하단 처럼 입력합니다.

```
SessionBean.java SessionBeanLocal.java SampleServlet.java MDBTest.java x
1 package com.ibm.juwlee.ejb;
2
3 import javax.ejb.ActivationConfigProperty;
4
5 /**
6  * Message-Driven Bean implementation class for: MDBTest
7  */
8
9 @MessageDriven(
10     activationConfig = { @ActivationConfigProperty(
11         propertyName = "destinationType", propertyValue = "javax.jms.Queue"
12     ) })
13 public class MDBTest implements MessageListener {
14
15     /**
16      * Default constructor.
17      */
18     public MDBTest() {
19         // TODO Auto-generated constructor stub
20     }
21
22     /**
23      * @see MessageListener#onMessage(Message)
24      */
25     public void onMessage(Message message) {
26         // TODO Auto-generated method stub
27         javax.jms.TextMessage txtMsg = (javax.jms.TextMessage)message;
28         try
29         {
30             System.out.println("MESSAGE -> " + txtMsg.getText());
31         }
32         catch (JMSEException e)
33         {
34             e.printStackTrace(System.out);
35         }
36     }
37 }
38
39
40
41
42 }
```

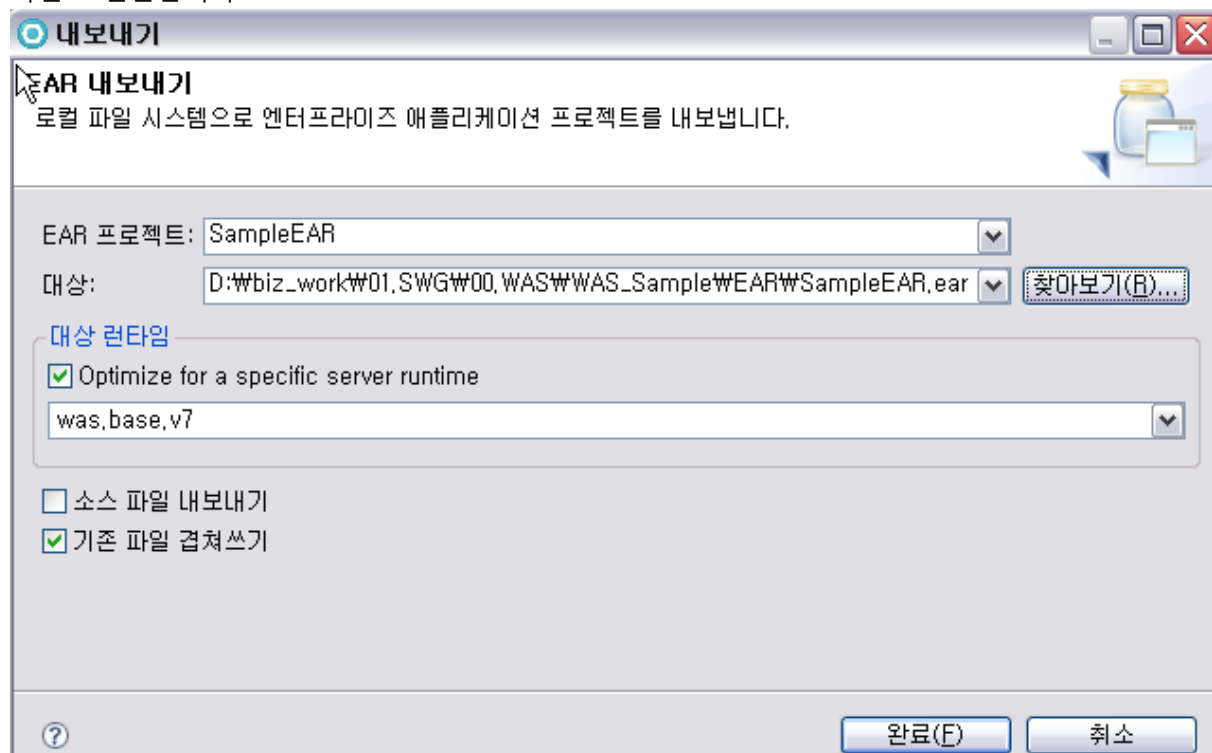
위에 추가하여 실제 호출할 Connection factory 와 Queue 의 JNDI 이름을 EJB 3.0 타입의 property 로 넣어줍니다.

```
SessionBean.java SessionBeanLocal.java SampleServlet.java MDBTest.java x
1 package com.ibm.juwlee.ejb;
2
3 import javax.ejb.ActivationConfigProperty;
4
5 /**
6  * Message-Driven Bean implementation class for: MDBTest
7  */
8
9 @MessageDriven(
10     activationConfig = { @ActivationConfigProperty(
11         propertyName = "destinationType", propertyValue = "javax.jms.Queue"
12     ),
13     @ActivationConfigProperty(propertyName="acknowledgeMode", propertyValue="Auto-acknowledge"),
14     @ActivationConfigProperty(propertyName = "ConnectionFactoryJndiName", propertyValue = "jms/JMSMsgCF"),
15     @ActivationConfigProperty(propertyName = "DestinationName", propertyValue = "jms/JMSMsg"),
16     }
17 )
18 public class MDBTest implements MessageListener {
19
20     /**
21      * Default constructor.
22      */
23     public MDBTest() {
24         // TODO Auto-generated constructor stub
25     }
26
27     /**
28      * @see MessageListener#onMessage(Message)
29      */
30     public void onMessage(Message message) {
31         // TODO Auto-generated method stub
32         javax.jms.TextMessage txtMsg = (javax.jms.TextMessage)message;
33         try
34         {
35             System.out.println("MESSAGE -> " + txtMsg.getText());
36         }
37         catch (JMSEException e)
38         {
39             e.printStackTrace(System.out);
40         }
41     }
42 }
43
44
45
46 }
```

여기까지가 MDB를 위한 간단한 소스 작성 완료입니다. 쉽죠...^^& 그럼 해당 EAR 이름에서 마우스 우 클릭하여 내보내기 > EAR 파일을 선택합니다.



내보내기 마법사가 나오면 반출할 위치를 선택하고 완료를 클릭하여 해당 EAR 프로젝트를 ear 파일로 반출합니다.



Part 3. MDB 어플리케이션 배포

MDB 어플리케이션 작성이 완료되면 이를 WAS에 배포하여야 합니다. 지난 강의에도 설명드렸기 때문에 이번에도 역시 간단하게 기존 어플리케이션을 업데이트 하는 것으로서 해당 MDB 어플리케이션을 WAS 에 배포하도록 하겠습니다. 관리콘솔 메뉴의 Applications > WebSphere enterprise applications 메뉴로 가서 해당 어플리케이션을 선택하고 업데이트를 클릭합니다.

Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

Preferences

Start	Stop	Install	Uninstall	Update	Rollout Update	Remove File	Export	Export DDL	Export File
Select	Name							Application Status	
You can administer the following resources:									
<input type="checkbox"/>	DefaultApplication							➡	
<input type="checkbox"/>	PlantsByWebSphere							➡	
<input checked="" type="checkbox"/>	SampleWeb_war							➡	
<input type="checkbox"/>	SamplesGallery							➡	
<input type="checkbox"/>	StockTrade_war							➡	
<input type="checkbox"/>	ivtApp							➡	
<input type="checkbox"/>	query							➡	
Total 7									

어플리케이션 파일 선택 메뉴가 나오면 위의 단계에서 반출한 EAR 파일을 선택한 후 다음을 클릭합니다.

Specify the EAR, WAR, JAR, or SAR module to upload and install.

Application to be updated:
SampleWeb_war

Application update options

☒ Replace the entire application

Upload an enterprise archive (*.ear) to replace the entire installed application.

Specify the path to the replacement ear file.

☒ Local file system

Full path

D:\Wbiz_work\W01.SW\GW00\

찾아보기...

☐ Remote file system

Full path

Browse...

☐ Replace or add a single module

If the path to the new module matches an existing path to a module in the installed application, the new module replaces the existing module. If the path to the module does not exist in the installed application, the new module is added to the application.

☐ Replace or add a single file

If the path to the new file matches an existing path to a file in the installed application, the new file replaces the existing file. If the path to the file does not exist in the installed application, the new file is added to the application.

☐ Replace, add, or delete multiple files

Use a compressed file format such as .zip or .gzip. The compressed file is unzipped into the installed application directory. If the uploaded files exist in the application with the same paths and file names, the uploaded files replace the existing files. If the uploaded files do not exist, the files are added to the application. You can remove existing files from the installed application by specifying metadata in the compressed file.

Next

Cancel

빠른 진행을 위해 'Fast path' 를 선택하고 다음을 누릅니다.

Preparing for the application update

How do you want to install the application?

☒ Fast Path - Prompt only when additional information is required.

☐ Detailed - Show all installation options and parameters.

Specify bindings to use

merge new and existing bindings ▼

☒ Choose to generate default bindings and mappings

어플리케이션 설치 마법사의 각 옵션을 확인하고 다음을 클릭합니다.

Specify options for installing enterprise applications and modules.

→ Step 1: Select installation options

[Step 2: Map modules to servers](#)

★ [Step 3: Bind listeners for message-driven beans](#)

[Step 4: Summary](#)

Select installation options

Specify the various options that are available to prepare and install your application.

☐ Precompile JavaServer Pages files

Directory to install application
\$(APP_INSTALL_ROOT)/l

☒ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

☒ Create MBeans for resources

☐ Override class reloading settings for Web and EJB modules

Reload interval in seconds
[]

☐ Deploy Web services

Validate Input off/warn/fail
warn ▼

☐ Process embedded configuration

File Permission

Allow all files to be read but not written to
Allow executables to execute
Allow HTML and image files to be read by everyone

.*\.*.dll=755#.*\.*.so=755#.*\.*.a=755#.*\.*.sl=755

Application Build ID
Unknown

☐ Allow dispatching includes to remote resources

☐ Allow servicing includes from remote resources

Asynchronous Request Dispatch Type
Disabled ▼

☐ Allow EJB reference targets to resolve automatically

어플리케이션 서버 매핑을 확인하고 다음을 클릭합니다.

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

→ **Step 2: Map modules to servers**

Step 3 Bind listeners for message-driven beans

Step 4 Summary

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and servers:

Select	Module	URI	Server
<input type="checkbox"/>	SampleEJB.jar	SampleEJB.jar,META-INF/ejb-jar.xml	WebSphere:cell=kr050578Node01Cell,node=kr050578Node01,server=server1
<input type="checkbox"/>	SampleWeb	SampleWeb.war,WEB-INF/web.xml	WebSphere:cell=kr050578Node01Cell,node=kr050578Node01,server=server1

이번 단계가 이전 강좌와 조금 다른 면인데 MDB 어플리케이션이 나타나면 해당 MDB 어플리케이션과 연결할 Activation specification 을 입력해야 합니다. 위에서 설정한 JNDI 이름 (jms/JMSMsgAS) 을 입력하고 다음으로 넘어갑니다.

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

→ **Step 3: Bind listeners for message-driven beans**

Step 4 Summary

Bind listeners for message-driven beans

Each message-driven enterprise bean in your application or module must be bound to a listener port name or to an activation specification JNDI name. When a message-driven enterprise bean is bound to an activation specification JNDI name you can also specify the destination JNDI name and authentication alias.

☒ Apply Multiple Mappings

Select	EJB module	EJB	URI	Messaging type	Listener Bindings
<input type="checkbox"/>	SampleEJB.jar	MDBTest	SampleEJB.jar,META-INF/ejb-jar.xml	javax.jms.MessageListener	<input type="radio"/> Listener port Name <input type="text"/> <input checked="" type="radio"/> Activation Specification Target Resource JNDI Name <input type="text" value="jms/JMSMsgAS"/> Destination JNDI Name <input type="text"/> ActivationSpec authentication alias <input type="text"/>

마지막으로 요약정보를 확인하고 종료를 눌러 어플리케이션 배포 작업을 마무리 합니다.

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

Step 3 Bind listeners for message-driven beans

→ **Step 4: Summary**

Summary

Summary of installation options

Options	Values
Precompile JavaServer Pages files	No
Directory to install application	\$(APP_INSTALL_ROOT)/kr050578Node01Cell
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	No
Create MBeans for resources	Yes
Override class reloading settings for Web and EJB modules	No
Reload interval in seconds	
Deploy Web services	No
Validate input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\,dll=755#.*\,so=755#.*\,a=755#.*\,sl=755
Application Build ID	Unknown
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No
Asynchronous Request Dispatch Type	Disabled
Allow EJB reference targets to resolve automatically	No
Application name	SampleWeb_war
Cell/Node/Server	Click here

Part 4. MDB 어플리케이션 테스트

MDB 어플리케이션 배포가 정상적으로 완료되었으면 이전 강좌에서 만들고 테스트 했던 JMS client 어플리케이션을 수행하여 해당 Queue 에 Message 를 하나 전달합니다.

```
D:\wbiz_work\01.SWC\00.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
MSCL0012I: 명령행 인수 처리 중.
MSCL0013I: Java EE 응용 프로그램 클라이언트 환경 초기화 중
[09. 6. 29 17:14:01:796 KST] 00000000 W UOW=null source=com.ibm.ws.ssl.config
.SSLConfig org=IBM prod=WebSphere component=Application Server thread=[P=240890:
0=0:CT]
CWPKI0041W: 하나 이상의 키 스토어에서 기본 암호를 사용하고 있습니다.
MSCL0035I: Java EE Application Client Environment의 초기화가 완료되었습니다.
MSCL0014I: 응용 프로그램 클라이언트 class Main 호출 중
Send completed
D:\wbiz_work\01.SWC\00.WAS\WAS_Sample\WEAR>
```

'Send completed' 메시지를 확인하면 정상적으로 Message 를 Queue에 전달한 것입니다. 그럼 WAS_ROOT\profiles\WAppSrv01\logs\server1\SystemOut.log 파일을 Text 편집기로 열어서 Log 를 확인합니다.



```
1951 [09. 6. 29 17:10:09:062 KST] 0000001d servlet I com.ibm.ws.webcontainer.servlet.ServletWrapper init SRVE02
1952 [09. 6. 29 17:10:11:015 KST] 0000001d servlet I com.ibm.ws.webcontainer.servlet.ServletWrapper init SRVE02
1953 [09. 6. 29 17:10:14:484 KST] 0000001d servlet I com.ibm.ws.webcontainer.servlet.ServletWrapper init SRVE02
1954 [09. 6. 29 17:10:14:516 KST] 0000001d servlet I com.ibm.ws.webcontainer.servlet.ServletWrapper init SRVE02
1955 [09. 6. 29 17:14:11:859 KST] 00000030 SystemOut O MESSAGE -> Package Received
1956 [207]
```

위의 화면처럼 'MESSAGE -> Package Received' 라는 출력이 있으면 정상적으로 JMS client 가 보낸 Message 를 Queue 에 보냈고 그 보내어진 Message 를 MDB 가 읽어들이어서 Text 를 출력한 것입니다. Buses> SIB > 해당 SIB > Destinations > 해당 Queue > queue points 를 선택해서 Queue depth 를 확인해 봅니다.

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	0
Total 1	

이전 강좌와 비교해 보면 아시겠지만 이전 강좌와는 다르게 Queue depth 의 숫자가 그대로 0 인 것을 확인하실 수 있습니다. 이전에 말씀드린 것처럼 JMS client 에서 보내진 Message 가 Queue 에 들어간 후 Event 가 발생해서 자동으로 해당 Queue 를 리스닝하고 있던 MDB 가 해당 Message 를 빼내어 읽어서 Output 을 출력했기 때문에 해당 Queue depth 가 그대로 0 으로 표시되는 것입니다. 여기까지 따라오셨으면 이번 MDB 강좌도 무사히 완료하신 것입니다. 벌써 강좌의 끝을 향해 달려가고 있네요. 그럼 조금만 더 노력하시길 바라며 이제 다음 강좌로. 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.multiplatform.doc/info/welcome_nd.html

참고 2) InforCenter – Message-driven beans - automatic message retrieval

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/cmb_ovrww.html

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