



# 웹 서비스를 사용하는 리치 Java 클라이언트 빌드



---

## 목차

웹 서비스를 사용하는 리치 <b>Java</b> 클라이언트 빌드 . . . 1	단원 2.3: 세부사항 필드를 테이블 선택사항에 바
소개: 웹 서비스를 사용하는 리치 Java 클라이언트 빌드 . . . 1	인드 . . . . . 20
모듈 1: Visual Editor에서 클라이언트 GUI 디자인 . 3	단원 2.4: 갱신 단추를 조치 바인더에 바인드 . . 24
단원 1.1: Java 프로젝트 설정 . . . . . 3	단원 2.5: 삭제 단추 및 확인 대화 상자 사용 . . 26
단원 1.2: 직원 테이블 추가 및 레이아웃 . . . . 4	단원 2.6: 새 직원을 추가하기 위한 조치 및 바인
단원 1.3: 비주얼 클래스 실행 . . . . . 9	딩 설정 . . . . . 29
모듈 2: 비주얼 구성요소를 웹 서비스에 바인드 . . 11	단원 2.7: 취소 단추 작동 프로그래밍 . . . . . 35
단원 2.1: 웹 서비스 설치 및 배치 . . . . . 11	단원 2.8: 직원 테이블에서 필터 설정 . . . . . 36
단원 2.2: 직원 테이블을 웹 서비스 데이터 소스	요약: 웹 서비스를 사용하는 리치 Java 클라이언트
에 바인드 . . . . . 13	빌드 . . . . . 37



---

## 웹 서비스를 사용하는 리치 Java 클라이언트 빌드

이 학습서에서는 Java Visual Editor를 사용하여 웹 서비스에 연결하는 리치 Java 클라이언트를 빌드하는 방법에 대해 학습합니다. 학습서에서 빌드한 클라이언트를 회사 디렉토리라고 합니다.

회사 디렉토리는 회사 직원 디렉토리를 유지보수하는 데 사용하는 Java 응용프로그램입니다. 이 응용프로그램은 직원 레코드를 작성, 검색, 갱신 및 삭제하는 방법을 제공하는 샘플 웹 서비스에 연결합니다.

클라이언트는 Swing 구성요소를 사용하여 Java Visual Editor에 시각적으로 빌드됩니다. Java Visual Editor는 웹 서비스에 연결하고 이 웹 서비스에 대해 작업하기 위한 헬퍼 클래스(데이터 소스, 데이터 오브젝트 및 바인더) 세트를 제공합니다. 웹 서비스는 사용자가 설치한 WebSphere Application Server V6.0에 로컬로 배치되며 도구는 WSDL(Web Services Description Language) 파일에 기초하여 클라이언트에 맞는 Java 프록시를 생성하는 데 도움을 줍니다.

완료된 제품 참조

### 학습 목표

이 학습서에서는 다음에 대해 설명합니다.

- 사용자 인터페이스를 디자인하고 레이아웃하기 위해 Java Visual Editor를 사용하는 방법
- 데이터 오브젝트 및 웹 서비스에 인터페이스 요소를 바인드하는 방법

### 소요 시간

2시간 15분

관련 정보

PDF 버전 보기

학습서: Hello World Java

---

## 소개: 웹 서비스를 사용하는 리치 Java 클라이언트 빌드

클라이언트용 GUI(Graphical User Interface)는 Swing 구성요소를 사용하여 대부분 시각적으로 사전 빌드되어 있습니다. 첫 번째 모듈에서는 Java Visual Editor를 사용하여 주요 GUI 구성요소의 레이아웃을 완료하고, 두 번째 모듈에서는 GUI 구성요소를 웹 서비스 데이터 소스, 서비스 및 데이터 소스가 리턴한 오브젝트에 바인드할 것입니다. Java Visual Editor로 응용프로그램을 빌드할 때, Java Visual Editor에서 생성되고 사용자 응용프로그램이 사용하는 헬퍼 클래스의 인스턴스인 바인더, 데이터 오브젝트 및 데이터 소스가 사용됩니다.

My Company Directory

File Help

Filter: (Last name)

Last name	First name	Email	Employee ID
Miller	Jeff	jeff.miller@mycodi...	24559
Garcia	Linda	linda.garcia@myc...	24560
Maxwell	Seth	seth.maxwell@m...	24561
Dehlavi	Madhubala	madhubala.dehlav...	27021
Huang	David	david.huang@my...	28225
Jensen	Kristin	kristin.jensen@my...	28290
Maxwell	Aubrey	aubrey.maxwell@...	30089

New Delete

Employee details

Last name: Huang

First name: David

Employee ID: 28225

Contact information

Email: david.huang@mycodir.biz

Phone: 555-8225

Work location

Office: 121

Building: C

Site: Atlanta, GA

Update Cancel

완료된 제품 그림 참조:

## 학습 목표

이 학습서에서는 다음에 대해 설명합니다.

- 사용자 인터페이스를 디자인하고 레이아웃하기 위해 Java Visual Editor를 사용하는 방법
- 데이터 오브젝트 및 웹 서비스에 인터페이스 요소를 바인드하는 방법

## 소요 시간

학습서 전체를 완료하는 데 약 2시간 30분이 소요됩니다.

## 시스템 요구사항

- WebSphere Application Server v6.1. 이 서버가 제품과 함께 이미 설치되었을 수도 있으며 또는 자체 독립형 설치를 사용할 수 있습니다. 이 학습서의 시나리오에서는 로컬 호스트에서 실행 중인 WebSphere Application Server에 샘플 웹 서비스를 배치하도록 요청합니다.

샘플 웹 서비스는 다른 서버에서 실행할 수 있지만 이 학습서는 WebSphere Application Server v6.0 및 v6.1에서만 테스트되었습니다.

## 전제조건

다음 개념에 대해 잘 알고 있어야 합니다.

- 기본 Java 개발

- 기본 웹 서비스 원리
- 기본 Workbench 기술(프로젝트에 대한 작업 및 Perspective와 보기의 탐색 등)

## 모듈 1: Visual Editor에서 클라이언트 GUI 디자인

이 모듈에서는 Java Visual Editor를 사용하여 비주얼 구성요소를 응용프로그램에 추가한 후 배열 제한조건을 시각적으로 레이아웃하고 설정하는 방법에 대해 학습합니다. 이 모듈의 마지막 단원에서는 Java 파일을 실행하여 어떻게 실제 응용프로그램으로 표시는지 확인할 수 있는 방법에 대해 설명합니다.

**알아두기:** 이 모듈을 시작하기 전에 학습서 소개에 요약된 전제조건 지식을 잘 알고 있어야 합니다.

### 학습 목표

이 모듈의 단원을 완료하였으므로 개념을 이해하고 다음을 수행할 수 있습니다.

- Java 인터페이스에서 JTable 추가 및 레이아웃 지정
- 비주얼 클래스를 실행하여 작업 테스트

### 소요 시간

이 모듈을 완료하는 데 약 15분이 소요됩니다.

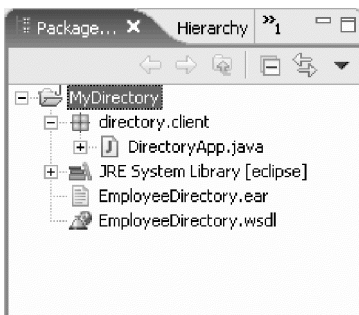
### 단원 1.1: Java 프로젝트 설정

이 단원에서는 프로젝트를 작업공간으로 가져와서 MyDirectory 프로젝트를 설정합니다. 프로젝트에는 단일 Java 클래스 및 나중에 사용할 기타 파일이 포함됩니다.

이 학습서의 주요 초점이 시각적 컴포넌트를 웹 서비스에 바인딩하는 것이지만 "My Company Directory" 응용프로그램에 대한 대부분의 Java GUI가 이미 설계되어 있습니다.

MyDirectory 프로젝트는 이 학습서에서 활용하는 기본 Java 프로젝트입니다. 여기에는 DirectoryApp.java 파일이 포함되어 있으며 이 파일은 빌드 중인 기본 Java 응용프로그램이 포함된 Java 파일입니다. 또한 이 학습서에는 사용자를 도와주는 MyDirectory 프로젝트의 몇몇 버전이 포함되어 있습니다. 예를 들어, 각 모듈을 시작하기 위한 시작 버전과 완료된 프로젝트가 포함되어 있는 완료 버전 등이 있습니다.

1. MyDirectory 프로젝트 가져오기를 수행하십시오.
2. Java Perspective의 패키지 탐색기에서 MyDirectory 프로젝트가 다음과 같이 표시됩니다.



## 학습 체크포인트

이 단원에서는 예제 MyDirectory 프로젝트를 임포트했으며 해당 학습서에 대한 시작점으로 작동합니다.

MyDirectory 프로젝트에는 다음과 같은 자원 파일이 포함되어 있습니다.

- DirectoryApp.java: 이 학습서에서 개발하고 있는 응용프로그램을 포함하는 Java 파일입니다. DirectoryApp.java 파일은 directory.client라는 이름의 Java 패키지에 있습니다.
- EmployeeDirectory.ear: 샘플 웹 서비스를 포함하는 엔터프라이즈 응용프로그램입니다. 모듈 2에서 이 웹 서비스를 IBM WebSphere Application Server V6.0의 로컬 설치에 배치할 것입니다.
- EmployeeDirectory.wsdl: 사용자가 배치하는 샘플 웹 서비스를 설명하는 데 WSDL(Web Services Description Language)을 사용하는 XML 파일입니다. 모듈 2에서 이 WSDL 파일을 사용하여 응용프로그램에서 사용할 Java 프록시를 생성합니다.

## 단원 1.2: 직원 테이블 추가 및 레이아웃

이 단원에서는 Java Visual Editor를 사용하여 JScrollPane 및 JTable을 응용프로그램에 추가합니다. 차후 연습에서는 회사 디렉토리의 모든 직원 목록을 리턴하는 웹 서비스로부터 데이터를 가져오도록 JTable을 프로그램합니다.

JTable을 추가한 후에는 Java Visual Editor의 디자인 보기를 사용하여 다음 스펙과 일치하도록 JTable의 레이아웃을 사용자 정의합니다.

- 가로로 세 개의 셀과 세로로 두 개의 셀에 JTable이 놓이도록 하십시오.
- 왼쪽에 15픽셀의 삽입을 추가하십시오.
- JTable의 이름을 employeesTable로 바꾸십시오.

데모 보기

### Java Visual Editor에서 DirectoryApp.java 파일을 여십시오.

Java Visual Editor에서 DirectoryApp.java 파일을 열려면 다음을 수행하십시오.

1. Java Perspective의 패키지 탐색기 보기에서 MyDirectory 프로젝트와 directory.client 패키지를 펼치십시오.
2. DirectoryApp.java 파일을 마우스 오른쪽 단추로 클릭한 후 열기 → **Visual Editor**를 선택하십시오. Java Visual Editor는 Java 클래스를 로드하고 디자인을 그래픽 캔버스 영역에 표시합니다.

팁:

- Java Visual Editor에서 사용하는 룩앤필을 변경하려면 창 → 환경 설정 → **Java** → **Visual Editor**로 이동하여 Swing 룩앤필을 지정하십시오. 환경 설정은 다음 클래스를 열 때 적용됩니다. 이 학습서에서는 Windows 룩앤필을 사용합니다.
- Visual Editor를 모든 Java 파일의 기본 편집기로 만들려면 창 → 환경 설정을 클릭한 후 **Workbench** → 파일 연관 페이지로 이동하여 환경 설정을 정의하십시오.

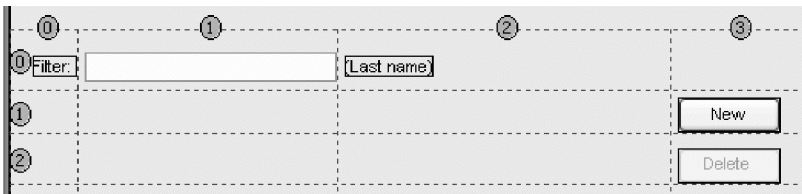


## JScrollPane에 JTable 추가

DirectoryApp.java의 기본 창은 JPanel이 있는 JFrame을 기본 콘텐츠 분할창으로 사용합니다. 이 때 작성하는 응용프로그램의 JPanel 이름은 jContentPane입니다. jContentPane은 GridBagLayout이라고 하는 레이아웃 관리자 유형을 사용하도록 설정되었습니다. GridBagLayout은 비주얼 구성요소가 점유할 수 있는 셀의 눈금에 기반한 강력한 레이아웃 체계입니다. Java Visual Editor는 눈금 경계를 표시하여 GridBagLayout에서의 작업을 쉽게 해줍니다. 또한 새 구성요소를 눈금에 놓을 때 위치 마커를 표시하며 크기를 조정하거나 GridBagLayout에서 이동 중인 구성요소에 핸들을 표시합니다.

직원 테이블(javax.swing.JTable)을 DirectoryApp.java 사용자 인터페이스에 추가하려면 다음을 수행하십시오.

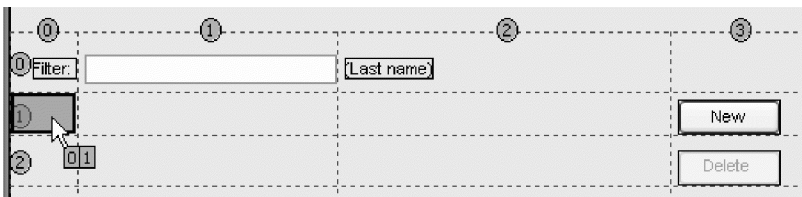
1. 디자인 보기 또는 Java Bean 보기에서 jContentPane을 마우스 오른쪽 단추로 클릭하고 눈금 표시를 선택하십시오. 빨간색 점선은 눈금 경계를 표시하며 파란색의 번호가 있는 원은 행 및 열 번호를 표시합니다. 예를 들어 새로 작성 단추의 셀은 행 1(눈금 y)과 열 3(눈금 x)입니다.



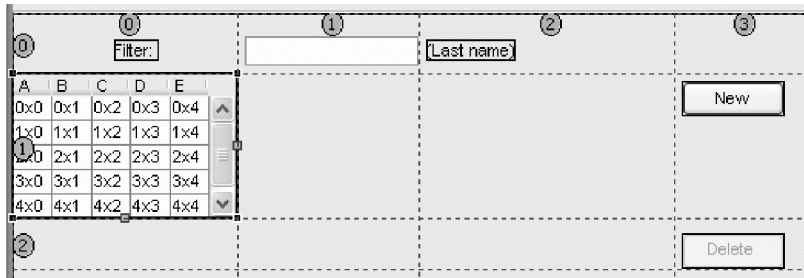
2. Java Visual Editor 팔레트에서 JScrollPane의 JTable Swing 컴포넌트를 선택하십시오. 이 컴포넌트는 팔레트의 Swing 컴포넌트 드로어에 분류되어 있습니다.

팁: 기본적으로 팔레트는 디자인 영역의 오른쪽에 접혀 있습니다. 팔레트의 크기를 조정하고 이동할 수 있습니다.

3. 마우스 포인터를 열 0, 행 1에 있는 눈금의 셀 위로 이동하십시오.



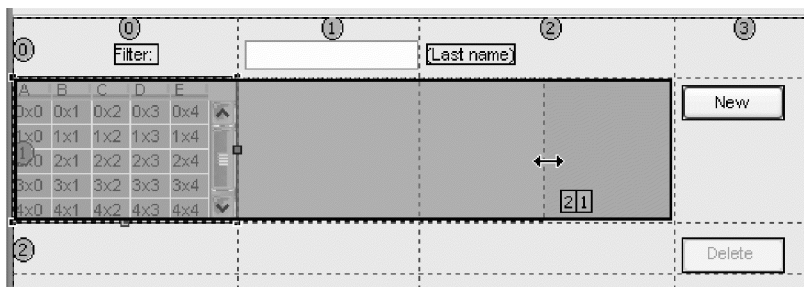
- 눈금 위에서 포인터를 이동하면 마우스 포인터는 포인터의 위치에 따라 눈금의 x와 y 좌표를 알려주는 두 개의 번호가 있는 사각형을 표시합니다.
  - 마우스 포인터를 눈금 경계 바로 위로 가져가면 새로운 행과 열이 작성될 수 있으며 기존 행과 열에는 번호가 재지정됩니다. 이 경우 마우스 포인터의 노란색 사각형, 눈금 사이의 노란색 막대 및 노란색 열과 행 레이블이 이러한 작동을 표시하며 이러한 배치가 갖게 되는 영향을 지적해 줍니다.
4. 왼쪽 마우스 단추를 클릭하여 JScrollPane 및 JTable을 열 0 및 행 1 셀에 끌어 놓으십시오.



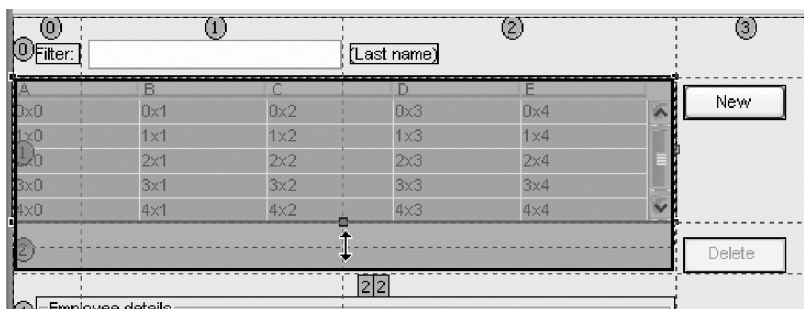
## JScrollPane 및 JTable을 눈금의 여러 행과 열로 늘리기

이제 더 나은 간격과 크기 조정을 위해 JScrollPane(및 해당되는 하위 JTable)을 세 개의 열과 두 개의 행으로 늘려보겠습니다. 테이블을 열 및 행으로 늘리려면 다음을 수행하십시오.

1. 디자인 영역이나 Java Bean 보기에서 JScrollPane을 선택하십시오. (방금 추가했으므로 계속 선택되어 있어야 합니다.) JScrollPane의 오른쪽과 맨 아래에 있는 작은 초록색 사각형을 보십시오. 이 크기 조정 핸들을 사용하여 JScrollPane을 끌어 여러 열과 행으로 늘려보십시오.
2. JScrollPane의 오른쪽에 있는 녹색 핸들을 왼쪽 마우스 단추로 클릭하고 계십시오.
3. 위치가 열 2, 행 1이 될 때까지 마우스 포인터를 오른쪽으로 끄십시오. 마우스 단추를 놓으면 어두운 회색 음영이 컴포넌트가 위치한 셀을 표시됩니다.



4. 마우스 단추를 놓으십시오. 이제 JScrollPane은 세 개 열로 늘어납니다.
5. 유사한 과정을 반복하여 JScrollPane이 행 2가 될 때까지 JScrollPane의 맨 아래 핸들을 끄십시오.



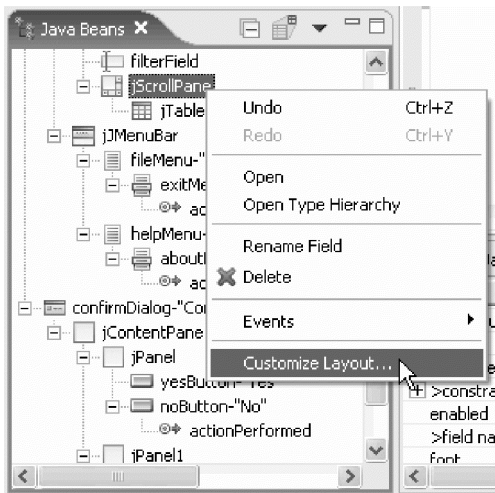
## 눈금 상자 내에서 JScrollPane의 간격을 사용자 정의하기

GridBagLayout 관리자의 또 다른 기능은 레이아웃을 추가로 사용자 정의하는 데 여러 가지 제한조건을 지정할 수 있다는 것입니다. 예를 들어, 다음 제한조건을 지정할 수 있습니다.

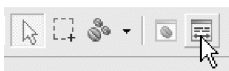
- **앵커:** 셀 안에서 컴포넌트에 앵커 방향을 지정할 수 있으며 이는 사용자가 응용프로그램의 크기를 다시 조정할 때 컴포넌트의 이동 방식에 영향을 줍니다. 예를 들어 구성요소를 상단 왼쪽, 가운데 왼쪽, 가운데 또는 하단 오른쪽에 정착시킬 수 있습니다.
- **채우기:** 셀 안에서 수평, 수직 또는 이 두 방향으로 사용 가능한 모든 공간을 차지하도록 컴포넌트에 지시할 수 있습니다.
- **삽입:** 컴포넌트와 눈금의 가장자리 사이에 간격을 두기 위해 맨 위, 맨 아래, 왼쪽 및 오른쪽에 채우기를 지정할 수 있습니다.

JScrollPane에 대한 앵커, 채우기 및 삽입을 사용자 정의하려면 다음을 수행하십시오.

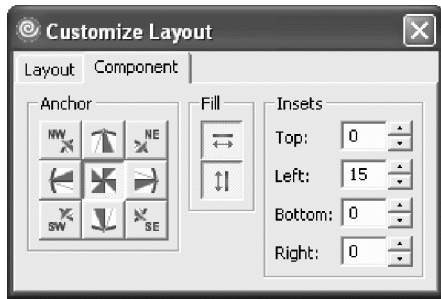
1. 디자인 보기 또는 Java Bean 보기에서 JScrollPane을 마우스 오른쪽 단추로 클릭하고 레이아웃 사용자 정의를 선택하십시오.



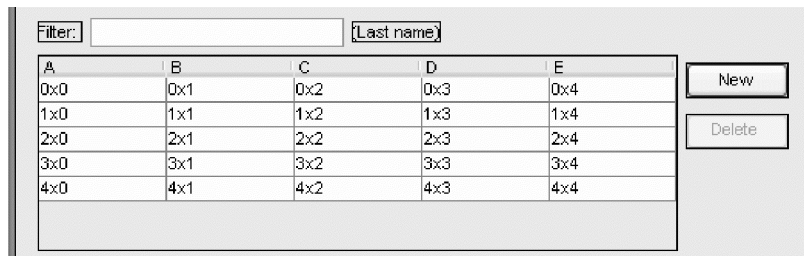
**팁:** 다른 컴포넌트에 대한 레이아웃을 선택하고 변경할 때 레이아웃 사용자 정의 대화 상자를 열어 둘 수 있습니다. 언제든지 메뉴 표시줄에서 레이아웃 사용자 정의 단추를 클릭하여 레이아웃 사용자 정의 대화 상자를 열 수 있습니다.



2. 레이아웃 사용자 정의 대화 상자의 컴포넌트 탭에서 앵커 가운데 단추를 눌렀는지 확인하십시오.
3. 가로 채우기 및 세로 채우기 단추가 모두 클릭되어 있는지 확인하십시오.
4. JScrollPane의 왼쪽 간격이 응용프로그램의 다른 비주얼 컴포넌트와 유사하도록 왼쪽 삽입으로 15(픽셀)를 추가하십시오.



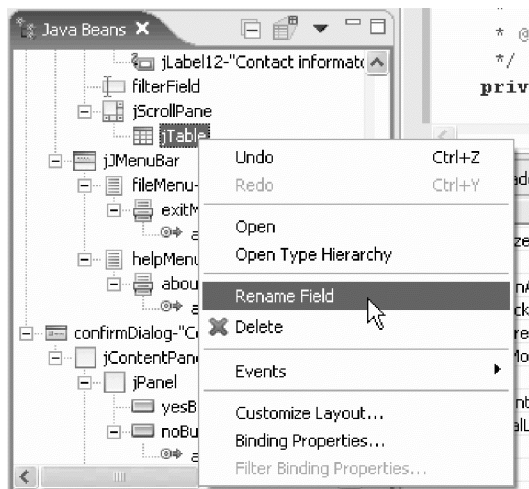
예를 들면 테이블이 필터 레이블에 맞게 정렬됩니다.



## 새 JTable의 이름을 유용한 값으로 바꾸고 단일 행을 선택하도록 설정

나중에 테이블에 대해 작업하게 되므로 JTable 인스턴스 및 해당되는 getter 메소드의 이름을 변경하는 것이 좋습니다. 테이블의 이름을 바꾸려면 다음을 수행하십시오.

1. Java Bean 보기에서 jTable 컴포넌트를 마우스 오른쪽 단추로 클릭하고 팝업 메뉴에서 필드 이름을 바꾸기를 선택하십시오.



2. employeesTable을 입력하고 확인을 클릭하십시오. JTable의 이름이 employeesTable로 변경되고 이를 인스턴스화하는 메소드는 getEmployeesTable이 됩니다.
3. 단일 행만 선택할 수 있도록 테이블을 설정하십시오.
  - a. 디자인 보기에서 employeesTable을 선택하십시오.
  - b. 특성 보기에서 **selectionMode** 특성을 선택하고 SINGLE\_SELECTION으로 설정하십시오.



Property	Value
preferredSize	375,80
rowHeight	16
rowSelectionAllowed	true
selectionBackground	49,106,197
selectionForeground	Color:white
selectionMode	SINGLE_SELECTION
showGrid	
showHorizontalLines	true
showVerticalLines	true
toolTipText	
visible	true

c. DirectoryApp.java 파일을 저장하십시오.

## 학습 체크포인트

Visual Editor를 사용하여 기존 사용자 인터페이스에 테이블을 삽입하는 방법을 학습했습니다. 따라서 레이아웃, 위치 및 간격 지정을 사용자 정의해 볼 수 있습니다.

## 단원 1.3: 비주얼 클래스 실행

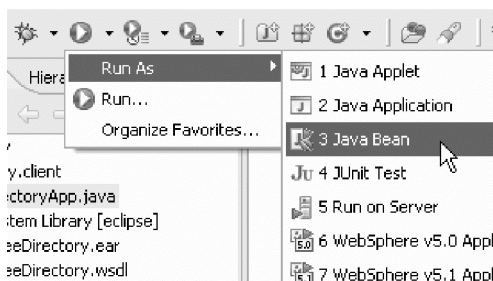
이제 Java 응용프로그램을 실행하여 해당 모양을 미리볼 수 있습니다. Workbench 및 Visual Editor를 사용하면 사용자 응용프로그램을 바로 실행하기가 아주 쉬우며 개발 과정의 어느 시점에서든지 이러한 단계를 반복하여 클래스의 실제 모양과 작동을 테스트할 수 있습니다.

데모 보기

Java Visual Editor는 main() 메소드 없이 클래스를 실행할 수 있는 Java Bean 실행기를 제공합니다. 비주얼 클래스를 실행하는 경우, Visual Editor는 독립된 가상 시스템(VM)에서 응용프로그램을 실행합니다. Java 응용프로그램과 같은 비주얼 클래스를 실행하는 경우, 실행기는 클래스의 main() 메소드를 실행하려고 합니다. 이 학습서의 경우 응용프로그램은 DirectoryApp JFrame을 호출하여 표시하는 main() 메소드를 포함하므로, 이 메소드를 응용프로그램이나 Java Bean으로 실행할 수 있습니다.

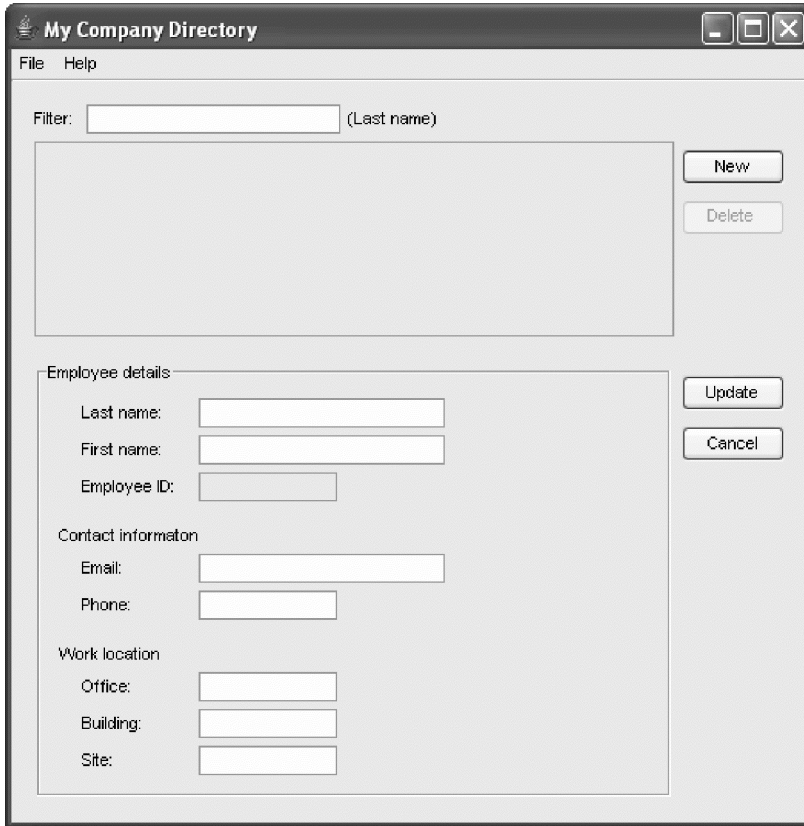
DirectoryApp.java 파일을 Java Bean으로 실행하려면 다음을 수행하십시오.

1. Java Visual Editor에서 DirectoryApp.java 파일이 열려 있는지 확인하십시오.
2. 메뉴 표시줄에서 실행 → 실행 도구 → **Java Bean**을 클릭하십시오.



**팁:** 응용프로그램은 Visual Editor 환경 설정(창 → 환경 설정 → **Java** → **Visual Editor**)에서 정의한 Swing 룩앤필을 사용하여 데스크탑에서 열립니다. 또는 실행 → 실행을 클릭하여 이 Java Bean을 시작할 특정 실행

행 구성의 룩앤필을 정의할 수도 있습니다. 이 응용프로그램을 Bean이 아닌 응용프로그램으로 실행할 경우 Windows 룩앤필이 main() 메소드에 정의되어 있으므로 이 룩앤필을 사용할 수도 있습니다. 이 학습서에 사용되는 화면 그림은 Windows 룩앤필을 보여줍니다.



## 학습 체크포인트

인터페이스를 지정했지만 데이터 연결이나 이벤트 기능은 프로그래밍하지 않았으므로 응용프로그램에 대해 어떤 것도 수행할 수 없습니다. 그러나 기본 레이아웃과 모양은 사용자에게 표시되므로 볼 수 있습니다. 몇몇 단추를 클릭해 볼 수도 있지만 이의 작동은 보지 못할 것입니다. 그러나 파일 메뉴와 도움말 메뉴는 이미 구현되어 있습니다. 이들 메뉴의 작동은 확인해 볼 수 있으며 Java 코드를 검사하여 actionPerformed 이벤트에서 이들이 어떻게 구현되는지도 볼 수 있습니다.

## 학습한 내용

이 모듈은 Java Visual Editor를 사용하여 리치 클라이언트용 인터페이스를 디자인하기 위한 것이었습니다. 그러나 클라이언트의 시각적 모양을 디자인하는 것 외에, 실제로 클라이언트를 유용하게 만들기 위해서는 수행해야 할 작업이 더 있습니다. 일반적인 경우 이벤트 작동이나 기타 로직(이 경우 비주얼 요소를 특정 정렬의 데이터 소스에 바인딩)을 포함시키기만 하면 됩니다.

이 모듈에서는 다음 작업을 수행하는 방법에 대해 학습하였습니다.

- 프로젝트 교환 가져오기를 사용하여 Java 프로젝트 가져오기
- JScrollPane의 JTable을 비주얼 클래스에 추가

- GridBagLayout 관리자를 사용하여 리치 클라이언트에 테이블을 시각적으로 레이아웃
- 응용프로그램을 실행하여 리치 Java 클라이언트의 모양 확인

다음 모듈인 모듈 2: 비주얼 컴포넌트를 웹 서비스에 바인드에서는 간단한 회사 디렉토리 인터페이스를 가져와서 웹 서비스 메소드에 액세스하는 강력한 리치 클라이언트로 전환함으로써 회사 디렉토리에서 직원 레코드를 작성, 검색, 갱신 및 삭제할 수 있습니다.

## 모듈 2: 비주얼 구성요소를 웹 서비스에 바인드

이 모듈에서는 회사 디렉토리의 비주얼 요소(단추, 직원 테이블, 필드 및 기타 조치)를 웹 서비스에 바인드하는 방법에 대해 설명합니다. 웹 서비스는 샘플 디렉토리에서 직원을 작성, 검색, 갱신 및 삭제할 수 있는 실제 기능을 제공합니다.

### 학습 목표

이 모듈의 단원을 완료하였으므로 개념을 이해하고 다음을 수행할 수 있습니다.

- 테이블을 데이터 웹 서비스 데이터 소스에 바인드
- 필드를 오브젝트에 바인드
- 조치 단추 프로그래밍

이 모듈을 완료하는 데 약 2시간이 소요됩니다.

### 단원 2.1: 웹 서비스 설치 및 배치

이 단원에서는 샘플 엔터프라이즈 응용프로그램(EAR) 파일을 WebSphere Application Server v6.1에 설치하고 EmployeeDirectory 웹 서비스를 배치합니다. 사용자의 응용프로그램은 이 웹 서비스를 사용하여 직원 레코드를 작성, 읽기, 갱신 및 삭제합니다.

시작하기 전에 다음 옵션중 하나를 완료하여 MyDirectory가 시작 준비가 된 프로젝트인지 확인해야 합니다.

- 3 페이지의 『모듈 1: Visual Editor에서 클라이언트 GUI 디자인』 완료

또는

- 모듈 2 시작점에서 MyDirectory 프로젝트 가져오기

팁: 가져올 때 프로젝트 이름을 다르게 지정하지 않으면 MyDirectory 프로젝트 콘텐츠를 겹쳐씹니다.

MyDirectory Java 프로젝트에는 EmployeeDirectory.ear 파일이 포함되어 있습니다. WebSphere 관리 콘솔을 사용하여 EAR 파일에 포함되는 EmployeeDirectory 엔터프라이즈 응용프로그램을 설치하십시오. 응용프로그램 설치 시, 응용프로그램에 포함된 웹 서비스도 배치하십시오. 완료된 회사 디렉토리 응용프로그램은 배치된 이 웹 서비스를 사용합니다.

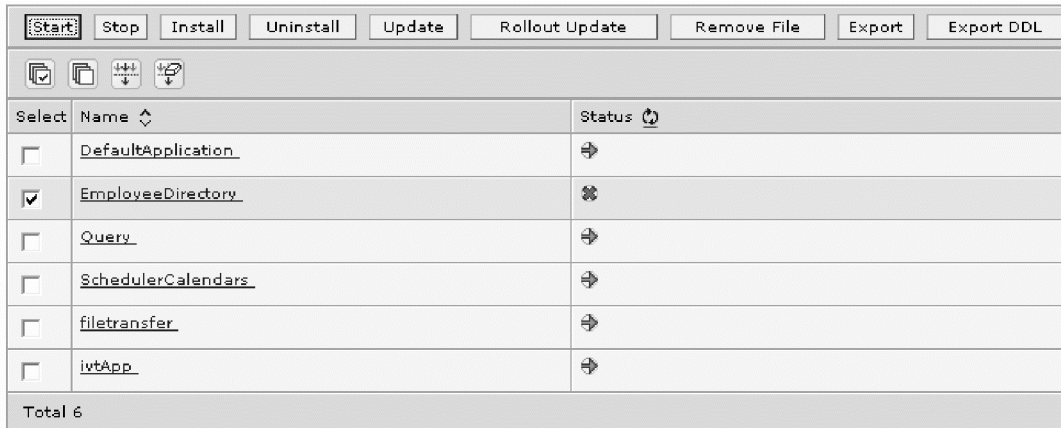
샘플 EmployeeDirectory 응용프로그램을 설치하고 WebSphere Application Server v6.1 환경에서 웹 서비스를 배치하려면 다음을 수행하십시오.

1. Workbench에서 응용프로그램 서버 인스턴스를 시작하십시오. 서버를 실행할 수 있는 몇 가지의 다른 방법이 있지만 여기에서는 Workbench에서 이를 수행하는 방법에 대해 설명합니다.
  - a. 서버 보기를 여십시오. 서버 보기를 Java Perspective에 추가하려면 창 → 보기 표시 → 기타를 클릭한 후 서버 → 서버를 선택하십시오.
  - b. 서버 보기에서 설치 및 설정된 서버를 나열합니다.
  - c. 서버를 마우스 오른쪽 단추로 클릭하고 시작을 선택하십시오. 서버 보기에 서버 상태가 시작됨으로 표시되거나 콘솔에 server1 서버가 e-business에 대해 열려 있음이라고 표시되면 서버가 시작된 것입니다. 이제 관리 콘솔을 실행할 수 있습니다.

주: 서버 보기에 서버 인스턴스가 없으면 다음과 같이 새 서버를 작성하십시오.

- a. 서버 보기에서 마우스 오른쪽 단추를 클릭하고 새로 작성 → 서버를 선택하십시오.
- b. WebSphere Application Server v6.1을 추가하려면 새 서버 마법사를 사용하십시오.
2. Websphere 관리 콘솔을 실행하십시오. 마찬가지로, 관리 콘솔을 실행할 수 있는 여러 방법이 있지만 여기에서는 Workbench에서 이를 수행하는 방법에 대해 설명합니다.
  - a. 서버 보기에서 방금 시작한 서버를 마우스 오른쪽 단추로 클릭하고 관리 콘솔 실행을 선택하십시오. Websphere 관리 콘솔이 브라우저 창에 열립니다.
  - b. 사용자 ID를 입력하고 로그인을 클릭하십시오. 관리 콘솔의 환영 페이지가 열립니다. 입력한 사용자 ID는 서버의 구성 데이터에 대한 사용자 특정 변경사항을 추적하는 데만 사용됩니다.
3. 엔터프라이즈 응용프로그램을 사용하여 MyDirectory 프로젝트에서 발견되는 EmployeeDirectory.ear 엔터프라이즈 응용프로그램을 설치하십시오. 관리 콘솔은 마법사 방식을 사용하여 사용자가 응용프로그램을 설치하는 것을 돕습니다. 사용자가 다음을 클릭하면 모든 옵션이 설정될 때까지 페이지 사이를 이동할 수 있습니다. 이 학습서에 필요한 웹 서비스의 샘플 엔터프라이즈 응용프로그램을 설치하려면 다음을 수행하십시오.
  - a. 관리 콘솔의 왼쪽에서 응용프로그램 메뉴 옵션을 펼치고 새 응용프로그램 설치를 클릭하십시오.
  - b. 로컬 파일 시스템을 선택하고 경로 지정 필드에 MyDirectory 프로젝트에 있는 EmployeeDirectory.ear 파일의 전체 경로를 입력하십시오. 팁: 전체 경로를 가져오려면 패키지 탐색기에서 EmployeeDirectory.ear 파일을 마우스 오른쪽 단추로 클릭하고 특성을 선택하십시오. 특성 페이지에 파일 위치가 나열됩니다. 파일 위치를 복사하여 경로 지정 필드에 붙여넣을 수 있습니다.
  - c. 설치 옵션 선택 페이지가 나올 때까지 다음을 클릭하십시오.
  - d. 웹 서비스 배치를 선택하십시오.
  - e. 요약 페이지가 나올 때까지 다음을 클릭한 다음 완료를 클릭하십시오.
  - f. 로컬 구성에 대해 수행한 변경사항을 적용하도록 요청하는 구성요소가 표시될 때 마스터 구성에 저장 링크를 클릭하십시오. 변경사항을 검토하고 저장 단추를 클릭하십시오.
4. 관리 콘솔을 사용하여 EmployeeDirectory 응용프로그램을 시작하려면 다음을 수행하십시오.
  - a. 응용프로그램 → 엔터프라이즈 응용프로그램을 클릭하십시오. EmployeeDirectory 응용프로그램은 서버에 설치된 응용프로그램으로 나열되지만 그 상태는 중지됨입니다.





- b. EmployeeDirectory 옆에 있는 선택란을 선택하고 시작을 클릭하십시오. EmployeeDirectory 응용프로그램이 시작되었다는 메시지가 표시되고, 상태 아이콘이 녹색 화살표로 변경됩니다.

이제 EmployeeDirectory 응용프로그램은 포트 9080에 있는 로컬 호스트에서 실행되며 웹 서비스에도 액세스할 수 있습니다. 이 학습서를 완료하고 나면 관리 콘솔로 돌아가서 EmployeeDirectory 응용프로그램을 중지한 후 설치 제거하십시오.

MyDirectory 프로젝트에서 발견된 EmployeeDirectory.wsdl 파일을 열 경우(기본적으로 그래픽 WSDL 편집기에 열려 있음), 방금 배치한 웹 서비스를 조사할 수 있습니다. WSDL 편집기에 WSDL 파일이 열려 있지 않으면 Workbench에서 웹 서비스 개발자 성능이 실행되지 않았을 수 있습니다. 환경 설정에서 Workbench 성능을 지정할 수 있습니다(창 → 환경 설정 → **Workbench** → 성능).

WSDL 편집기의 다음 이미지는 EmployeeDirectory 서비스에서 사용 가능한 조작을 표시합니다.



WSDL 편집기를 사용하여 각 조작 및 해당되는 요청 메시지와 리턴 메시지를 조사할 수 있습니다. 이는 사용자가 웹 서비스와 나머지 연습에서 웹 서비스를 사용하는 방법을 이해하는 데 도움을 줍니다.

## 단원 2.2: 직원 테이블을 웹 서비스 데이터 소스에 바인드

회사 디렉토리 응용프로그램은 디렉토리에 있는 현재의 모든 직원 레코드 목록을 표시합니다. 레코드는 JTable(employeesTable)에서 정렬 가능한 열(성, 이름, 전자 우편 및 직원 ID)과 함께 표시됩니다. 테이블에 대한 레코드를 가져오려면 샘플 웹 서비스 데이터 소스에서 리턴된 데이터 오브젝트에 employeesTable을 바인드해야 합니다.

데모 보기

## 데이터 오브젝트, 데이터 소스 및 바인더의 개요

작업할 `employeesTable`의 로컬 데이터 오브젝트를 가져오기 위해 Visual Editor를 사용하여 데이터 소스를 응용프로그램에 추가합니다. 데이터 소스는 샘플 웹 서비스 프록시에 연결하고 응용프로그램에 사용할 수 있는 서비스 메소드를 발견합니다. 그런 다음 데이터 소스에서 사용 가능한 `getLightEmployeeRecord` 서비스를 선택하십시오. 마지막으로 사용자 응용프로그램의 `employeesTable`을 행 데이터 오브젝트(`lightEmployeeRecord`)로 리턴되는 필드에 바인드합니다.

Java Visual Editor의 내장 바인더 클래스를 사용하여 빠르고 쉽게 이 모든 데이터 소스 및 데이터 오브젝트를 작성할 수 있습니다. Visual Editor는 비주얼 구성요소를 데이터 팩토리에 바인드하는 대로 프로젝트에 생성되는 일반 인터페이스 및 클래스 세트를 제공합니다. 기본적으로 `jve.generated` 패키지에 바인더 클래스가 생성됩니다. Visual Editor는 추가로 응용프로그램 필요에 맞도록 사용자 정의하고 향상시킬 수 있는 일반 구현으로 바인더 클래스를 제공합니다. 이 학습서에서는 기본적이고 간단하지만 기본 바인더 클래스 사용의 유동성 및 강력한 기능에 대해 설명합니다.

**중요사항:** 이 연습을 시작하기 전에 다음 도움말 항목을 꼭 읽어 보는 것이 좋습니다. 이 항목은 데이터 오브젝트, 데이터 소스, Java Visual Editor가 제공하는 바인더 외에 기능 및 로직에 대해 자세히 학습하는 데 도움이 될 수 있습니다.

- 데이터 바인더 개요
- 바인더 API 참조

이 학습서에서는 웹 서비스 데이터 소스, 몇 가지 유형의 데이터 오브젝트 및 몇 가지 유형의 바인더를 사용자 응용프로그램에서 사용합니다. 이러한 오브젝트의 인스턴스를 응용프로그램에 추가할 때 Visual Editor는 필요한 클래스를 프로젝트의 `jve.generated` 패키지에 추가합니다. 이 때 데이터 바인딩 로직을 확장하거나, 바꾸거나 다시 작성할 수 있습니다. Java Visual Editor는 사용자 응용프로그램이 사용 중인 데이터 오브젝트, 데이터 소스 및 바인더를 디자인 보기의 자유 양식 영역에 표시하여 바인더 오브젝트에 관한 시각적 지원을 제공합니다. Visual Editor는 선택한 오브젝트의 현재 바인더를 표시하기 위해 비주얼 구성요소와 데이터 오브젝트 및 데이터 소스 사이에 선을 그립니다.

다음 다이어그램은 비주얼 구성요소, 바인더, 데이터 오브젝트 및 데이터 소스가 상호작용하는 방법에 대한 간단한 개요입니다. 이 학습서에서 빌드할 응용프로그램은 약간 더 복잡하고 창조적으로 바인더를 사용합니다. 다음 다이어그램은 샘플 응용프로그램에서 빌드 중인 바인더, 데이터 오브젝트 및 데이터 소스를 정확히 표시한 것은 아닙니다.

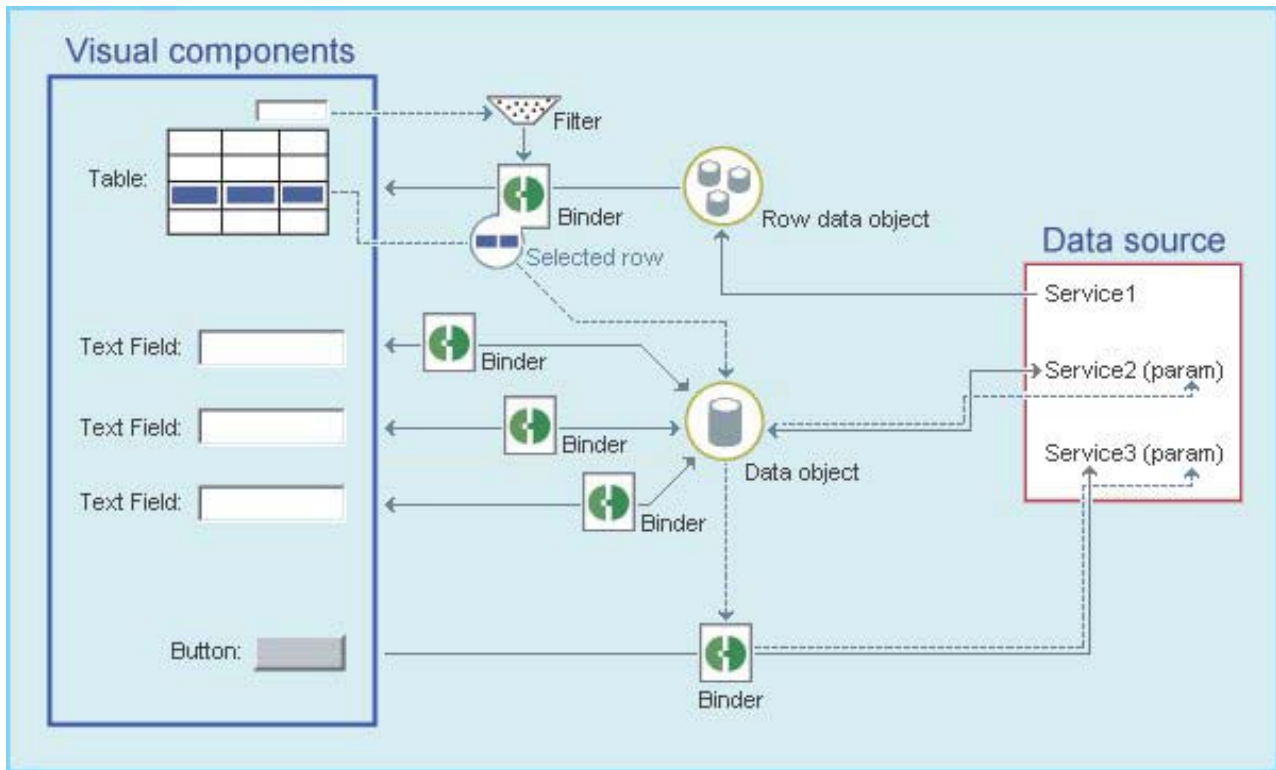


그림 1. 다음 다이어그램은 비주얼 구성요소, 바인더, 데이터 오브젝트 및 데이터 소스 간의 샘플 관계를 보여줍니다.

그림 1에서, 각 비주얼 구성요소에는 데이터 오브젝트(단추의 경우에는 데이터 소스)와 연관시키는 자체 바인더가 있습니다. 텍스트 필드의 바인더는 데이터 오브젝트의 특정 특성에 필드를 바인드합니다. 이 다이어그램의 데이터 오브젝트와 행 데이터 오브젝트 모두 데이터 소스에서 서비스에 대한 직접 호출을 통해 데이터를 가져옵니다. 텍스트 필드의 데이터 오브젝트는 테이블에서 선택한 행의 키 값을 Service2를 호출하기 위한 인수로 사용합니다. Service2는 테이블에서 선택한 행에 대한 더 많은 정보를 포함하고 있을 수 있는 전체 레코드를 리턴합니다. 이 전체 레코드는 다시 Service3을 호출할 때 단추의 조치 바인더에 대한 인수로 사용합니다. Service3은 필드에 입력한 값을 갱신하는 메소드일 수 있습니다. 데이터 오브젝트, 데이터 바인더 및 데이터 소스에 대한 자세한 설명은 앞에 제공된 링크를 참조하십시오.

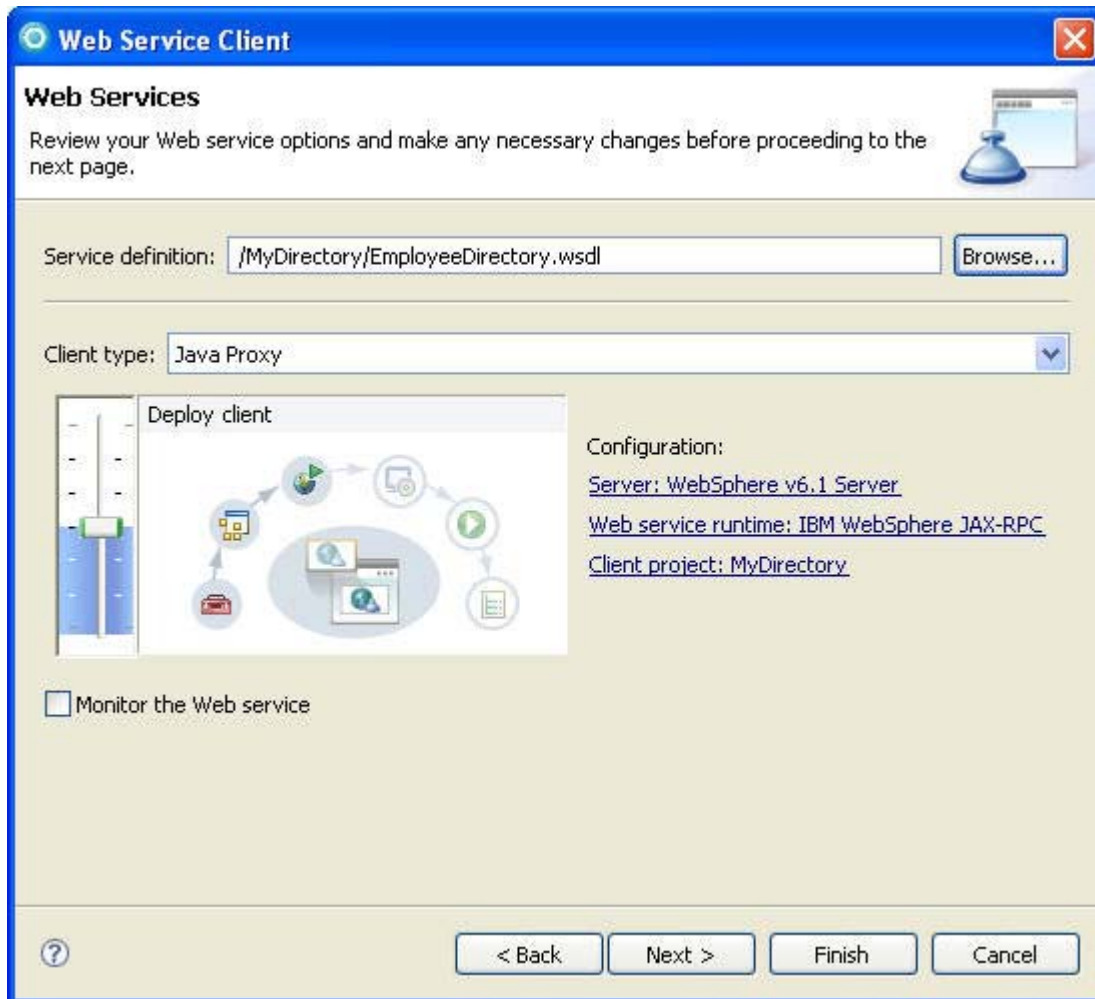
### 제공된 WSDL 파일을 사용하여 프로젝트에서 웹 서비스 Java 프록시 생성

서버에서 실행 중인 웹 서비스에 대해 작업하려면 Java 응용프로그램이 상호작용할 Java 프록시나 클라이언트가 필요합니다. WSDL 파일을 사용할 경우, 웹 서비스 클라이언트 마법사를 사용하여 Java 프로젝트에 Java 프록시를 생성할 수 있습니다. MyDirectory 프로젝트에는 이 프록시를 생성하기 위해 사용할 EmployeeDirectory.wsdl 파일이 포함되어 있습니다. Java 프록시를 생성한 후 웹 서비스를 나타내는 데이터 소스를 작성하고 바인딩 비주얼 구성요소를 시작할 수 있습니다.

**중요사항:** 이 연습에서 사용하는 WSDL 파일은 WebSphere Application Server의 로컬 설치에 웹 서비스를 배치하고 로컬 호스트(http://localhost:9080)의 기본 포트를 사용했다고 가정합니다. EAR 파일을 다르게 배치 했으면 진행하기 전에 적절하게 WSDL 파일을 편집해야 합니다.

프로젝트에서 웹 서비스 Java 프록시를 생성하려면 다음을 수행하십시오.

1. 기본 메뉴에서 파일 → 새로 작성 → 기타를 클릭한 후 웹 서비스 → 웹 서비스 클라이언트 마법사를 선택하십시오. 웹 서비스 카테고리가 표시되지 않으면 모든 마법사 표시를 선택하십시오.
2. 다음과 같이 마법사를 사용하여 웹 서비스 클라이언트를 정의하십시오.
  - a. 서비스 정의의 경우, MyDirectory 프로젝트에 제공된 WSDL 파일(/MyDirectory/EmployeeDirectory.wsdl)을 입력하십시오.
  - b. 클라이언트 유형 필드에서 **Java** 프록시를 선택하십시오.
  - c. 슬라이더 막대를 클라이언트 배치로 설정하십시오.
  - d. 서버가 실행 중이며 웹 서비스 런타임이 실행 중인 서버에 대해 적절히 설정되었는지 확인하십시오. 이 학습서는 IBM WebSphere JAX-RPC 런타임을 사용하여 WebSphere v6.0 및 WebSphere v6.1에서 테스트되었습니다.
  - e. Java 프록시 클라이언트가 MyDirectory 프로젝트로 출력되는지 확인하십시오.



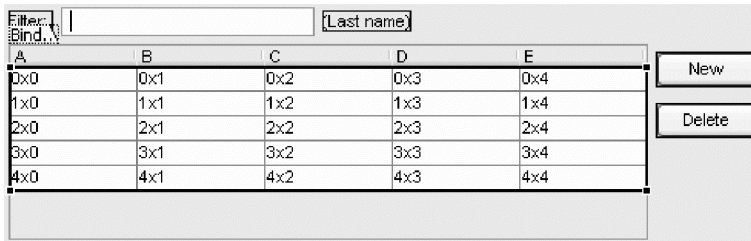
3. 완료를 클릭하십시오. 웹 서비스 클라이언트 마법사는 Java 프록시를 생성한 후 프로젝트의 새 패키지 (directory.service)에 추가합니다.

## employeesTable을 웹 서비스에서 리턴된 행 데이터 오브젝트에 바인드

employeesTable은 이 응용프로그램에서 바인드하는 첫 번째 비주얼 구성요소이므로 방금 프로젝트에 추가한 샘플 웹 서비스 프록시를 지시하는 데이터 소스를 작성해야 합니다. 나중 연습에서 다른 비주얼 구성요소를 바인드할 때 이 데이터 소스를 재사용합니다. 이 단계에서는 웹 서비스 데이터 소스와 lightEmployeeRecordRows 데이터 오브젝트를 추가합니다.

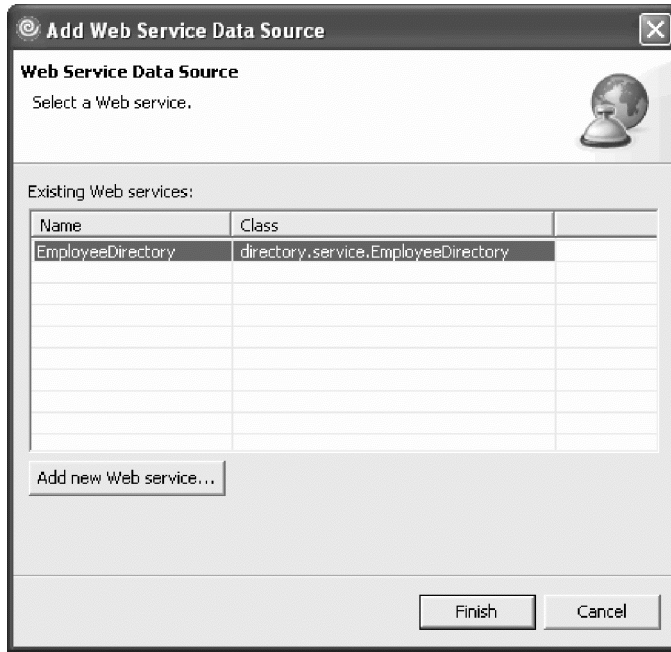
직원 테이블을 바인드하려면 다음을 수행하십시오.

1. Java Bean 보기 또는 디자인 보기에서 employeesTable을 선택하십시오. (해당되는 JScrollPane 상위 요소는 선택하지 마십시오.) 바인드라는 레이블의 작은 탭이 디자인 영역에서 employeesTable의 맨 위에 표시됩니다.

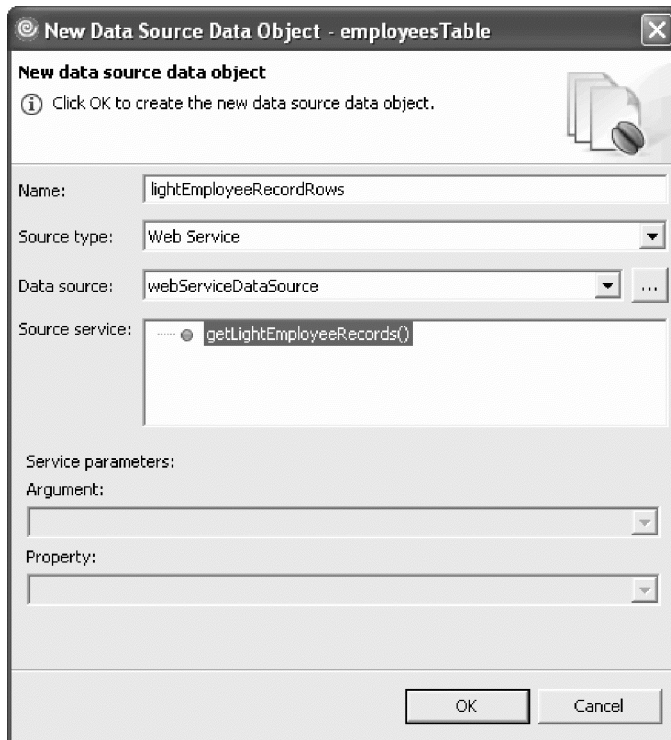


2. employeesTable에서 바인드 탭을 클릭하십시오. 대신 employeesTable을 마우스 오른쪽 단추로 클릭한 다음 바인딩 특성을 선택할 수도 있습니다.
3. 사용자 응용프로그램에는 데이터 오브젝트가 없으므로 새로 추가해야 합니다. 새 데이터 소스 데이터 오브젝트를 클릭하십시오.
4. 소스 유형 필드에서 웹 서비스를 선택하십시오.
5. 아직 웹 서비스 데이터 소스를 사용자의 응용프로그램에 추가하지 않았으므로 지금 추가해야 합니다. 데이터 소스 필드 옆에 있는 ... 단추를 클릭하여 웹 서비스 데이터 소스 추가 대화 상자를 여십시오. 이 대화 상자는 프로젝트에서 사용 가능한 웹 서비스 클라이언트나 프록시를 찾습니다.
6. EmployeeDirectory 웹 서비스를 선택하고 완료를 클릭하십시오. 다음과 같이 새 데이터 소스가 DirectoryApp.java 파일에 추가됩니다.



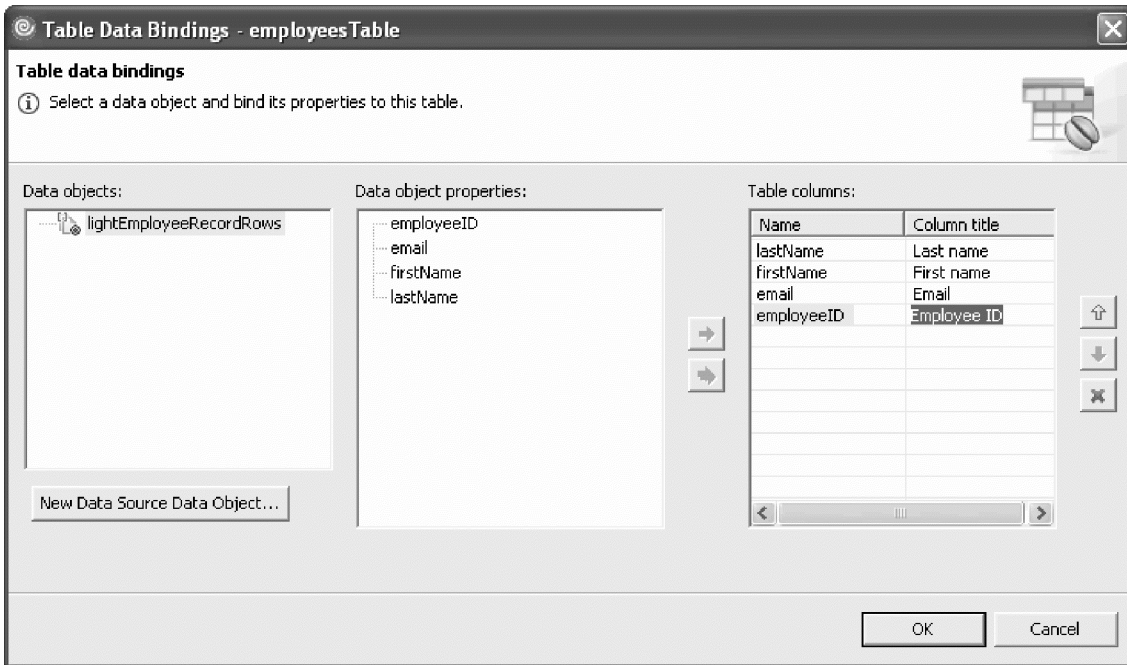


7. 새 데이터 소스 데이터 오브젝트 대화 상자의 소스 서비스 필드에서 `getLightEmployeeRecords()`를 선택하고 새 데이터 오브젝트의 기본 이름으로 `lightEmployeeRecordRows`를 사용하십시오. 이 서비스 메소드에 필요한 매개변수는 없습니다. 확인을 클릭하십시오. 새 데이터 오브젝트가 작성되어 디자인 보기의 자유 양식 영역에 표시됩니다.



팁: 테이블을 바인드하는 중이므로 새 데이터 소스 데이터 오브젝트 대화 상자는 행 데이터 오브젝트를 리턴하는 서비스만 표시합니다. 이 경우 `getLightEmployeeRecords()` 메소드는 오브젝트 배열을 리턴하는 유일한 사용 가능한 서비스입니다.


8. 테이블 데이터 바인딩 대화 상자에서 `lightEmployeeRecordRows` 데이터 오브젝트를 선택하십시오.
9. `employeesTable`에 표시하려는 `lightEmployeeRecordRows` 데이터 오브젝트의 특성을 선택하십시오.

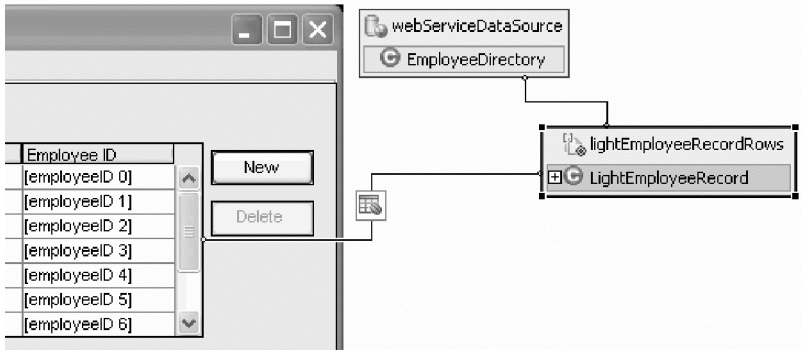


- a. 이중 화살표 ⇄ 단추를 클릭하여 모든 오브젝트 특성을 테이블 열 목록에 추가하십시오.
- b. 위로 및 아래로 화살표를 사용하여 위에서 아래 순(즉, `lastName`, `firstName`, `email`, `employeeID`)으로 열을 배열하십시오.
- c. 열 제목(성, 이름, 전자 우편, 직원 ID)의 이름을 변경하십시오.

팁: 테이블 바인드를 완료하고 나면 항상 바인딩 특성으로 돌아가서 언제든지 열의 이름을 바꾸거나 순서를 재지정할 수 있습니다.

- d. 확인을 클릭하십시오.

`employeesTable`이 `JRowTableBinder`를 사용하여 `lightEmployeeRecordRows` 데이터 오브젝트에 바인드되었습니다. 자유 양식 영역에서 `lightEmployeeRecordRows` 데이터 오브젝트를 클릭하면 Visual Editor는 데이터 오브젝트에서 테이블로 선을 그립니다. 이 선에서 `JRowTableBinder`는 테이블 바인더  아이콘으로 표시됩니다. 다른 선은 데이터 오브젝트가 `webServiceDataSource`를 해당 데이터 소스로 사용한다는 것을 표시합니다.



## 학습 체크포인트

프로젝트 및 응용프로그램 변경사항을 확인하십시오. 이 단원에서는 웹 서비스 데이터 소스, 행 데이터 오브젝트, `employeesTable`을 행 데이터 오브젝트에 바인드하는 바인더를 추가했습니다.

Java Visual Editor가 생성한 모든 바인더 클래스를 보유하는지 프로젝트에서 작성된 새 패키지(`jve.generated`)를 조사하십시오. 웹 서비스의 Java 프록시를 보유하는 새 패키지(`directory.service`)도 확인하십시오. 이 단원에서 배운 내용을 설명하거나 요약하십시오.



이제 회사 디렉토리 응용프로그램을 실행하면 직원 테이블은 웹 서비스를 통해 기존 직원 레코드로 채워집니다.

## 단원 2.3: 세부사항 필드를 테이블 선택사항에 바인드

이전 연습에서 `employeesTable`을 웹 서비스의 `getLightEmployeeRecords()` 서비스가 리턴하는 `lightEmployeeRecordRows` 데이터 오브젝트에 바인드했습니다. 이제는 테이블에서 선택하는 직원에 따라 세부사항 필드를 채워야 합니다.

선택된 각 직원에 대한 추가 세부사항을 가져오는 데 또 다른 데이터 오브젝트가 사용됩니다. 추가할 `selectedEmployeeRecord` 데이터 오브젝트는 `getFullEmployeeRecord()` 서비스에서 리턴됩니다. 이 서비스는 테이블에서 선택된 직원의 ID를 매개변수로 사용하여 직원에 대한 추가 세부사항(예: 전화번호 및 작업 위치)을 보충합니다.

테이블을 행 데이터 오브젝트에 바인드할 때 사용했던 `JRowTableBinder`를 사용하면 이 단계를 간소화할 수 있습니다. `JRowTableBinder` 바인더는 테이블에서 선택한 요소를 `getFullEmployeeRecord(Integer)` 메소드에 대한 매개변수로 사용할 수 있는 별도의 데이터 오브젝트로 내놓습니다. 그러면 각 텍스트 필드를 `selectedEmployeeRecord` 데이터 오브젝트의 해당 특성에 쉽게 바인드할 수 있습니다.

**이 웹 서비스에 대한 추가 학습:** 웹 서비스에 포함된 두 개의 서비스로 각 직원의 모든 세부사항을 가져옵니다. 테이블에는 모든 직원이 나열되기 때문에 테이블에는 데이터의 서브세트만 표시됩니다. 이 때 단일 직원을



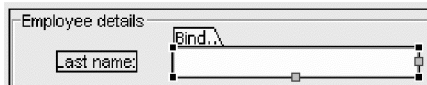
선택하면 선택된 해당 직원에 대해서만 나머지 직원 정보를 검색할 수 있습니다. 테이블이 데이터를 요청할 때 웹 서비스가 각 직원에 대한 모든 정보를 송신한 경우, 웹 통신량이 과도해져 응용프로그램 성능 저하를 초래할 수 있습니다.

예를 들어, 직원 레코드에 사진이나 첨부이 포함된 경우, 단순히 전체 직원 목록을 가져올 때 모든 사진을 검색하고 싶지 않을 수 있습니다. 따라서 테이블을 채우는 데에는 `getLightEmployeeRecord` 서비스가 사용되고 `getFullEmployeeRecord`는 테이블에 선택된 직원에 대한 전체 레코드를 가져옵니다.

## 성 필드 바인드

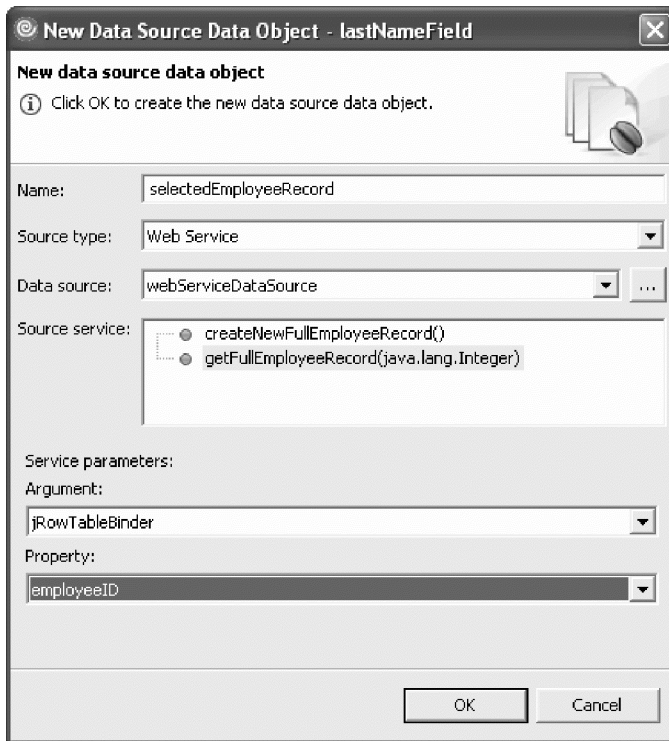
이 단계에서는 `selectedEmployeeRecord` 데이터 오브젝트의 `lastName` 특성에 성 필드를 바인드합니다.

1. Java Beans 보기나 디자인 보기에서 성(`lastNameField`)에 대해 `JTextField`를 선택하십시오. 디자인 영역의 텍스트 필드에 바인드 탭이 표시됩니다.

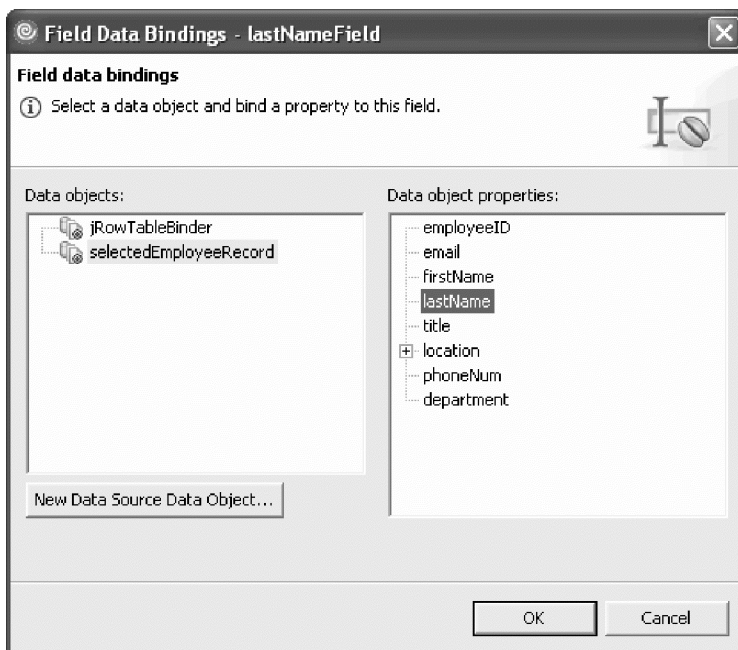


2. 바인드 탭을 클릭하여 필드 데이터 바인딩 대화 상자를 여십시오.
3. 새 데이터 소스 데이터 오브젝트를 클릭하십시오. 기존 `JRowTableBinder` 데이터 오브젝트가 정확한 성을 리턴하더라도 여기에는 전체 직원 레코드가 포함되어 있지 않습니다. 따라서 전체 직원 레코드를 나타내는 새 데이터 오브젝트를 작성해야 합니다.
4. 소스 유형 필드에서 웹 서비스가 선택되었는지, 데이터 소스에 대해 `webServiceDataSource`가 선택되었는지 확인하십시오.
5. 소스 서비스 목록에서 `getFullEmployeeRecord(java.lang.Integer)`를 선택하십시오. 새 데이터 소스 데이터 오브젝트 대화 상자는 텍스트 필드와 호환 가능한 데이터 오브젝트를 리턴하는 서비스를 나열합니다.
6. 이름 필드에 `selectedEmployeeRecord`를 입력하십시오.
7. 인수 필드에서 `JRowTableBinder`를 선택하고 특성 필드에서 `employeeID`를 선택하십시오. 이제 선택된 행의 직원 ID가 `getFullEmployeeRecord()` 서비스 메소드의 인수로 설정됩니다.

주: `getFullEmployeeRecord(java.lang.Integer)`는 인수로 정수를 필요로 합니다. 직원 테이블에서 현재 선택한 직원 ID를 사용하여 전체 레코드를 검색할 수 있습니다. 테이블을 바인드했을 때 Visual Editor가 자동으로 `JRowTableBinder`를 생성했으며 이를 통해 직원 테이블의 현재 선택 상태를 청취할 수 있습니다. 정수 매개변수의 경우 `JRowTableBinder`에서 선택된 행의 직원 ID를 사용합니다.



8. 확인을 클릭하십시오.
9. 필드 데이터 바인딩 대화 상자에서 데이터 오브젝트 목록에 selectedEmployeeRecord가 선택되었는지 확인하십시오. jRowTableBinder 데이터 오브젝트보다 selectedEmployeeRecord 데이터 오브젝트에 사용 가능한 특성이 더 많습니다.
10. 데이터 오브젝트 특성 목록에서 lastName 특성을 선택하십시오.



11. 확인을 클릭하십시오. 응용프로그램의 성 필드는 이제 `selectedEmployeeRecord` 데이터 오브젝트의 `lastName` 특성에 바인드됩니다. 이 오브젝트는 `getFullEmployeeRecord()`에 의해 리턴됩니다.

`selectedEmployeeRecord`라는 이름의 새 데이터 오브젝트가 작성되어 사용자 응용프로그램에 추가됩니다. 데이터 오브젝트의 비주얼 표시는 다음 이미지에 표시된 대로 디자인 보기의 자유 양식 영역에 추가됩니다.



이제 디자인 영역의 `lastName` 필드를 선택하면 이 필드가 `selectedEmployeeRecord`에 바인드되었음을 표시하는 선이 나타납니다. 이 선 가운데에 있는 텍스트 바인더 아이콘은 이 바인딩에 사용된 `SwingTextComponentBinder`를 나타냅니다. 디자인 영역에서 바인더를 나타내는 아이콘이나 선을 선택하는 경우 특성 보기에서 바인더의 특성을 조사할 수 있습니다.

## 나머지 세부사항 필드 바인드

직원에 대한 나머지 세부사항 필드 각각을 바인드하려면 성 필드와 유사한 프로세스를 따르면 되지만 데이터 오브젝트를 추가할 필요는 없습니다. 이미 `selectedEmployeeRecord` 데이터 오브젝트를 추가했으므로 각 필드를 `selectedEmployeeRecord` 데이터 오브젝트의 해당 특성에 간단하게 바인드할 수 있습니다.

필드를 바인드하려면 응용프로그램의 직원 세부사항 섹션에 있는 각 필드에서 다음 단계를 완료하십시오.

1. 디자인 보기에서 필드를 선택하고 바인드 탭을 클릭하십시오.
2. 필드 데이터 바인딩 대화 상자에서 데이터 오브젝트 목록으로부터 `selectedEmployeeRecord`를 선택하십시오.
3. 데이터 오브젝트 특성 목록에서 바인드 중인 필드에 적절한 특성을 선택하십시오. 다음 도표에 각 텍스트 필드를 다음에 바인드해야 하는 특성이 나열되어 있습니다.

필드	<code>selectedEmployeeRecord</code> 데이터 오브젝트의 특성
<code>lastNameField</code>	<code>lastName</code>
<code>firstNameField</code>	<code>firstName</code>
<code>idField</code>	<code>employeeID</code>
<code>emailField</code>	<code>email</code>
<code>phoneField</code>	<code>phoneNum</code>
<code>officeField</code>	<code>location.office</code>
<code>buildingField</code>	<code>location.building</code>
<code>siteField</code>	<code>location.site</code>

4. 확인을 클릭하십시오.

텍스트 필드 바인딩을 완료하면 디자인 영역이 다음 이미지와 유사하게 표시됩니다.

Employee details

Last name: {lastName}

First name: {firstName}

Employee ID: {employeeID}

Contact information

Email: {email}

Phone: {phoneNum}

Work location

Office: {location.office}



Building: {location.building}

Site: {location.site}

## 직원 ID 필드를 읽기 전용으로 만들기

필드의 편집 가능 특성이 `false`로 설정되어 있으므로 직원 ID 필드는 사용 불가능합니다. 그러나 텍스트 필드 바인더의 기본 작동은 데이터 오브젝트에 값이 있을 때 필드의 사용 가능 상태를 변경시킵니다. 따라서 필드가 초기 읽기 전용 상태로 남아 있도록 이 바인더 동작을 끌 수 있습니다.

바인더가 편집 가능한 특성으로 자동 전환되지 않게 하려면 다음을 수행하십시오.

1. 직원 ID 필드를 선택하십시오. 필드의 바인더를 표시하는  아이콘이 있는 선이 디자인 영역에 표시됩니다.
2. 직원 ID 필드에 대한 바인더 아이콘()을 클릭하십시오.
3. 특성 보기에서 `autoEditable` 특성을 **false**로 변경하십시오. **Enter**를 누르십시오.

## 학습 체크포인트

이제 응용프로그램을 실행하고 테이블에서 직원을 선택하면 해당 직원 레코드의 세부사항이 세부사항 필드에 표시됩니다.

## 단원 2.4: 갱신 단추를 조치 바인더에 바인드

Java Visual Editor는 단추를 클릭할 때 데이터 소스에 대해 서비스를 호출하기 위한 조치 바인더를 제공합니다. 예를 들어, 갱신 단추를 클릭하면 응용프로그램은 세부사항 필드에 입력된 변경사항으로 웹 서비스에서 `modifyEmployee()` 메소드를 실행해야 합니다. 이 단원에서는 갱신 단추를 조치 바인더에 바인드합니다.

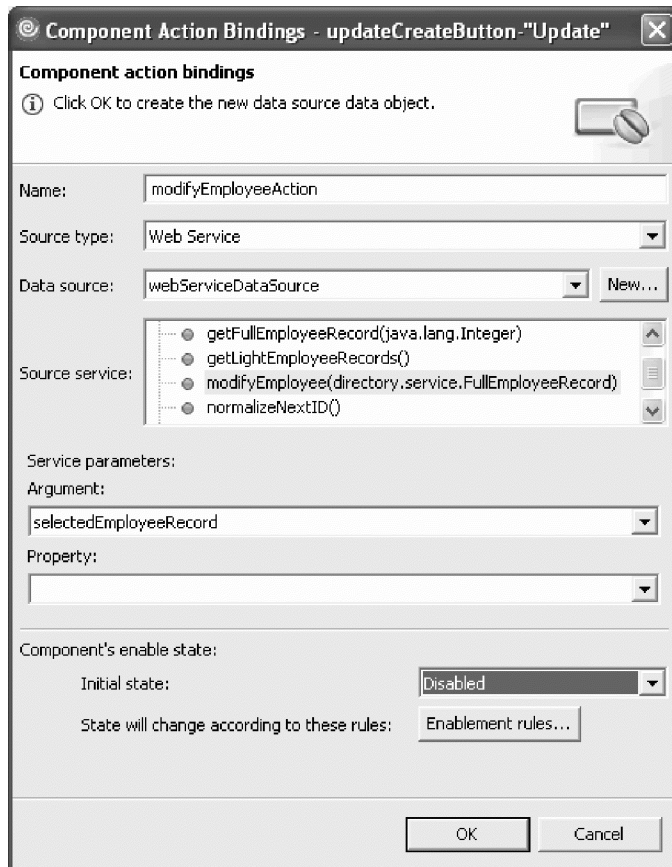
갱신 단추를 바인드하려면 다음을 수행하십시오.

1. 디자인 영역에서 갱신 단추를 선택하고 바인드 탭을 클릭하여 컴포넌트 조치 바인딩 대화 상자를 여십시오.

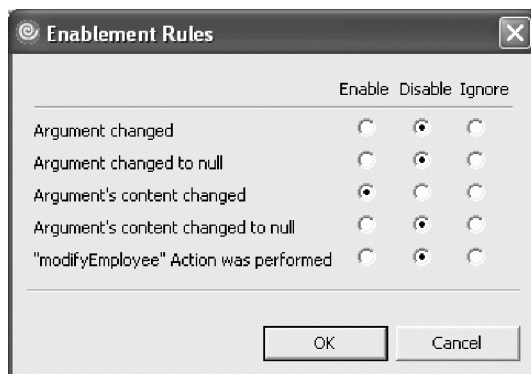


2. 소스 유형 필드에서 웹 서비스를 선택하십시오.
3. 데이터 소스 필드에서 **webServiceDataSource**를 선택하십시오.

4. 소스 서비스 목록에서 **modifyEmployee(directory.service.FullEmployeeRecord)**를 선택하십시오.
5. 이름 필드는 **modifyEmployeeAction**으로 자동 변경됩니다. 이 기본값을 사용하십시오.
6. 인수 필드에서 **selectedEmployeeRecord**를 선택하십시오.
7. modifyEmployee() 메소드는 전체 직원 레코드를 인수로 사용하므로 특성 필드를 공백으로 남겨두어야 합니다.
8. 단추의 초기 상태를 사용 불가능으로 설정하십시오.



9. 단추 상태를 어떻게 변경할지 정의하려면 **활성화 규칙**을 클릭하십시오. 인수 콘텐츠가 변경될 경우에만 단추를 사용하고 기타 모든 인스턴스에서는 사용할 수 없도록 지정하십시오. 확인을 클릭하십시오.





이는 `selectedEmployeeRecord` 콘텐츠를 변경해야 갱신 단추를 사용할 수 있음을 의미합니다. 즉, `selectedEmployeeRecord`에 바인드되는 세부사항 필드 중 하나에 새로운 값이 입력되면 바인더는 이 단추를 사용 가능하게 합니다. 새 레코드를 선택하거나 갱신을 클릭하면 단추는 다시 사용할 수 없게 됩니다.

#### 10. 확인을 클릭하십시오.

갱신 단추에 대해 새 `SwingDataServiceAction` 바인더가 추가됩니다. 디자인 영역에서 단추를 선택할 경우, Visual Editor는 단추가 웹 서비스 데이터 소스에 바인드됨을 표시하는 선을 그립니다. 분홍색 점선 화살표가 `selectedEmployeeRecord` 오브젝트에서 그 선 방향으로 표시됩니다. 이 화살표는 `selectedEmployeeRecord`가 서비스를 호출하기 위한 인수임을 나타내는 것입니다.

### 학습 체크포인트

이제 응용프로그램을 실행하면 직원 레코드를 갱신할 수 있습니다.

테이블에서 직원을 선택하고 성을 변경하십시오. 성을 변경하는 즉시 갱신 단추가 사용 가능하게 됩니다. 갱신을 클릭하면 `modifyEmployee` 서비스가 호출되고 직원이 갱신됩니다. 직원 테이블에 새로운 성이 반영됩니다.

## 단원 2.5: 삭제 단추 및 확인 대화 상자 사용

이 연습에서는 직원 레코드를 삭제할 수 있도록 회사 디렉토리 응용프로그램을 프로그램합니다.

이 응용프로그램을 사용하여 다음을 수행할 수 있습니다.

- 테이블에서 직원을 선택하면 삭제 단추가 사용 가능하게 됩니다.
- 삭제 단추를 클릭하면 삭제 확인 대화 상자가 열려 삭제를 확인하도록 요청합니다.
- 삭제 확인 대화 상자에서 예 단추를 클릭할 경우, 직원 레코드가 삭제되고 삭제 확인 대화 상자가 닫히며 직원 목록이 새로 고쳐집니다.
- 아니오를 클릭하면 삭제가 취소되고 삭제 확인 대화 상자가 닫힙니다.

### 테이블에서 행을 선택했는지 여부에 따라 삭제 단추가 사용 가능 또는 사용 불가능하게 되도록 프로그래밍

삭제 단추가 사용 가능 또는 사용 불가능하도록 프로그래밍하려면 행을 선택할 때 단추를 사용할 수 있게 하는 리스너를 테이블에 추가하십시오.

1. Java Bean 보기에서 `employeesTable`을 선택하십시오. 소스 보기에서 다음 행이 강조표시됩니다.

```
employeesTable = new JTable();
```

2. 이 행 바로 다음에 새로운 `ListSelectionListener` 및 `valueChanged` 이벤트를 `employeesTable`에 추가하십시오.

```
employeesTable.getSelectionModel().addListSelectionListener(new ListSelectionListener() {  
    public void valueChanged(ListSelectionEvent e) {  
        getDeleteButton().setEnabled(getEmployeesTable().getSelectedRowCount() != 0);  
    }  
});
```

- 이 코드 행을 추가하고 나면 소스 편집기는 사용자가 ListSelectListener 및 ListSelectionEvent를 가져올 때까지 이 행을 오류로 표시합니다. 필수 가져오기를 추가하려면 기본 메뉴에서 소스 → 가져오기 구성을 클릭하십시오. 다음 행이 클래스의 가져오기 섹션에 추가됩니다.

```
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
```

이제 테이블에서 행을 선택하면 삭제 단추가 사용 가능하게 됩니다.

## 삭제를 클릭하면 삭제 확인 대화 상자가 열리도록 프로그래밍

actionPerformed 이벤트를 삭제 단추에 추가한 다음 삭제 확인 대화 상자가 열리도록 이벤트를 프로그래밍하십시오.

- 삭제 단추를 마우스 오른쪽 단추로 클릭하고 이벤트 → actionPerformed를 선택하십시오. getDeleteButton() 메소드에 다음 이벤트 스텝이 추가됩니다.

```
deleteButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        System.out.println("actionPerformed()");
        // TODO Auto-generated Event stub actionPerformed()
    }
});
```

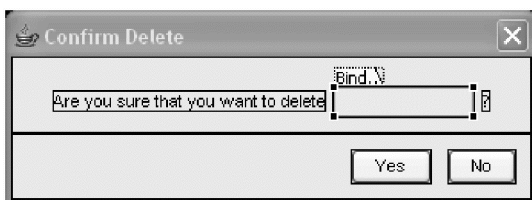
- 생성된 이 스텝을 다음 코드로 변경하십시오. 이는 단추를 클릭할 때 삭제 확인 대화 상자가 표시되도록 설정합니다.

```
deleteButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        getConfirmDialog().setVisible(true);
    }
});
```

## 삭제 확인 대화 상자에서 텍스트 필드 바인드

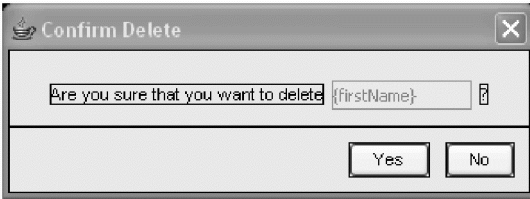
삭제할 직원의 이름을 표시하려면 삭제 확인 대화 상자의 텍스트 필드를 바인드하십시오.

- Java Bean 보기 또는 디자인 영역에서 employeeToDeleteField 텍스트 필드를 선택하고 바인드 탭을 클릭하십시오.



- 필드 데이터 바인딩 대화 상자에서 selectedEmployeeRecord 데이터 오브젝트와 firstName 필드를 선택하고 확인을 클릭하십시오.

이제 텍스트 필드가 employeesTable에서 선택된 행의 firstName 열에 바인드되었습니다.



3. 이 필드를 읽기 전용으로 하려면 필드 바인더의 **autoEditable** 특성을 **false**로 설정하십시오.

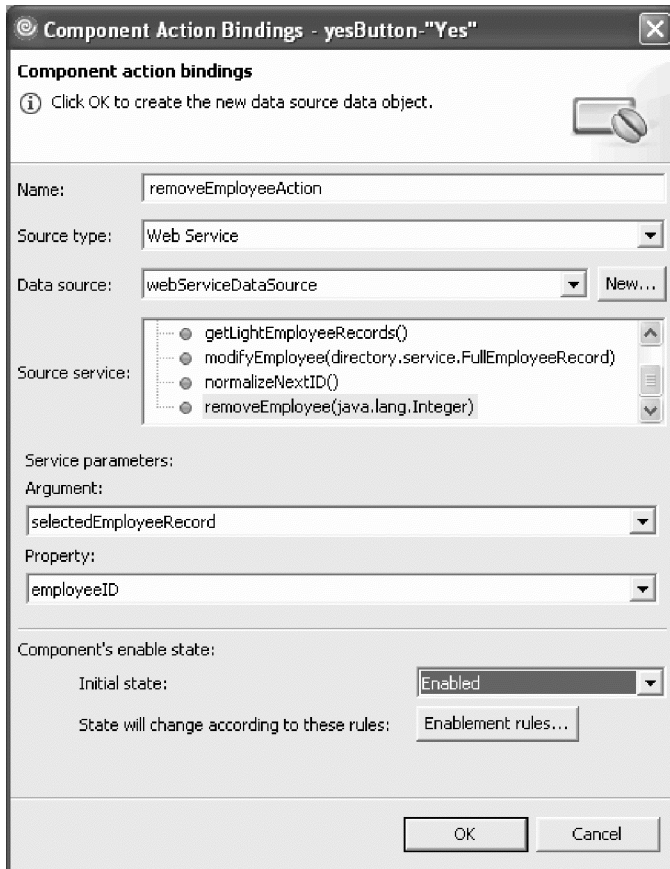
### 삭제를 수행하도록 예 단추 바인드

웹 서비스에서 `removeEmployee(java.lang.Integer)` 메소드를 호출하도록 예 단추를 바인드하십시오.

1. 예 단추를 선택하고 바인드 탭을 클릭하여 컴포넌트 조치 바인딩 대화 상자를 여십시오.
2. 소스 유형 필드에서 웹 서비스를 선택하십시오.
3. 데이터 소스 필드에서 **webServiceDataSource**를 선택하십시오.
4. 소스 서비스 목록에서 **removeEmployee(java.lang.Integer)**를 선택하십시오.
5. 이름 필드는 **removeEmployeeAction**으로 자동 변경됩니다. 이 기본값을 사용하십시오.
6. 인수 필드에서 **selectedEmployeeRecord**를 선택하십시오.
7. 특성 필드에서 **employeeID**를 선택하십시오. `removeEmployee()` 메소드는 정수를 인수로 취하므로 `selectedEmployeeRecord`의 직원 ID를 사용합니다.
8. 단추의 초기 상태를 사용 불가능으로 설정하십시오.
9. 활성화 규칙에 대해 조건에 따라 무시를 선택하십시오.

이 컴포넌트 상태는 예 단추가 항상 사용 가능함을 의미합니다. 그 상태를 변경할 필요가 없기 때문입니다.





10. 확인을 클릭하십시오.

## 직원이 삭제된 후 삭제 확인 대화 상자를 숨기도록 이벤트 추가

이 단계에서는 예 단추의 바인더(예 단추 자체가 아님)에 이벤트를 추가합니다. 직원이 제거된 후, 즉 바인더가 데이터 소스에 대해 서비스를 성공적으로 호출한 후 삭제 확인 대화 상자가 닫으려고 합니다.

getRemoveEmployeeAction() 메소드에 다음 코드를 추가하십시오.

```
removeEmployeeAction.addActionBinderListener(new jve.generated.IActionBinder.ActionBinderListener() {
    public void afterActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {
        getConfirmDialog().setVisible(false);
    }
    public void beforeActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {}
});
```

이 이벤트 코드는 바인더의 조치가 수행된 후 삭제 확인 대화 상자를 숨깁니다.

## 학습 체크포인트

회사 디렉토리 응용프로그램을 실행하여 테이블에서 직원을 선택하려면 삭제 단추를 클릭하고 예를 클릭하여 삭제를 확인하십시오. 그러면 직원 레코드가 디렉토리에서 제거되고 직원 목록에 제거가 반영됩니다.

## 단원 2.6: 새 직원을 추가하기 위한 조치 및 바인딩 설정

이 단원에서는 회사 디렉토리 응용프로그램을 사용하여 새 직원 레코드를 추가합니다.

새 직원을 추가하기 위한 응용프로그램의 작동이 상당히 복잡하고 동적이므로 이 연습은 더 복잡해질 수 밖에 없고 일부 소스 코드는 수동으로 변경해야 합니다. 또한, 이 연습에서는 데이터 오브젝트의 몇몇 고급 기능을 보여주고 사용자 필요에 맞도록 바인더 및 데이터 오브젝트를 사용할 수 있는 방법에 대한 창의적 예제를 제공합니다.

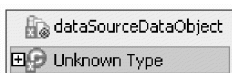
다음 목록은 응용프로그램의 필수 작동을 설명합니다.

- 새로 작성 단추를 클릭하면 다음과 같은 동작이 발생합니다.
  - 직원 테이블에서 선택사항이 지워지고 테이블은 사용 불가능하게 됩니다.
  - 테이블 선택사항을 지우면 삭제 단추를 사용할 수 없게 됩니다.
  - 필터 필드가 사용 불가능합니다.
  - 새 직원 ID를 제외하고 세부사항 필드에서 값이 모두 지워집니다.
  - 갱신 단추의 텍스트가 추가로 전환됩니다.
- 추가 단추를 클릭하면 다음과 같은 동작이 발생합니다.
  - 세부사항 필드에 입력한 값은 새 직원 레코드로 디렉토리에 추가됩니다.
  - 테이블은 사용 가능하게 되고 값은 새로 고쳐집니다.
  - 필터 필드가 사용 가능합니다.
  - 추가 단추의 텍스트가 갱신으로 다시 전환됩니다.

### createNewFullEmployeeRecord()를 호출하는 새 데이터 소스 데이터 오브젝트 추가

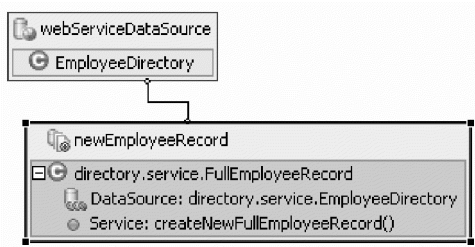
샘플 웹 서비스는 사용 가능한 다음 직원 ID 번호로 채워지도록 공백의 새 직원 레코드를 제공하는 createNewFullEmployeeRecord 서비스를 제공합니다. 이 공백 레코드는 새 직원 정보로 채우고 다시 웹 서비스에 제출할 수 있습니다.

1. Java Visual Editor의 팔레트에서 데이터 오브젝트 드로어를 펼치고 데이터 소스 데이터 오브젝트를 선택하십시오.
2. 마우스 포인터를 디자인 보기의 공백 영역 또는 자유 양식 영역으로 이동하고 왼쪽 마우스 단추를 클릭하여 데이터 소스 데이터 오브젝트를 놓으십시오. 새 데이터 소스 데이터 오브젝트가 자유 양식 영역에 추가되어 다음과 같이 표시됩니다.



3. 마우스 오른쪽 단추로 데이터 소스 데이터 오브젝트를 클릭하고 필드 이름 바꾸기를 선택하십시오. 데이터 오브젝트의 이름을 newEmployeeRecord로 바꾸십시오.
4. newEmployeeRecord 데이터 오브젝트를 마우스 오른쪽 단추로 클릭하고 바인딩 특성을 선택하십시오. 데이터 바인딩 대화 상자가 열립니다.
5. 데이터 소스 필드에서 webServiceDataSource를 선택하십시오.
6. 서비스 필드에서 createNewFullEmployeeRecord()를 선택하십시오.
7. 확인을 클릭하십시오.

자유 양식 영역에서 newEmployeeRecord 데이터 소스 데이터 오브젝트가 웹 서비스에 바인드되었는지 확인할 수 있습니다.

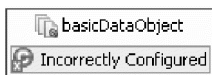


## 데이터 오브젝트를 쉽게 전환하기 위한 기본 데이터 오브젝트 추가

세부사항 필드와 갱신 단추는 모드를 전환해야 하므로(둘 다 갱신을 수행하고 새 직원을 작성하도록), 다른 시간에 다른 두 데이터 오브젝트에 바인드해야 합니다. 이 단계를 용이하게 하려면 기본 데이터 오브젝트 switchingDataObject를 추가하십시오. 이 기본 데이터 오브젝트를 사용하면 selectedEmployeeRecord 및 newEmployeeRecord 사이에 텍스트 필드에 대한 바인딩이 전환됩니다.

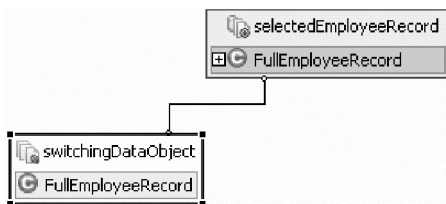
새 기본 데이터 오브젝트는 단지 이전 연습에서 정의한 다른 데이터 오브젝트(selectedEmployeeRecord)를 지시합니다. 또한 이 오브젝트는 이전에 작성한 newEmployeeRecord를 사용하도록 기본 데이터 오브젝트에 지시하는 메소드를 작성할 때 유용하게 사용됩니다. 즉, 이 기본 데이터 오브젝트는 selectedEmployeeRecord 데이터 오브젝트와 newEmployeeRecord 데이터 오브젝트 사이에 전환되는 중간 데이터로 작동하여, 응용프로그램의 비주얼 구성요소가 두 개의 다른 데이터 오브젝트에 대해 작동할 수 있습니다.

1. Visual Editor 팔레트에서 기본 데이터 오브젝트를 선택하고 자유 양식 영역에 놓으십시오. basicDataObject가 추가됩니다.



2. 데이터 오브젝트의 이름을 switchingDataObject로 바꾸십시오.
3. switchingDataObject에 대한 특성 보기에서 **sourceObject** 특성을 **selectedEmployeeRecord**로 설정하십시오. 특성에 대한 값 열의 드롭 다운 메뉴에서 selectedEmployeeRecord를 선택할 수 있습니다.

이제 switchingDataObject는 selectedEmployeeRecord를 참조하여 같은 값을 반영합니다.

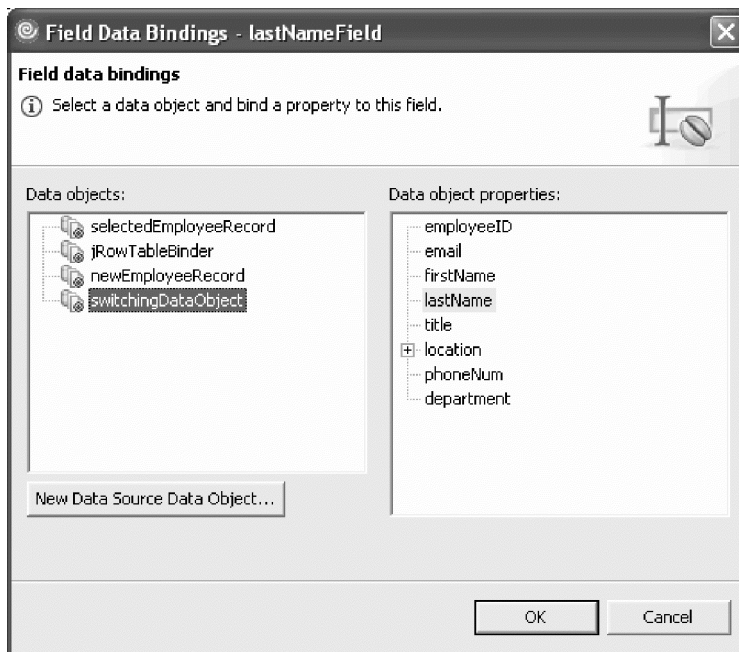


## 각 직원 필드를 switchingDataObject에 리바인드

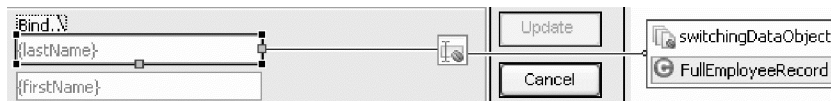
각 직원 세부사항 필드가 이미 selectedEmployeeRecord에 바인드되어 있어도 이들을 바로 switchingDataObject에 바인드할 수 있습니다. 필드를 바인드하고 나면 기존 직원 레코드를 수정하고 있는지 아니면 새로운 직원 레코드를 추가하고 있는지에 따라 필드의 데이터 오브젝트 사이를 동적으로 전환할 수 있습니다.

직원 세부사항 섹션의 필드마다 다음 단계를 완료하십시오.

1. 필드를 선택하고 바인드 탭을 클릭하십시오.
2. 필드 데이터 바인딩 대화 상자에서 switchingDataObject를 선택하십시오. 이 필드는 selectedEmployeeRecord에 이미 바인드되어 있습니다.



3. 필드가 정확한 데이터 오브젝트 특성에 바인드되어 있는지 확인하고 확인을 클릭하십시오. 디자인 보기에서 필드를 선택하면 지금은 바인더 선이 switchingDataObject를 향하고 있는 것을 볼 수 있습니다.



## 모드를 갱신하고 전환하기 위한 플래그 및 메소드 정의

다음의 updateMode() 메소드는 모드 플래그가 새로 작성으로 설정되어 있는지 확인하고 그에 따라 응용프로그램 작동을 변경합니다. 기본적으로 isNewMode 부울 플래그는 거짓으로 설정되므로 updateMode() 메소드는 직원 테이블 및 필터 필드를 사용 가능하게 하고, 갱신 단추의 텍스트를 "갱신"으로 설정합니다. isNewMode가 참으로 설정되면 직원 테이블은 사용할 수 없게 되고 선택한 내용이 모두 지워집니다. 필터 필드를 사용할 수 없으며 갱신 단추의 텍스트는 "추가"로 설정됩니다.

DirectoryApp.java 클래스에서 마지막 닫는 중괄호 바로 앞에 다음 코드를 추가하십시오.

```

private boolean isNewMode = false;
private void updateMode() {
    if (isNewMode) {
        getEmployeesTable().clearSelection();
        getEmployeesTable().setEnabled(false);
        getFilterField().setEditable(false);
        getUpdateCreateButton().setText("Add");
    } else {
        getEmployeesTable().setEnabled(true);
        getFilterField().setEditable(true);
        getUpdateCreateButton().setText("Update");
    }
}
}

```

## actionPerformed 이벤트를 새로 작성 단추에 추가

이 단계에서는 새로 작성 단추를 클릭할 경우의 이벤트 코드를 추가합니다. 이벤트는 `switchingDataObject`가 `newEmployeeRecord` 데이터 오브젝트를 사용하도록 통지하고 모드 플래그를 "새로 작성"으로 설정한 후 앞 단계에서 추가한 `updateMode()` 메소드를 실행합니다.

1. 디자인 보기에서 새로 작성 단추를 마우스 오른쪽으로 클릭하고, 이벤트 → **actionPerformed**를 선택하십시오. `getNewButton()` 메소드에 다음 코드가 생성됩니다.

```

newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        System.out.println("actionPerformed()"); // TODO Auto-generated Event stub actionPerformed()
    }
});

```

2. 생성된 이 스텝을 다음 코드로 변경하십시오.

```

newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        getSwitchingDataObject().setSourceObject(getNewEmployeeRecord());
        getNewEmployeeRecord().refresh();
        isNewMode = true; //sets application to new mode
        updateMode(); //changes UI according to new mode
        getLastNameField().grabFocus();
    }
});

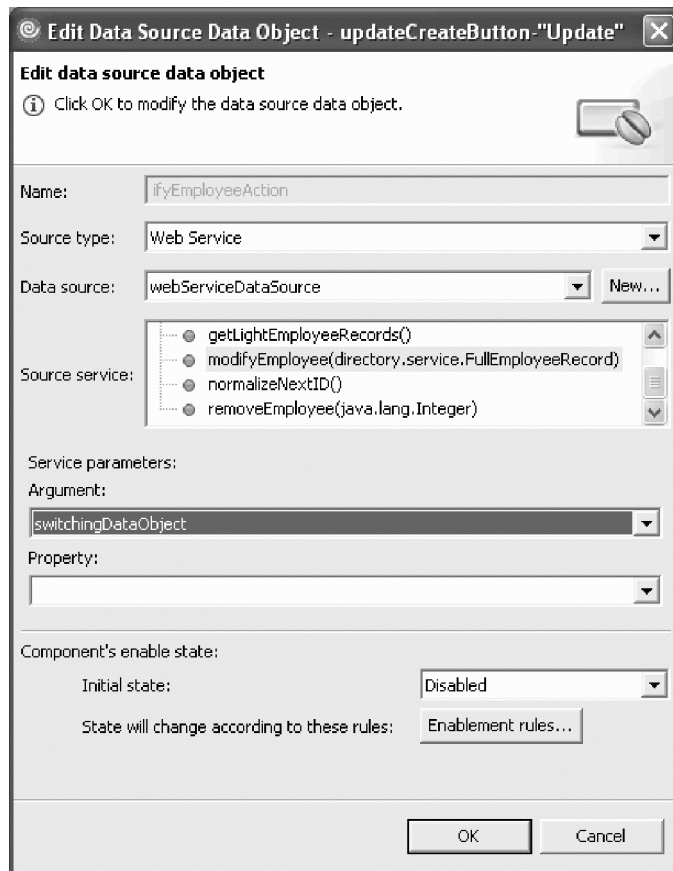
```

## 갱신 단추 리바인드

이전 단원에서는 웹 서비스에서 `modifyEmployee` 메소드를 사용하도록 갱신 단추를 프로그램했습니다. 그 조치는 `SwingDataServiceAction`으로 구현됩니다. `SwingDataServiceAction` 특성 중 하나가 서비스에 대한 인수로 작동하는 소스 오브젝트입니다. 수정 조치를 위한 소스 오브젝트는 현재 `selectedEmployeeRecord`로 설정되어 있습니다. 갱신 및 추가를 모두 제어하도록 단추를 프로그램하려면 단추 조치를 재구성하여 `switchingDataObject`를 `modifyEmployee` 서비스에 대한 인수로 사용해야 합니다.

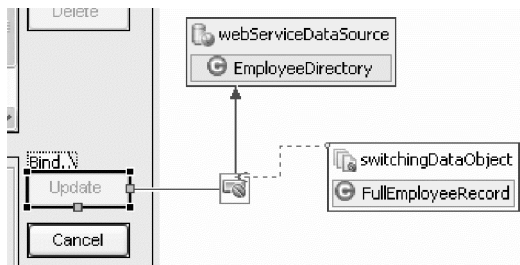
1. 디자인 보기에서 갱신 단추를 선택하십시오. `selectedEmployeeRecord`가 서비스 호출의 인수임을 표시하는 분홍색 점선 회살표에 주목하십시오.
2. 갱신 단추의 바인드 탭을 클릭하십시오.
3. 인수 필드에서 `switchingDataObject`를 선택하십시오.





#### 4. 확인을 클릭하십시오.

이제는 modifyEmployee 메소드의 인수로 switchingDataObject를 사용하도록 단추 조치가 구성된 점에 주목하십시오.



#### 모드를 재설정하도록 갱신 단추의 바인더에 이벤트 추가

갱신 단추를 클릭하고 웹 서비스에서 조치를 완료한 후, 기본 모드 및 작동으로 다시 이동할 수 있습니다. 이를 수행하려면 갱신 또는 추가를 수행한 후 모드를 갱신하고 테이블을 새로 고칠 단추 조치 바인더에 대한 이벤트 리스너를 추가하십시오.

갱신 단추에 대한 getModifyEmployeeAction() 메소드에 다음 코드를 추가하십시오.

```
modifyEmployeeAction.addActionBinderListener(new jve.generated.IActionBinder.ActionBinderListener() {
    public void afterActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {
        if (isNewMode) {
            //Go back to using the selectedEmployeeRecord
```

```

        getSwitchingDataObject().setSourceObject(getSelectedEmployeeRecord());
        //Revert out of new mode
        isNewMode = false;
        updateMode();
    }
    // Refresh the table's data object
    getLightEmployeeRecordRows().refresh();    }
    public void beforeActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {}
});

```

## 학습 체크포인트

이제 회사 디렉토리 응용프로그램을 실행할 때 새로 작성 단추를 클릭하여 새 직원 레코드를 추가할 수 있습니다.

## 단원 2.7: 취소 단추 작동 프로그래밍

응용프로그램을 사용할 때 변경사항을 제출하지 않도록 하면 직원 레코드에 대해 변경 전의 상태로 돌아갈 수 있습니다. 즉, 필드를 처음 상태로 돌아가려면 취소 및 지우기가 가능해야 합니다. 이 기능을 추가하려면 취소 단추에서 일부 actionPerformed 이벤트를 설정하면 됩니다.

다음 목록은 취소 단추의 필수 작동에 대해 설명합니다.

- 새 모드에 있는 동안 취소 단추를 클릭하면 응용프로그램은 새 모드에서 이전으로 복귀합니다.
- 직원 레코드를 수정하는 동안 취소 단추를 클릭하면 변경한 값이 원래 값으로 돌아갑니다.

actionPerformed 이벤트를 취소 단추에 추가하여 필수 작동을 수행하려면 다음을 수행하십시오.

1. 디자인 보기에서 취소 단추를 마우스 오른쪽으로 클릭하고, 이벤트 → actionPerformed를 선택하십시오. getCancelButton() 메소드에 다음 코드가 생성됩니다.

```

cancelButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        System.out.println("actionPerformed()"); // TODO Auto-generated Event stub actionPerformed()
    }
});

```

2. 생성된 이벤트 스텝을 다음 코드로 변경하십시오.

```

cancelButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        if (isNewMode) {
            getSwitchingDataObject().setSourceObject(getSelectedEmployeeRecord());
            isNewMode = false;
            updateMode();
        } else {
            getSelectedEmployeeRecord().refresh();
        }
    }
});

```

## 학습 체크포인트

actionPerformed 이벤트를 통해 취소 단추 프로그래밍 방법을 학습했습니다.

## 단원 2.8: 직원 테이블에서 필터 설정

텍스트 필터 바인더를 사용하여 직원 테이블 콘텐츠를 필터할 수 있습니다. 필터는 텍스트 필드에서 입력을 취하여 특정의 특성이나 테이블의 열을 기초로 테이블을 필터합니다.

응용프로그램에서는 필터 필드에 입력한 문자를 사용하여 직원 성별로 필터합니다. 필터 필드에 입력한 정확한 값이 직원 레코드의 성으로 제시되면 직원 레코드는 다음과 같이 테이블에 표시됩니다.

Filter:  (Last name)

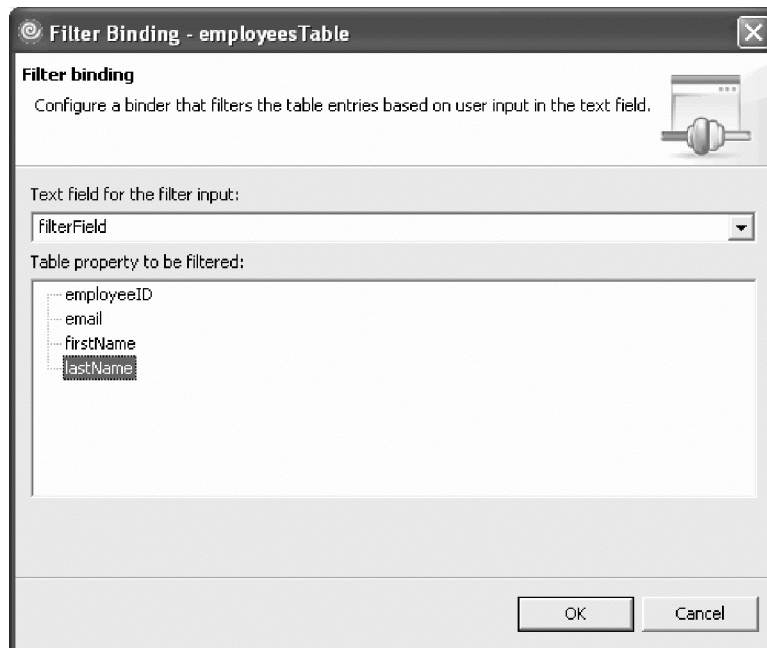
Last name	First name	Email	Employee ID
Maxwell	Seth	seth.maxwell@my...	24561
Maxwell	Aubrey	aubrey.maxwell@...	30089
Martinez	James	james.martinez@m...	31780

New

Delete

테이블에 대한 필터를 작성하려면 다음을 수행하십시오.

1. employeesTable에 대한 바인더 아이콘을 선택한 후 바인딩 필터 특성을 선택하십시오. 바인딩 필터 대화 상자가 열립니다.
2. 필터 입력에 대한 텍스트 필드 목록에서 **filterField**를 선택하십시오.
3. 필터할 테이블 특성 목록에서 **lastName**을 선택하십시오.



4. 확인을 클릭하십시오.

새 `SwingPropertyFilter`가 생성됩니다. 테이블 바인더의 필터 특성은 새 필터를 사용하도록 설정됩니다. 새 필터는 입력으로 필터 필드를 사용하고 테이블의 `lastName` 특성에 따라 필터하도록 구성됩니다.

## 학습 체크포인트

테이블에 대한 필터 설정 방법을 학습했습니다.



이제 회사 디렉토리 응용프로그램을 실행하면 필터 필드에 문자를 입력할 수 있으며 테이블은 성에 입력한 문자열 행을 표시하도록 필터됩니다.

축하합니다! 회사 디렉토리 응용프로그램을 완료했습니다.

## 요약: 웹 서비스를 사용하는 리치 Java 클라이언트 빌드

축하합니다! Java Visual Editor를 사용하여 회사 디렉토리 응용프로그램 즉, 샘플 웹 서비스에 연결하여 직원 디렉토리를 유지보수하는 리치 Java 클라이언트를 빌드하는 방법에 대해 학습했습니다.

Last name	First name	Email	Employee ID
Miller	Jeff	jeff.miller@mycodi...	24559
Garcia	Linda	linda.garcia@myc...	24560
Maxwell	Seth	seth.maxwell@m...	24561
Dehlavi	Madhubala	madhubala.dehlav...	27021
Huang	David	david.huang@my...	28225
Jensen	Kristin	kristin.jensen@my...	28290
Maxwell	Aubrey	aubrey.maxwell@...	30089

Employee details:

Last name:

First name:

Employee ID:

Contact information:

Email:

Phone:

Work location:

Office:

Building:

Site:

완료된 제품 그림 참조:

## 학습한 내용

Visual Editor를 사용하여 그래픽 사용자 인터페이스를 성공적으로 작성했습니다. 이때 GridBagLayout을 사용하여 직원 테이블을 배열했습니다. 그런 다음 테이블, 필드 및 단추를 적절한 데이터 오브젝트와 데이터 소스를 바인드하여 응용프로그램이 사용자가 생성한 웹 서비스 Java 프록시에 대해 작동하도록 했습니다. 또한 응용프로그램을 쉽게 사용하고 직관할 수 있도록 하기 위해 응용프로그램이 적절하게 작동하도록 일부 복잡한 코딩을 수행했습니다. 그리고 WebSphere Application Server v6.0에 엔터프라이즈 응용프로그램을 설치하고 웹 서비스를 배치하는 방법에 대해 학습했습니다.

가장 두드러지게 주목할 것은 데이터에 대해 작업할 수 있도록 Java Visual Editor가 제공하는 강력한 바인더 클래스에 대한 모든 것을 학습했다는 것입니다. 이로써 사용자는 사용자 고유의 바인더를 실제로 사용할 수 있습니다.

이제 다음 작업을 수행할 수 있습니다.

- Java Visual Editor를 사용하여 GridBagLayout에서 구성요소 레이아웃
- 비주얼 클래스를 Java Bean으로 실행
- Java 응용프로그램의 비주얼 구성요소를 웹 서비스에서 리턴된 데이터 오브젝트 및 메소드에 바인드
- 이벤트를 비주얼 구성요소에 추가

## 추가 자원

완료된 My Directory 응용프로그램 버전 가져오기

이 프로젝트에는 완료된 응용프로그램, 바인더 클래스를 갖춘 jve.generated 패키지 및 WebSphere Application Server v6.1에 대해 구성된 웹 서비스 Java 클라이언트가 포함됩니다. 학습서를 통해 작업하지 않고 완료된 프로젝트를 가져올 경우, Java 빌드 경로 변수를 구성해야 할 수도 있습니다. 경로 변수는 WebSphere v6.1 웹 서비스 thin 클라이언트 JAR 파일을 가리켜야 합니다.

팁: 가져올 때 프로젝트 이름을 다르게 지정하지 않으면 MyDirectory 프로젝트 콘텐츠를 겹쳐씹습니다.