

하나씩 쉽게 따라 해보는 IBM WebSphere Application Server(WAS) v7 – 21

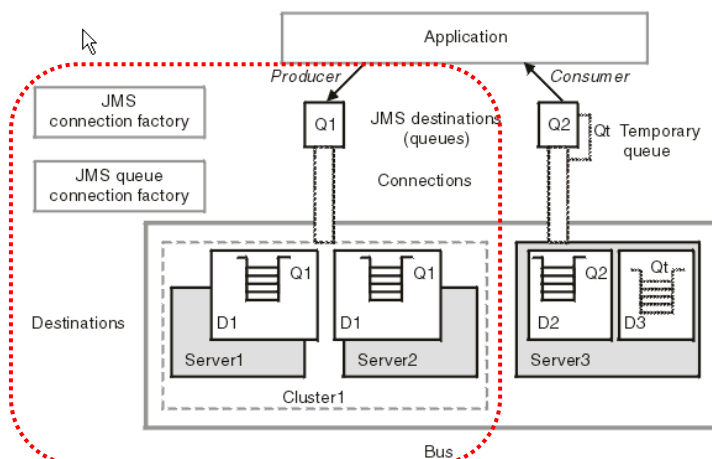
이정운 (juwlee@kr.ibm.com)

하나씩 쉽게 따라 해보는 IBM WAS v7 드디어 그 스물 한번째 이야기를 시작합니다. 스무번째 강좌까지 2부를 마무리하고 3부를 생각중이었는데, 아시는 분을 통해서 IBM WAS v7 의 JMS Clustering 에 대해 강좌를 해주면 좋을 것 같다는 의견을 받았습니다. 그래서 고민하다가 3부로 넘어가는 중간에 add-on 형식의 추가판 처럼, 스물 한번째 강좌를 JMS Clustering 이라는 주제로 시작하도록 하겠습니다.

이미 7강에서 강좌를 진행했지만 WAS 는 기본적으로 JMS 서비스를 제공하도록 되어 있습니다. 그런데 이 JMS 서비스가 점점 발전되면서 마치 MOM(메시지 지향 미들웨어) 제품인 MQ 처럼 clustering 이 가능해 졌습니다. 기존 7강에서는 단순히 하나의 Message Engine 에서만 Queue 를 받고 처리하는 예제를 보여드렸습니다. 이런 구성의 경우, 평상시에는 문제가 없으나 해당 Message Engine 의 장애가 발생하거나 해당 서버에 장애가 발생하게 되면 서비스가 안된다는 단점이 있습니다. 그러나 Message Engine 을 clustering 하게 되면 이런 장애상황에서도 다른쪽에 미리 clustering 해둔 Message Engine 을 통해서 JMS 서비스를 지속적으로 운영할 수 있습니다.

기존 IBM WAS v6.1 에서도 JMS Clustering 이 되었지만, Active-active 모드가 아닌 Active-standby 모드만 지원이 가능했었습니다. 즉, 이전 버전에서는 Active 상태인 Message Engine 이 장애가 난 경우에만 바로 standby 모드인 Message Engine 이 active 상태를 넘겨받아 JMS 서비스를 유지 할 수 있었던 것이었죠. (그러다 보니 standby 상태에서 active 상태로 절체되기 위한 시간이 별도로 필요했습니다.) 그러나 IBM WAS v7 부터는 JMS Clustering 도 MQ 처럼 Active-active 모드로 유지할 수 있으므로 그런 시간 자체가 없어지며, 보다 중요한 것은 동시에 여러개의 Message engine 이 구동되므로 보다 빠른 Message 처리 성능을 얻을 수 있는 길이 생긴 것입니다.

그럼 이제 서론은 각설하고 실제 강좌를 진행하면서 직접 눈으로 확인해보시죠.



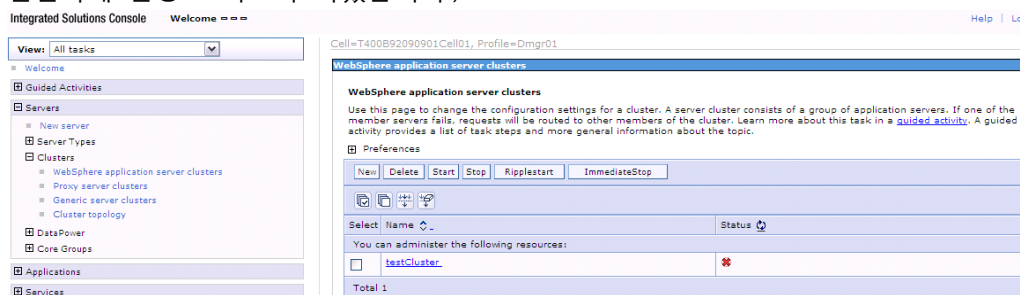
참조 : IBM WAS v7 Information Center

Part 1. JMS Clustering 설정

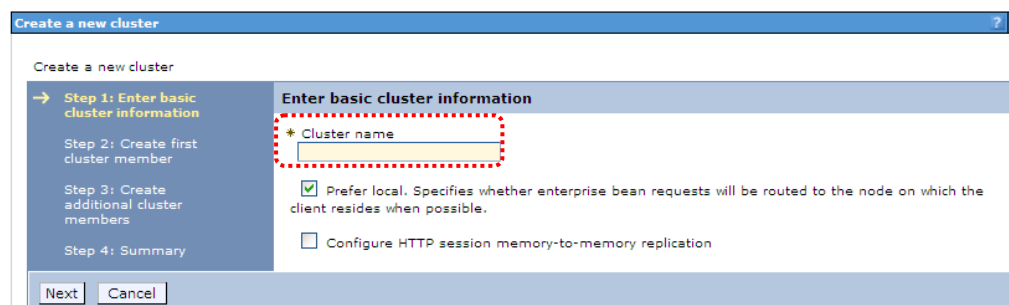
JMS Clustering 을 사용하기 위해서는 당연히게도 IBM WAS 관리콘솔을 이용해서 해당 설정을 해야합니다. 7장에서 논의 된 것처럼 Messaging engine 을 구성하고 SIB, Connection factory, queue 등을 설정해야 합니다. 가장 먼저 Messaging Engine 이 실제로 거주할 Server Cluster 를 정의하도록 하겠습니다.

(Server Cluster 를 정의하고 그 안에 Messaging Engine 이 각 WAS Server 에 존재하는 구성을 통하여 JMS Clustering 샘플을 보여드리도록 하겠습니다.)

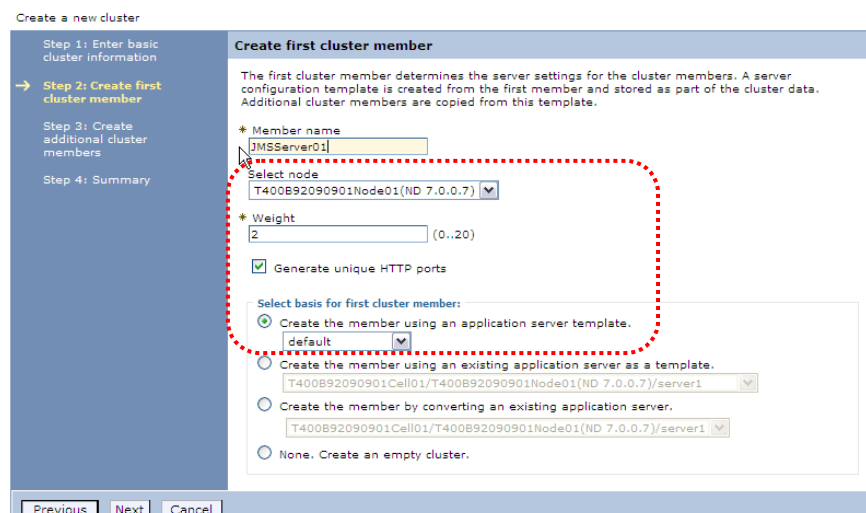
관리콘솔에서 Servers > WebSphere application server clusters 메뉴를 클릭하여 cluster 메뉴를 확인한 후 server cluster 작성을 위하여 new 버튼을 누릅니다.(이전 강좌에서 이미 다룬 내용이므로 간단하게 설명 드리도록 하겠습니다.)



사용할 Server cluster 의 이름을 입력하고 다음을 클릭합니다.



Server Cluster 의 멤버로 사용될 WAS 서버 이름을 입력하고 다음을 클릭합니다.



단순 single mode 의 Server 가 아니라, Server Cluster 이기 때문에 추가적으로 하나의 server 멤버를 더 추가하고(add member 클릭) 다음으로 넘어갑니다.

Create a new cluster

Step 1: Enter basic cluster information

Step 2: Create first cluster member

→ Step 3: Create additional cluster members

Step 4: Summary

Create additional cluster members

Enter information about this new cluster member, and click Add Member to add this cluster member to the member list. A server configuration template is created from the first member, and stored as part of the cluster data. Additional cluster members are copied from this template.

* Member name

Select node
T400B92090901Node01(ND 7.0.0.7) ▼

* Weight
 (0..20)

☒ Generate unique HTTP ports

Add Member

Use the Edit function to edit the properties of a cluster member that is already included in this list. Use the Delete function to remove a cluster member from this list. You are not allowed to edit or remove the first cluster member or an already existing cluster member.

Edit Delete

Select	Member name	Nodes	Version	Weight
<input type="checkbox"/>	JMSServer01	T400B92090901Node01	ND 7.0.0.7	2
<input type="checkbox"/>	JMSServer02	T400B92090901Node01	ND 7.0.0.7	2

Total 2

Previous Next Cancel

마지막으로 요약정보를 확인하고 완료를 클릭하면 Messaging engine cluster 가 거주하게 될 Server cluster 구성을 완료하신 것 입니다.

Create a new cluster

Step 1: Enter basic cluster information

Step 2: Create first cluster member

Step 3: Create additional cluster members

→ Step 4: Summary

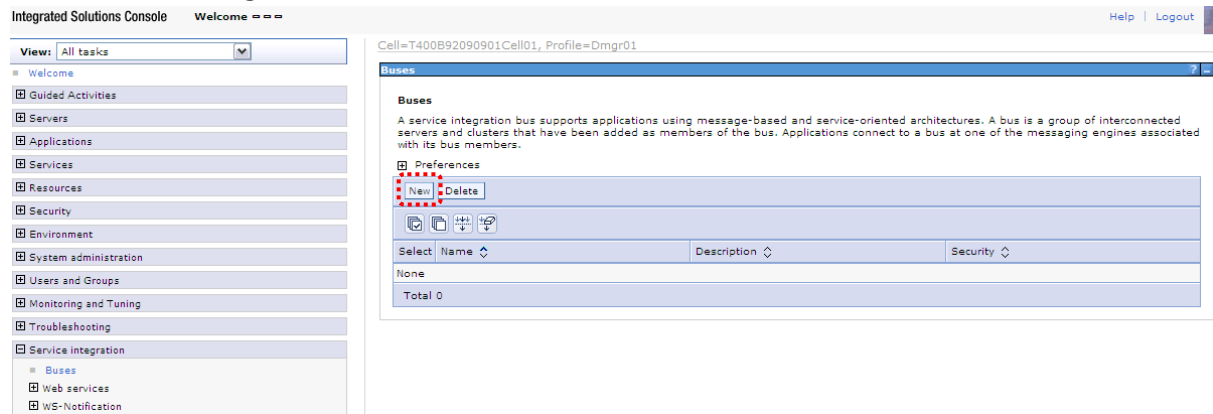
Summary

Summary of actions:

Options	Values
Cluster Name	JMSCluster01
Core Group	DefaultCoreGroup
Node group	DefaultNodeGroup
Prefer local	true
Configure HTTP session memory-to-memory replication	false
Server name	JMSServer01
Node	T400B92090901Node01(ND 7.0.0.7)
Weight	2
Clone Template	default
Clone Basis	Create the member using an application server template.
Generate unique HTTP ports	true
Server name	JMSServer02
Node	T400B92090901Node01(ND 7.0.0.7)
Weight	2
Clone Template	Version 7 member template
Generate unique HTTP ports	true

Previous Finish Cancel

Server cluster 작성을 완료하였으면 이제 Message 전송을 위한 Bus 를 만들어야 합니다. 관리콘솔에서 Service integrations > Buses 메뉴를 클릭하여 Bus 메뉴로 들어와 New 를 클릭합니다.



사용할 Bus 의 이름을 지정하고 다음을 클릭합니다.

Create a new Service Integration Bus.

→ **Step 1: Create a new bus**

(The next step of the wizard depends on decisions made in the current step)

Step 2: Confirm create of new bus

Create a new bus

Configure the attributes of your new bus.

* Enter the name for your new bus.

☐ Bus security

Next Cancel

요약정보를 확인하고 완료를 누르면 Bus 생성을 완료하였습니다. 간단하죠^^&

Create a new Service Integration Bus.

Step 1: Create a new bus

→ **Step 2: Confirm create of new bus**

Confirm create of new bus

The following is a summary of your selections. To complete the bus creation, click Finish. If there are settings you wish to change, click Previous to review bus settings.

Summary of actions:

New bus "SIB01" will be created with bus security setting "Disabled".

Previous Finish Cancel

Bus 생성이 완료되었으면, 이름을 클릭하여 Bus 설정 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

New Delete		
Select	Name	Description
You can administer the following resources:		
<input type="checkbox"/>	SIB01	Disabled
Total 1		

Bus 설정 세부메뉴에서 Bus 의 실제 member 를 설정하기 위하여 Bus members 를 클릭합니다.

[Buses](#) > [SIB01](#)

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	Topology
Name <input type="text" value="SIB01"/>	Bus members
UUID <input type="text" value="DA37153ADCF1A942"/>	Messaging engines
Description <input type="text"/>	Foreign bus connections
Inter-engine transport chain <input type="text"/>	Bootstrap members
<input type="checkbox"/> Discard messages	Destination resources
<input checked="" type="checkbox"/> Configuration reload enabled	Destinations
Default messaging engine high message threshold <input type="text" value="50000"/> messages	Mediations
Limit the range of available bootstrap members to: <input checked="" type="radio"/> All members of the cell with the Service Integration Bus Service enabled	Services
	Inbound services
	Outbound services
	WS-Notification services
	Reliable messaging state
	Additional Properties
	Custom properties
	Security
	Web service gateway instances

하단과 같이 Bus members 메뉴가 나오면 Add 를 클릭합니다.

[Buses](#) > [SIB01](#) > [Bus members](#)

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

Preferences

Add Remove		
Select	Name	Type
None		
Total 0		

추가할 member 로 기존에 만들었던 Cluster 를 선택하고 다음을 클릭합니다.

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**

(The next step of the wizard depends on decisions made in the current step)

Step 2: Summary

Select server, cluster or WebSphere MQ server

Choose the server, cluster or WebSphere MQ server to add to the bus

☐ Server

☒ Cluster

☐ WebSphere MQ server

그러면, 하단과 같은 policy 설정 메뉴를 확인할 수 있습니다. Policy 설정이란, 어떤 식으로 Messaging engine 을 구성할지 정책을 정하는 것입니다. 이전 버전인 WAS v6.1 과 동일하게 High availability 만을 위한 구축이 가능하며(active-standby 모드), active-active 모드로 구성되는 Scalability 모드, 두 모드를 섞어서 사용하는 모드등 원하시는 모드를 선택하면 구성도를 바로 확인할 수 있습니다. (각 모드를 선택하면 바로 구성도가 보이므로 각각의 구성을 공부차원으로 한번 확인해 보시기 바라며, 여기서는 이미 말씀드린대로 active-active 모드로 구성되는 scalability 모드를 선택하도록 하겠습니다.)

Select a predefined messaging engine policy to apply to the selected cluster when it is added as a bus member.

Step 1: Select server, cluster or WebSphere MQ server

→ **Step 1.1: Messaging engine policy assistance settings**

(The next step of the wizard depends on decisions made in the current step)

Step 2: Summary

Messaging engine policy assistance settings

Enabling messaging engine policy assistance enables a predefined or custom policy to be applied to the selected server cluster. Tooling will be enabled to assist in maintaining the policy if the server cluster changes in size. Restrictions will be placed on the changes that can be made to associated core group policies.

☒ Enable messaging engine policy assistance?

Select	Policy type	Is further configuration required?
<input type="radio"/>	High availability	⚠ The current configuration has a single point of failure because there is only a single node. Consider adding a cluster member configured on a separate node.
<input checked="" type="radio"/>	Scalability	No
<input type="radio"/>	Scalability with high availability	⚠ The current configuration has a single point of failure because there is only a single node. Consider adding a cluster member configured on a separate node.
<input type="radio"/>	Custom	Advice is not available for a custom configuration.

다음으로 Message 저장소로 사용할 타입을 정하는 메뉴가 나오는데 테스트의 편의성을 위하여 File 저장소를 선택하고 다음을 클릭합니다.

Choose the type of message store for the persistence of message state

Step 1: Select server, cluster or WebSphere MQ server

Step 1.1: Messaging engine policy assistance settings

→ **Step 1.1.1: Select the type of message store**

(The next step of the wizard depends on decisions made in the current step)

Step 2: Summary

Select the type of message store

Choose the type of message store for the persistence of message state

☒ File store

☐ Data store

Previous Next Cancel

그러면, 이전 7강하고는 다르게 하단처럼 두 개의 Messaging engine 을 확인할 수 있습니다. 여기서는 바로 다음을 클릭하면 안되고 Messaging engine 의 이름을 클릭하여 세부 설정 메뉴로 들어갑니다.

Configure the messaging engines that will be created when the server cluster is added as a bus member.

Step 1: Select server, cluster or WebSphere MQ server

Step 1.1: Messaging engine policy assistance settings

→ **Step 1.1.2: Configure messaging engines**

(The next step of the wizard depends on decisions made in the current step)

Step 2: Summary

Configure messaging engines

The collection table shows the messaging engines that will be created when the server cluster is added as a bus member. At least one messaging engine must be created and message store settings must be configured for each messaging engine.

Name	Failover?	Fail back?	Preferred order of servers to run on	Only run on preferred servers?	Is the message store configured?
JMSSCluster01.000-SIB01	No	No	T400B92090901Node01:JMSServer01	Yes	No
JMSSCluster01.001-SIB01	No	No	T400B92090901Node01:JMSServer02	Yes	No

Previous Next Cancel

세부 설정에서 file 저장소의 위치와 사이즈 등을 설정하고 다음을 클릭합니다.

Provide the properties for the file store

Step 1: Select server, cluster or WebSphere MQ server

Step 1.1: Messaging engine policy assistance settings

Step 1.1.1: Select the type of message store

Step 1.1.2: Configure messaging engines

→ **Step 1.1.2.1: Configure file store**

Step 2: Summary

Configure file store

Specify the properties for the file store

Log

* Log size
10 MB

* Log directory path
server\profiles\AppSrv01\logs

Store

☒ Same settings for permanent and temporary stores

Permanent and temporary stores

* Minimum permanent store size
20 MB

☐ Unlimited permanent store size

* Maximum permanent store size
50 MB

* Permanent store directory path
server\profiles\AppSrv01\logs

Previous

Next

Cancel

그러면 하단의 그림처럼 테이블의 마지막에 Message 저장소 설정이 되었냐는 질문에 답이 No 에서 Yes 로 변경된 것을 확인할 수 있습니다. 나머지 Messaging engine 의 이름을 클릭하여 동일한 작업을 반복하고 다음을 클릭합니다.

Configure the messaging engines that will be created when the server cluster is added as a bus member.

Step 1: Select server, cluster or WebSphere MQ server

Step 1.1: Messaging engine policy assistance settings

Step 1.1.1: Select the type of message store

→ **Step 1.1.2: Configure messaging engines**

(The next step of the wizard depends on decisions made in the current step)

Step 2: Summary

Configure messaging engines

The collection table shows the messaging engines that will be created when the server cluster is added as a bus member. At least one messaging engine must be created and message store settings must be configured for each messaging engine.

Name	Failover?	Fail back?	Preferred order of servers to run on	Only run on preferred servers?	Is the message store configured?
JMSSCluster01.000-SIB01	No	No	T400B92090901Node01:JMSServer01	Yes	Yes
JMSSCluster01.001-SIB01	No	No	T400B92090901Node01:JMSServer02	Yes	No

Previous

Next

Cancel

다음으로 성능을 위해서 IBM 에서 가이드 된 수치대로 heap 사이즈를 변경할 것이라고 물어보는 메뉴가 나오는데 기본 테스트만 진행할 것이므로 변경 없이 바로 다음을 클릭합니다.

Tune application server for messaging performance.

Step 1: Select server, cluster or WebSphere MQ server

Step 1.1: Messaging engine policy assistance settings

Step 1.1.1: Select the type of message store

Step 1.1.2: Configure messaging engines

→ **Step 1.1.2.1: 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 maybe 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 요약정보를 확인하고 완료를 클릭합니다.

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: Messaging engine policy assistance settings

Step 1.1.1: Select the type of message store

Step 1.1.2: Configure messaging engines

→ **Step 2: Summary**

Summary

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

Adding server cluster "JMSSCluster01" as member of bus "SIB01".

The cluster is being configured using messaging engine assistance for policy: "Scalability".

The following messaging engines have been configured:

- Messaging engine name "JMSSCluster01.000-SIB01"
- The messaging engine should failover to other servers in the server cluster. "false"

Previous Finish Cancel

위의 과정을 잘 따라오셨다면, 하단의 그림처럼 Server Cluster 멤버 내에 두 개의 message engine 이 생성된 것을 확인할 수 있습니다.

[Buses](#) > [SIB01](#) > [Messaging engines](#)

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

Preferences

Start Stop

Select Name Description Status

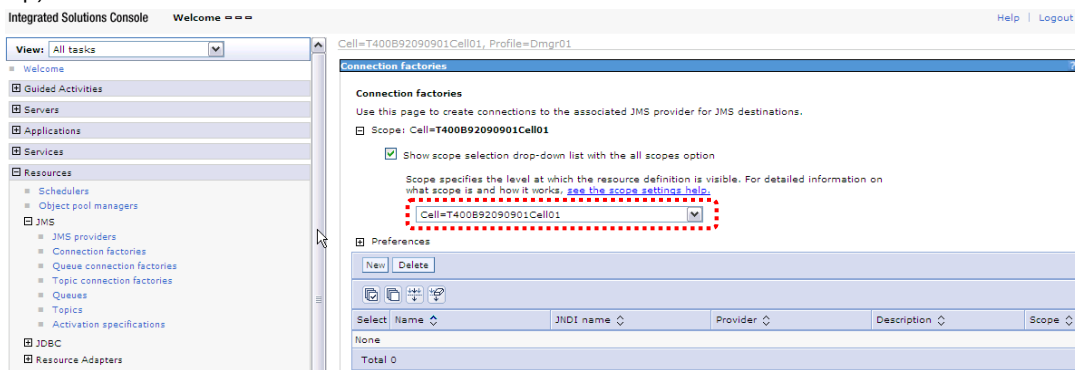
You can administer the following resources:

<input checked="" type="checkbox"/>	JMSSCluster01.000-SIB01		⊗
<input checked="" type="checkbox"/>	JMSSCluster01.001-SIB01		⊗

Total 2

지금까지는 Server cluster 생성, Bus 생성 및 member 등록을 통한 Message engine 설정을 완료했습니다. 이제는 실제 Queue 사용을 위한 Connection factory 와 Queue 설정을 진행하도록 하겠습니다.

관리콘솔에서 Resources > JMS > Connection factories 메뉴를 클릭하고 범위를 Cell 로 조정한 후에 New 를 클릭합니다. (여기서는 편의상 Cell 로 했지만 원하시는 범위로 충분히 제한 가능합니다.)



JMS resource provider 를 선택하는 메뉴에서는 WAS 에서 기본으로 제공하는 Default message provider 를 선택합니다.

[Connection factories](#) > [Select JMS resource provider](#)

Scope:

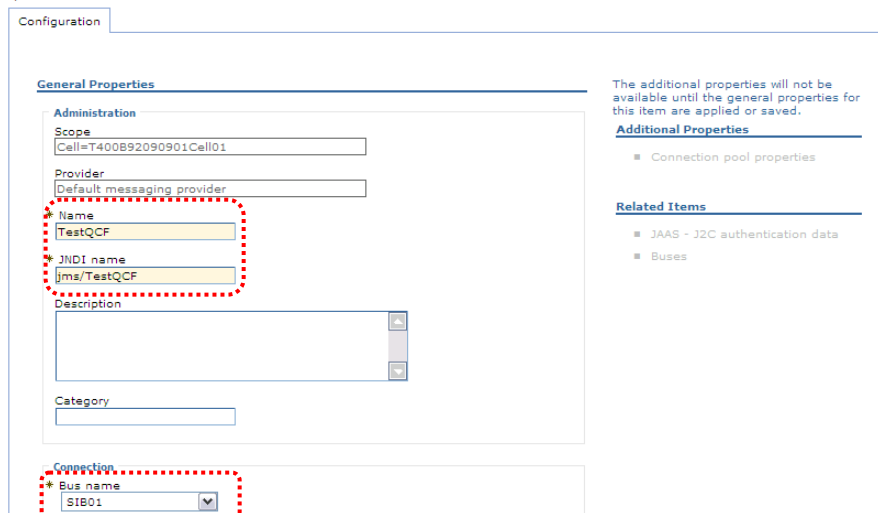
Select the provider with which to create the Connection factory. The following providers support the selected resource type and are available at the selected scope.



Connection factory 의 이름과 JNDI 이름을 입력하고 미리 만들어둔 Bus 이름을 설정합니다.

[Connection factories](#) > [Default messaging provider](#) > [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.



여기서 추가적으로 provider endpoints 정보를 하단처럼 입력해야 합니다. provider endpoints 정보는 실제 Messaging engine 과 연결하기 위한 실제 정보이므로 다음 예제처럼 두 대의 Server cluster 의 정보를 각각 하단처럼 입력하면 됩니다. (서버가 거주하는 호스트이름:port이름:chain이름)

이때, 사용되는 port 는 SIB_ENDPOINT_ADDRESS 포트를 입력하면 됩니다.

<input type="checkbox"/>	SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	T400B92090901.kr.ibm.com	9416	No associated transports
<input type="checkbox"/>	SIB_ENDPOINT_ADDRESS	*	7281	View associated transports
<input type="checkbox"/>	SIB_ENDPOINT_SECURE_ADDRESS	*	7290	View associated transports
<input type="checkbox"/>	SIB_MQ_ENDPOINT_ADDRESS	*	5562	View associated transports

여기까지 잘 따라오셨다면 Connection factory 설정이 완료된 것입니다. 그리고 이제 Connection factory 에서 실제 연결될 Queue 를 설정해보도록 하겠습니다. 관리콘솔에서 Resources > JMS > Queue 메뉴를 클릭한 후 범위를 조정하고 New 버튼을 클릭합니다.

Queues

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

Scope: Cell=T400B92090901Cell01

☒ Show scope selection drop-down list with the all scopes option

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=T400B92090901Cell01

Preferences

New Delete					
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>					
Select	Name	JNDI name	Provider	Description	Scope
None					
Total 0					

JMS resource provider 타입을 선택하는 메뉴에서는 역시 이전과 동일하게 Default messaging provider 를 선택합니다.

[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

OK Cancel

Connection factory 와 마찬가지로 이름과 JNDI 이름, Bus 이름을 지정하고 Queue name 을 지정하는 콤보박스를 클릭하여 하단처럼 새로운 Queue 를 생성할 수 있는 Create Service Integration Bus destination 메뉴를 선택합니다.

Configuration

General Properties

Administration

Scope

Provider

* Name

* JNDI name

Description

Connection

Bus name

* Queue name

Select...
_SYSTEM.Exception.Destination.JMSCluster01.000-SIB01
_SYSTEM.Exception.Destination.JMSCluster01.001-SIB01
Create Service Integration Bus destination
other, please specify

Related Items

■ Buses

그러면, 하단처럼 Queue 를 생성하는 마법사로 바로 넘어가는데 여기서는 사용하고자 하는 Queue 의 이름을 입력하고 다음을 선택하면 됩니다.

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

Description

Next Cancel

해당 Queue 에 할당되는 Bus 멤버를 하단처럼 Bus 에서 추가했던 Cluster 멤버로 선택합니다.

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

The screenshot shows the 'Assign the queue to a bus member' step of the JMS Queue creation wizard. On the left, a sidebar lists three steps: 'Step 1: Set queue attributes', 'Step 2: Assign the queue to a bus member' (which is highlighted with a yellow arrow), and 'Step 3: Confirm queue creation'. The main area has a title bar 'Assign the queue to a bus member' and a description: 'Assign the queue to a bus member that will store and process the messages for the queue.' Below this is a 'Bus member' dropdown menu with 'Cluster=JMSCluster01' selected. At the bottom are 'Previous', 'Next', and 'Cancel' buttons.

요약화면을 확인하고 완료를 클릭합니다.

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

The screenshot shows the 'Confirm queue creation' step of the JMS Queue creation wizard. The sidebar on the left highlights 'Step 3: Confirm queue creation' with a yellow arrow. The main area has a title bar 'Confirm queue creation' and a description: 'To complete creation of the queue, click Finish. If you want to change any selections, click Previous.' Below this is a 'Summary of actions:' box containing two lines of text: 'New queue "ClusterQ" will be created.' and 'Queue points for "ClusterQ" will be created for bus member "JMSCluster01" of bus "SIB01".' At the bottom are 'Previous', 'Finish', and 'Cancel' buttons.

위의 작업을 진행하면 Queue name 을 선택하는 콤보박스에서 이제 방금 만들었던 Queue 를 선택할 수 있습니다. 해당 Queue 를 선택하고 설정을 저장합니다.

[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.

The screenshot shows the JMS Queue configuration page. At the top is a 'Configuration' tab. Below it is a 'General Properties' section with two sub-sections: 'Administration' and 'Connection'. The 'Administration' section contains fields for 'Scope' (Cell=T400B92090901Cell01), 'Provider' (Default messaging provider), 'Name' (TestQ), 'JNDI name' (jms/TestQ), and a 'Description' text area. The 'Connection' section contains a 'Bus name' dropdown (SIB01), a 'Queue name' dropdown (ClusterQ) which is highlighted with a red dashed box, and a 'Delivery mode' dropdown (Application). On the right side, there is a 'Related Items' section with a link to 'Buses'.

이렇게 하시면 다음처럼 Queue 도 정상적으로 잘 만들어진 것을 확인할 수 있습니다.

Queues

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

☐ Scope: Cell=T400B92090901Cell01

☒ Show scope selection drop-down list with the all scopes option

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=T400B92090901Cell01 ▼

Preferences

New Delete					
Select	Name	JNDI name	Provider	Description	Scope
You can administer the following resources:					
<input type="checkbox"/>	TestQ	jms/TestQ	Default messaging provider		Cell=T400B92090901Cell01
Total 1					

여기까지가 JMS Clustering 을 위한 기본설정을 모두 완료하신 것입니다. Policy 부분만을 제외하고는 7강과 거의 동일한 부분이 많으므로 여기서 잘 이해가 되지 않으신다면 7강을 한번 더 읽어 보시길 권고드립니다.

Part 2. JMS Clustering 테스트

JMS Clustering 을 테스트 하기 위해서는 Messaging engine 이 구동되는 WAS Server 를 시작해서 Messaging engine 을 구동해야 합니다. 하단의 그림처럼 Clustering 되어 있는 WAS Server 를 시작시켜주면 바로 Clustering 이 된 Messaging engine 이 시작된 것을 확인할 수 있습니다.

Application servers

Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use it to change the status of a specific application server.

Preferences

New Delete Templates... Start Stop Restart ImmediateStop Terminate						
Select	Name	Node	Host Name	Version	Cluster Name	Status
You can administer the following resources:						
<input type="checkbox"/>	JMSServer01	T400B92090901Node01	T400B92090901.kr.ibm.com	ND 7.0.0.7	JMSCluster01	
<input type="checkbox"/>	JMSServer02	T400B92090901Node01	T400B92090901.kr.ibm.com	ND 7.0.0.7	JMSCluster01	

Buses > SIB01 > Messaging engines

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

Preferences

Start Stop			
Select	Name	Description	Status
You can administer the following resources:			
<input type="checkbox"/>	JMSCluster01.000-SIB01		
<input type="checkbox"/>	JMSCluster01.001-SIB01		
Total 2			

그리고 중요한 것 중의 하나는 실제로 Message 를 보낼 client 소스 작성입니다. 이는 7강에서 사용한 소스를 그대로 이용하도록 하겠습니다. (단, Queue Connection Factory 와 Queue 의 JNDI 이름은 당연히 변경되어야 하겠지요.^^&);

```
public class Main
{
    private static final String JMSCF_JNDI_NAME = "cell/clusters/JMSCluster01/jms/TestQCF";
    private static final String JMSQ_JNDI_NAME = "cell/clusters/JMSCluster01/jms/TestQ";
    private static final String messageText = "Package Received";

    public static void main(String args[])
        throws Exception
    {
        Hashtable table = new Hashtable();
        table.put("java.naming.factory.initial", "com.ibm.websphere.naming.WsnInitialContextFactory");
        table.put("java.naming.provider.url", "iiop://was-il:9809");
        Context initCtx = new InitialContext(table);
        ConnectionFactory qcf = (ConnectionFactory)initCtx.lookup(JMSCF_JNDI_NAME);
        Destination q = (Destination)initCtx.lookup(JMSQ_JNDI_NAME);
        Connection connection = qcf.createConnection();
        Session session = connection.createSession(false, 1);
        MessageProducer queueSender = session.createProducer(q);
        TextMessage outMessage = session.createTextMessage();
        outMessage.setText(messageText);
        outMessage.setJMSType("package_received");
        outMessage.setJMSDestination(q);
        queueSender.setDeliveryMode(1);
        queueSender.send(outMessage);
        connection.close();
        System.out.println("Send completed");
    }
}
```

Client 소스 작성을 완료하였으면 간편하게 실행해서 테스트하기 위하여 WAS 의 bin 디렉토리에 있는 launchClient.bat 을 이용해서 바로 클라이언트를 실행합니다. 실행이 완료되어 Message 를 JMS Cluster 에 정확하게 send 하면 하단처럼 Send completed 를 확인할 수 있습니다.

```
C:\IBM\WebSphere\AppServer\profiles\Dmgr01\bin>launchClient.bat JMSTestEAR.ear
IBM WebSphere Application Server, 릴리스 7.0
Java EE Application Client Tool
Copyright IBM Corp., 1997-2008
WSCL00121: 명령행 인수 처리 중.
WSCL00131: Java EE 응용프로그램 클라이언트 환경 초기화 중
[10. 3. 26 16:48:42:093 KST] 00000000 W UOW=null source=com.ibm.ws.ssl.config.SSLConfig org=IBM
rod=WebSphere component=Application Server thread=[P=721859:0=0:CT]
CWPk10041W: 하나 이상의 키 스토어에서 기본 암호를 사용하고 있습니다.
[10. 3. 26 16:48:43:156 KST] 00000000 W UOW=null source=com.ibm.ws.sib.utils.ras.SibMessage org=
BM prod=WebSphere component=Application Server thread=[P=721859:0=0:CT]
[!] CWSJY0004W: 클래스 경로에서 wmq.client.props 특정 파일을 찾을 수 없습니다.
WSCL00351: Java EE Application Client Environment의 초기화가 완료되었습니다.
WSCL00141: 응용프로그램 클라이언트 class Main 호출 중
Send completed
```

Message 가 정상적으로 전송되었다면, 관리콘솔에서 Buses > 해당 버스 > Messaging engines > 해당 Messaging engine > Queue points 를 클릭하시면 하단처럼 Queue depth 가 0 에서 1 로 증가된 것을 확인할 수 있습니다. (해당 Messaging engine 이라는 말은 2 개로 Cluster 된 Messaging engine 중의 하나를 의미합니다.)

Buses > **SIB01** > Messaging engines > **JMSCluster01.000-SIB01** > Queue points

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

Preferences

Identifier	Queue Depth
You can administer the following resources:	
ClusterQ@JMSCluster01.000-SIB01	1
SYSTEM.Exception.Destination.JMSCluster01.000-SIB01@JMSCluster01.000-SIB01	0
Total 2	

다시 한번 Message 를 send 하는 client 를 수행하고 동일한 Messaging engine 의 queue point 를 보면 위에서 처럼 변화가 없고, Clustering 된 다른 Messaging engine 을 확인하시면 하단과 같이 이번에는 다른 쪽 Messaging engine 의 queue 에 Message 가 들어온 것을 확인할 수 있습니다.

Buses > **SIB01** > Messaging engines > **JMSCluster01.001-SIB01** > Queue points

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

Preferences

Identifier	Queue Depth
You can administer the following resources:	
ClusterQ@JMSCluster01.001-SIB01	1
SYSTEM.Exception.Destination.JMSCluster01.001-SIB01@JMSCluster01.001-SIB01	0
Total 2	

그것은 이미 언급 드린 것처럼, 이전과 같이 High availability 구성을 사용하지 않고(Active-standby) Scalability 구성(active-active 모드) 을 사용하였기 때문에 Cluster 된 두 개의 Messaging engine 을 모두 다 사용할 수 있기 때문입니다. 결국, Cluster 가 가지고 있는 두 개의 Queue(하지만, 동일한 이름으로 접근이 가능한 Queue) 에 다 넣을 수 있는 것입니다. 이렇듯, IBM WAS v7 에서는 이전 버전과는 다르게 여러가지 다양한 모드의 JMS Clustering 을 지원하므로 필요한 환경에 맞추어 적합한 것을 선택적으로 사용할 수 있을 듯 합니다.

여기까지 잘 따라오셨나요? 약간 어려운 내용일 수도 있지만 지금까지 잘 따라왔다면 이전 강좌들의 내용까지 엮어서 쉽게 이해할 수 있는 부분도 많을 것이라고 생각됩니다. 좀더 쉽게 생각해 보면 기업환경의 시스템에서 Clustering 이 중요해 지면서 JMS 도 Server cluster 처럼 Clustering 된다 라고 생각하시면 됩니다. 또한, 강좌를 진행하면서 느끼셨겠지만 IBM WAS 의 경우는 점점 발전하면서 마법사형식으로 step-by-step 으로 쉽게 설정을 할 수 있는 부분들을 점점 더 많이 제공하므로 설정의 편리함, 역시 많이 느낄 수 있을 듯 합니다. 그럼 여기서 add-on 형식으로 진행되었던 이번 강좌를 마무리 하도록 하겠습니다. 이만~~~~~휘리릭... ^^&

참고 1) IBM WebSphere Application Server v7.0 InfoCenter

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

참고 2) IBM WebSphere Application Server v7.0 InfoCenter

- Messaging engine policy assistance

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.pmc.nd.multipatform.doc/concepts/cjt1005_.html?resultof=%22%6d%65%73%73%61%67%69%6e%67%22%20%22%6d%65%73%73%61%67%22%20%22%65%6e%67%69%6e%65%22%20%22%65%6e%67%69%6e%22%20%22%70%6f%6c%69%63%79%22%20%22%70%6f%6c%69%63%69%22%20

참고 3) IBM WebSphere Application Server v7.0 InfoCenter

- Scalability messaging engine policy

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.pmc.nd.multipatform.doc/concepts/cjt1002_.html?resultof=%22%6d%65%73%73%61%67%69%6e%67%22%20%22%6d%65%73%73%61%67%22%20%22%65%6e%67%69%6e%65%22%20%22%65%6e%67%69%6e%22%20%22%70%6f%6c%69%63%79%22%20%22%70%6f%6c%69%63%69%22%20

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