

하나씩 쉽게 따라 해보는 IBM WebSphere Application Server(WAS) v7 – 19

이정운 (juwlee@kr.ibm.com)

하나씩 쉽게 따라 해보는 IBM WAS v7 그 열 아홉번째 이야기를 시작합니다. 열 아홉번째 이야기는 지난 강좌와의 연속 기획으로 WAS Server 설정 옵션에 이은 Application 설정 옵션들에 대한 설명을 하도록 하겠습니다. WAS 서버의 옵션 설정을 가장 많이 사용한다고 하면, 그 다음으로 많이 사용하는 옵션 설정은 바로 Application 에 대한 옵션 설정입니다. Application 에 대한 옵션을 어떻게 설정하느냐는 어떻게 Application 을 잘 관리할 것인가와 맞닿아 있는 이야기 이므로, 해당 옵션을 이해하고 본인이 사용하는 환경에 보다 더 적합하게 Application 을 관리할 수 있게 되기를 바라겠습니다. 이미 이전 강좌에서 옵션의 중요성은 많이 설명했으니 간단히 넘어가기로 하고 실제 강좌를 진행하도록 하겠습니다.^&

The screenshot displays the IBM WebSphere Integrated Solutions Console interface. On the left is a navigation tree with categories like Welcome, Guided Activities, Servers, Applications, Services, Resources, Security, Environment, System administration, Users and Groups, Monitoring and Tuning, Troubleshooting, Service integration, and UDDI. The 'Applications' category is expanded, showing 'New Application', 'Application Types' (with 'WebSphere enterprise applications' selected), 'Business-level applications', and 'Assets'. The main panel is titled 'Enterprise Applications' and includes a header 'Cell=T400B92090901Node01Cell, Profile=AppSrv03'. Below the header, there's a section for 'Enterprise Applications' with a description and a 'Preferences' button. A row of action buttons (Start, Stop, Install, Uninstall, Update, Rollout Update, Remove File, Export, Export DDL, Export File) is visible. A table lists installed applications with columns for 'Select', 'Name', and 'Application Status'. The table contains 8 entries, all with a status of 'Running' (indicated by a green arrow). At the bottom, it says 'Total 8'.

Select	Name	Application Status
<input type="checkbox"/>	ClassLoaderOutEJB.ear	Running
<input type="checkbox"/>	ClassLoaderTestEAR	Running
<input type="checkbox"/>	DefaultApplication	Running
<input type="checkbox"/>	EJBCallTestEAR	Running
<input type="checkbox"/>	EJBCallerJAR.ear	Running
<input type="checkbox"/>	IBMWTC	Running
<input type="checkbox"/>	IvtApp	Running
<input type="checkbox"/>	SWERV	Running

Part 1. 기본적인 Application 옵션

WAS 의 관리콘솔에서 Applications > Application types > WebSphere enterprise applications 메뉴를 클릭하여 나타난 Application 중에서 하나의 Application 을 클릭하면 하단과 같이 Application 에 대한 세부 설정 메뉴를 확인할 수 있습니다. (이번 강좌에서는 기본적으로 설치시에 같이 설치되는 DefaultApplication 을 선택하여 확인하도록 하겠습니다.)

[Enterprise Applications](#) > [DefaultApplication](#)

Use this page to configure an enterprise application. Click the links to access pages for further configuring of the application or its modules.

Configuration

General Properties

★ Name

Application reference validation

Detail Properties

- [Target specific application status](#)
- [Startup behavior](#)
- [Application binaries](#)
- [Class loading and update detection](#)
- [Request dispatcher properties](#)
- [Security role to user/group mapping](#)
- [View Deployment Descriptor](#)
- [Last participant support extension](#)

References

- [EJB references](#)
- [Shared library references](#)
- [Shared library relationships](#)

Modules

- [Manage Modules](#)

Web Module Properties

- [Session management](#)
- [Context Root For Web Modules](#)
- [JSP and JSF options](#)
- [Virtual hosts](#)

Enterprise Java Bean Properties

- [Default messaging provider references](#)
- [Stateful session bean failover settings](#)
- [Application profiles](#)
- [Map data sources for all 2.x CMP beans](#)
- [Provide default data source mapping for modules containing 2.x entity beans](#)
- [EJB JNDI names](#)

Database Profiles

- [SQL profiles and pureQuery bind files](#)



처음에 살펴볼 옵션은 Modules 카테고리에 있는 **Manage Modules** 옵션입니다. 해당 옵션을 클릭하면 하단과 같은 모듈 �핑 화면을 확인할 수 있습니다.

[Enterprise Applications](#) > [DefaultApplication](#) > [Manage Modules](#)

Manage Modules

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:

<input type="button" value="Remove"/> <input type="button" value="Update"/> <input type="button" value="Remove File"/> <input type="button" value="Export File"/>				
 				
Select	Module	URI	Module Type	Server
<input type="checkbox"/>	Increment EJB module	Increment.jar,META-INF/ejb-jar.xml	EJB Module	WebSphere:cell=T400B92090901Node01Cell,node=T400B92090901Node03,server=server1
<input type="checkbox"/>	Default Web Application	DefaultWebApplication.war,WEB-INF/web.xml	Web Module	WebSphere:cell=T400B92090901Node01Cell,node=T400B92090901Node03,server=server1

해당 옵션은 모듈 매핑에 관련된 설정으로서 실제로 EAR 로 배포된 어플리케이션 내부의 각 모듈들이 어떤 서버와 매핑되는 가의 설정을 할 수 있는 메뉴입니다. 예를 들어 위의 그림에서는 Web 모듈과 EJB 모듈, 두 가지를 확인할 수 있으면 해당 모듈들이 server1 과 매핑되어 있는 것을 그림으로 확인할 수 있습니다. 해당 서버와 매핑되어 있다라는 말은 해당 어플리케이션의 Web 모듈과 EJB 모듈은 해당 서버 위에서 운영된다는 의미입니다. (현재 해당 어플리케이션은 server1 에서 운영된다는 의미) 이 메뉴를 통해서 해당 어플리케이션의 각 모듈에 대한 서버 매핑을 확인하고 변경할 수 있습니다.

Session management 옵션은 이전 강좌에서 이미 다루었으니 넘어가고 다음으로 체크할 메뉴는 **Context root for Web Modules** 옵션입니다.

[Enterprise Applications](#) > [DefaultApplication](#) > **Context Root For Web Modules**

Context Root For Web Modules

Context root defined in the deployment descriptor can be edited.

Web module	URI	Context Root
Default Web Application	DefaultWebApplication.war,WEB-INF/web.xml	/

OK Cancel

Context root 는 이전에도 한 번 설명드린 것 같은데 해당 웹 모듈의 컨텍스트 루트 위치를 의미합니다. 즉, 요청이 들어오면 어디를 컨텍스트 루트로서 인식하고 요청을 처리할 것인지를 결정하게 됩니다. 또한 컨텍스트 루트는 각 어플리케이션의 웹 모듈에 대한 유일한 구분자 역할을 합니다. 따라서 두 개의 어플리케이션의 웹 모듈이 동일한 컨텍스트 루트를 가질 수 없습니다. 예를 들어 위에 첨부된 그림처럼 컨텍스트 루트가 "/" 로 되어 있다면 들어오는 모든 요청들은 해당 어플리케이션으로 라우트 될 수 있습니다. <http://127.0.0.1/>, <http://127.0.0.1/Test>, <http://127.0.0.1/Test.jsp> 같은 요청들이 위의 "/" 를 컨텍스트 루트로 가진 DefaultApplication 에서 처리할 수 있습니다. 만약 컨텍스트 루트를 "Default" 로 변경한다면 위의 요청들은 DefaultApplication 에서 처리될 수 없으며 <http://127.0.0.1/Default> , <http://127.0.0.1/Default/Test>, <http://127.0.0.1/Default/Test.jsp> 와 같은 요청들이 처리될 수 있습니다.

JSP and JSF options 는 내용 그대로 JSP 와 JSF 에 대한 옵션입니다.

[Enterprise Applications](#) > [DefaultApplication](#) > **JSP and JSF options**

JSP and JSF options

JSP reloading options for Web modules

Servlet and JSP reload attributes can be specified per module.

Web module	URI	JSP enable class reloading	JSP reload interval in seconds
Default Web Application	DefaultWebApplication.war,WEB-INF/web-ext.xml	<input checked="" type="checkbox"/>	10

JSF implementation

Select a JSF implementation that the container will use for this application.

- ☒ Sun Reference Implementation 1.2
- ☐ MyFaces 1.2

JSP reloading options for Web modules 는 JSP 리로드에 대한 옵션입니다. JSP 리로드란 JSP 에 대한 변경사항이 있을 경우에 한해서, 재시작 없이 바로 반영하기 위한 옵션입니다. JSP 의 경우에는 정적 콘텐츠이기 때문에 서버나 클래스로더의 재시작 단계 없이 해당 옵션을 통해서 바로 변경사항을 반영할 수 있습니다. 또한, 몇 초 주기로 변경사항을 체크할지에 대한 시간 범위를 변경할 수도 있습니다.

JSF implementation 옵션은 JSF 구현을 위해서 사용할 구현체를 지정하는 옵션입니다. Sun에서 제공하는 Sun Reference Implementation 1.2 이 기본으로 사용되지만 원하실 경우 이 옵션을 통해서 Apache 프로젝트에서 구현한 MyFaces 1.2 로 구현체를 변경하실 수 있습니다.

다음으로 소개할 옵션은 **Virtual hosts** 관련 옵션입니다. Virtual host 는 가상 호스트를 의미하며 이미 웹서버에서 사용해 보신분들도 있어서 익숙하실 수도 있는데 가상으로 호스트를 만든다는 의미입니다.

[Enterprise Applications](#) > [DefaultApplication](#) > **Virtual hosts**

Virtual hosts

Specify the virtual host where you want to install the Web modules that are contained in your application. You can install Web modules on the same virtual host or disperse them among several hosts.

☒ Apply Multiple Mappings

 		
Select	Web module	Virtual host
<input type="checkbox"/>	Default Web Application	default_host ▼

위의 설명이 이해가 좀 어려우신가요? 호스트는 하나의 물리적인 서버를 의미하기도 하지만 결국은 하나의 IP 로 서비스를 받는 것을 의미합니다. 그러므로 보통의 경우에는 하나의 물리적인 하드웨어에 하나의 IP가 매핑되고, 이는 하나의 호스트가 존재하는 것으로 표현되며 하나의 서비스를 하게 됩니다. 그러나, 아무리 그렇다고 해도 하나의 서버에 하나의 서비스만 하는 것은 너무 낭비가 심한 경우가 많죠. 그래서 가상호스트라는 것이 존재하는 것입니다. 하나의 서버에 하나의 IP 만을 받는다고 하여도, 특정 포트를 호스트에 매핑하여 가상호스트를 등록하고 그것을 마치 또 다른 호스트처럼 사용될 수 있게 하는 것입니다.

가상 호스트는 결국, 하나의 서버와 IP 를 가지고 여러 서비스를 가능하게 합니다. (따라서, WAS에서 실제 서비스가 되려면 가상 호스트에 해당 서비스 포트가 등록되어 있어야합니다. 보통 많은 분들이 잊어먹고 있는 경우가 많은데 어플리케이션 테스트 했을 경우에 가상호스트를 매핑할 수 없다는 관련 예외가 발생하면 90% 정도는 해당 서버의 서비스 포트와 가상호스트가 매핑되지 않아서 발생하는 경우입니다.)

Enterprise Java Bean Properties 카테고리에 포함된 Application 옵션은 여러가지가 있지만 실제적으로 잘 사용되지는 않습니다. (실제로 변경하는 경우가 많지 않음) 그래도 몇가지는 이번 강좌에서 소개하고 넘어가도록 하겠습니다.

먼저 **Stateful session bean failover settings** 입니다. 기본으로 설정되는 옵션인데, 말 그대로

Stateful session bean 에 대한 페일오버에 대한 설정을 할 수 있습니다. 그리고 필요하면 복제 설정도 여기서 가능합니다. 단, 요즘 프로젝트에서 Stateful session bean 자체를 그리 많이 사용하지 않으니 바로 넘어가도록 하겠습니다.

[Enterprise Applications](#) > [DefaultApplication](#) > [Stateful session bean failover settings](#)

Each EJB container provides a method for stateful session beans to fail over to other servers. This section enables you to specify when failover occurs for the stateful session beans in this application. You can also override the default replication settings that are configured for the EJB container on the server with values that are specific to the application.

Configuration

General Properties

☒ Enable stateful session bean failover using memory to memory replication

Replication settings:

- ☒ Use replication settings from EJB container
- ☐ [Use application replication settings](#) (Replication domains are defined, but the memory to memory settings have not been selected.)

그 다음의 Map data sources for all 2.x CMP beans 와 Provide default data source mapping for modules containing 2.x entity beans 는 모두 해당 bean에 대한 기본 데이터 소스를 설정하는 부분이므로 역시 그냥 넘어가시면 됩니다. (추가로 데이터소스를 연결안해주고 해당 bean 들을 사용하면 여기서 설정되어 있는 기본 데이터소스로 맵핑됩니다.)

[Enterprise Applications](#) > [DefaultApplication](#) > [Map data sources for all 2.x CMP beans](#)

Map data sources for all 2.x CMP beans

Specify an optional data source for each 2.x container-managed persistence (CMP) bean. Mapping a specific data source to a CMP bean overrides the default data source for the module that contains the enterprise bean.

<div> <div>Set Multiple JNDI Names</div> <div>Set Authorization Type</div> <div>Modify Resource Authentication Method...</div> </div>					
<div> <div></div> <div></div> </div>					
Select	EJB	EJB module	URI	Target Resource JNDI Name	Resource authorization
<input type="checkbox"/>	Increment	Increment EJB module	Increment.jar,META-INF/ejb-jar.xml	<div>DefaultDatasource</div> <div>Browse...</div>	Resource authorization: Per application

[Enterprise Applications](#) > [DefaultApplication](#) > [Provide default data source mapping for modules containing 2.x entity beans](#)

Provide default data source mapping for modules containing 2.x entity beans

Specify the default data source for the Enterprise JavaBeans (EJB) 2.x module that contains 2.x container-managed persistence (CMP) beans.

<div> <div>Set Multiple JNDI Names</div> <div>Set Authorization Type</div> <div>Modify Resource Authentication Method...</div> <div>Extended Properties...</div> </div>				
<div> <div></div> <div></div> </div>				
Select	EJB module	URI	Target Resource JNDI Name	Resource authorization
<input type="checkbox"/>	Increment EJB module	Increment.jar,META-INF/ejb-jar.xml	<div>DefaultDatasource</div> <div>Browse...</div>	Resource authorization: Per application

마지막으로 EJB JNDI names 옵션은 각 EJB 의 JNDI name 을 설정하는 옵션입니다. 보통 EJB 어플리케이션을 개발이나 배포할 때 JNDI 이름을 입력하면 이 옵션에서 해당 이름을 확인할 수 있으며, 필요하면 여기서 변경도 가능합니다.

[Enterprise Applications](#) > [DefaultApplication](#) > **EJB JNDI names**

EJB JNDI names

Each non-message-driven enterprise bean in your application or module must be bound to a Java Naming and Directory Interface (JNDI) name. For beans in a pre-EJB 3.0 module, you have to use JNDI name for the bean to provide the binding. For beans in a EJB 3.0 module, you can optionally provide binding through JNDI name for the bean or local/remote home JNDI names. If JNDI name for the bean is specified, you cannot specify binding for its local/remote home and any business interface. If no JNDI name is specified for beans in a EJB 3.0 module, runtime will provide a container default.

EJB module	EJB	URI	Target Resource JNDI Name
Increment EJB module	Increment	Increment.jar,META-INF/ejb-jar.xml	Target Resource JNDI Name <input type="text" value="Increment"/>

Part 2. Application 세부 설정

이전 파트에서 Application 기본 설정에 대해서 살펴보았다면 이번 파트에서는 Application 세부 설정에 대한 부분을 살펴보도록 하겠습니다. 관리콘솔에서 Detail Properties 라고 카테고리 되어 있는 부분입니다.

[Enterprise Applications](#) > [DefaultApplication](#)

Use this page to configure an enterprise application. Click the links to access pages for further configuring of the application or its modules.

Configuration

General Properties

★ Name
DefaultApplication

Application reference validation
Issue warnings

Detail Properties

- Target specific application status
- Startup behavior
- Application binaries
- Class loading and update detection
- Request dispatcher properties
- Security role to user/group mapping
- View Deployment Descriptor
- Last participant support extension

References

- EJB references
- Shared library references
- Shared library relationships

Modules

- Manage Modules

Web Module Properties

- Session management
- Context Root For Web Modules
- JSP and JSF options
- Virtual hosts

Enterprise Java Bean Properties

- Default messaging provider references
- Stateful session bean failover settings
- Application profiles
- Map data sources for all 2.x CMP beans
- Provide default data source mapping for modules containing 2.x entity beans
- EJB JNDI names

Database Profiles

- SQLJ profiles and pureQuery bind files

먼저 **Target specific application status** 는 해당 어플리케이션이 맵핑된 WAS 서버의 상태와 현재 어플리케이션의 상태를 하단처럼 표시해 줍니다. 해당 어플리케이션은 testCluster 와 webserver1 에 매핑되어 있기 때문에 두 개의 서버를 보여주고 있습니다.

[Enterprise Applications](#) > [DefaultApplication](#) > [Target specific application status](#)

Use this page to view a mapping of a deployed object, such as an application or module, into a target server or cluster environment. This page displays the status of the enterprise application or module on each server or cluster.

Preferences

Enable Auto Start Disable Auto Start					
Select	Target	Node	Version	Auto Start	Application Status
You can administer the following resources:					
<input type="checkbox"/>	testCluster	Not applicable	Not applicable	Yes	
<input type="checkbox"/>	webserver1	T400B92090901Node01	ND 7.0.0.7	Yes	
Total 2					

Startup behavior 는 시작할 때의 행동들에 대한 옵션 설정입니다. 시작할 때의 행동들이란 Application 이 시작되는 것에 대한 옵션 설정을 의미합니다.

[Enterprise Applications](#) > [DefaultApplication](#) > **Startup behavior**

Use this page to configure settings that determine how quickly an application starts compared to other applications and the server.

Configuration

General Properties

* Startup order

1

☐ Launch application before server completes startup

☒ Create MBeans for resources

Apply

OK

Reset

Cancel

Startup order 는 WAS 가 시작될 경우에 해당 어플리케이션에 대한 시작 순서입니다. 시작에 대한 가중치의 개념으로 작은 가중치를 가질수록 먼저 시작됩니다. 즉, WAS 서버에 여러 어플리케이션이 디플로이 되어 있을 경우 1이 가장 먼저 시작되며 더 큰 수 일수록 더 늦게 시작합니다.

Launch application before server completes startup 옵션은 WAS 서버의 시작이 완료되기 전에 어플리케이션을 실행할 것인가를 의미합니다. 좀 말이 애매모호하게 느껴질 수도 있는데 좀 더 이해하기 쉽게 해당 옵션에 대한 false 와 true 를 직접 설명드리겠습니다. 기본은 false 인데 이 경우는, 어플리케이션이 시작이 완료될 때까지 WAS 서버 시작이 완료되지 않는다는 의미입니다. 즉 어플리케이션이 정상적으로 시작이 되어야지만 해당 WAS 서버의 시작도 완료될 수 있습니다. True 의 경우는 이와 반대로 어플리케이션이 시작되든 안되든 상관없이 WAS 서버는 시작이 완료될 수 있다는 의미입니다. 즉, 어플리케이션의 시작은 별도의 스레드로 구동되어 WAS 서버의 시작과 연관되지 않는다는 의미입니다.

Create MBeans for resources 옵션은 해당 Application 자원에 대한 MBean 객체를 만든다는 옵션이며 WAS에서 MBean 객체는 보통 관리나 모니터링을 위해 사용됩니다. 따라서, 기본적으로 true 로 선택되어 있습니다.

다음으로 보실 것은 **Application binaries** 옵션입니다. 해당 옵션은 이전 17강에서 폴더에 대해서 설명하면서 이미 설명했기 때문에 해당 강좌를 참조하시기 바라며 간단히만 짚고 넘어가도록 하겠습니다.

[Enterprise Applications](#) > [DefaultApplication](#) > **Application binaries**

Use this page to configure the location and distribution of application binary files.

Configuration

General Properties

★ Location (full path)

`${APP_INSTALL_ROOT}/${CELL}`

☐ Use configuration information in binary

☒ Enable binary distribution, expansion and cleanup post uninstallation

File permissions

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 level

Unknown

Apply

OK

Reset

Cancel

Use configuration information in binary 옵션을 사용하면 config 밑에 있는 설정 파일을 참조하는 것이 아니라 Application 이 설치된 installedApps 에 설치된 어플리케이션 파일 및 설정 파일을 참조합니다.

Enable binary distribution, expansion and cleanup post uninstallation 옵션은 해당 어플리케이션이 맵핑된 서버의 node 마다 어플리케이션을 자동으로 배포하고 설치하는 옵션입니다.

이미 말씀드린 것 처럼 위의 옵션의 자세한 사항은 하나씩 따라해보는 IBM WAS 강좌 중 17강을 참조하시기 바라겠습니다.

그 뒤가 **Class loader** 인데 이는 클래스로더에 관련된 옵션입니다.

[Enterprise Applications](#) > [DefaultApplication](#) > **Class loader**

Use this page to configure the reloading of classes when application files are updated.

Configuration

General Properties

Class reloading options

☐ Override class reloading settings for Web and EJB modules

Polling interval for updated files

Seconds

Class loader order

☒ Classes loaded with parent class loader first

☐ Classes loaded with local class loader first (parent last)

WAR class loader policy

☒ Class loader for each WAR file in application

☐ Single class loader for application

Apply

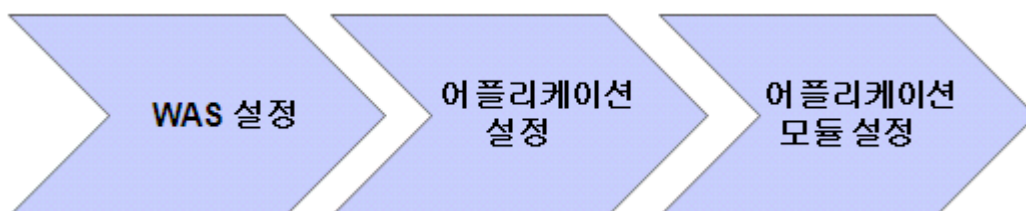
OK

Reset

Cancel

Class reloading options 카테고리에 있는 옵션은 클래스 리로드를 위한 옵션입니다. 클래스 리로드란 WAS 관리 콘솔이나 wsadmin 셸을 통해서가 아니라 Application 이 실제 설치된 폴더에서 직접 클래스를 변경했을 경우, 동적으로 클래스를 다시 로드하기 위한 옵션입니다. 해당 옵션이 설정되고 Polling interval for updated files 옵션에 주기가 입력되면, 그 주기만큼 돌면서 클래스를 체크하고 변경사항이 있다면 해당 어플리케이션을 다시 로드합니다. 단, 여기서 주의할 점은 JVM의 특성상 어플리케이션이 해당 클래스를 메모리에 다시 로드하기 위해서는 클래스로더가 내려갔다가 다시 올라온다는 점을 유념하셔야 합니다.

여기서 하나 더 짚고 넘어갈 것이 있는데 클래스 리로드에 대한 옵션은 WAS 에서 뿐만 아니라 어플리케이션을 개발할 때 RAD 같은 툴에서도 먼저 줄 수 있습니다. 이런식으로 여러 군데에서 동일한 옵션을 줄 수 있는 경우라면 WAS 에서는 다음의 단계를 따릅니다.



즉, 어플리케이션 각 모듈에서 설정된 것이 최우선으로 적용되며, 그것이 없을 경우에는 어플리케이션에 있는 설정이, 그 다음이 WAS 에 있는 설정이 적용됩니다. 따라서 기존 WAS v6.1 의 경우에는 아무리 이 옵션을 변경하여도 어플리케이션에 해당 설정이 들어가 있는 경우 원치 않는 클래스 리로드가 발생하는 경우가 많았습니다. 그렇기 때문에 WAS v7 에서는, 이 부분의 경우 옵션 자체에 덮어쓰기(override) 기능이 들어가면서 어플리케이션의 설정 자체를 덮어쓰게 되어있습니다. 즉, 다른 옵션과는 다르게 어플리케이션에 설정이 있다고 하여도 WAS 의 설정이 가장 우선적으로 적용됩니다. (이런 것을 볼때마다, WAS 가 점점 나아지고 있다는 느낌이 조금씩 듭니다.^.^-)

다음의 **Class loader order** 는 18장에서 설명한 Class loading mode 옵션과 동일한 옵션이며 WAR class loader policy 옵션은 Class loader policy 옵션과 동일한 옵션입니다. (다만, 단위가 EAR 이 아니라 WAR 인 것만 다릅니다.)

Security role to user/group mapping 은 Application 에 접근하기 위한 보안 역할에 대한 설정을 할 수 있는 옵션입니다. 개인적으로 테스트 할 때는 많이 안쓰이지만 기업의 경우에는 많이 사용되는 옵션입니다. 사용자나 그룹 단위로 역할을 부여할 수 있습니다.

[Enterprise Applications](#) > [DefaultApplication](#) > **Security role to user/group mapping**

Security role to user/group mapping

Each role that is defined in the application or module must map to a user or group from the domain user registry. accessIds: The accessIds are required only when using cross realm communication in a multi domain scenario. For all other scenarios the accessId will be determined during the application start based on the user or group name. The accessIds represent the user and group information that is used for Java Platform, Enterprise Edition authorization when using the WebSphere default authorization engine. The format for the accessIds is user:realm/uniqueUserID, group:realm/uniqueGroupID. Entering wrong information in these fields will cause authorization to fail. AllAuthenticatedInTrustedRealms: This indicates that any valid user in the trusted realms be given the access. AllAuthenticated: This indicates that any valid user in the current realm be given the access.

Select	Role	Special subjects	Mapped users	Mapped groups
<input type="checkbox"/>	All Role	All Authenticated in Application's Realm		

Deployment Descriptor 는 EAR 어플리케이션의 DD 파일을 관리콘솔에서 확인할 수 있는 옵션입니다. 즉, 여기서 application.xml 파일을 바로 확인할 수 있습니다.

[Enterprise Applications](#) > [DefaultApplication](#) > **Deployment Descriptor**

Expand and collapse the application deployment descriptor data to view.

[Expand All](#) [Collapse All](#)

```
<application id="Application_ID" >
  <display-name> DefaultApplication.ear</display-name>
  <description> This is the IBM WebSphere Application Server Default Application.</description>
  <module id="WebModule_1" >
    <web>
      <web-uri> DefaultWebApplication.war</web-uri>
      <context-root> /</context-root>
    </web>
  </module>
  <module id="EjbModule_1" >
    <ejb> Increment.jar</ejb>
  </module>
  <security-role id="SecurityRole_1262060543765" >
    <description> All Authenticated users role.</description>
    <role-name> All Role</role-name>
  </security-role>
</application>
```

Detail Properties 카테고리의 마지막 옵션은 **Last participant support extension** 이라는 옵션입니다. 해당 옵션은 마지막 **참여자** 지원이라는 기능의 사용여부를 결정하는 설정입니다.

[Enterprise Applications](#) > [DefaultApplication](#) > **Last participant support extension**

Use this page to configure settings for last participant support. Last participant support is an extension to the transaction service that enables a single one-phase resource to participate in a two-phase transaction with one or more two-phase resources. Values on this panel are ignored if you select Use configuration information in binary on the Application binaries panel.

Configuration

General Properties

☐ Accept heuristic hazard

Apply

OK

Reset

Cancel

여기서 마지막 참여자 지원 기능을 좀 더 자세히 설명드리자면, 이전 데이터 연결 강좌에서 보셨겠지만 데이터 소스를 구성할 때 XA 와 Non-XA 를 선택하셨을 것입니다. 이중 XA 의 경우에는 여러가지 자원에 동시에 접속할 수 있는 2 Phase commit 이 가능하다고 이미 말씀드렸습니다. 그런데 모든 자원이 이렇게 2 Phase commit 을 사용할 수 있으면 고민도 없고 좋겠지만 그렇지 않은 경우도 있고, 필요에 의해서(주로, 성능을 위해서) Non-XA 데이터 소스를 써야만 하는 경우도 있습니다. 근데 또 문제는, 이런 경우에도 2 Phase commit 으로 트랜잭션을 묶고 싶은 소망이 생긴다는 것입니다.—,.; Non-XA 를 썼으면서도 2 Phase commit 을 쓰고 싶은 모순적인 상황이 발생하는 것입니다. 이런 경우 하나의 Non-XA 자원에 한해서 다른 여러 개의 XA 자원과 2 Phase commit 이 가능하도록 트랜잭션을 묶을 수 있는 기능이 바로 지금보고 있는 마지막 참여자 지원 기능입니다.

마지막으로 **Shared library references** 옵션은 해당 어플리케이션에 공유 라이브러리를 연결하기 위한 옵션입니다. 특히, 이 때 생각하셔야 할 것중의 하나는 어떤 클래스로더에 공유 라이브러리를 연결할 것인가 입니다. 하단의 그림에서 보는 것 처럼 어플리케이션 클래스로더 쪽에도 연결할 수 있고, WAR 클래스 로더 쪽에도 연결할 수 있습니다.

[Enterprise Applications](#) > [DefaultApplication](#) > **Shared library references**

Shared Library Mapping for Modules

Specify shared libraries that the application or individual modules reference. These libraries must be defined in the configuration at the appropriate scope.

Reference shared libraries			
Select	Application	URI	Shared Libraries
<input type="checkbox"/>	DefaultApplication	META-INF/application.xml	
Select	Module	URI	Shared Libraries
<input type="checkbox"/>	Default Web Application	DefaultWebApplication.war,WEB-INF/web.xml	

OK

Cancel

공유 라이브러리를 연결할 위치를 선택한 후 Reference shared libraries 메뉴를 클릭하면 하단처럼 선택 가능한 공유라이브러리를 선택하여 손쉽게 연결할 수 있습니다.

[Enterprise Applications](#) > [DefaultApplication](#) > [Shared library references](#) > **Shared Library Mapping**

Map shared libraries to an entire application or to one or more modules.

Map libraries to the application or module listed

Default Web Application

Select the library in the Available list. Move it to the Selected list by clicking >>.

Available: ClassLoaderTestLIB	 	Selected: <div></div>
<div>New</div>		
<div>OK Cancel</div>		

여기까지 잘 이해가 되시나요? 지난 강좌에 의해서 개략적인 옵션 설명만 주~욱 하다 보니 아마도 조금은 지루하셨을 수 있을거라고 생각합니다. 그러나 이미 강조드렸지만 Application 옵션은 WAS 서버 옵션 다음으로 가장 많이 사용되는 옵션이고, 이를 잘 이해하면 어떻게 WAS 서버에서 Application 을 잘 관리할 수 있을까에 대한 답을 조금 보여주기도 하므로 잘 이해하게 될 때까지 반복해서 읽어보고 직접 테스트 해보시길 바라겠습니다. 그럼 이번 강좌도 여기서 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.multipiplatform.doc/info/welcome_nd.html

참고 2) IBM WebSphere Application Server v7.0 InfoCenter

- Enterprise application settings

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.doc/info/aes/ae/urun_rapp_inst.html?resultof=%22%57%65%62%53%70%68%65%72%65%22%20%22%77%65%62%73%70%68%65%72%22%20%22%65%6e%74%65%72%70%72%69%73%65%22%20%22%65%6e%74%65%72%70%72%69%73%22%20%22%61%70%70%6c%69%63%61%74%69%6f%6e%73%22%20%22%61%70%70%6c%69%63%22%20

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