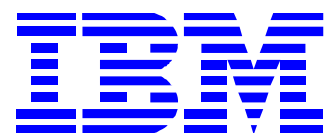


WebSphere Application Server v8.5.5
IBM WAS Base 버전과 WXS 를 이용한
Session Cluster 구축

(2013. 08.)

IBM SWG WebSphere Technical Sales

이정운 과장(juwlee@kr.ibm.com)



0) IBM WAS Base 버전과 WXS 를 이용한 Session Cluster 구축 간략 소개

안녕하세요 freeman 입니다.

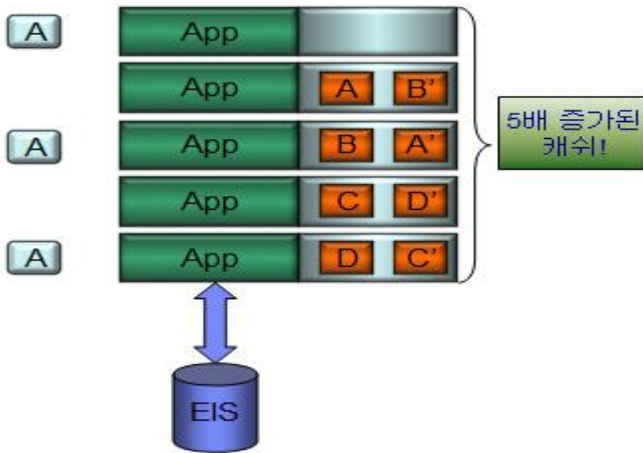
이번 강좌에서 언급드릴 내용은 IBM WAS Base 버전과 WXS 를 이용한 Session Cluster 구축입니다. Session Cluster 는 이미 잘 알고 계시겠지만 Session persistence 를 제공할 수 있는 Server Cluster 를 의미합니다. 이를 좀 더 쉽게 설명하면 장애가 발생하면 Cluster 내의 가용한 다른 서버로 fail-over 가 되며 이때 Session 또한 소실되지 않고 유지할 수 있는 고가용성 환경을 의미합니다. (이전 버전까지 Memory-to-memory 방식으로 Session Cluster 를 구축하기 위해서는 IBM WAS ND 버전이 반드시 필요했습니다.) IBM WAS v8.5.5 에서 가장 변화된 부분중의 하나는 WebSphere eXtreme Scale(WXS) 이라는 In-Memory Data Grid 솔루션의 라이선스가 IBM WAS 랑 통합되어 별도의 비용 추가 없이 IBM WAS 와 WXS 를 통합해서 활용할 수 있습니다. (ND 버전의 경우에는 제한 없이 WXS 사용이 가능하며 Base 버전의 경우에는 라이선스 적으로 HTTP Session 과 DynaCache 목적의 사용으로만 WXS 사용이 제한됩니다.) 그렇기 때문에, Session Cluster 를 구축하기 위해서 반드시 IBM WAS ND 버전을 구매하는 것이 아니라 IBM WAS Base 버전만 구매한다고 해도 IBM WAS Base 와 통합 제공되는 WXS 를 이용해서 Session Cluster 를 구축할 수 있습니다.

여기서, IBM WXS 란 IBM WebSphere 제품 중에서 In-Memory Data Grid(분산 캐싱 플랫폼) 를 제공할 수 있는 솔루션입니다. 여기서 분산 캐싱 플랫폼이란 다수의 JVM 에 분산되어 있는, 사용하지 않는 메모리를 취합하여 하나의 논리적인 캐시처럼 사용하는 기술을 의미하며 Gartner 에서도 언급되었지만 향후 가장 주목되는 기술중의 하나입니다. 이를 이용하면 거대한 트랜잭션 볼륨을 가진 중요한 비즈니스 어플리케이션에 지속적인 고성능과 확장성, 가용성을 실현해 줄 수 있습니다.

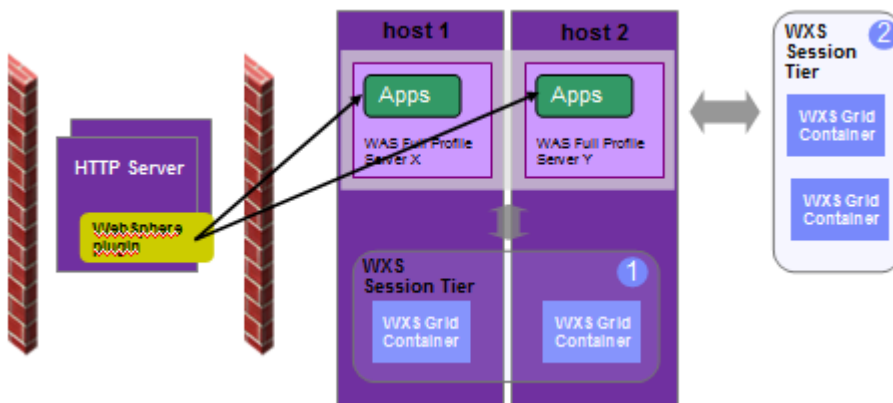
좀 더 자세히 이야기 하자면 분산 캐싱 기술을 이용해서 하단과 같이 아키텍처적으로 DB layer 앞 단에 위치하여 고비용이며 빈번하게 발생하는 DB 작업 대신에 미리 만들어 둔 Cache layer 에서 실제 데이터를 가지고 와서 고성능, 저비용으로 작업이 가능한 기술을 의미합니다. 여기서 Cache layer 에는 DB 데이터가 들어가며 아키텍처적으로 Pre-loader 라는 것을 사용하여 사전에 미리 다 가져다 놓을 수도 있고 요청이 있을 때마다 가지고 올 수도 있습니다. 미리 언급한 것처럼 비싸게 물리적인 DB 에 직접 access 하는 것보다 Cache layer 에서 데이터를 가지고 오는 것이 훨씬 저비용으로 고성능을 제공할 수 있습니다.



또한, WXS 로 구성되는 Cache layer 는 분산 캐싱 기술을 활용하였기 때문에 분리되어 있는 여러 개의 JVM 인스턴스를 마치 하나의 JVM pool 처럼 인지 가능하고 JVM 인스턴스가 추가되는 개수만큼 병렬적으로 확장이 가능하여 마치 충분한 메모리 저장소로 활용 가능합니다.



이와 같은 WXS 의 분산 캐싱 기능을 통해서 여러 대의 WAS 에서 사용되는 Session 을 WXS 의 Session Layer(메모리) 에 저장하고 이를 앞에 위치한 모든 WAS 가 접속하고 활용하여 Session Cluster 를 구축할 수 있습니다. (원하는 아키텍처 형태에 따라 IBM WAS 와 같은 노드에 위치할 수도 있고 외부의 별도 H/W 에 구성될 수도 있습니다. 기본적으로 IBM 에서 권고하는 아키텍처는 외부의 별도 H/W 에 WXS 를 구축하는 것입니다.)



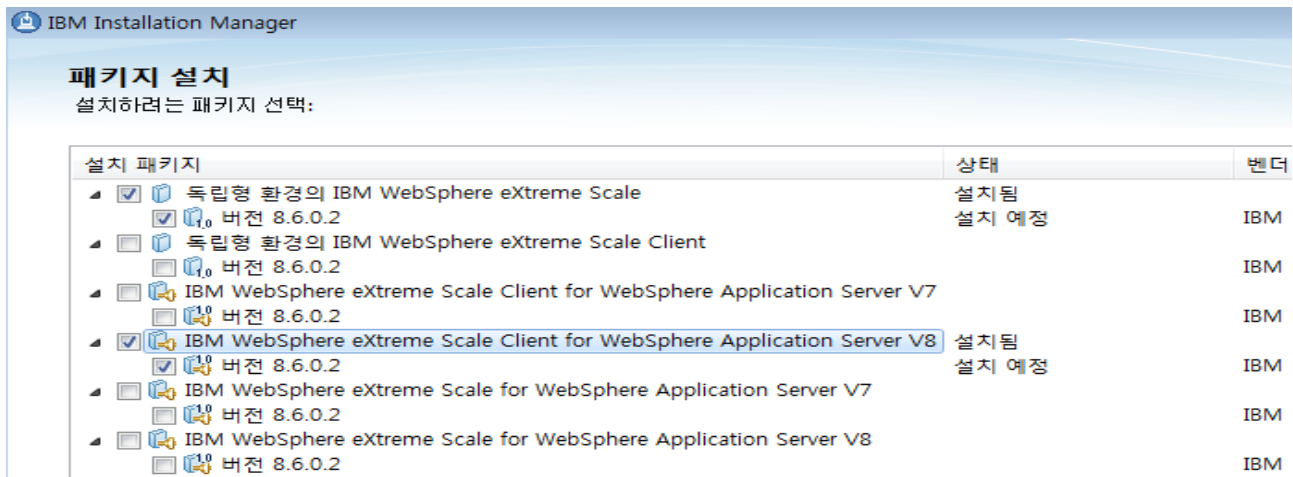
그럼 좀 더 자세한 사항은 이제부터 진행되는 강좌를 참고해주시기 바라면서 지금부터 강좌를 진행하도록 하겠습니다.

1) IBM WAS & WXS profile 생성

1. 해당 강좌를 진행하기 전에 IBM WAS 와 WXS 설치에 대해서는 이미 많이 다른 강좌에서 진행하였기 때문에 본 강좌에서는 IBM WAS 와 WXS 가 설치되어 있다는 것을 가정하고 해당 강좌를 진행하도록 하겠습니다.

(IBM WXS 경우에는 독립 서버로 사용하기 위해서 독립적인 폴더에 WXS 서버를 설치해야 하며 IBM WAS profile 을 WXS 용으로 강화하기 위해서 IBM WAS 가 설치된 폴더에 WXS 서버(WebSphere eXtreme Scale for Websphere Application Server)를 추가 설치해야 합니다. 그러나, IBM WAS 의 관리 콘솔에서 WXS 와 좀 더 쉽게 통합하는 형태를 굳이 원하지 않는다면 WAS 설치 폴더에 추가로 WXS 서버를 설치할 필요가 없으며 이후에 언급될 WXS 용으로 profile 강화작업도 수행하지 않으셔도 됩니다.

강화 없이 IBM WAS 환경 그대로 사용하고자 한다면 WXS 라이브러리들을 가지고 있는 WXS client (WebSphere eXtreme Scale Client for WebSphere Application Server) 만 설치하면 됩니다. (즉 원래의 WAS profile 을 가지고 독립 설치되어 있는 WXS 연결가능)

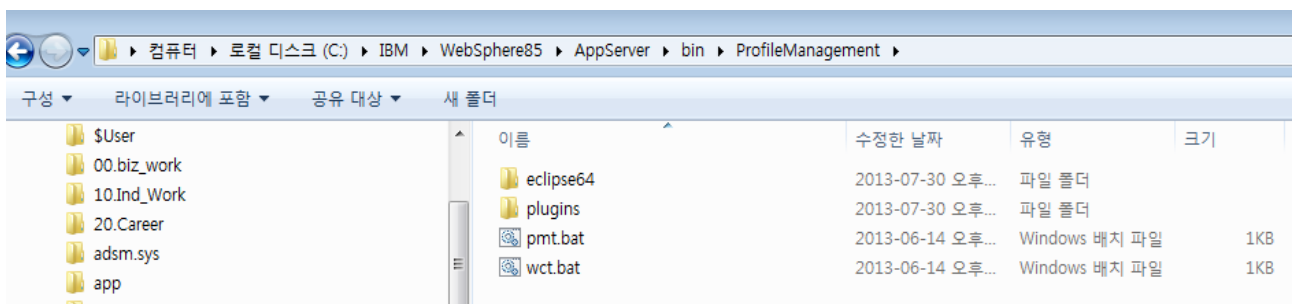


단, WXS 강화작업을 진행하지 않으셨다면 향후 강좌에서 살펴보게 될 splicer.properties 의 "catalogHostPort" 를 사용해서 카탈로그 서비스의 호스트 명과 포트를 web.xml 에 반드시 추가해주어야 합니다.)

IBM WAS v8.5.5 ND Trial Download site : <http://www.ibm.com/developerworks/downloads/ws/wasnetwork/>

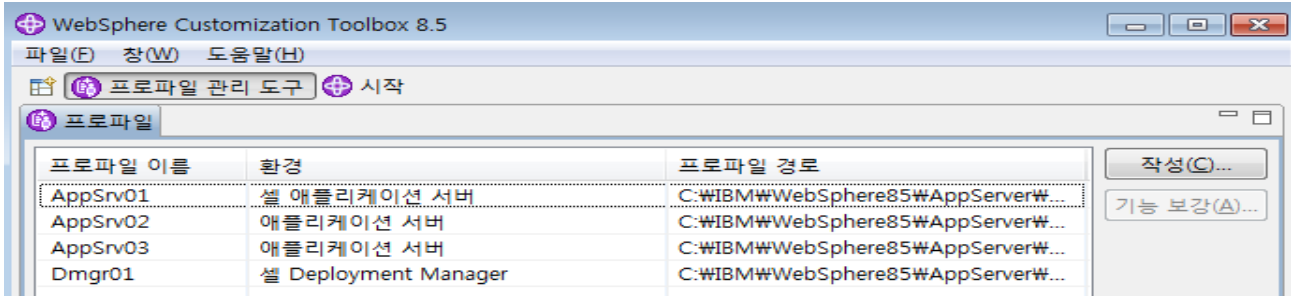
IBM WXS v8.6 Trial Download site : <http://www.ibm.com/developerworks/downloads/ws/wsdg/>

2. IBM WAS 가 설치된 폴더의 bin 디렉토리의 ProfileManagement 폴더에 가서 pmt.bat 을 클릭하여 WebSphere Customization Toolkit 을 수행합니다.

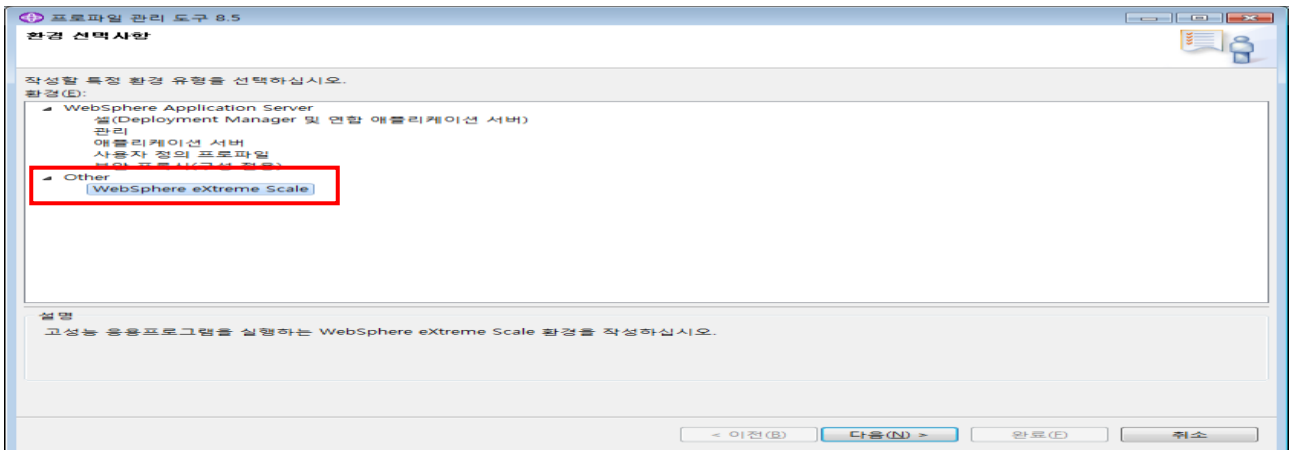


3. WCT 가 하단과 같이 수행되면 profile 을 추가하기 위해서 작성 버튼을 클릭합니다.

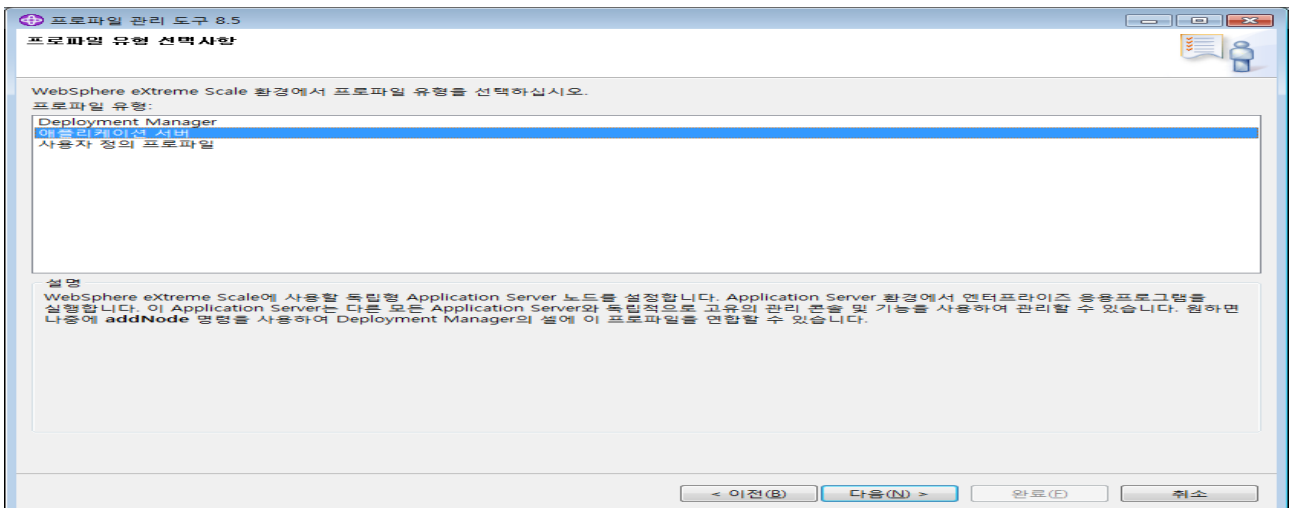
(이번 강좌에서는 IBM WAS Base 가 두 대 설치된 환경을 가정하여 IBM WAS ND 를 이용해서 WXSSvr01 과 WXSSvr02 두개의 단독 profile 을 만들어서 사용할 것 입니다.)



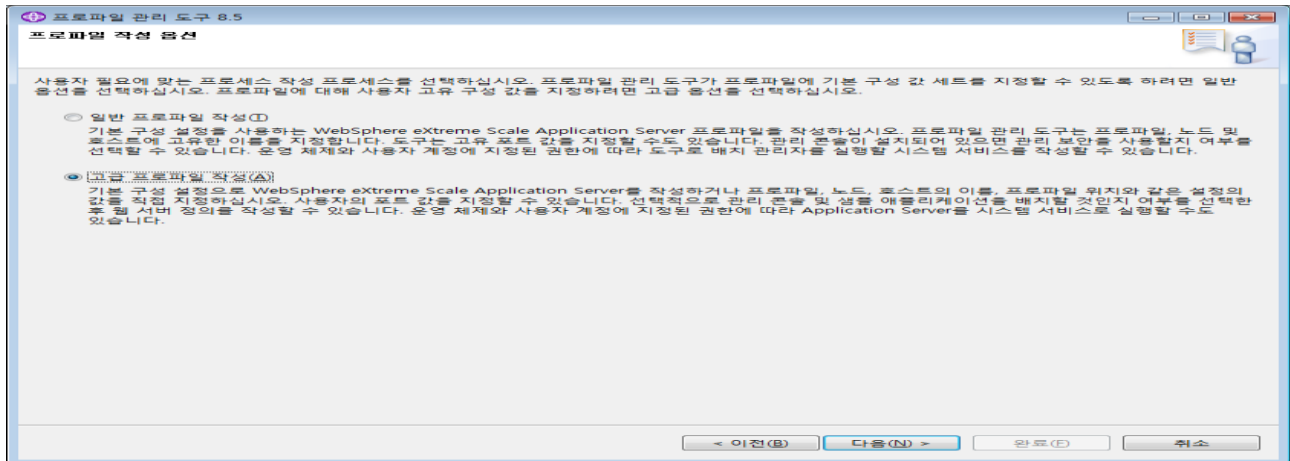
4. WXS client 를 이미 WAS 에 설치한 상태라 profile 선택사항에서 Others > WebSphere eXtreme Scale 용 profile 을 만들수 있는 메뉴가 생성된 것을 확인할 수 있으며 해당 type 을 선택하고 다음을 클릭합니다. (이렇게 profile 을 만드시면 바로 WXS 용 profile 로 강화를 시킬 수 있습니다. 아니면 WAS profile 을 만든 후에 동일 메뉴에서 "기능 보강" 버튼을 클릭하여 WXS 용 profile 로 강화도 가능합니다.)



5. profile 유형을 선택하는 부분에서는 애플리케이션 서버를 선택하고 다음을 클릭합니다.



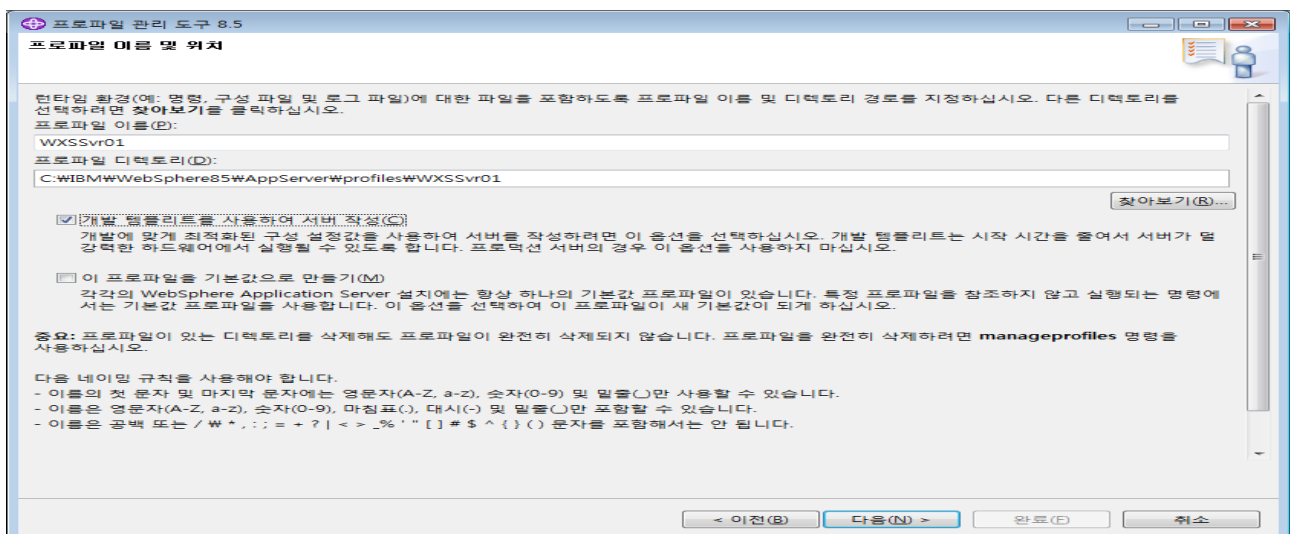
6. profile 작성 옵션은 이름을 지정하기 위해서 고급 프로파일 작성을 선택하고 다음을 클릭합니다.



7. 기본적으로 애플리케이션 배치를 선택하고 다음을 클릭합니다.



8. 원하는 profile 이름과 위치를 선택하고 다음을 클릭합니다.



9. 원하시는 서버 이름을 입력하고 다음을 클릭합니다.

프로파일 관리 도구 8.5
노드 및 호스트 이름

이 프로파일에 대한 노드 이름, 서버 이름 및 호스트 이름을 지정하십시오.

노드 이름(N):
kr050578Node05

서버 이름(S):
server1

호스트 이름(H):
kr050578

노드 이름: 노드 이름은 관리를 위해 사용됩니다. 노드가 연합된 경우 이름은 셀 내에서 고유해야 합니다.
서버 이름: 서버 이름은 애플리케이션 서버에 대한 논리 이름입니다.
호스트 이름: 호스트 이름은 이 컴퓨터의 DNS 이름(전체 또는 축약형) 또는 IP 주소이며 밑줄(_)을 포함할 수 없습니다.

다음 네이밍 규칙을 사용해야 합니다.

- 이름의 첫 문자 및 마지막 문자에는 영문자(A-Z, a-z), 숫자(0-9) 및 밑줄(_)만 사용할 수 있습니다.
- 이름은 영문자(A-Z, a-z), 숫자(0-9), 마침표(.), 대시(-) 및 밑줄(_)만 포함할 수 있습니다.
- 이름은 공백 또는 / \ * . : ; = + ? | < > _ % ' " [] # \$ ^ { } () 문자를 포함해서는 안 됩니다.

기존 프로파일을 작성 중인 프로파일로 마이그레이션할 경우, Information Center에서 마이그레이션 이전 고려사항 기사를 읽으십시오.
[온라인 Information Center 보기](#)

< 이전(B) 다음(N) > 완료(F) 취소

10. 테스트 환경이기 때문에 관리 보안 사용은 해제하고 다음을 클릭합니다.

프로파일 관리 도구 8.5
관리 보안

관리 보안을 사용 가능으로 설정할 것인지 여부를 선택하십시오. 보안을 사용 가능으로 설정하려면 관리 도구에 로그인할 사용자 이름 및 비밀번호를 제공하십시오. 이 관리 사용자가 애플리케이션 서버 내의 저장소에 작성됩니다. 프로파일 작성을 완료한 후 더 많은 사용자, 그룹 또는 외부 저장소를 추가할 수 있습니다.

☐ 관리 보안 사용(E)

사용자 이름(U):

비밀번호(P):

비밀번호 확인(F):

관리 보안에 대한 자세한 정보는 Information Center를 참조하십시오.
[온라인 Information Center 보기](#)

< 이전(B) 다음(N) > 완료(F) 취소

11. 포트 값 지정을 확인하고 다음을 클릭합니다.

프로파일 관리 도구 8.5
포트 값 지정

다음 포트에서 활동이 발견되었습니다: 9060, 9043, 9080, 9443, 2809, 5060, 5061, 8880, 9633, 9401, 9403, 9402, 9100, 9353, 7276, 7286, 5558, 5578, 11003, 11004

다음 필드에 있는 값은 애플리케이션 서버에 대한 포트를 정의하며 이 설정에 있는 다른 프로파일과 충돌하지 않습니다. WebSphere Application Server 또는 기타 프로그램의 다른 설치에서 같은 포트를 사용할 수 있습니다. 힌타입 포트 충돌을 피하려면 각 포트 값이 고유한지 확인하십시오.

관리 콘솔 포트(기본값 9060)(A):	9060
관리 콘솔 보안 포트(기본값 9043)(G):	9043
HTTP 전송 포트(기본값 9080)(H):	9080
HTTPS 전송 포트(기본값 9443)(I):	9443
부트스트랩 포트(기본값 2809)(P):	2809
SIP 포트(기본값 5060)(U):	5060
SIP 보안 포트(기본값 5061)(U):	5061
SOAP 커넥터 포트(기본값 8880)(S):	8880
관리 프로세스 간 통신 포트(기본값 9633)(Q):	9633
SAS SSL ServerAuth 포트(기본값 9401)(L):	9401
CSV2 ServerAuth 리스너 포트(기본값 9403)(Z):	9403
CSV2 MultiAuth 리스너 포트(기본값 9402)(W):	9402
ORB 리스너 포트(기본값 9100)(Q):	9100
고가용성 관리자 통신 포트(DCS)(기본값 9353)(Y):	9353
서비스 통합 포트(기본값 7276)(E):	7276

< 이전(B) 다음(N) > 완료(F) 취소

12. 마찬가지로 Windows 서비스로 실행을 해제하고 다음을 클릭합니다.

프로파일 관리 도구 8.5
Windows 서비스 정의

WebSphere Application Server를 실행하기 위해 Windows 서비스를 사용할 것인지를 선택하십시오. Windows 서비스에서는 WebSphere Application Server를 시작 및 중지할 수 있고 시작 및 복구 조치를 구성할 수 있습니다.

☒ 애플리케이션 서버 프로세스를 Windows 서비스로 실행(B)
☐ 로컬 시스템 계정으로 로그인(L)
☐ 지정된 사용자 계정으로 로그인(G)
 사용자 이름(U): kr050578
 비밀번호(P): kr050578
 시작 유형(S): 자동
 Windows 서비스를 실행하는 사용자 계정은 다음 사용자 권한이 있어야 합니다.
 - 서비스로 로그인

< 이전(B) 다음(N) > 완료(F) 취소

13. 웹 서버 정의는 나중에 별도로 작성할 것이기 때문에 해당 설정을 해제하고 다음을 클릭합니다.

프로파일 관리 도구 8.5
웹 서버 정의

동적 콘텐츠에 대한 요청을 애플리케이션 서버로 라우트하는 데 웹 서버를 사용하는 경우에는 선택적으로 웹 서버 정의를 작성하십시오. 또는 웹 서버 플러그인 설치 중 생성되는 스크립트나 관리 콘솔에서 웹 서버 정의를 작성할 수도 있습니다.

☒ 웹 서버 정의 작성(D)
 웹 서버 유형(T): IBM HTTP Server
 웹 서버 운영 체제(O): Windows
 웹 서버 이름(S): webserver1
 웹 서버 호스트 이름 또는 IP 주소(H): kr050578
 웹 서버 포트(기본값 80)(P): 80

< 이전(B) 다음(N) > 완료(F) 취소

14. 마지막으로 profile 작성 요약을 확인하고 문제가 없으면 작성 버튼을 클릭하여 실제로 profile 을 작성합니다.

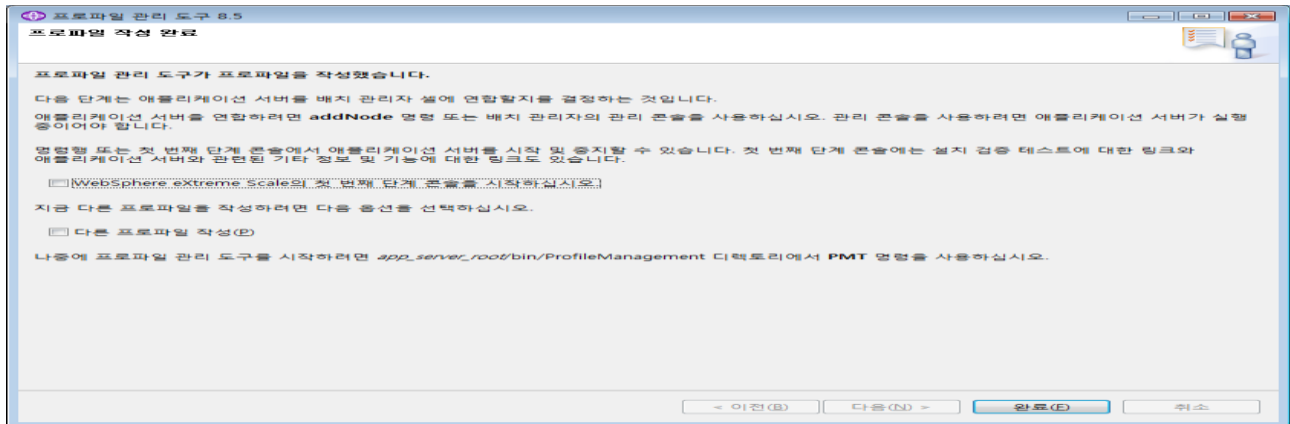
프로파일 관리 도구 8.5
프로파일 작성 요약

정정을 위해 요약에 있는 정보를 검토하십시오. 정보가 올바른 경우 작성을 클릭하여 새 프로파일 작성을 시작하십시오. 이전 패널에 있는 값을 변경하려면 이전을 클릭하십시오.

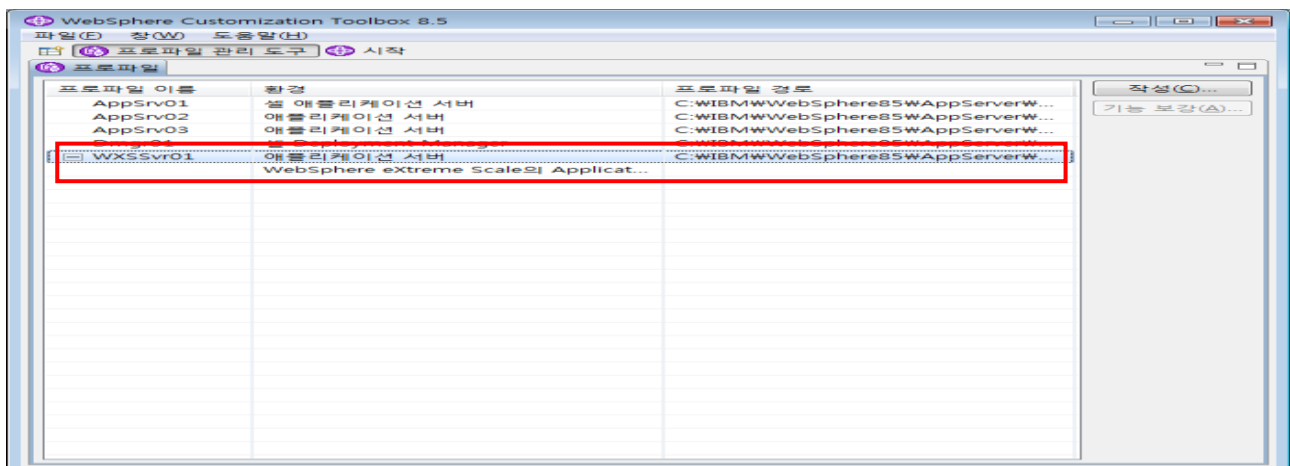
작성할 애플리케이션 서버 환경: WebSphere eXtreme Scale의 Application Server
 위치: C:\IBM\WebSphere85WAppServer\profiles\WXSSvr01
 필요한 디스크 공간: 10 MB
 프로파일 이름: WXSSvr01
 이 프로파일을 기본값으로 만들기: False
 성능 튜닝 설정: ProfileNameAndDirectoryPanel.tuningSelection.null
 노드 이름: kr050578Node05
 서버 이름: server1
 포트 이름: kr050578
 관리 콘솔 배치(광장)(A): True
 기본 애플리케이션 배치(D): True
 설치 검증 도구 애플리케이션을 배치하십시오(I): True
 관리 보안 사용(권장): False
 관리 콘솔 포트: 9060
 관리 콘솔 보안 포트: 9043
 HTTP 전송 포트: 9080

< 이전(B) 작성(B) 완료(F) 취소

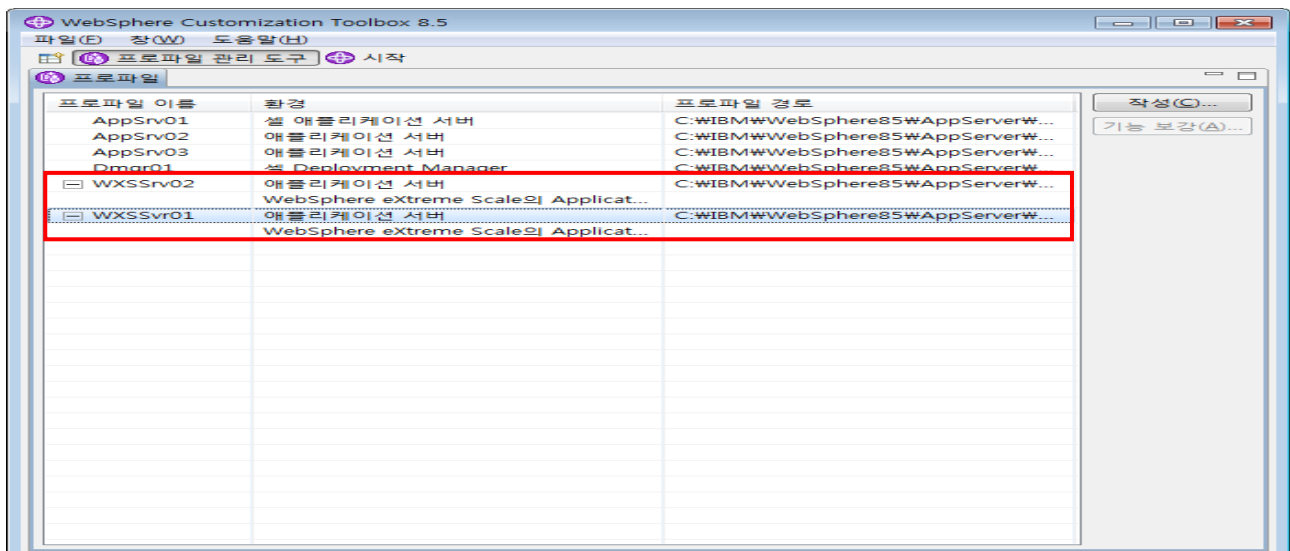
15. profile 작성이 완료되면 하단과 같은 완료 메시지를 확인할 수 있으며 완료 버튼을 클릭하여 profile 작성을 완료합니다.



16. profile 작성이 성공적으로 완료되면 하단과 같이 방금 만든 WXSSvr01 profile 을 WCT 에서 확인이 가능합니다.



17. 이미 언급한 것처럼 IBM WAS Base 두 대간의 Session Cluster 를 테스트하기 위하여 WXSSvr02 라는 단독 profile 을 위의 단계를 반복하여 하나 더 생성합니다.

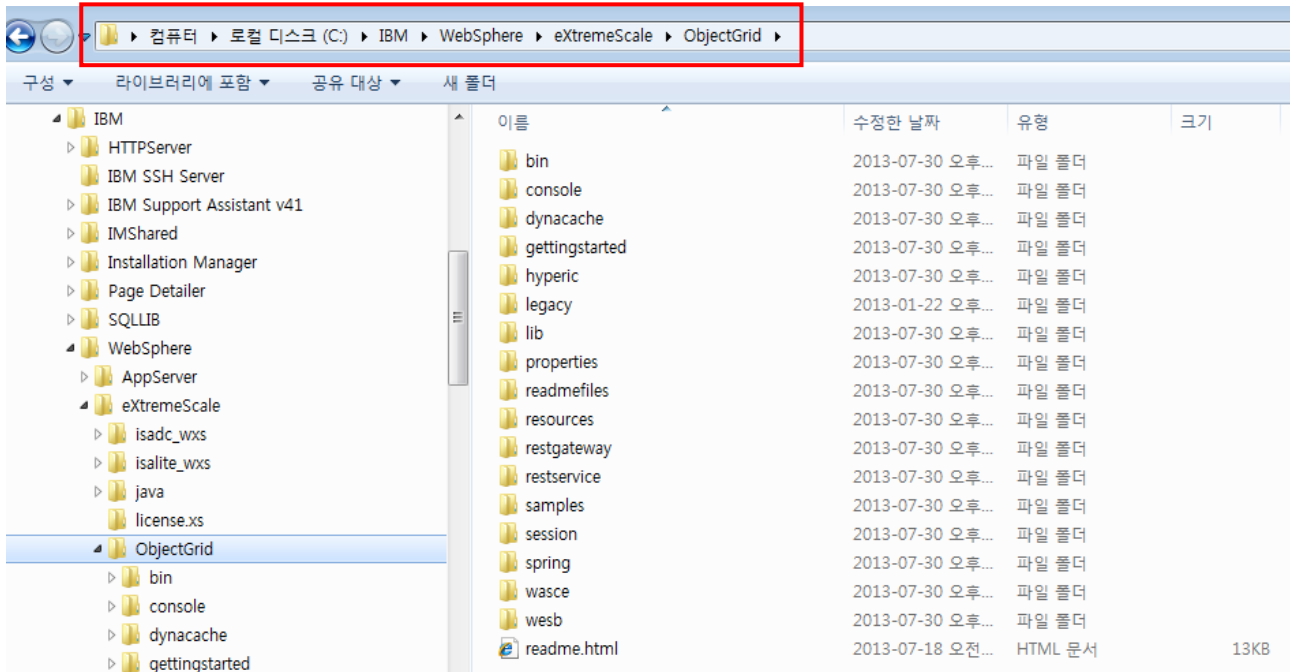


18. 이렇게 두 개의 독립적인 profile 을 생성함으로써 하나의 서버에 IBM WAS Base 가 두 대 설치된 것과 동일한 환경을 만들 수 있습니다.

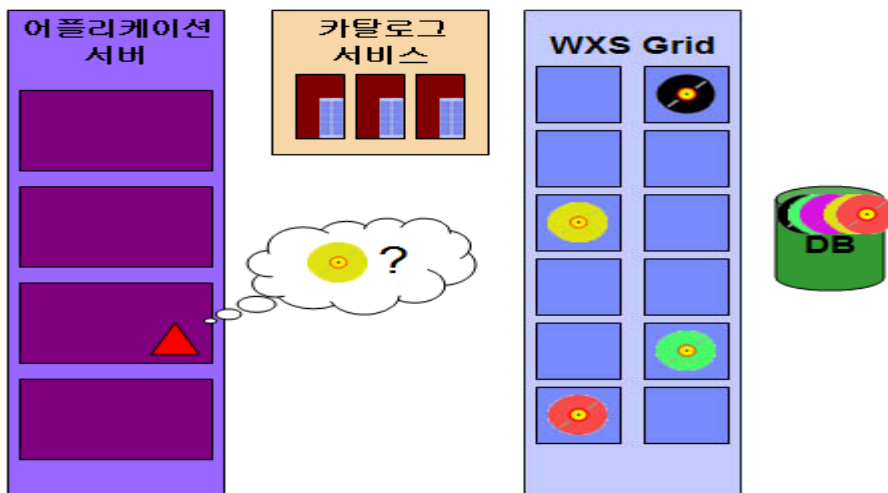
(기 언급드린 것처럼 IBM WAS 의 관리 콘솔에서 좀 더 쉽게 통합하는 형태를 굳이 원하지 않는다면 WXS 형태의 강화된 profile 을 만들 필요는 없으며 WXS client 설치 후에 그냥 일반 WAS profile 형태 그대로도 사용 가능합니다.)

2) WXS 설정 및 시작

1. 독립형 서버로 설치된 WXS 서버를 Session 저장소로 활용하기 위하여 기본 설정 및 시작 작업을 수행합니다. (일반적으로 WXS 를 독립 서버 모델로 설치하면 하단과 같은 구조를 가지게 됩니다.)



2. WXS 설정 부분을 시작하기 전에 대해서 해당 솔루션의 구조에 대해서 조금 더 자세히 설명하면 WXS 는 Catalog Server 라는 형태와 Container Server 라는 두 가지 형태로 분리됩니다. Container Server 는 실제로 Data 들이 Grid 형태로 들어가 있는 Server 를 의미하며(하단의 WXS Grid 는 이런 Container Server 의 집합을 의미) Catalog Server 는 쉽게 Master Server 처럼 분산된 Container Server 들에 어떤 Data 가 어떤 Container Server 에 있는지에 대한 index 정보와 실제 Container Server 의 분배/관리 작업을 수행합니다. (하단의 카탈로그 서비스) 일반적으로 WXS client 는 최초에 init 될 때 Catalog server 에 접속해서 WXS Grid 에 대한 index 를 가지고 오며 이 후 이 index 를 가지고 실제 필요한 Data 가 있는 Container Server 에 접속하는 형태로 진행됩니다.



3. 상단에 언급된 것처럼 먼저 Catalog Server 를 실행합니다. Catalog Server 를 실행할 경우에 별도 설정이 필요하지는 않으며 하단과 같이 listen port 만 명시해주면 됩니다.

startXSSEServer cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2899

The first screenshot shows the command prompt window with the command: `startXSSEServer cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2899`. The output displays system information including Java version (JRE 1.6.0 IBM J9 2.6 Windows 7 amd64-64 Compressed References 20130301_140166), JVM arguments, and classpath details.

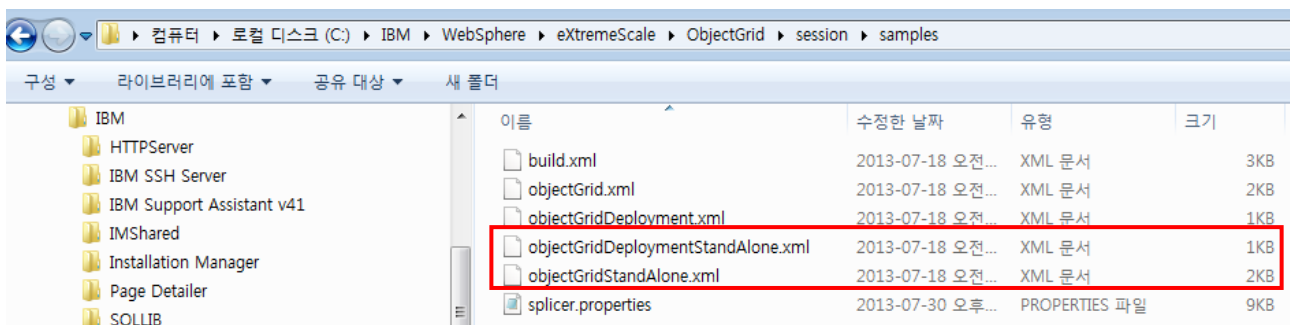
The second screenshot shows the output of the `startXSSEServer` command, displaying a list of services and their status. The output includes:

```

kr050578 이<가> 완전한 것으로 표시되지 않습니다. 다중 호스트 연결이 올바르게
작동하지 않을 수 있습니다.
[13. 7. 30 16:39:37:535 KST] 00000000 XIORegistry W CWOBJ0208W: eXtremeIO
전송은 호스트 주소 kr050578/169.254.124.34 이<가> 링크 로컬이고 적절히 작동되지
않는다는 것을 감지했습니다.
[13. 7. 30 16:39:38:420 KST] 00000000 ServerImpl I CWOBJ2518I: ObjectGrid
카탈로그 서비스 시작 중: 도메인 DefaultDomain의 cs0.
[13. 7. 30 16:39:38:423 KST] 00000000 ServerImpl I CWOBJ1252I: 카탈로그
서비스에서 쿼리를 사용할 수 없습니다.
[13. 7. 30 16:39:39:623 KST] 00000000 JvmMemoryUtil I CWOBJ4542I: 기본 Backi
ngMap 메모리 크기가 사용되었습니다.
[13. 7. 30 16:39:39:654 KST] 00000000 ServerImpl I CWOBJ3001I: ObjectGrid
: BalanceGrid 및 컨테이너 또는 서버: cs0에 대한 요청을 처리하기 위해 ObjectGrid
EntityManager 서비스를 사용할 수 있습니다.
[13. 7. 30 16:39:39:666 KST] 00000000 CatalogService I CWOBJ8263I: 관리 집선
기 MBean이 cs0 카탈로그 서버에 사용됩니다.
[13. 7. 30 16:39:39:731 KST] 00000000 CatalogService I CWOBJ8196I: 마스터 카
탈로그 서비스 클러스터가 DefaultDomain 클러스터로 활성화되었습니다.
[13. 7. 30 16:39:40:877 KST] 00000000 MBeanService I CWOBJ0068I: MBeanServe
r가 JMX URL service:jmx:rmi://169.254.124.34:1099/jndi/rmi://169.254.124.34:1099
/objectGrid/MBeanServer<으>로 시작되었습니다.
[13. 7. 30 16:39:40:903 KST] 00000000 ServerImpl I CWOBJ1001I: cs0 Object
Grid 서버가 요청을 처리할 준비가 되었습니다.
C:\IBM\WebSphere\ExtremeScale\ObjectGrid\bin>

```

4. Catalog Server 의 구동이 완료되면 이제 Container Server 를 구동하면 됩니다. Container Server 를 구동할 때는 Container Server 의 내부 구성을 어떻게 할 지에 대한 설정 정보인 objectGrid.xml, objectGridDeployment.xml 이라는 두가지 설정 파일이 필요합니다. objectGridDeployment.xml 은 Container Server 안에 어떻게 Object Grid 를 구성할 것인가(container 개수 mapset 등)에 대한 설정이며 objectGrid.xml 은 그렇게 구성된 Object Grid 에 대한 세부 설정에 관련된 파일입니다. 기본적으로 Session 저장에 대한 설정 부분은 WXS 제품 설치 디렉토리에 session > sample 디렉토리에 하단과 같이 이미 완성해 놓은 버전을 같이 제공하고 있으므로 해당 설정을 사용하면 됩니다. (본 강좌에서는 objectGridStandAlone.xml 과 objectGridDeploymentStandAlone.xml 을 사용할 예정입니다.)




```

관리자: C:\windows\system32\cmd.exe
oreGroup 코어 그룹의 DCS 코어 스택 전송이 시작되었습니다. 1 구성원이 있습니다.
[13. 7. 30 16:40:43:174 KST] 00000000 HAController I CWOBJ1710I: 독립형 고
가용성 관리자가 시작되었습니다.
[13. 7. 30 16:40:43:195 KST] 00000000 PeerManagerSe I CWOBJ7700I: 피어 관리
자 서비스가 코어 그룹(ObjectGridCoreGroup)을 가진 서버 (com.ibm.ws.objectgrid.se
rver.leader.PeerManager0550173d3)에서 시작되었습니다.
[13. 7. 30 16:40:43:196 KST] 00000000 ServerAgent I CWOBJ7800I: 코어 그룹<
ObjectGridCoreGroup>, 호스트(kr050578), 포트(56743)가 있는 ObjectGrid 고가용성
제어기를 시작합니다.
[13. 7. 30 16:40:43:753 KST] 00000002 PeerManager I CWOBJ8600I: PeerManage
r가 1 크기의 피어를 발견했습니다.
[13. 7. 30 16:40:43:760 KST] 00000002 ServerAgent I CWOBJ1772I: 고가용성(HA)
관리자 및 DCS(Distribution and Consistency Services)가 이 코어 그룹에서 실행
중인 서버 목록이 kr050578:56743(으)로 변경되었음을 extreme Scale에 알렸습니다.
[13. 7. 30 16:40:44:190 KST] 00000000 ServerAgent I CWOBJ1770I: 이 프로세
스는 이제 ObjectGridCoreGroup 코어 그룹에 대한 코어 그룹 리더입니다.
[13. 7. 30 16:40:43:790 KST] 00000003 ServerImpl I CWOBJ1773I: 현재 다음
서버를 포함한 버전 1의 카탈로그 서버로부터 이 서버가 업데이트된 고가용성(HA) 정
의 세트를 수신함: kr050578:56743
[13. 7. 30 16:40:44:787 KST] 00000000 ServerImpl I CWOBJ8000I: 영역(DefaultZone) 및 코어 그룹(DefaultZoneCG0)에 등록되었습니다.
[13. 7. 30 16:40:44:793 KST] 00000000 ServerImpl I CWOBJ1001I: c0 ObjectG
rid 서버가 요청을 처리할 준비가 되었습니다.

C:\IBM\WebSphere\ExtremeScale\ObjectGrid\bin>

```

6. 정상적으로 구동되었는지 확인하기 위해서 xsadmin 명령을 이용해서 Container Server 의 map 구성을 확인합니다. 하단과 같은 화면이 나오면 정상적으로 Container Server 구성이 완료된 것입니다.

xsadmin -g session -m sessionMapSet -mapsizes

```

관리자: C:\windows\system32\cmd.exe
C:\IBM\WebSphere\ExtremeScale\ObjectGrid\bin>xsadmin -g session -m sessionMapSet
-map sizes

This administrative utility is provided as a sample only and is not to be
considered a fully supported component of the WebSphere eXtreme Scale product

Starting at: 2013-07-30 16:41:19.000000733

Connecting to Catalog service at localhost:1099

*****Displaying Results for Grid - session, MapSet - sessionMapSet*****
*****

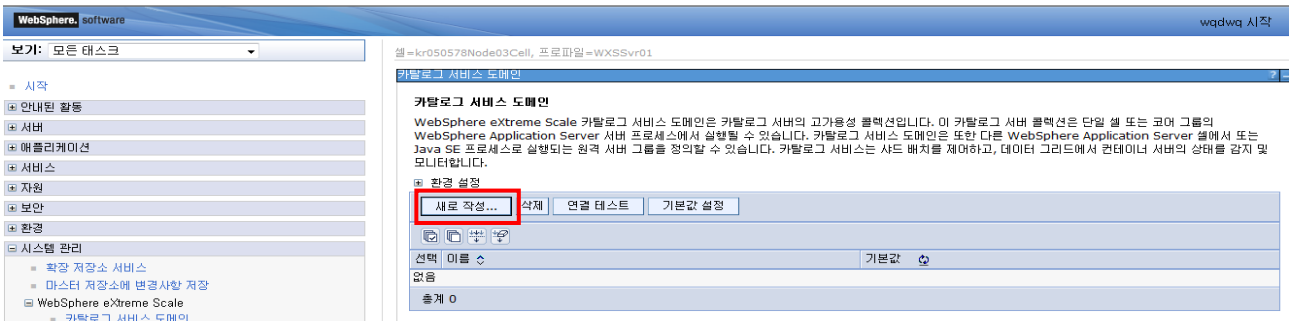
*** Listing Maps for c0 ***
Map Name Partition Map Entries Used Bytes Shard Type
objectgridSessionMetadata 0 0 0 Primary
objectgridSessionMetadata 1 0 0 Primary
objectgridSessionMetadata 2 0 0 Primary
objectgridSessionMetadata 3 0 0 Primary
objectgridSessionMetadata 4 0 0 Primary
objectgridSessionMetadata 5 0 0 Primary
objectgridSessionMetadata 6 0 0 Primary
objectgridSessionMetadata 7 0 0 Primary
objectgridSessionMetadata 8 0 0 Primary

```

3) WXS 연결을 위한 WAS 설정

0. WAS profile 을 WXS 형태로 강화하지 않은 경우 이번에 설명드릴 카탈로그 서비스 연결이 같은 WAS 에서 WXS 연결을 위한 메뉴가 생성되지 않습니다. WXS 강화를 하지 않고 WXS client 만을 설치하여 WAS profile 형태를 그대로 사용하시는 분은 WAS 콘솔에서 WXS 설정하는 부분은 그냥 넘어가시면 됩니다. 그러나, 9번에 언급된 사용자 정의 옵션인 HttpSessionCloneId, HttpSessionIdReuse 입력은 반드시 수행되어야 합니다.

1. WXSSvr01, WXSSvr02 서버에서 하나의 WAS 를 시작한 후 WXS 연결 설정을 하기 위하여 관리콘솔에서 시스템 관리 > WebSphere eXtreme Scale > 카탈로그 서비스 도메인 메뉴를 클릭한 후 새로작성을 클릭합니다.



2. 새로 작성 메뉴가 나오면 카탈로그 서비스 이름을 지정하고 이전 파트에서 생성한 Catalog Server 의 위치와 리스너 포트를 입력하고 확인을 클릭합니다.

카탈로그 서비스 도메인 > 새로 작성...

WebSphere eXtreme Scale 카탈로그 서비스 도메인은 카탈로그 서버의 고가용성 플렉션입니다. 이 카탈로그 서버 플렉션은 단일 셀 또는 코어 그룹의 WebSphere Application Server 서버 프로세스에서 실행될 수 있습니다. 카탈로그 서비스 도메인은 또한 다른 WebSphere Application Server 셀에서 또는 Java SE 프로세스로 실행되는 원격 서버 그룹을 정의할 수 있습니다. 카탈로그 서비스는 사드 배치를 제어하고, 데이터 그리드에서 컨테이너 서버의 상태를 감지 및 모니터링합니다.

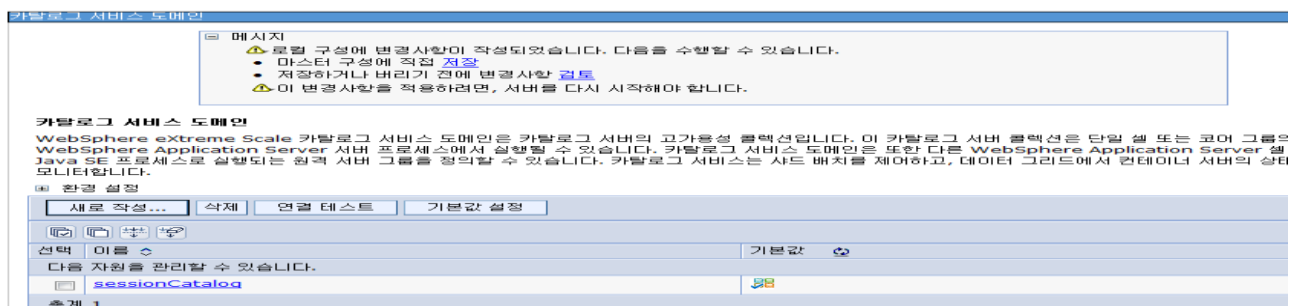
일반 특성																						
이름 <input type="text" value="sessionCatalog"/>																						
<input checked="" type="checkbox"/> 다른 카탈로그 서비스 도메인이 명시적으로 지정되지 않은 경우 이 카탈로그 서비스 도메인을 기본값으로 사용합니다.																						
<input checked="" type="checkbox"/> IBM eXtremeIO(XIO) 통신 사용. 원격 서버 멤버로 구성된 도메인에는 이 구성 옵션을 적용할 수 없습니다.																						
카탈로그 서버																						
<div> <div> <div>새로 작성</div> <div>삭제</div> </div> <table border="1"> <thead> <tr> <th>선택</th> <th>이름</th> <th>클라이언트 포트</th> <th>리스너 포트</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/></td> <td>기존의 Application Server</td> <td></td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td>kr050578Node03Cell\kr050578Node05\server1</td> <td></td> <td></td> </tr> <tr> <td><input checked="" type="radio"/></td> <td>원격 서버</td> <td></td> <td>2899</td> </tr> <tr> <td></td> <td>localhost</td> <td></td> <td></td> </tr> </tbody> </table> </div>			선택	이름	클라이언트 포트	리스너 포트	<input type="radio"/>	기존의 Application Server			<input type="radio"/>	kr050578Node03Cell\kr050578Node05\server1			<input checked="" type="radio"/>	원격 서버		2899		localhost		
선택	이름	클라이언트 포트	리스너 포트																			
<input type="radio"/>	기존의 Application Server																					
<input type="radio"/>	kr050578Node03Cell\kr050578Node05\server1																					
<input checked="" type="radio"/>	원격 서버		2899																			
	localhost																					
<div> <div>적용</div> <div>확인</div> <div>재설정</div> <div>취소</div> </div>																						

이 항목에 대한 일반 특성을 적용하거나 저장하기 전에는 추가 특성이 사용 가능하지 않습니다.

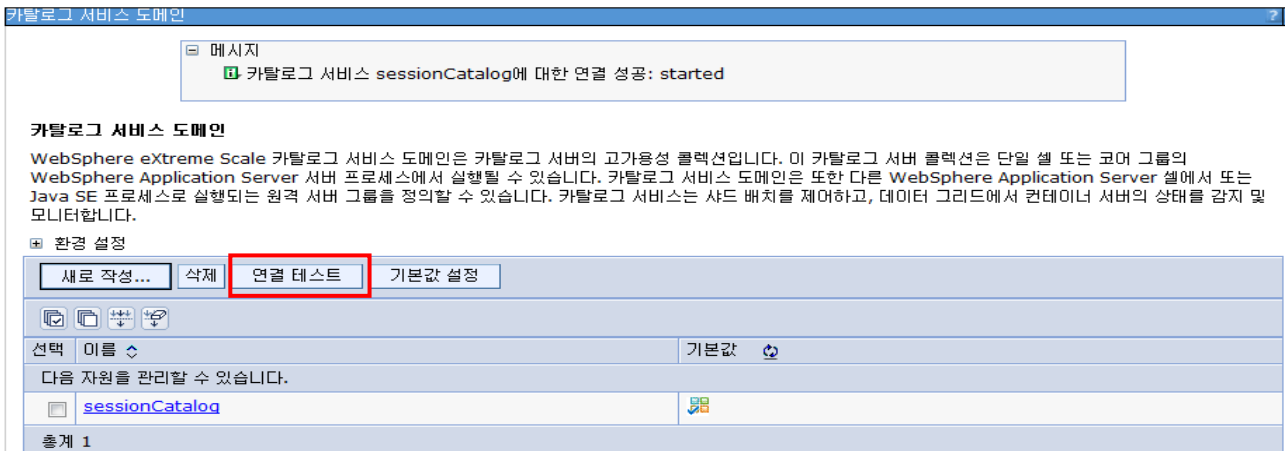
추가 특성

- 클라이언트 보안 특성
- 사용자 정의 특성

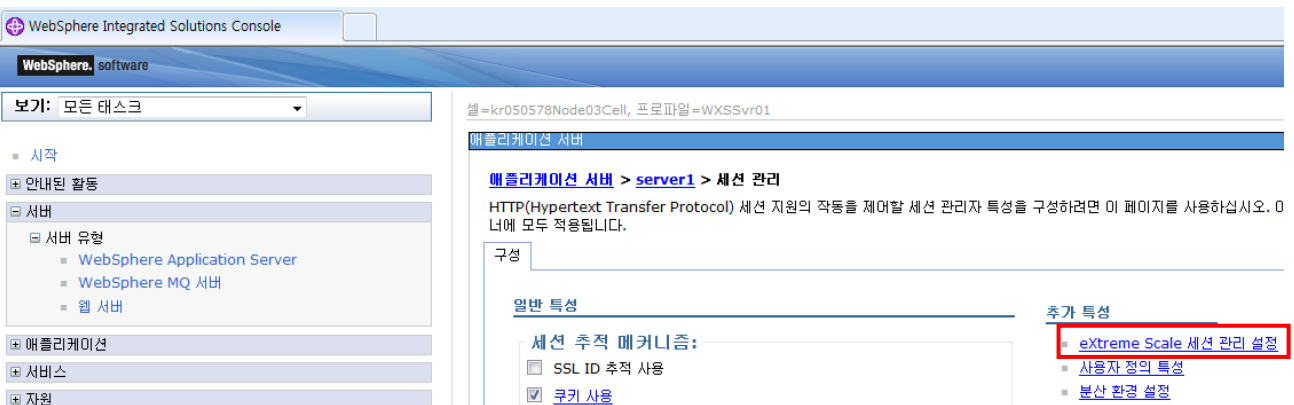
3. 해당 작업을 완료하면 하단과 같이 카탈로그 서비스 도메인이 만들어진 것을 확인할 수 있습니다.



4. 실제로 WXS 의 Catalog Server 와 정상적으로 연결되는지 확인하기 위하여 연결 테스트를 수행합니다.



5. 이제 실제로 WAS 의 Session 관리 부분을 WXS 를 이용해서 하기 위하여 관리 콘솔에서 서버 > WebSphere Application Server > 해당 서버 이름 > 세션 관리 메뉴를 클릭한 후 eXtreme Scale 세션 관리 설정 메뉴를 클릭합니다.

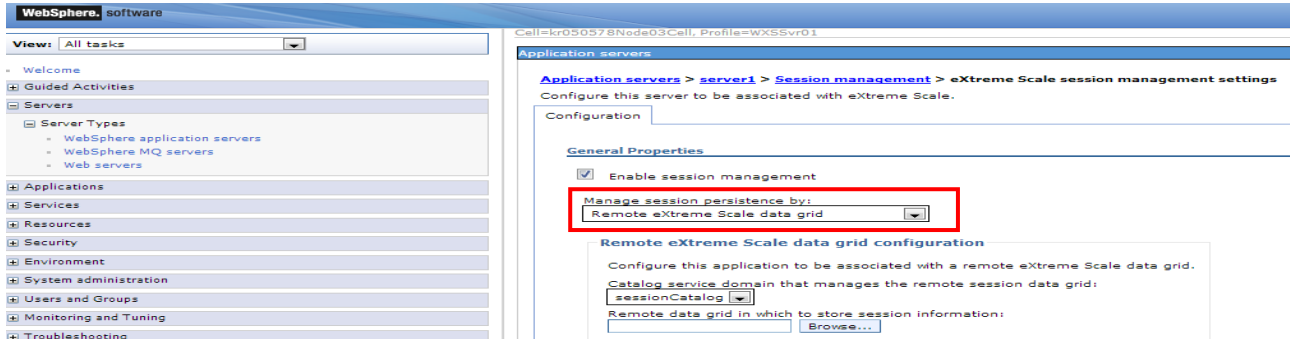


6. 세부 메뉴가 나오면 기본적으로 해제되어 있는 세션 관리 사용을 체크합니다.

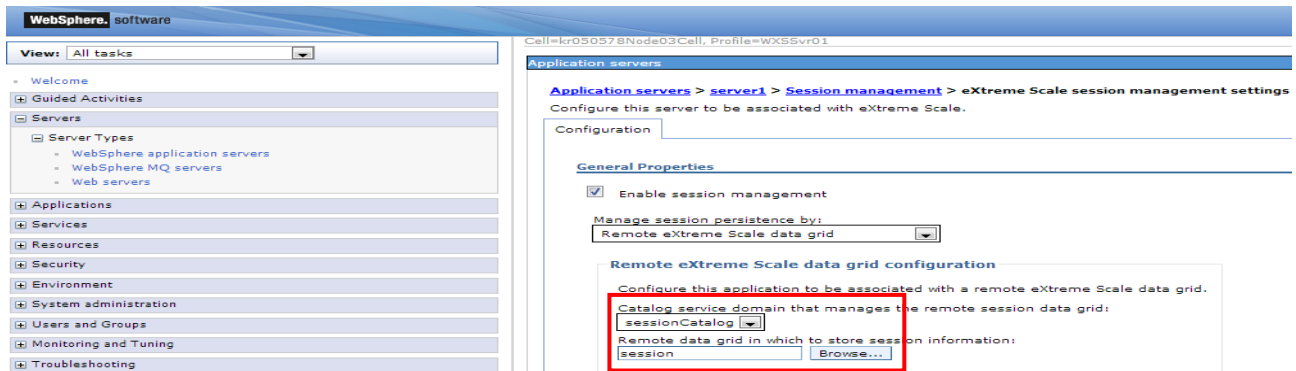


7. 본 강좌에서는 같은 서버에 있지만 instance 가 분리되어 있는 WXS 서버를 Session 저장 목적으로 사용할 것이기 때문에 두 번째에 있는 Remote eXtreme Scale data grid 를 선택합니다.

(현 버전에 해당 메뉴 부분의 한글 표시 이슈가 있어서 영문으로 화면을 캡처했습니다.)



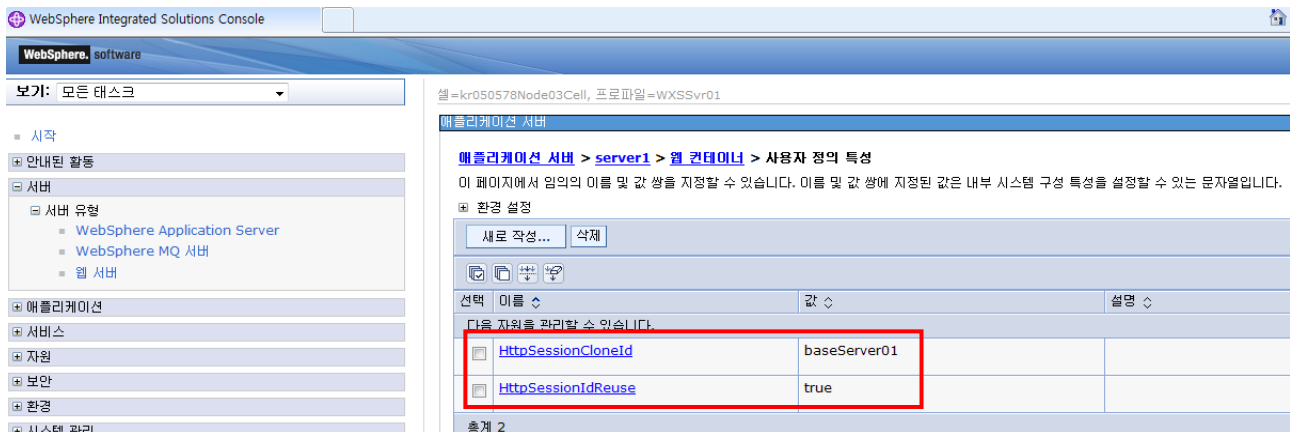
8. 이전에 만든 카탈로그 서비스 도메인("sessionCatalog")을 선택하고 브라우저를 클릭하면 Session 정보를 저장할 map 을 선택하는 화면이 나오는데 session 을 선택하여 입력하면 기본적인 Session 관리를 위한 WXS 설정을 완료된 것입니다.



9. 추가적으로 IBM WAS Base 버전의 Session Cluster 구축시에 각 WAS 서버 구분을 위해서 하단과 같은 옵션을 서버 > WebSphere Application Server > 해당 서버 > 웹 컨테이너 > 사용자 정의 특성에 추가합니다.(해당 옵션은 WXS server 를 설치했거나 client 를 설치와 상관없이 동일 적용되어야 합니다.)

HttpSessionCloneId = "구분 되는 이름" – Session Affinity 를 위한 구분되는 이름을 넣는 옵션

HttpSessionIdReuse = "true" – 두 대의 WAS 서버를 교차할 때도 SessionId 를 재사용하는 옵션



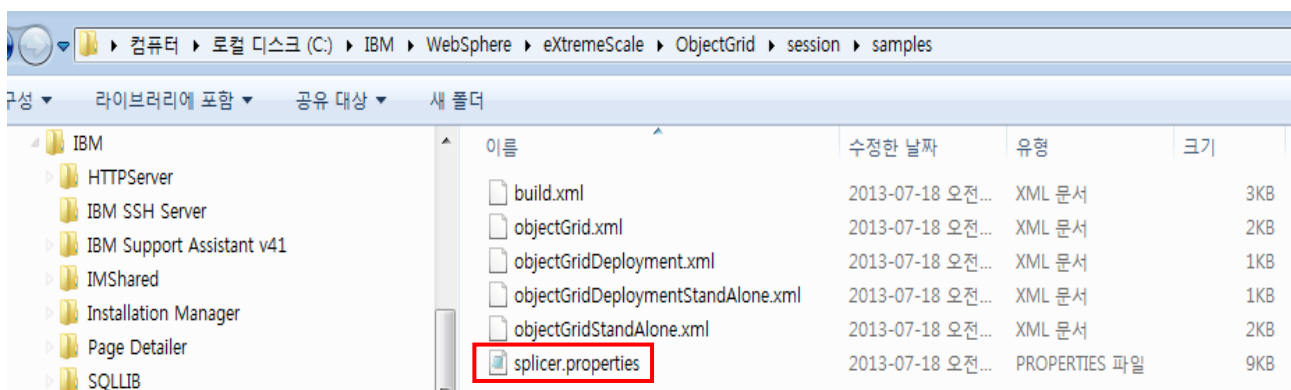
4) 테스트를 위한 샘플 애플리케이션 작성

1. 테스트를 위한 샘플 애플리케이션을 작성하기 위해서 session 을 사용하는 JSP 를 하단과 같이 하나 작성합니다.



```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<meta name="GENERATOR" content="Rational Software Architect">
</head>
<body><p>Your session details:</p>
<table border="1">
  <tbody>
    <tr>
      <td>Session Id:</td>
      <td><%=session.getId() %></td>
    </tr>
    <tr>
      <td>Session creation date:</td>
      <td><%=new Date(session.getCreationTime())%></td>
    </tr>
    <tr>
      <td>Last time session accessed:</td>
      <td><%=new Date(session.getLastAccessedTime())%></td>
    </tr>
    <tr>
      <td>Server Name:</td>
      <td><%= com.ibm.websphere.runtime.ServerName.getFullName() %></td>
    </tr>
    <tr>
      <td>Any data:</td>
      <td><%= (session.getAttribute("DATA")!=null)?session.getAttribute("DATA"):"&nbsp;" %></td>
    </tr>
  </tbody>
</table>
```

2. EAR 형태로 샘플 애플리케이션을 export 한 후 WXS 를 Session 저장소로 활용하기 위해서는 추가적으로 web.xml 설정에 대한 변경 작업이 필요합니다. (WXS 를 Session 으로 사용한다고 하여도 개발 소스의 변경은 필요하지 않으며 web.xml 에 단순히 filter 와 설정만 추가되면 됩니다.) 이를 자동으로 수행하기 위해서 “C:\IBM\WebSphere\eXtremeScale\ObjectGrid\session\samples” 폴더에 splicer.properties 파일을 활용합니다.



3. 일반적으로 splicer.properties 가 권고 값들이므로 대부분 그대로 사용하면 되며 reuseSessionId=true 부분의 '#' 을 제거하여 enable 해야 합니다.

```

142 # session ID's across requests to different hosts. Default is
143 # false. The value of this should be the same as what is set
144 # in the web container.
145 reuseSessionId=true
146 # A string value of either "true" or "false". The default is
147 # "false". Per the servlet specification, HTTP Sessions cannot
148 # be shared across web applications. An extension to the servlet
149 # specification is provided to allow this sharing.
150 shareSessionsAcrossWebApps = false
151 # Set to true if you want to enable urlRewriting. Default is
152 # false. The value of this should reflect what is set in the
153 # web container settings for session management.
154 useURLEncoding = false

```

대부분 옵션에 대한 설명이 같이 나와있지만 몇 가지만 간단히 정리 드리자면

- **catalogHostPort** = localhost:2809 (카탈로그 서비스가 수행되는 호스트 명과 port)

관리콘솔에서 카탈로그 서비스에 대한 설정을 이미 수행했다면 해당 부분은 '#' 으로 주석으로 되어 있는 것을 그대로 두면 되며, 위 강좌에 언급된 관리콘솔에서 카탈로그 서비스 설정을 하지 않았다면(WXS client 만 설치된 IBM WAS profile 을 그냥 쓰는 경우) 해당 주석을 풀어주어 카탈로그 서비스의 위치를 명시해주면 됩니다.

- **fragmentedSession** = true

true 인 경우는 session.setAttribute 가 호출될 때만 해당 attribute 가 복제되며 해당 값이 false 면 getSession 이 호출될 때마다 전체 session data 가 복제됩니다.

- **replicationInterval** = 10

Session 복제가 비동기식으로 10초 마다 수행됨. 값이 0인 경우 Servlet의 종료시점에 동기식으로 복제되기 때문에 성능저하 발생될 수 있습니다만 session 이 반드시 유지될 경우라면 0 설정도 가능합니다.

- **sessionTableSize** = 10000

WAS 에 설정된 session table size 최대값보다 큰 수 지정

4. WXS 가 제공하는 addObjectGridFilter 스크립트를 이용해서 자동으로 EAR 파일의 web.xml 변경작업을 수행합니다.

C:\IBM\WebSphere\WeXtremeScale\ObjectGrid\session\bin\addObjectGridFilter .\MySessionTest.ear

C:\IBM\WebSphere\WeXtremeScale\ObjectGrid\session\samples\splicer.properties

```

CWWSM0027I: .war 파일 처리: MyWebApp
CWWSM0028I: 컨텍스트 매개변수는 다음과 같습니다.
CWWSM0029I: 컨텍스트 이름: securityEnabled 값: false
CWWSM0029I: 컨텍스트 이름: fragmentedSession 값: true
CWWSM0029I: 컨텍스트 이름: sessionTableSize 값: 1000
CWWSM0029I: 컨텍스트 이름: objectGridType 값: REMOTE
CWWSM0029I: 컨텍스트 이름: reuseSessionId 값: true
CWWSM0029I: 컨텍스트 이름: replicationInterval 값: 10
CWWSM0029I: 컨텍스트 이름: objectGridName 값: session
CWWSM0030I: 응용프로그램 결함이 완료되었습니다.
D:\W00.biz_work\01.SVGW09.SampleSource\session>

```

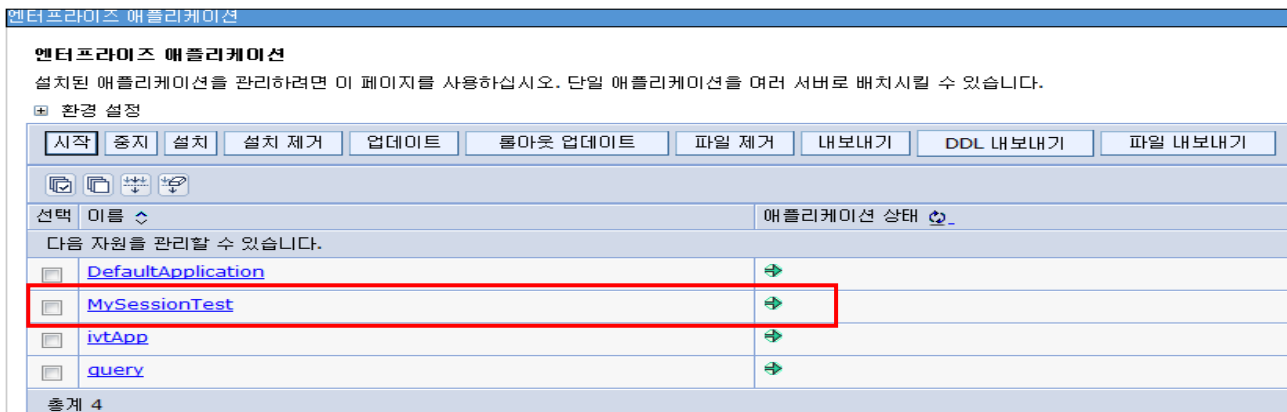
5. 해당 작업이 완료되면 EAR 안의 web.xml 에 하단과 같은 WXS 정보들이 추가된 것을 확인할 수 있습니다 .

```

0 10 20 30 40 50 60 70 80 90
22 </context-param>
23 <context-param>
24     <param-name>replicationInterval</param-name>
25     <param-value>10</param-value>
26 </context-param>
27 <context-param>
28     <param-name>objectGridName</param-name>
29     <param-value>session</param-value>
30 </context-param>
31 <filter>
32     <description>Filter that provides for an ObjectGrid based Session Manager.</description>
33     <display-name>HttpSessionFilter</display-name>
34     <filter-name>HttpSessionFilter</filter-name>
35     <filter-class>com.ibm.ws.xs.sessionmanager.HttpSessionFilter</filter-class>
36 </filter>
37 <filter-mapping>
38     <filter-name>HttpSessionFilter</filter-name>
39     <url-pattern>*</url-pattern>
40     <dispatcher>REQUEST</dispatcher>
41 </filter-mapping>
42 <listener>
43     <description>IBMSessionListener</description>
44     <display-name>IBMSessionListener</display-name>
45     <listener-class>com.ibm.ws.xs.sessionmanager.IBMHttpSessionListener</listener-class>
46 </listener>

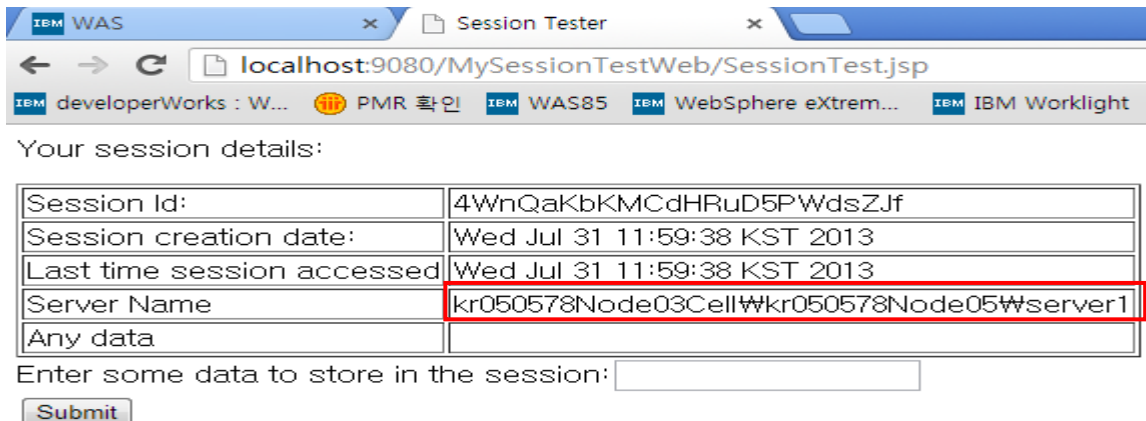
```

6. 해당 샘플 애플리케이션을 만들어둔 WAS 에 배포합니다.



7. 샘플 애플리케이션이 정상적으로 수행되는지 테스트를 수행합니다.

(http://localhost:9080/MySessionTestWeb/SessionTest.jsp)



8. 실제로 Session 에 특정 data 를 저장하는 테스트도 같이 수행합니다.

Your session details:

Session Id:	4WnQaKbKMCdHRuD5PWdsZJf
Session creation date:	Wed Jul 31 11:59:38 KST 2013
Last time session accessed:	Wed Jul 31 11:59:38 KST 2013
Server Name	kr050578Node03CellWkr050578Node05Wserver1
Any data	test01

Enter some data to store in the session:

9. 해당 테스트 완료후에 WXS 의 Container Server 에 정상적으로 Session 객체가 저장되었는지 xscmd 명령을 이용해서 확인해 봅니다.

xscmd -c findByKey -g session -m objectgridSessionMetadata -fs "*" -rv -cep localhost:2899

```
C:\W\IBM\WebSphere\WeXtremeScale\WObjectGrid\Win>xscmd -c findByKey -g session -m ob
jectgridSessionMetadata -fs "*" -rv -cep localhost:2899
시작 시간 :2013-07-31 12:01:34.476

CWXS10068I: 명령 실행 중: findByKey
1 일치 키를 찾았습니다.

파티션 키                                값
-----
25      4WnQaKbKMCdHRuD5PWdsZJf/MySessionTestWeb IBMHttpSessionData<4WnQaKbKMCdHR
uD5PWdsZJf>[sessionHandle=19, attributesSize=2, webApp=/MySessionTestWeb, lastAc
cessTime...
```

CWXS10040I: findByKey 명령이 완료되었습니다.

종료 시간: 2013-07-31 12:01:37.984

xscmd -c findByKey -g session -m objectgridSessionAttribute -fs "*" -rv -cep localhost:2899

```
C:\W\IBM\WebSphere\WeXtremeScale\WObjectGrid\Win>xscmd -c findByKey -g session -m ob
jectgridSessionAttribute -fs "*" -rv -cep localhost:2899
시작 시간 :2013-07-31 12:02:23.722

CWXS10068I: 명령 실행 중: findByKey
1 일치 키를 찾았습니다.

파티션 키                                값
-----
25      4WnQaKbKMCdHRuD5PWdsZJf/MySessionTestWeb_DATA test01

CWXS10040I: findByKey 명령이 완료되었습니다.

종료 시간: 2013-07-31 12:02:26.752
```

5) Cluster 구성을 위한 IBM HTTP Server 설정

1. IBM WAS Base 서버 두 대의 Cluster 구성을 했을 경우 부하 분산을 위한 IBM HTTP Server 설정을 하기 위하여 GenPluginCfg 라는 스크립트를 활용하여 각각의 서버에서 plugin-cfg.xml 을 생성합니다.

```
GenPluginCfg -server.name server1 -output.file.name C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg.xml
```

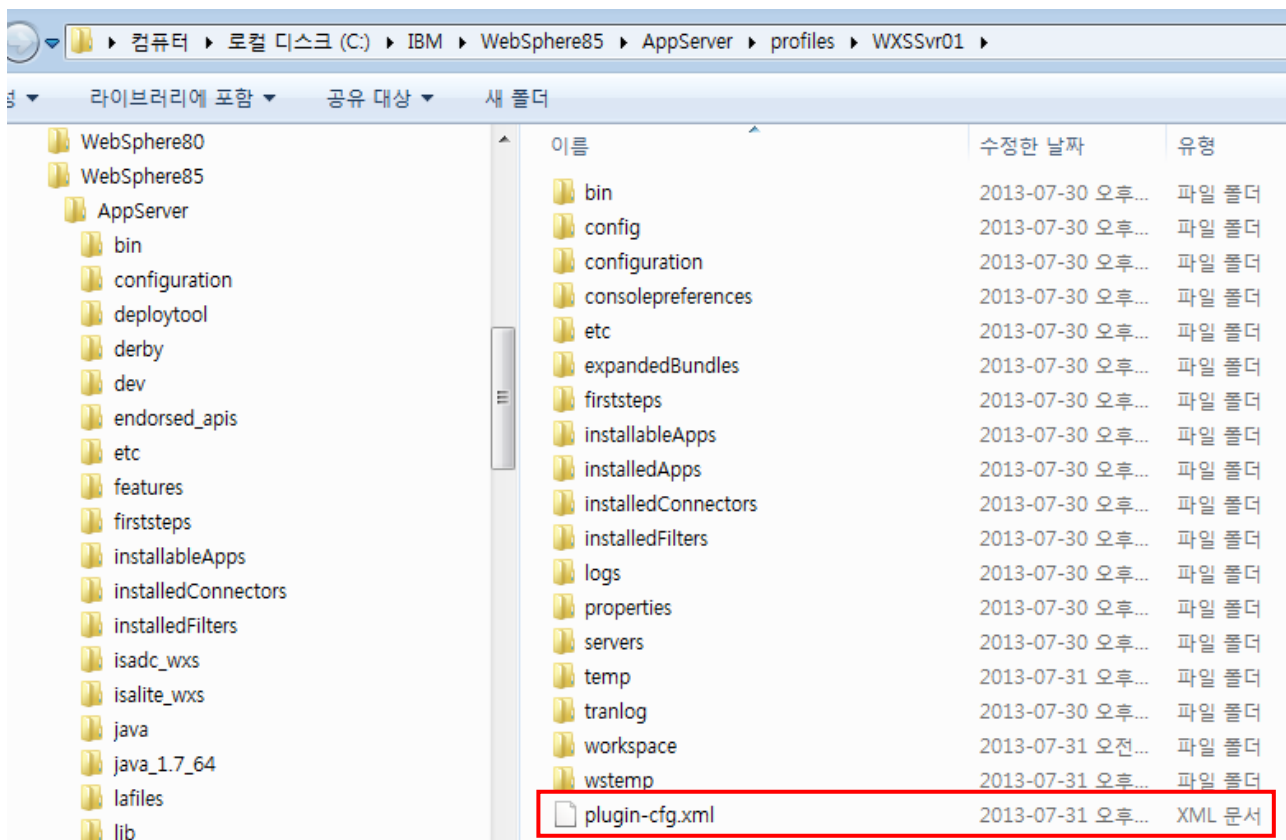
```
C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\bin>GenPluginCfg -server.name server1 -output.file.name C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg.xml

IBM WebSphere Application Server, 릴리스 8.0
WebSphere 플러그인 구성 생성 프로그램
Copyright IBM Corp., 1997-2008

PLGC0008I: 플러그인이 셀에 대한 단일 서버 플러그인 구성 파일을 생성 중입니다. kr050578Node03Cell, 노드 kr050578Node05, 서버 server1.
PLGC0005I: 플러그인 구성 파일 = C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg.xml

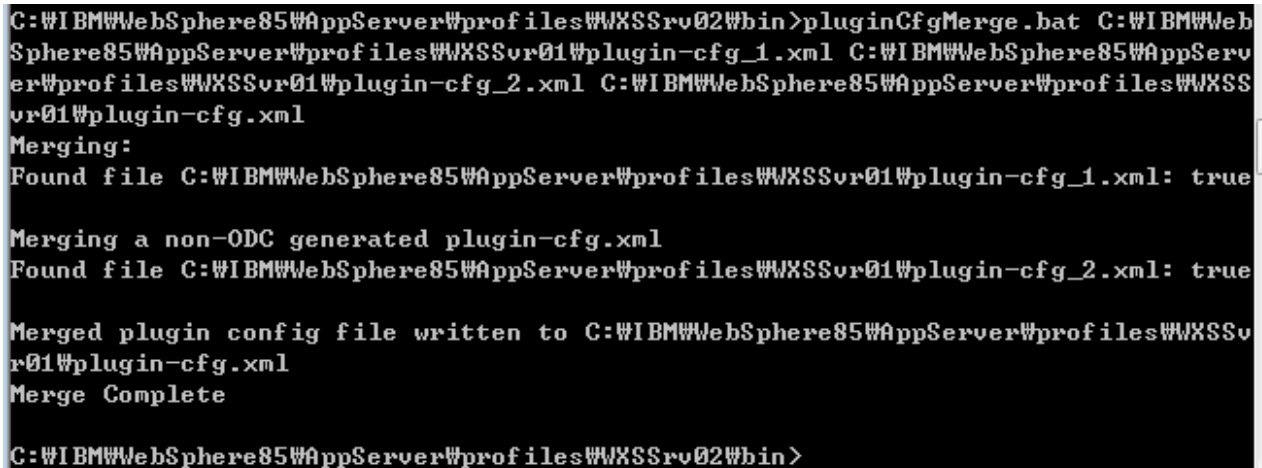
C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\bin>
```

2. 위의 작업이 정상적으로 완료되면 하단과 같이 plugin-cfg.xml 파일이 생성된 것을 볼 수 있습니다.



3. 분리된 두 대의 서버에 대한 plugin-cfg.xml 생성이 각각 완료되면 이를 하나로 합치기 위하여 pluginCfgMerge.bat 스크립트를 수행합니다.

```
pluginCfgMerge.bat C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg_1.xml
C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg_2.xml
C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg.xml
```



```
C:\IBM\WebSphere85\AppServer\profiles\WXSSvr02\bin>pluginCfgMerge.bat C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg_1.xml C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg_2.xml C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg.xml
Merging:
Found file C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg_1.xml: true
Merging a non-ODC generated plugin-cfg.xml
Found file C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg_2.xml: true
Merged plugin config file written to C:\IBM\WebSphere85\AppServer\profiles\WXSSvr01\plugin-cfg.xml
Merge Complete
C:\IBM\WebSphere85\AppServer\profiles\WXSSvr02\bin>
```

4. IBM HTTP Server 가 위에서 병합한 plugin-cfg.xml 을 볼 수 있는 위치로 해당 파일을 이동하고 반영하기 위하여 IBM HTTP Server 를 재시작합니다.

참고: <http://www-01.ibm.com/support/docview.wss?uid=swg21318463>

Session cluster 를 사용해보신 분들은 이미 잘 아시겠지만 Memory-to-memory session replication 을 사용하는 경우에는 plugin-cfg.xml 설정에 GetDWLMTable 옵션을 true 로 설정해야 합니다.

(GetDWLMTable="true")

```
<ServerCluster CloneSeparatorChange="false" GetDWLMTable="true" IgnoreAffinityRequests="false"
  <Server CloneID="17f69k519" ConnectTimeout="5" ExtendedHandshake="false" LoadBalanceWeight=
    <Transport Hostname="kr050578" Port="9083" Protocol="http"/>
    <Transport Hostname="kr050578" Port="9446" Protocol="https">
      <Property Name="keyring" Value="C:\IBM\WebSphere85\Plugins\config\webserver1\plugin-k
      <Property Name="stashfile" Value="C:\IBM\WebSphere85\Plugins\config\webserver1\plugin
    </Transport>
  </Server>
  <Server CloneID="17f69k8ot" ConnectTimeout="5" ExtendedHandshake="false" LoadBalanceWeight=
    <Transport Hostname="kr050578" Port="9084" Protocol="http"/>
    <Transport Hostname="kr050578" Port="9447" Protocol="https">
      <Property Name="keyring" Value="C:\IBM\WebSphere85\Plugins\config\webserver1\plugin-k
      <Property Name="stashfile" Value="C:\IBM\WebSphere85\Plugins\config\webserver1\plugin
    </Transport>
  </Server>
  <PrimaryServers>
    <Server Name="kr050578Node01_Server01"/>
    <Server Name="kr050578Node04_Server02"/>
  </PrimaryServers>
</ServerCluster>
```

6) WXS 를 이용한 Session Cluster 테스트

1. 이전 파트까지 모든 설정과 준비가 완료되었기 때문에 실제 테스트를 위하여 샘플 요청을 IBM HTTP Server 를 통해서 수행하고 생성된 Session Id 와 Server Name 을 확인합니다. – server1 에서 서비스 (<http://localhost/MySessionTestWeb/SessionTest.jsp>)

IBM WAS x Session Tester x

localhost/MySessionTestWeb/SessionTest.jsp

IBM developerWorks : W... PMR 확인 IBM WAS85 IBM WebSphere eXtrem... IBM IBM Worklight

Your session details:

Session Id:	4WnQaKbKMCdHRuD5PWdsZJf
Session creation date:	Wed Jul 31 13:00:09 KST 2013
Last time session accessed:	Wed Jul 31 13:00:09 KST 2013
Server Name	kr050578Node03CellWkr050578Node05Wserver1
Any data	

Enter some data to store in the session:

2. Session 테스트를 위해서 'test02' 라는 값을 Session 에 저장합니다.

IBM WAS x Session Tester x

localhost/MySessionTestWeb/SessionTest.jsp

IBM developerWorks : W... PMR 확인 IBM WAS85 IBM WebSphere eXtrem... IBM IBM Worklight

Your session details:

Session Id:	4WnQaKbKMCdHRuD5PWdsZJf
Session creation date:	Wed Jul 31 13:00:09 KST 2013
Last time session accessed:	Wed Jul 31 13:00:39 KST 2013
Server Name	kr050578Node03CellWkr050578Node05Wserver1
Any data	test02

Enter some data to store in the session:

3. 장애 상황을 가정하고 지금 Session 이 맺혀진 server1 서버를 종료합니다.

```
C:\IBM\WebSphere85\AppServer\profiles\WXSrv01\bin>stopserver server1
ADMU0116I: 도구 정보
C:\IBM\WebSphere85\AppServer\profiles\WXSrv01\logs\server1\stopServer.log
파일에 로그 중입니다.
ADMU0128I: WXSrv01 프로파일로 도구 시작 중
ADMU3100I: 다음 서버에 대한 구성을 읽는 중: server1
ADMU3201I: 서버 중지 요청이 발행되었습니다. 중지 상태 대기 중입니다.
ADMU4000I: server1 서버 중지가 완료되었습니다.
```


4. 동일한 요청을 다시 보내면 정상적으로 fail-over 되어서 server2 가 해당 요청을 처리한다는 것을 확인할 수 있습니다. 뿐만 아니라, 이전 요청과 동일한 Session Id 와 Session 데이터 값('test02') 이 해당 서버가 죽어도 소실되지 않고 유지되는 것을 확인할 수 있습니다.

IBM WAS Session Tester

localhost/MySessionTestWeb/SessionTest.jsp

developerWorks : W... PMR 확인 WAS85 WebSphere eXtrem... IBM Worklight

Your session details:

Session Id:	4WnQaKbKMCdHRuD5PWdsZJf
Session creation date:	Wed Jul 31 13:00:09 KST 2013
Last time session accessed:	Wed Jul 31 13:01:48 KST 2013
Server Name	kr050578Node04CellWkr050578Node06Wserver2
Any data	test02

Enter some data to store in the session:

5. 장애 복구 상황을 가정하여 이전에 중지한 server1 서버를 다시 시작합니다.

```
C:\IBM\WebSphere85\AppServer\profiles\WXSrv01\bin>startserver server1
ADMU0116I: 도구 정보가
C:\IBM\WebSphere85\AppServer\profiles\WXSrv01\logs\server1\startServer.log
파일에 로그 중입니다.
ADMU0128I: WXSrv01 프로파일로 도구 시작 중
ADMU3100I: 다음 서버에 대한 구성을 읽는 중: server1
ADMU3200I: 서버가 실행되었습니다. 초기화 상태 대기 중입니다.
ADMU3000I: server1 서버가 e-business 용으로 열렸습니다. 프로세스 ID: 15132
C:\IBM\WebSphere85\AppServer\profiles\WXSrv01\bin>
```

6. 해당 서버 재시작 후 동일한 요청을 다시 보내면 fail-back 이 수행되어 이전에 처리하던 server1 에서 요청이 처리되는 것을 확인할 수 있습니다.

IBM WAS Session Tester

localhost/MySessionTestWeb/SessionTest.jsp

developerWorks : W... PMR 확인 WAS85 WebSphere eXtrem... IBM Worklight

Your session details:

Session Id:	4WnQaKbKMCdHRuD5PWdsZJf
Session creation date:	Wed Jul 31 13:00:09 KST 2013
Last time session accessed:	Wed Jul 31 13:04:34 KST 2013
Server Name	kr050578Node03CellWkr050578Node05Wserver1
Any data	test02

Enter some data to store in the session:

7) 결론

이번 강좌에서는 IBM WAS v8.5.5 에서 라이선스 적으로 통합된 WXS 를 이용해서 IBM WAS Base 버전과 WXS 를 이용한 Session Cluster 구축에 대하여 살펴보는 시간을 가져봤습니다. 이전까지 IBM WAS Base 버전을 이용해서 제한된 범위에서 Cluster 구축이 가능하긴 했지만 fail-over 만 지원했고 Session Persistence 는 DB 를 활용하는 경우만 지원하였기 때문에 많이 활용되는 경우가 드물었습니다.

그러나, IBM WAS v8.5.5 부터는 Base 버전이라고 해도 WXS 라이선스가 Session Cluster 용으로 통합 제공되므로 이번 강좌에서 맛 보신것과 같이 두 대 이상의 IBM WAS Base 버전을 사용한다고 해도 완벽한 Session Cluster 를 구축할 수 있습니다. 결국, 이를 통해 Base 버전이라고 해도 점점 더 다양한 용도와 안정성이 필요한 시스템에 사용될 수 있는 길이 열렸다고 생각합니다.

여기서는 WXS 의 Session 저장 목적 부분만 강조했지만 해당 제품은 맨 처음에 언급된 것처럼 In-Memory Data Grid 솔루션으로 해당 기능 말고도 다양한 기능들이 더 있습니다. 해당하는 좋은 기능은 향후 다른 강좌에서 살펴볼 수 있게 되길 바라며 이번 강좌는 마무리하고 다음 강좌에서 뵈도록 하겠습니다. Go Go !!!

9) 참고 자료

1. 이 가이드는 IBM WAS v8.5.5 최초 사용자를 위한 기본 가이드입니다.
2. IBM WAS 자체에 아직 익숙하지 않으신 분들은 가급적 기본강좌인 '하나씩 쉽게 따라 해보는 IBM WAS v7' 강좌와 '제대로 맛보는 IBM WAS v8.5' 강좌를 먼저 읽고 이 강좌를 읽으시는 것이 이해에 훨씬 도움이 됩니다.
(http://www.websphere.pe.kr/xe/?mid=was_info_re&page=3&document_srl=800
http://www.websphere.pe.kr/xe/?mid=was_info_re&page=2&document_srl=134863)
3. 가급적 IBM WAS v8.5 InfoCenter 의 해당 카테고리를 한 번 읽어보고 난 후에 작업하시기 바랍니다.
4. InfoCenter – WebSphere Application Server v8.5
(<http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp>)
5. InfoCenter – WebSphere eXtreme Scale Version 8.6
(http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/index.jsp?topic=%2Fcom.ibm.websphere.extreme.scale.doc%2Fwelcome%2Fwelcome_xs.html)
6. InfoCenter – HTTP session management
(<http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/index.jsp?topic=%2Fcom.ibm.websphere.extreme.scale.doc%2Fcxshhttpsession.html>)
7. Deep dive into WebSphere eXtreme Scale HTTP session management, Part 1: Understanding HTTP session management and how it works in WebSphere eXtreme Scale
(http://www.ibm.com/developerworks/websphere/techjournal/1301_ying/1301_ying.html)