

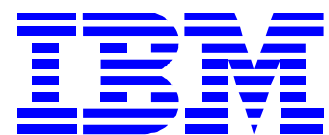
WebSphere Application Server v8.0

Multi-DataSource Failover

(2011. 07.)

IBM SWG WebSphere CTP

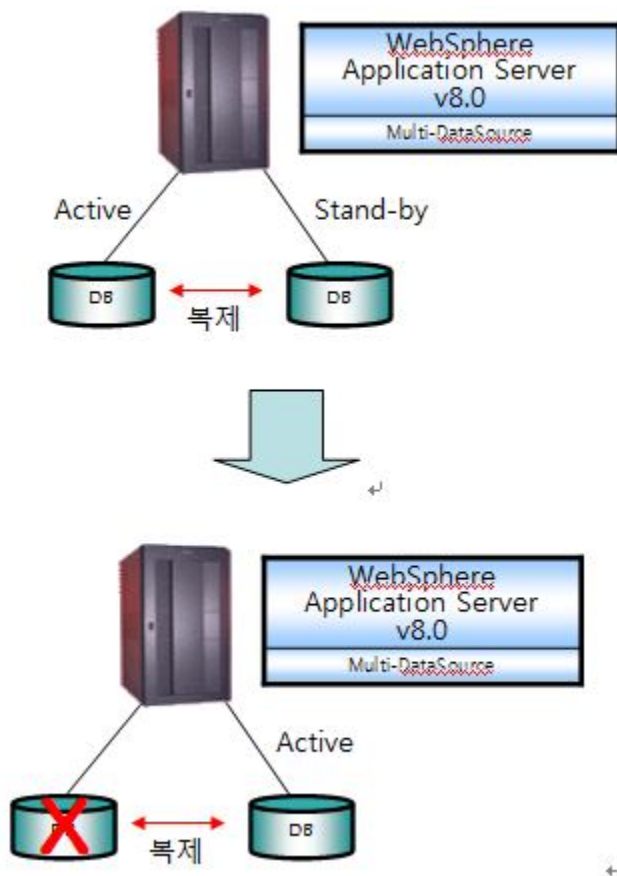
이정운 과장(juwlee@kr.ibm.com)



0) Multi-DataSource Failover 란?

안녕하세요! Freeman 입니다. ^^&

이번에 시작하는 '먼저 해보는 IBM WAS v8.0' 시리즈의 다섯 번째 강좌는 " Multi-DataSource Failover" 입니다. Multi-DataSource Failover 란 말 그대로 어플리케이션에서 실제로 DB 에 연결되는 DataSource 를 지정할 때 해당 DB 에 대한 작업을 수행하는 것에 추가하여 별도로 대체 DataSource 를 미리 지정할 수 있는 기능입니다. 이 기능을 활용하면 해당 DataSource 에 연결된 DB 부분에 문제가 있을 때 어플리케이션 코딩 없이 자동으로 원래의 DataSource 가 아니라 기 지정된 대체 Datasource 를 활용할 수 있습니다. 즉, 장애 시점에 Datasource 를 활용하여 DB 를 스위치하여 문제가 없는 백업 DB 를 활용할 수 있는 방안을 제공할 수 있습니다.



Oracle RAC 와 같이 DB 가 자체적인 고가용성(High Availability) 기능을 제공하는 경우를 제외하고 DB 에서 자체적인 고가용성을 제공하지 못할 경우에 요건에 따라서 프로그래밍 코드가 아닌 WAS 를 통해서 고가용성을 제공해 줄 수 있는 유용한 방안입니다.

참고적으로 이번 강좌에서 사용하는 테스트 DB 는 IBM DB2 로 진행하도록 하겠습니다.

1) 사전 DB2 준비

0. DB 의 경우에는 로컬에 DB2 v9.7 Express-C 버전이 깔려있는 것을 가정으로 하겠습니다. DB2 v9.7 Express-C 은 무료로 하단의 링크된 사이트에서 다운로드 받을 수 있습니다.

(<http://www.ibm.com/developerworks/downloads/im/udbexp/>)

1. DB2 CMD 에 접속하여 하단과 같이 sample 로 사용할 table 을 하나 작성합니다.

```
db2 => create table TEST(ID integer generated always as identity (start with 1,
increment by 1, no cache ), NAME varchar(15))
DB20000I  SQL 명령이 완료되었습니다.
db2 => select * from TEST

ID          NAME
-----
0 레코드가 선택되었습니다.

db2 =>
```

2. table 이 작성되면 체크를 위해서 sample 값을 insert 합니다.

```
db2 => insert into TEST(NAME) values ('TESTer01')
DB20000I  SQL 명령이 완료되었습니다.
db2 => select * from TEST

ID          NAME
-----
1 TESTer01
1 레코드가 선택되었습니다.

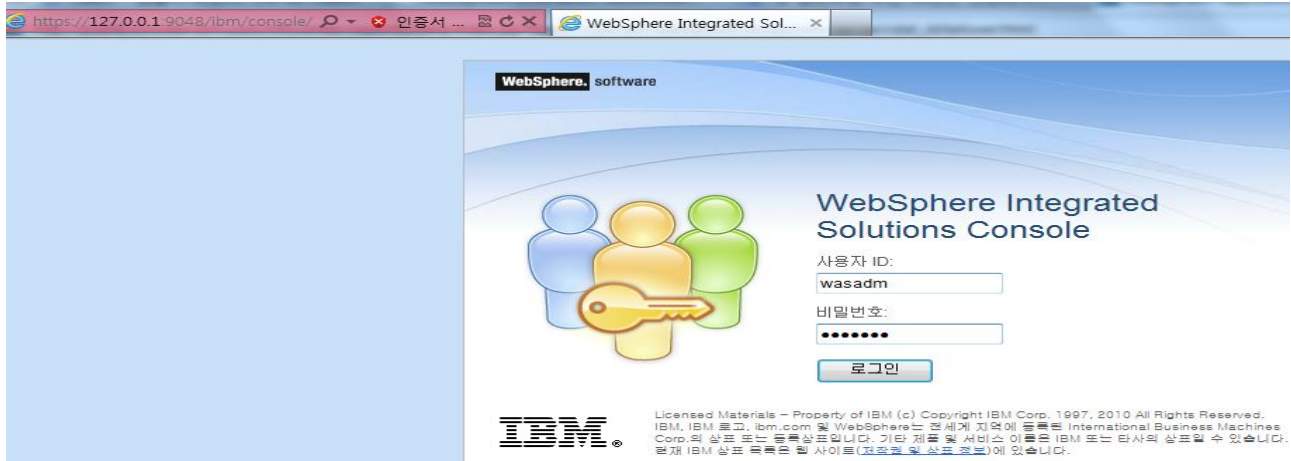
db2 => _
```

3. 변경 사항의 반영을 위하여 commit 을 수행합니다.

```
db2 => commit
DB20000I  SQL 명령이 완료되었습니다.
```

2) WAS 에서 DataSource 설정

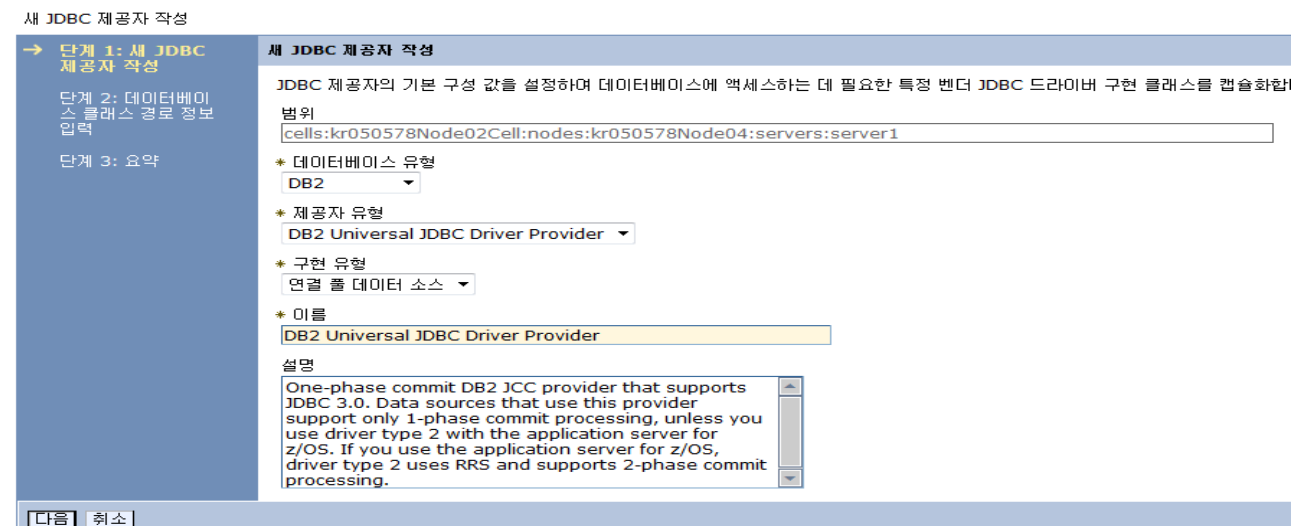
1. WAS 관리콘솔에 접속합니다.



2. 자원 > JDBC > JDBC 제공자 메뉴를 선택한 후 범위를 해당 server1 으로 변경합니다. 변경 후에 새로 작성을 클릭합니다.



3. 새 JDBC 제공자 작성 마법사가 나오면 DB2 관련 정보를 선택하고 다음을 클릭합니다.



4. 다음으로 DB2 관련 JDBC 드라이버에 대한 클래스 경로 화면이 나옵니다.

새 JDBC 제공자 작성

단계 1: 새 JDBC 제공자 작성
→ **단계 2: 데이터베이스 클래스 경로 정보 입력**
단계 3: 요약

데이터베이스 클래스 경로 정보 입력

WebSphere Application Server가 JDBC 프로바이더를 정의하는 데 사용하는 JDBC 드라이버 클래스 파일의 클래스 경로를 설정하십시오. 이 항목은 ENTER 키를 사용하여 구분되며, 경로 분리자 문자(예: ';' 또는 ':')를 포함해서는 안 됩니다. 값이 지정되어 있으면 다음을 클릭하여 값을 선택하십시오.

클래스 경로:

WebSphere 변수 (으)로 저장되는 "db2jcc.jar, db2jcc_license_cisuz.jar"의 디렉토리 위치

기본 라이브러리 경로

WebSphere 변수 (으)로 저장되는 디렉토리 위치

5. 탐색기를 활용하여 DB2 가 설치된 위치에서 하단과 같이 JDBC 드라이버 위치를 확인합니다.

C:\IBM\SQLLIB\java

라이브러리에 포함 공유 대상 새 폴더

찾기	이름	수정된 날짜	유형	크기
폴더	images	2011-05-23 오후...	파일 폴더	
폴더	jdk	2011-05-23 오후...	파일 폴더	
파일	Common	2011-04-19 오후...	ALZip JAR File	2,229KB
파일	db2dbgm	2011-04-19 오후...	ALZip JAR File	60KB
파일	db2java	2011-04-19 오후...	ALZip ZIP File	852KB
파일	db2jcc	2011-04-19 오후...	ALZip JAR File	3,332KB
파일	db2jcc_license_cu	2011-04-19 오후...	ALZip JAR File	1KB
파일	db2jcc4	2011-04-19 오후...	ALZip JAR File	3,510KB
파일	db2policy	2011-04-19 오후...	ALZip JAR File	120KB
파일	db2qgjava	2011-04-19 오후...	ALZip JAR File	176KB
파일	db2umplugin	2011-04-19 오후...	ALZip JAR File	10KB
파일	sqlj	2011-04-19 오후...	ALZip ZIP File	1,605KB
파일	sqlj4	2011-04-19 오후...	ALZip ZIP File	1,636KB

6. 이전에 확인한 DB2 JDBC 드라이버의 위치를 클래스 경로에 하단처럼 입력하고 적용을 클릭합니다.

새 JDBC 제공자 작성

단계 1: 새 JDBC 제공자 작성
→ **단계 2: 데이터베이스 클래스 경로 정보 입력**
단계 3: 요약

데이터베이스 클래스 경로 정보 입력

WebSphere Application Server가 JDBC 프로바이더를 정의하는 데 사용하는 JDBC 드라이버 클래스 파일의 클래스 경로를 설정하십시오. 이 항목은 ENTER 키를 사용하여 구분되며, 경로 분리자 문자(예: ';' 또는 ':')를 포함해서는 안 됩니다. 값이 지정되어 있으면 다음을 클릭하여 값을 선택하십시오.

클래스 경로:

WebSphere 변수 (으)로 저장되는 "db2jcc.jar, db2jcc_license_cisuz.jar"의 디렉토리 위치

기본 라이브러리 경로

WebSphere 변수 (으)로 저장되는 디렉토리 위치

7. 기본 라이브러리 경로를 확인하고 다음을 클릭합니다.

새 JDBC 제공자 작성

단계 1: 새 JDBC 제공자 작성

→ 단계 2: 데이터베이스 클래스 경로 정보 입력

단계 3: 요약

데이터베이스 클래스 경로 정보 입력

WebSphere Application Server가 JDBC 프로바이더를 정의하는 데 사용하는 JDBC 드라이버 클래스 파일의 항목은 ENTER 키를 사용하여 구분되며, 경로 분리자 문자(예: ';' 또는 ':')를 포함해서는 안 됩니다. 값이 지정된 기본 라이브러리 경로

WebSphere 변수 `${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}`(으)로 저장되는 디렉토리 위치

이전 다음 취소

8. 요약 정보를 확인하고 문제가 없으면 완료를 클릭합니다.

새 JDBC 제공자 작성

단계 1: 새 JDBC 제공자 작성

단계 2: 데이터베이스 클래스 경로 정보 입력

→ 단계 3: 요약

요약

조치 요약:

옵션	값
범위	cells:kr050578Node02Cell:nodes:kr050578Node04:servers:server1
JDBC 제공자 이름	DB2 Universal JDBC Driver Provider
설명	One-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support only 1-phase commit processing, unless you use driver type 2 with the application server for z/OS. If you use the application server for z/OS, driver type 2 uses RRS and supports 2-phase commit processing.
클래스 경로	C:\IBM\SQLLIB\java\db2jcc.jar C:\IBM\SQLLIB\java\db2jcc_license_cu.jar
기본 경로	<code>\${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}</code>
<code>\${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}</code> 구현 클래스 이름	com.ibm.db2.jcc.DB2ConnectionPoolDataSource

이전 완료 취소

9. 위의 작업을 완료하면 다음과 같이 DB2 JDBC 제공자가 설치된 것을 확인할 수 있습니다.

JDBC 제공자

JDBC 제공자의 특성을 편집하려면 이 페이지를 사용하십시오. JDBC 제공자 오브젝트는 환경의 특정 벤더 데이터베이스에 액세스할 수 있도록 특정 JDBC 드라이버 구현 클래스를 캡슐화합니다. [안내된 활동](#) 내 이 태스크에 대한 자세한 내용을 학습하십시오. 안내된 활동에서는 태스크 단계의 목록과 주제에 대한 좀더 일반적인 정보가 제공됩니다.

범위: 셀=**kr050578Node02Cell**, 노드=**kr050578Node04**, 서버=**server1**

범위는 자원 정의를 볼 수 있는 레벨을 지정합니다. 범위 및 작동 방법에 관한 자세한 정보는 [범위 설정 도움말을 참조하십시오](#).

노드=**kr050578Node04**, 서버=**server1**

환경 설정

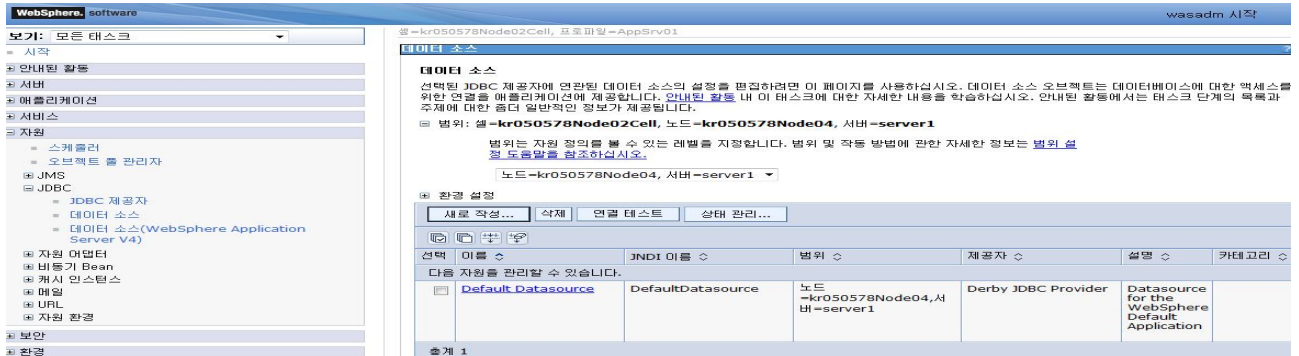
새로 작성... 삭제

선택 이름 범위 설명

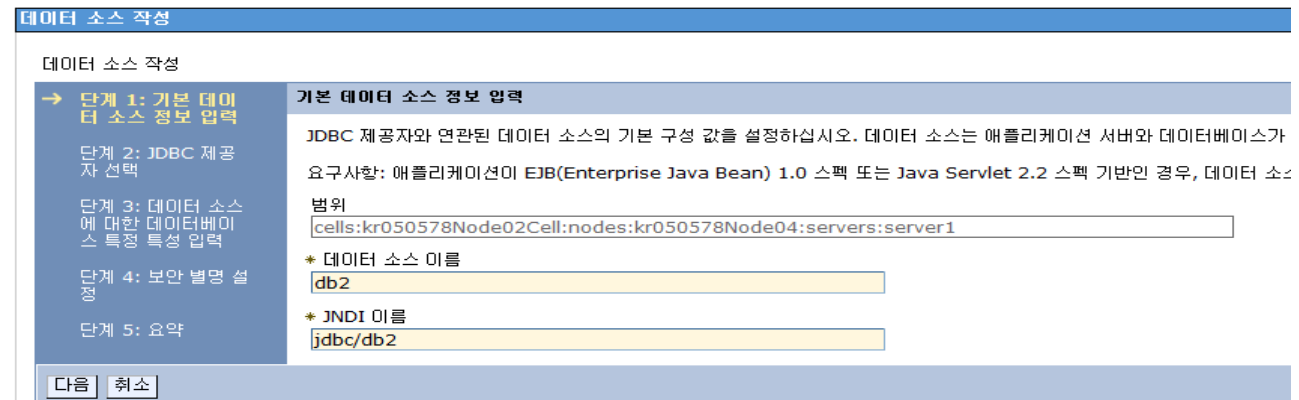
다음 자원을 관리할 수 있습니다.

선택	이름	범위	설명
<input type="checkbox"/>	DB2 Universal JDBC Driver Provider	노드=kr050578Node04,서버=server1	One-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support only 1-phase commit processing, unless you use driver type 2 with the application server for z/OS. If you use the application server for z/OS, driver type 2 uses RRS and supports 2-phase commit processing.

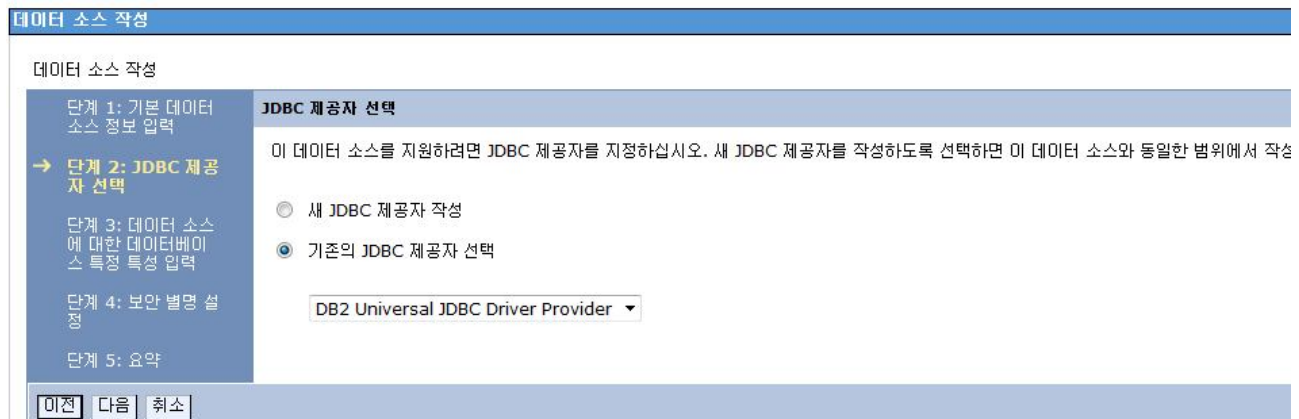
10. 관리콘솔에서 자원 > JDBC > 데이터 소스 메뉴를 클릭한 후 데이터 소스의 범위를 server1 으로 변경합니다. 그 후 새로 작성을 클릭합니다.



11. 데이터 소스 작성 마법사 화면이 나오면 데이터 소스 이름과 JNDI 이름을 입력하고 다음을 클릭합니다.



12. JDBC 제공자 선택 화면에서 기존의 JDBC 제공자 선택을 클릭하고 이전에 만들어 둔 DB2 JDBC 제공자를 선택합니다.



13. 데이터베이스 이름, 서버 이름, 포트 번호를 다음과 같이 입력하고 다음을 클릭합니다.

데이터 소스 작성

데이터 소스 작성

단계 1: 기본 데이터 소스 정보 입력
 단계 2: JDBC 제공자 선택
 → 단계 3: 데이터 소스에 대한 데이터베이스 특정 특성 입력
 단계 4: 보안 별명 설정
 단계 5: 요약

데이터 소스에 대한 데이터베이스 특정 특성 입력

이 데이터 소스에 의한 연결을 지원하는 데이터베이스 벤더 JDBC 드라이버에 필요한 해당 데이터베이스 특정 특성을 지정하십시오.

이름	값
* 드라이버 유형	4
* 데이터베이스 이름	SAMPLE
* 서버 이름	localhost
* 포트 번호	50000

☒ 컨테이너 관리 지속성(CMP)에서 이 데이터 소스 사용

14. 보안 별명 설정 부분은 이후에 다시 입력할 것이므로 우선 아무것도 매핑하지 말고 다음을 클릭합니다.

데이터 소스 작성

데이터 소스 작성

단계 1: 기본 데이터 소스 정보 입력
 단계 2: JDBC 제공자 선택
 단계 3: 데이터 소스에 대한 데이터베이스 특정 특성 입력
 → 단계 4: 보안 별명 설정
 단계 5: 요약

보안 별명 설정

이 자원의 인증 값을 선택하십시오.

컴포넌트 관리 인증 별명
 (없음)

맵핑 구성 별명
 (없음)

컨테이너 관리 인증 별명
 (없음)

주의: 다음 링크 중 하나에 액세스하여 J2C 인증 별명을 새로 작성할 수 있습니다. 링크를 클릭하면 마법사가 취소되고 현재 [글로벌 J2C 인증 별명](#) [보안 도메인](#)

15. 요약 정보를 확인하고 이상이 없으면 완료를 클릭합니다.

데이터 소스 작성

데이터 소스 작성

단계 1: 기본 데이터 소스 정보 입력
 단계 2: JDBC 제공자 선택
 단계 3: 데이터 소스에 대한 데이터베이스 특정 특성 입력
 단계 4: 보안 별명 설정
 → 단계 5: 요약

요약

조치 요약:

옵션	값
범위	cells:kr050578Node02Cell:nodes:kr050578Node04:servers:server1
데이터 소스 이름	db2
JNDI 이름	jdbc/db2
기존의 JDBC 제공자 선택	DB2 Universal JDBC Driver Provider
구현 클래스 이름	com.ibm.db2.jcc.DB2ConnectionPoolDataSource
드라이버 유형	4
데이터베이스 이름	SAMPLE
서버 이름	localhost
포트 번호	50000
컨테이너 관리 지속성(CMP)에서 이 데이터 소스 사용	true
컴포넌트 관리 인증 별명	(없음)
맵핑 구성 별명	(없음)
컨테이너 관리 인증 별명	(없음)

16. 하단과 같이 변경 정보에 대한 저장 메뉴가 나오면 아직 저장하지 말고 지금 만들어진 데이터 소스 이름을 클릭합니다.

데이터 소스

메시지

- 로컬 구성에 변경사항이 작성되었습니다. 다음을 수행할 수 있습니다.
 - 마스터 구성에 직접 저장
 - 저장하거나 버리기 전에 변경사항 검토
 - 이 변경사항을 적용하려면, 서버를 다시 시작해야 합니다.

데이터 소스

선택된 JDBC 제공자에 연결된 데이터 소스의 설정을 편집하려면 이 페이지를 사용하십시오. 데이터 소스 오브젝트는 데이터베이스에 대한 액세스를 위한 연결을 애플리케이션에 제공합니다. [안내된 활동](#) 내 이 태스크에 대한 자세한 내용을 학습하십시오. 안내된 활동에서는 태스크 단계의 목록과 주제에 대한 좀더 일반적인 정보가 제공됩니다.

범위: 셀=**kr050578Node02Cell**, 노드=**kr050578Node04**, 서버=**server1**

범위는 자원 정의를 볼 수 있는 레벨을 지정합니다. 범위 및 작동 방법에 관한 자세한 정보는 [범위 설정 도움말을 참조하십시오](#).

노드=**kr050578Node04**, 서버=**server1**

환경 설정

새로 작성... 삭제 연결 테스트 상태 관리...

선택	이름	JNDI 이름	범위	제공자	설명	카테고리
<input type="checkbox"/>	Default Datasource	DefaultDatasource	노드= kr050578Node04 , 서버= server1	Derby JDBC Provider	Datasource for the WebSphere Default Application	
<input type="checkbox"/>	db2	jdbc/db2	노드= kr050578Node04 , 서버= server1	DB2 Universal JDBC Driver Provider	DB2 Universal Driver Datasource	

17. 데이터 소스의 세부 설정 메뉴에서 JAAS - J2C 인증 데이터 메뉴를 클릭합니다.

데이터 소스 > db2

선택된 JDBC 제공자에 연결된 데이터 소스의 설정을 편집하려면 이 페이지를 사용하십시오. 데이터 소스 오브젝트는 데이터베이스에 대한 액세스를 위한 연결을 애플리케이션에 제공합니다.

구성

일반 특성

* 범위
cells:kr050578Node02Cell:nodes:kr050578Node04:servers:server1

* 제공자
DB2 Universal JDBC Driver Provider

* 이름
db2

JNDI 이름

추가 특성

- 연결 풀 특성
- WebSphere Application Server 데이터 소스 특성
- 사용자 정의 특성

관련 항목

- JAAS - J2C 인증 데이터

18. JAAS - J2C 인증 데이터 리스트에서 새로 작성을 클릭합니다.

데이터 소스 > db2 > JAAS - J2C 인증 데이터

사용할 Java(TM) 2 커넥터 보안의 사용자 ID 및 비밀번호 목록을 지정합니다.

☒ 이전 릴리스와 호환되도록 셀의 노드 이름으로 새 별명의 접두부 지정

적용

환경 설정

새로 작성... 삭제

선택	별명	사용자 ID	설명
없음			
총계 0			

19. 해당 DB 에 접속할 수 있는 ID 와 Password 를 입력하고 확인을 클릭합니다.

[데이터 소스](#) > [db2](#) > [JAAS - J2C 인증 데이터](#) > 새로 작성...

사용할 Java(TM) 2 커넥터 보안의 사용자 ID 및 비밀번호 목록을 지정합니다.

일반 특성

* 별명
juwlee

* 사용자 ID
juwlee

* 비밀번호
.....

설명

20. 이 후 데이터 소스 세부 설정 메뉴 하단에서 보안 설정 매핑을 하단과 같이 변경합니다. 이때 반드시, 컴포넌트 관리 인증 별명과 컨테이너 관리 인증 별명을 설정하셔야지만 정상적인 동작을 확인할 수 있습니다.

보안 설정

이 자원의 인증 값을 선택하십시오.

컴포넌트 관리 인증 별명
kr050578Node02/juwlee

맵핑 구성 별명
(없음)

컨테이너 관리 인증 별명
kr050578Node02/juwlee

21. 데이터 소스 설정을 완료하면 이번에 만든 데이터 소스를 선택한 후 연결 테스트를 클릭합니다.

환경 설정

선택	이름	JNDI 이름	범위	제공자	설명	카테고리
<input type="checkbox"/>	Default Datasource	DefaultDatasource	노드 =kr050578Node04,서 버=server1	Derby JDBC Provider	Datasource for the WebSphere Default Application	
<input checked="" type="checkbox"/>	db2	jdbc/db2	노드 =kr050578Node04,서 버=server1	DB2 Universal JDBC Driver Provider	DB2 Universal Driver Datasource	

총계 2

22. 다음과 같이 연결 테스트 조작에 성공했다는 메시지가 나오면 데이터 소스 설정을 완료하신 것입니다.

데이터 소스

메시지

⚠ 1 경고와 함께 kr050578Node04 노드에 있는 server1 서버의 db2 데이터 소스에 대한 연결 테스트 조작에 성공했습니다. [JVM 로그 보기](#) 추가 세부사항의 경우,

3. 해당 서블릿에서 데이터 소스를 받아오기 위하여 @Resource 어노테이션을 활용하여 다음과 같이 코딩합니다.

```
public class DB2SelectServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @Resource(name="DB2Datasource", type=javax.sql.DataSource.class)
    private javax.sql.DataSource db2DS;
```

4. 받아온 Connection 에서 실제 select 문 수행을 위해서 하단과 같이 코딩합니다.

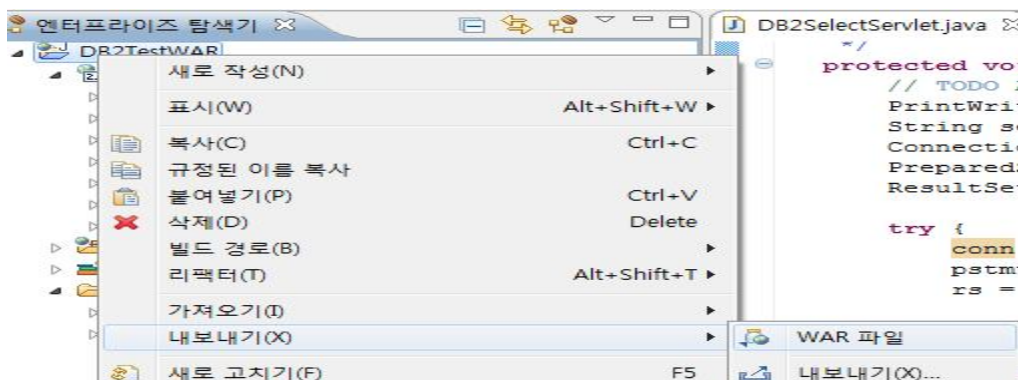
```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException
// TODO Auto-generated method stub
PrintWriter out = response.getWriter();
String select_sql = "select * from TEST";
Connection conn = null;
PreparedStatement pstmt = null;
ResultSet rs = null;

try {
    conn = db2DS.getConnection();
    pstmt = conn.prepareStatement(select_sql);
    rs = pstmt.executeQuery();

    while(rs.next())
    {
        int Id = rs.getInt("ID");
        String Name = rs.getString("NAME");
        out.println("ID : " + Id + " NAME : " + Name + "<br>");
    }

} catch (SQLException e) {
    e.printStackTrace(out);
} finally {
    try {
        if (pstmt != null)
            pstmt.close();
    } catch (SQLException e) {
        // TODO 자동 생성된 catch 블록
        e.printStackTrace(out);
    }
    try {
        if (conn != null)
            conn.close();
    }
}
```

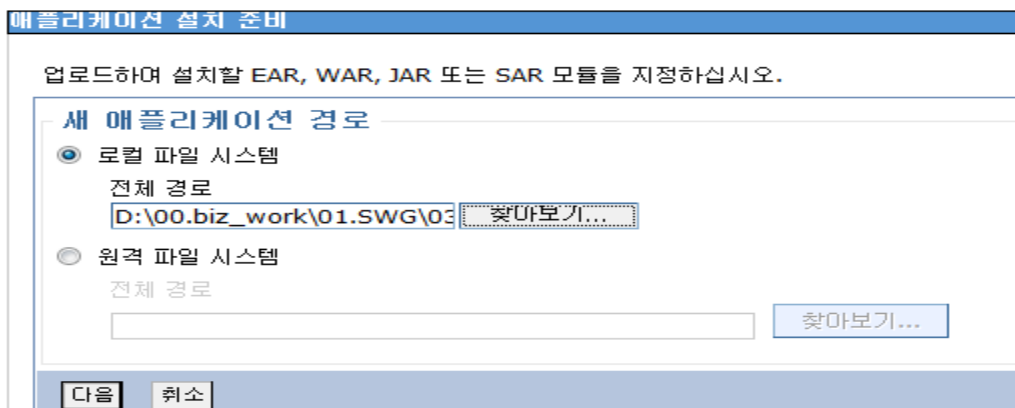
5. 만들어진 동적 웹 프로젝트를 WAR 파일 형태로 내보내기 합니다.



6. 방금 만든 어플리케이션을 WAS 에 설치하기 위하여 관리콘솔에서 어플리케이션 > 어플리케이션 유형 > WebSphere 엔터프라이즈 어플리케이션 메뉴를 클릭한 후 설치를 클릭합니다.



7. 새 어플리케이션 경로에 방금 내보내기 한 WAR 파일의 위치를 넣어준 후 다음을 클릭합니다.



8. 어플리케이션 설치 방법을 확인하고 그냥 다음을 클릭합니다.



9. 설치 옵션 선택 마법사에서는 디폴트 값을 사용하고 다음을 클릭합니다.

→ 단계 1: 설치 옵션 선택

단계 2 서버에 모듈 맵핑

★ 단계 3 자원으로 자원 참조 맵핑

★ 단계 4 웹 모듈에 대한 컨텍스트 루트 맵핑

★ 단계 5 모듈의 메타데이터

단계 6 요약

설치 옵션 선택

사용자 애플리케이션에 사용 가능한 다양한 옵션을 지정하십시오.

☐ JavaServer 페이지 파일 사전 컴파일

애플리케이션을 설치할 디렉토리

☒ 애플리케이션 분배

☐ 2진 구성 사용

☐ 엔터프라이즈 Bean 배치

애플리케이션 이름

☒ 자원에 대한 MBeans 작성

☐ 웹 및 EJB 모듈의 설정을 다시 로드하여 클래스 대체

다시 로드 간격(초)

☐ 웹 서비스 배치

입력 유효성 검증 값(깨짐/경고/실패)

☐ 프로세스 임베디드 구성

10. 서버에 모듈 맵핑은 server1 으로 되어 있는 것을 확인하고 다음을 클릭합니다.

새 애플리케이션 설치

엔터프라이즈 애플리케이션과 모듈에 대한 설치 옵션을 지정합니다.

단계 1 설치 옵션 선택

→ 단계 2: 서버에 모듈 맵핑

★ 단계 3 자원으로 자원 참조 맵핑

★ 단계 4 웹 모듈에 대한 컨텍스트 루트 맵핑

★ 단계 5 모듈의 메타데이터

단계 6 요약

서버에 모듈 맵핑

애플리케이션에 포함된 모듈을 설치할 애플리케이션 서버나 애플리케이션 서버 클러스터와 같은 대상을 지정하십시오. 모듈은 동일한 애플리케이션 서버에 설치되거나 여러 애플리케이션 서버 간에 분산될 수 있습니다. 또한 이 애플리케이션에 대한 요청의 라우터로 서비스를 제공할 웹 서버를 대상으로 지정하십시오. 웹 서버 각각의 플러그인 구성 파일(plugin-cfg.xml)은 웹 서버가 라우트되는 애플리케이션을 기초로 생성됩니다.

클러스터 및 서버:

선택	모듈	URI	서버
<input type="checkbox"/>	DB2TestWAR	DB2TestWAR.war,WEB-INF/web.xml	WebSphere:cell=kr050578Node02Cell,node=kr050578Node04,server=server1

11. 자원 참조 맵핑에서는 찾아보기를 클릭하여 지금 방금 만든 DB2 데이터 소스를 선택한 후 다음을 클릭합니다.

새 애플리케이션 설치

엔터프라이즈 애플리케이션과 모듈에 대한 설치 옵션을 지정합니다.

단계 1 설치 옵션 선택
단계 2 서버에 모듈 맵핑
→ 단계 3: **자원으로 자원 참조 맵핑**
단계 4 웹 모듈에 대한 컨텍스트 루트 맵핑
단계 5 모듈의 메타데이터
단계 6 요약

자원으로 자원 참조 맵핑

애플리케이션에 정의된 각 자원 참조가 자원에 맵핑되어야 합니다.

javax.sql.DataSource

복수 JNDI 이름 설정... 자원 인증 메소드 수정... 확장 특성...

선택	모듈	Bean	URI	자원 참조	대상 자원 JNDI 이름	로그인 구성
<input checked="" type="checkbox"/>	DB2TestWAR		DB2TestWAR.war,WEB-INF/web.xml	DB2Datasource	jdbc/db2 찾아보기...	자원 경형: 컨테이너 인증 메소드: 없음

이전 다음 취소

12. 웹 모듈에 대한 컨텍스트 루트를 입력한 후 다음을 클릭합니다.

새 애플리케이션 설치

엔터프라이즈 애플리케이션과 모듈에 대한 설치 옵션을 지정합니다.

단계 1 설치 옵션 선택
단계 2 서버에 모듈 맵핑
단계 3 자원으로 자원 참조 맵핑
→ 단계 4: **웹 모듈에 대한 컨텍스트 루트 맵핑**
단계 5 모듈의 메타데이터
단계 6 요약

웹 모듈에 대한 컨텍스트 루트 맵핑

웹 모듈의 컨텍스트 루트에 대한 값을 구성합니다.

웹 모듈	URI	컨텍스트 루트
DB2TestWAR	DB2TestWAR.war,WEB-INF/web.xml	/DB2

이전 다음 취소

13. 모듈의 메타데이터 부분을 확인하고 다음을 클릭합니다.

새 애플리케이션 설치

엔터프라이즈 애플리케이션과 모듈에 대한 설치 옵션을 지정합니다.

단계 1 설치 옵션 선택
단계 2 서버에 모듈 맵핑
단계 3 자원으로 자원 참조 맵핑
단계 4 웹 모듈에 대한 컨텍스트 루트 맵핑
→ 단계 5: **모듈의 메타데이터**
단계 6 요약

모듈의 메타데이터

메타데이터 완료 속성은 이 모듈의 배치 디스크립터가 완료되었는지 여부를 정의합니다. 메타데이터 완료 속성이 "true"로 설정되면 배치 정보를 지정하는 주석이 무시됩니다.

모듈	URI	메타데이터 완료 속성
DB2TestWAR	DB2TestWAR.war,WEB-INF/web.xml	<input type="checkbox"/>

이전 다음 취소

14. 어플리케이션 설치 옵션 요약을 확인하고 완료를 클릭합니다.

엔터프라이즈 애플리케이션과 모듈에 대한 설치 옵션을 지정합니다.

단계 1 선택	요약	
	설치 옵션 요약	
단계 2 서버에 모 듈 맵핑	옵션	값
	JavaServer 페이지 파일 사전 컴파일	아니오
단계 3 자원으로 참조 맵핑	애플리케이션을 설치할 디렉토리	
	애플리케이션 분배	예
단계 4 웹 모듈에 대한 컨텍스트 루트 맵핑	2진 구성 사용	아니오
	엔터프라이즈 Bean 배치	아니오
단계 5 모듈의 메 타데이터	애플리케이션 이름	DB2TestWAR_war
	자원에 대한 MBeans 작성	예
→ 단계 6: 요약	웹 및 EJB 모듈의 설정을 다시 로드하여 플래스 트체	아니오
	다시 로드 간격(초)	
	웹 서비스 배치	아니오
	입력 유효성 검증 값(꺼짐/경고/실패)	경고
	프로세스 임베디드 구성	아니오
	파일 권한	.*\dll=755#.*\so=755#.*\a=755#.*\sl=755
	애플리케이션 빌드 ID	Unknown
	원격 자원으로 포함(include) 디스패치 허용	아니오
	원격 자원에서 포함(include) 서비스 허용	아니오
	비즈니스 레벨 애플리케이션 이름	
	비동기 요청 디스패치 유형	사용 불가능
	EJB 참조 대상이 자동으로 해석하도록 허용	아니오
	Deploy client modules	아니오
	클라이언트 배치 모드	isolated
	Validate schema	아니오
	셀/노드/서버	여기를 클릭하십시오.

15. 실제 어플리케이션 설치를 확인 한 후 저장을 클릭합니다.

설치 중...
애플리케이션에 엔터프라이즈 Bean이 있는 경우, EJB 배치 프로세스에 몇 분이 소요될 수 있습니다. 프로세스가 완료될 때까지 구성을 저장하지 마십시오.
EJB 배치 프로세스 발생 시 이 프로세스에 대한 특정 정보는 애플리케이션이 배치되는 서버 또는 Deployment Manager에서 SystemOut.log를 확인하십시오.
ADMA5016: DB2TestWAR_war 설치가 시작되었습니다.
ADMA5067: DB2TestWAR_war 애플리케이션의 자원 유효성 검증이 완료되었습니다.
ADMA5058: 배치 대상 버전에 대해 애플리케이션 및 모듈 버전의 유효성을 검증했습니다.
ADMA5005: WebSphere Application Server 저장소에 DB2TestWAR_war 애플리케이션이 구성되어 있습니다.
ADMA5005: WebSphere Application Server 저장소에 DB2TestWAR_war 애플리케이션이 구성되어 있습니다.
ADMA5081: The bootstrap address for client module is configured in the WebSphere Application Server repository.
ADMA5053: 설치된 선택적 패키지에 대한 라이브러리 참조가 작성됩니다.
ADMA5005: WebSphere Application Server 저장소에 DB2TestWAR_war 애플리케이션이 구성되어 있습니다.
ADMA5001: 애플리케이션 2진 C:\IBM\WebSphere8\AppServer\profiles\AppSrv01\wstemp\794994399\workspace\cells\kr050578Node02Cell\applications\DB2TestWA C.
ADMA5005: WebSphere Application Server 저장소에 DB2TestWAR_war 애플리케이션이 구성되어 있습니다.
SECJ0400: DB2TestWAR_war 애플리케이션이 appContextIDForSecurity 정보로 업데이트되었습니다.
ADMA5005: WebSphere Application Server 저장소에 DB2TestWAR_war 애플리케이션이 구성되어 있습니다.
ADMA5005: WebSphere Application Server 저장소에 DB2TestWAR_war 애플리케이션이 구성되어 있습니다.
ADMA5113: 활성화 계획이 작성되었습니다.
ADMA5011: 애플리케이션 DB2TestWAR_war의 temp 디렉토리 정리가 완료되지 않았습니다.
ADMA5013: DB2TestWAR_war 애플리케이션이 설치되었습니다.
DB2TestWAR_war 애플리케이션이 설치되었습니다.
애플리케이션을 시작하려면 우선 마스터 구성의 변경사항을 저장하십시오.
로컬 구성에 변경사항이 작성되었습니다. 다음을 수행할 수 있습니다.
마스터 구성에 직접 저장
저장하거나 버리기 전에 변경사항 검토

16. 해당 어플리케이션을 시작 시킵니다.



17. 다음과 같이 브라우저를 이용하여 Sample 요청을 날려서 결과를 확인합니다.



ID : 1 NAME : TESTer01

18. 정상적인 결과가 리턴된다면 어플리케이션 설치를 완료하신 것 입니다.

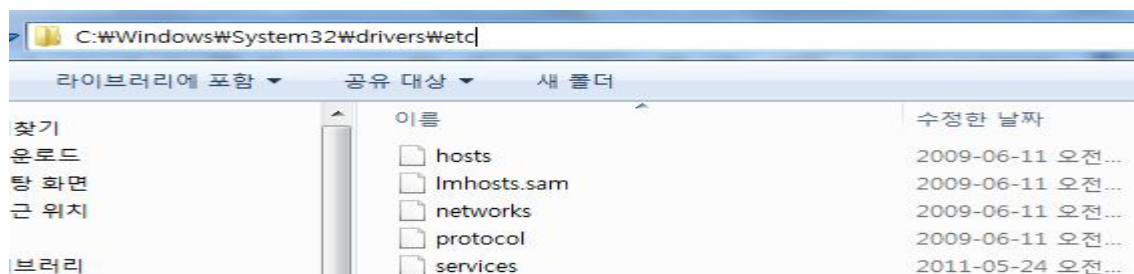
4) Failover 테스트를 위한 DB2 인스턴스 추가 생성

1. 실제적인 failover 테스트를 위하여 다음과 같은 명령어를 이용해서 DB2INST1 이라는 DB2 인스턴스를 추가로 생성합니다. (Windows 환경을 기준으로 설명)

```
C:\IBM\SQLLIB\BIN>db2icrt -u db2admin2 db2inst1
db2admin2의 현재 암호 입력하십시오:
DB20000I DB2ICRT 명령이 완료되었습니다.
```

```
C:\IBM\SQLLIB\BIN>db2ilist
DB2INST1
DB2
```

2. 추가로 만들어진 DB2 인스턴스에 서비스 포트를 매핑하기 위하여 c:\Windows\System32\drivers\etc 폴더에 가서 services 파일을 edit 모드로 엽니다.



3. services 파일 맨 밑에 "db2c_인스턴스명 포트/tcp" 양식으로 새로운 서비스 포트를 입력합니다.

```
db2c_DB2 50000/tcp
db2c_db2inst1 60000/tcp # DB2 connection service port
```

4. 현재 환경의 DB2 인스턴스를 확인한 후 현재 환경을 새로 만든 DB2 인스턴스 환경으로 변경합니다. 변경 후 서비스 포트를 DB2 인스턴스와 연결하기 위하여 Database manage configuration 정보를 변경합니다.

```
C:\IBM\SQLLIB\BIN>db2 get instance
현재 데이터베이스 관리 프로그램 인스턴스: DB2

C:\IBM\SQLLIB\BIN>set DB2INSTANCE=DB2INST1
C:\IBM\SQLLIB\BIN>db2 get instance
현재 데이터베이스 관리 프로그램 인스턴스: DB2INST1

C:\IBM\SQLLIB\BIN>db2 update database manager configuration using sucename db2c_
db2inst1
DB20000I UPDATE DATABASE MANAGER CONFIGURATION 명령이 완료되었습니다.

C:\IBM\SQLLIB\BIN>db2start
SQL1063N DB2START 처리가 완료되었습니다.

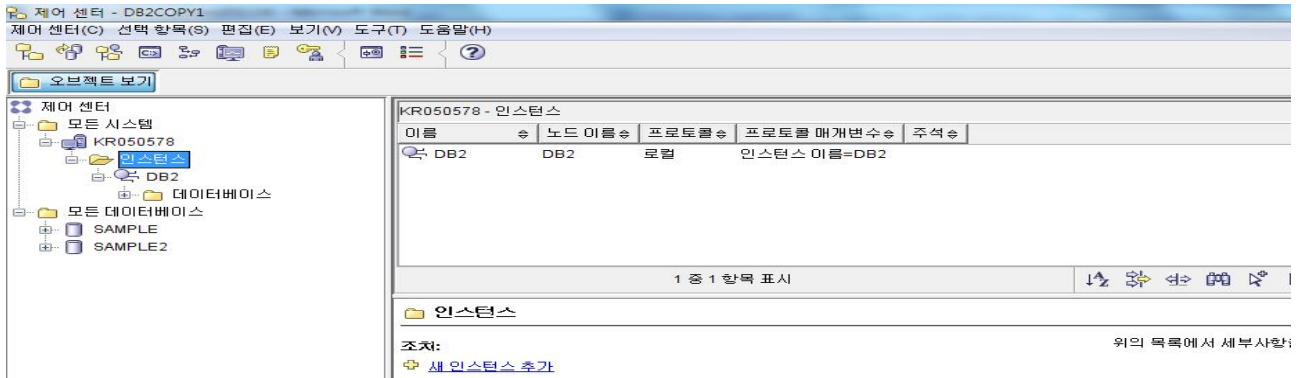
C:\IBM\SQLLIB\BIN>db2stop
SQL1064N DB2STOP 처리가 완료되었습니다.

C:\IBM\SQLLIB\BIN>db2set DB2COMM=tcpip
```

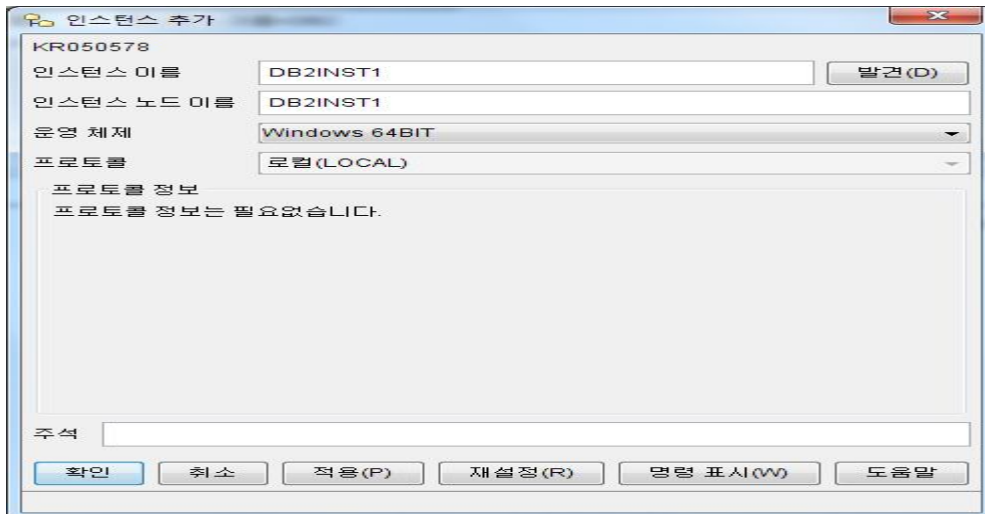
5. 변경이 완료되면 db2start 명령어를 통해 새로운 DB2 인스턴스를 시작합니다.

```
C:\IBM\SQLLIB\BIN>db2start
SQL1063N DB2START 처리가 완료되었습니다.
```

6. DB2 제어 센터를 오픈하여 인스턴스 메뉴에서 새 인스턴스 추가를 클릭합니다.



7. 발견을 클릭하여 새로 추가된 DB2 인스턴스를 선택하고 확인을 클릭합니다.



8. 새로운 DB2 인스턴스에 대한 카탈로그 작업이 완료하면 새 데이터베이스 작성을 클릭하여 Sample 데이터베이스를 하나 생성합니다.



9. 새로 만든 sample 데이터베이스에 접속하여 이전에 만든 Table 과 동일한 구조를 가진 Table 을 생성하고 테스트 결과 확인을 위한 임시 값을 입력합니다.

```
db2 => connect to sample

데이터베이스 연결 정보

데이터베이스 서버                = DB2/NT64 9.7.4
SQL 권한 부여 ID                 = JUWLEE
로컬 데이터베이스 별명          = SAMPLE

db2 => select * from TEST
SQL0204N  "JUWLEE.TEST"은(는) 정의되지 않은 이름입니다.  SQLSTATE=42704
db2 => create table TEST(ID integer generated always as identity(start with 1, i
ncrement by 1, no cache), NAME varchar(15))
DB20000I  SQL 명령이 완료되었습니다.
db2 => insert into TEST(NAME) values('TESTer02')
DB20000I  SQL 명령이 완료되었습니다.
db2 => select * from TEST

ID          NAME
-----
1 TESTer02

1 레코드가 선택되었습니다.

db2 =>
```

5) Failover 테스트를 추가 설정

1. 관리콘솔에서 자원 > JDBC > 데이터 소스 메뉴를 클릭하여 새로 만든 DB2 인스턴스를 위한 데이터 소스를 작성합니다.

관리콘솔에서 자원 > JDBC > 데이터 소스 메뉴를 클릭하여 새로 만든 DB2 인스턴스를 위한 데이터 소스를 작성합니다.

선택된 JDBC 제공자에 연관된 데이터 소스의 설정을 변경하려면 이 페이지를 사용하십시오. 데이터 소스 오브젝트는 데이터베이스에 대한 액세스를 위한 연결을 애플리케이션에 제공합니다. **안내된 활동** 내 이 태스크에 대한 자세한 내용을 학습하십시오. 안내된 활동에서는 태스크 단계의 목록과 주제에 대한 좀 더 일반적인 정보가 제공됩니다.

범위: 셀=**kr050578Node02Cell**, 노드=**kr050578Node04**, 서버=**server1**

범위는 자원 정의를 볼 수 있는 레벨을 지정합니다. 범위 및 작동 방법에 관한 자세한 정보는 [범위 설정 도움말을 참조하십시오](#).

노드=**kr050578Node04**, 서버=**server1**

환경 설정

새로 작성... 삭제 연결 테스트 상태 관리...

선택	이름	JNDI 이름	범위	제공자	설명	카테고리
<input type="checkbox"/>	Default Datasource	DefaultDatasource	노드= kr050578Node04 , 서버= server1	Derby JDBC Provider	Datasource for the WebSphere Default Application	
<input type="checkbox"/>	db2	jdbc/db2	노드= kr050578Node04 , 서버= server1	DB2 Universal JDBC Driver Provider	DB2 Universal Driver Datasource	

2. 다른 정보는 동일하게 입력하되 서비스 포트를 새로 만들었으므로 그에 맞는 포트를 입력합니다.

데이터 소스 작성

데이터 소스 작성

단계 1: 기본 데이터 소스 정보 입력

단계 2: JDBC 제공자 선택

→ 단계 3: 데이터 소스에 대한 데이터베이스 특정 특성 입력

단계 4: 보안 별명 설정

단계 5: 요약

이 데이터 소스에 대한 데이터베이스 특정 특성 입력

이 데이터 소스에 의한 연결을 지원하는 데이터베이스 벤더 JDBC 드라이버에 필요한 해당 데이터베이스 특정 특성

이름	값
* 드라이버 유형	4
* 데이터베이스 이름	SAMPLE
* 서버 이름	localhost
* 포트 번호	60000

☒ 컨테이너 관리 지속성(CMP)에서 이 데이터 소스 사용

이전 다음 취소

3. 새로운 DB2 인스턴스를 위한 데이터 소스가 추가된 것을 확인합니다.

환경 설정

새로 작성... 삭제 연결 테스트 상태 관리...

선택	이름	JNDI 이름	범위	제공자	설명	카테고리
<input type="checkbox"/>	Default Datasource	DefaultDatasource	노드= kr050578Node04 , 서버= server1	Derby JDBC Provider	Datasource for the WebSphere Default Application	
<input type="checkbox"/>	db2	jdbc/db2	노드= kr050578Node04 , 서버= server1	DB2 Universal JDBC Driver Provider	DB2 Universal Driver Datasource	
<input type="checkbox"/>	db2_2	jdbc/db2_2	노드= kr050578Node04 , 서버= server1	DB2 Universal JDBC Driver Provider	DB2 Universal Driver Datasource	

4. 이전 데이터 소스와 마찬가지로 연결 테스트를 클릭하여 연결 조작 정상 여부를 확인합니다.

데이터 소스

메시지
 1 경고와 함께 kr050578Node04 노드에 있는 server1 서버의 db2_2 데이터 소스에 대한 연결 테스트 조작에 성공했습니다. [JVM 로그 보기](#) 추가 세부사항의 경우,

데이터 소스
 선택된 JDBC 제공자에 연관된 데이터 소스의 설정을 편집하려면 이 페이지를 사용하십시오. 데이터 소스 오브젝트는 데이터베이스에 대한 액세스를 위한 연결을 애플리케이션에 제공합니다. [안내된 활동](#) 내 이 태스크에 대한 자세한 내용을 학습하십시오. 안내된 활동에서는 태스크 단계의 목록과 주제에 대한 좀더 일반적인 정보가 제공됩니다.

범위: 셀=**kr050578Node02Cell**, 노드=**kr050578Node04**, 서버=**server1**

범위는 자원 정의를 볼 수 있는 레벨을 지정합니다. 범위 및 작동 방법에 관한 자세한 정보는 [범위 설정 도움말을 참조하십시오](#).

노드=**kr050578Node04**, 서버=**server1**

5. 해당 데이터 소스 > 연결 풀 > 사용자 정의 특성 메뉴를 클릭한 후 새로 작성을 클릭합니다.

데이터 소스

[데이터 소스](#) > [db2](#) > [연결 풀](#) > [사용자 정의 특성](#)

이 페이지에서 임의의 이름 및 값 쌍을 지정할 수 있습니다. 이름 및 값 쌍에 지정된 값은 내부 시스템 구성 특성을 설정할 수 있는 문자열입니다.

환경 설정

☐ ☐ ☐ ☐

선택	이름	값	설명
	없음		

총계 0

6. Multi-DataSource failover 를 위하여 필요한 사용자 정의 속성을 하단처럼 입력합니다.

alternateResourceJNDIName – 대체하기를 원하는 데이터 소스의 JNDI 이름

failureThreshold – 몇 번의 failure 후에 대체 JNDI 로 넘어갈지에 대한 임계값

[데이터 소스](#) > [db2](#) > [연결 풀](#) > [사용자 정의 특성](#)

이 페이지에서 임의의 이름 및 값 쌍을 지정할 수 있습니다. 이름 및 값 쌍에 지정된 값은 내부 시스템 구성 특성을 설정할 수 있는 문자열입니다.

환경 설정

☐ ☐ ☐ ☐

선택	이름	값	설명
	다음 자원을 관리할 수 있습니다.		
<input type="checkbox"/>	alternateResourceJNDIName	jdbc/db2_2	
<input type="checkbox"/>	failureThreshold	5	

총계 2

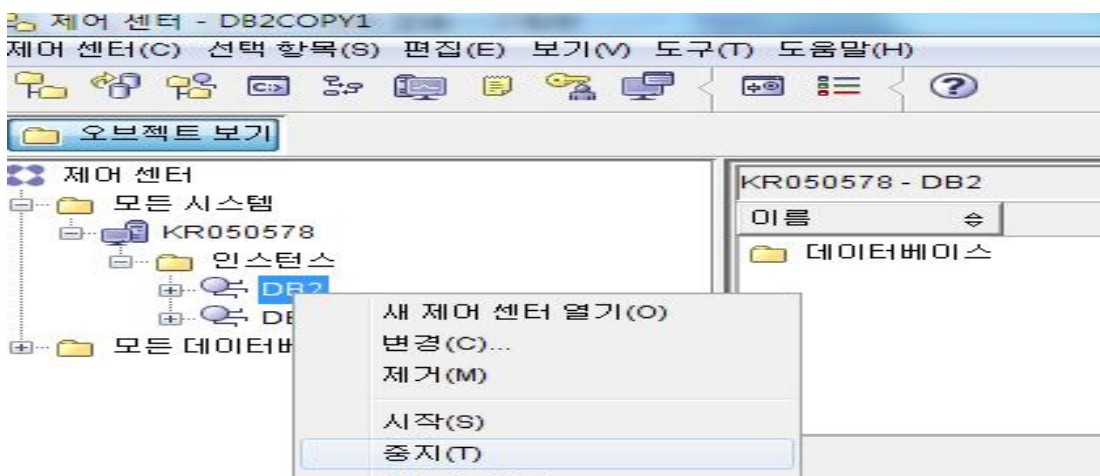
6) Failover 테스트

1. Sample 페이지를 요청하여 하단처럼 정상적인 결과가 나오는 것을 확인합니다.

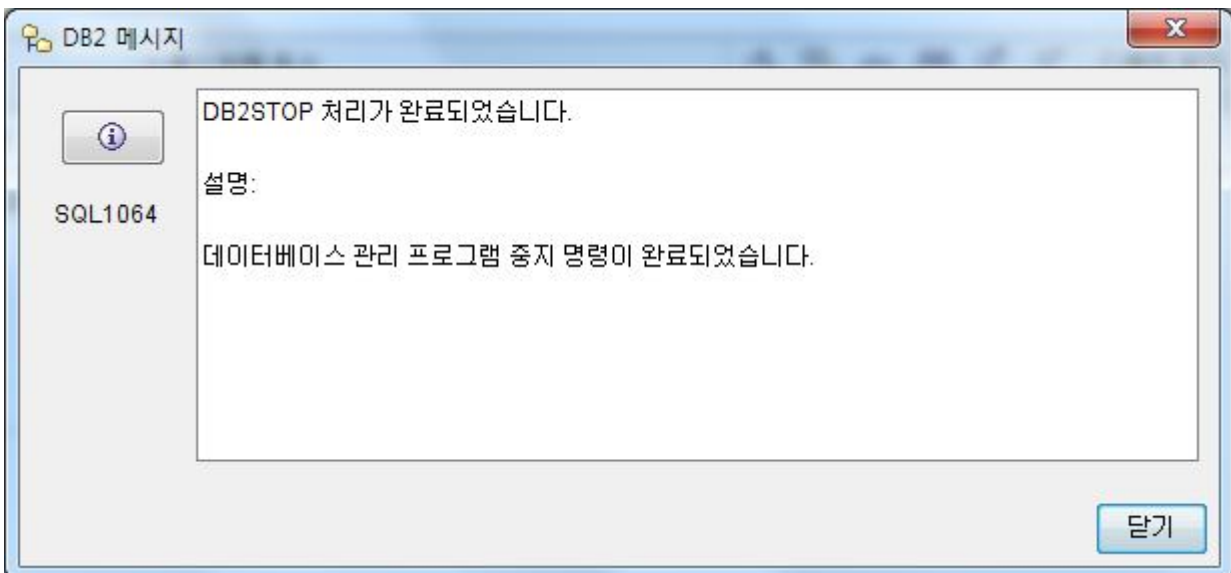


ID : 1 NAME : TESTer01

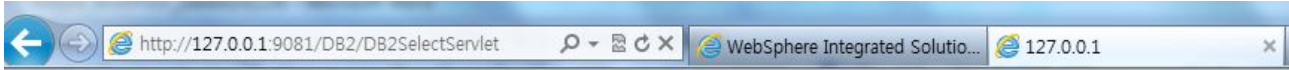
2. Failover 테스트를 위하여 DB2 제어 센터에 접속하여 현재 Sample 페이지의 요청을 받고 있는 DB2 인스턴스를 중지 합니다.



3. 정상적으로 DB2 인스턴스가 중지되면 하단과 같은 중지완료 메시지를 확인할 수 있습니다.



8. 원래 자원을 참조했던 DB2 인스턴스가 정상으로 돌아오는 것을 인지한 순간부터 즉시, 요청이 원래 DB2 인스턴스를 참조하여 하단에 보는 것처럼 TESTer01 출력을 보여줍니다. 즉, 인지된 순간 Failback 가 정상적으로 이루어집니다.



ID : 1 NAME : TESTer01

9. 마찬가지로 하단과 같은 로그가 SystemOut.log 에 남게 됩니다.

DataSourceCon A J2CA0682I: JNDI 이름이 jdbc/db2인 구성된 자원이 JNDI 이름이 jdbc/db2인 자원의 새 요청을 처리할 수 있습니다..

7) 참고 자료

1. 이 가이드는 IBM WAS v8.0 최초 사용자를 위한 기본 가이드입니다.
2. IBM WAS 자체에 아직 익숙하지 않으신 분들은 가급적 기본강좌인 '하나씩 쉽게 따라 해보는 IBM WAS v7' 강좌를 먼저 읽고 이 강좌를 읽으시는 것이 훨씬 이해에 됩니다.
(http://www.websphere.pe.kr/xe/?mid=was_info_re&page=3&document_srl=800)
3. 가급적 IBM WAS v8.0 InfoCenter 의 해당 카테고리를 한 번 읽어보고 난 후에 작업하시기 바랍니다.
4. InfoCenter – Resource workload routing
(http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/cdat_dsfailover.html?resultof=%22%61%6c%74%65%72%6e%61%74%65%52%65%73%6f%75%72%63%65%4a%4e%44%49%4e%61%6d%65%22%20%22%61%6c%74%65%72%6e%61%74%65%72%65%73%6f%75%72%63%65%6a%6e%64%69%6e%61%6d%22%20)