



iSeries 소매 상점 시나리오 어플리케이션

Windows용 버전 5.1



iSeries 소매 상점 시나리오 어플리케이션

Windows용 버전 5.1

목차

제 1 장 iSeries 소매 상점 시나리오 어플리케이션 소개	1
프로젝트 기반 시나리오	2
제 2 장 기술 개념	5
제 3 장 시나리오 실행	9
시작하기 전에	9
샘플 파일 설치	10
.savf 파일 복원	10
라이브러리 목록 수정	12
Development Studio Client에서 어플리케이션 가져오기 및 실행	13
로컬 시스템에 .zip 파일 추출	13
웹 프로젝트 작성	14
웹 프로젝트 파일 가져오기	15
WebFacing 프로젝트 파일 가져오기 및 WebFacing 프로젝트 작성	15
서버 구성	16
WebFacing 프로젝트 구성	18
서버 정보 정의	18
Workbench에서 어플리케이션 실행	19
Workbench에서 고객으로 어플리케이션 실행	19
Workbench에서 관리자로 어플리케이션 실행	20
WebSphere Application Server에 시나리오 어플리케이션 전개	21
WebSphere Application Server 구성	21
관리자 페이지 보호	22
iSeries WebSphere Application Server 전개용 EAR 파일 작성	24
iSeries WebSphere Application Server에 EAR 파일 전개	24
WebSphere Application Server에서 어플리케이션 실행	25
제 4 장 단계별 모듈 1: 제품 가격을 리턴하는 웹 서비스 작성(SV000514)	27
소개	27
시작하기 전에	27
새 웹 프로젝트 작성	28
iSeries 서버 정보 정의	28
RPG 서비스 프로그램 작성	29

매개변수 작성 및 Java bean 생성	29
Java bean에서 웹 서비스 작성	31
샘플 테스트	32
제 5 장 단계별 모듈 2: 재고를 열람하고 항목을 주문하기 위한 인터페이스 작성(SV000501)	35
소개	35
시작하기 전에	36
WebFacing 프로젝트 작성	36
DDS 소스 변환	37
Workbench에서 UTF-8 지원 구성 — WAS 버전 4.0 사용자 전용	38
WebSphere Application Server용 UTF-8 지원 구성	39
스타일 시트 작성	40
수동으로 캐스캐이딩 스타일 시트의 사용자 정의 (선택적)	40
웹 상호작용을 사용한 WebFacing 프로젝트 확장 및 향상	41
웹 상호작용에 대한 프로젝트 링크	44
파일 공개 및 서버 다시 시작	46
인터페이스 테스트	46
제 6 장 고급 모듈 1: iSeries 서버에 고객 주문을 배치하는 HTML, servlet 및 JSP 파일 작성 (SV001585)	49
소개	49
상위 레벨 단계 요약	50
시작하기 전에	50
웹 페이지, servlet 및 JSP 파일 작성	51
제 7 장 고급 모듈 2: SV000514 및 SV001586 웹 서비스를 사용하는 웹 프로젝트 작성(SV000618)	59
소개	59
시작하기 전에	61
웹 페이지, servlet, JSP 및 RPG 코드 작성	61
WebSphere Application Server에 전개하기 전에	63
제 8 장 주의사항	65
저작권	67
프로그래밍 인터페이스 정보	67
상표 및 서비스표	67

제 1 장 iSeries 소매 상점 시나리오 어플리케이션 소개

시나리오 패키지는 iSeries용 IBM WebSphere Development Studio Client, iSeries 서버용 Eclipse 기반 기술을 사용하여 개발된 샘플 어플리케이션입니다. 시나리오는 다음과 같은 목적으로 iSeries 서버를 사용하고자 하는 개발자를 위해 설계되었습니다.

- Java 개발
- 웹 개발
- RPG 코드 관리 및 웹 어플리케이션으로 변형

어플리케이션은 iSeries 서버로 복원할 두 개의 .savf 파일과 클라이언트 제품으로 가져올 다섯 개의 .zip 파일로 이루어져 있습니다. 이들은 진행 중인 여러 개의 지점에 대한 URL과 함께 일련의 웹 페이지를 형성합니다.

중요한 참고: 모든 서비스 팩, PTF 및 기타 정보를 다음 지원 페이지를 확인하십시오.
ibm.com/software/awdtools/wdt400/support/

이 시나리오는 다음과 같은 iSeries 특정 구성요소를 강조하며 사용자에게 제품의 다양한 부분을 경험할 수 있게 합니다.

- IBM WebFacing 도구
- 웹 서비스
- iSeries용 웹 개발 도구(웹 상호작용 마법사 및 프로그램 호출 마법사 포함)
- iSeries용 Java 개발 도구
- IBM Toolbox for Java

시나리오는 함께 비즈니스를 수행하며 비즈니스 논리 및 데이터에 대해 iSeries 서버를 사용하는 두 회사(도매 공급업체와 의류 소매 상점)의 상황을 설명합니다. 과거에는 전자 우편, 전화 및 팩스를 통해 재고를 점검하고, 주문을 제출하며, 주문 처리 과정을 추적하는 등의 통신을 수행했습니다. 이제는 웹을 사용하여 정기적인 비즈니스 트랜잭션을 수행하고자 합니다.

소매 상점은 다음과 같은 웹 사이트를 원합니다.

- 고객이 제품 구입에 사용할 수 있음
- 직원이 도매 공급업체로부터 재고를 주문하는 데 사용할 수 있음

도매 공급업체는 다음을 원합니다.

- 추적을 위한 소매업자 주문 수신
- 다수의 잠재적 고객에 대한 서비스

이 시나리오에서 사용자는 이들 두 회사에 대한 프로그래밍 컨설턴트의 역할을 수행하며, 비즈니스를 웹으로 이동하는 데 도움을 줍니다.

어플리케이션에는 사용자의 유형에 따라 두 가지 서로 다른 시작점이 있습니다. 고객으로서, 사용자는 상점에서 제공하는 제품(이 경우, 캐주얼 의류)을 보는데서 시작합니다. 제품을 구입하고자 할 경우, 링크를 클릭하여 주문 화면을 액세스합니다. 주문 후, 요약 페이지가 생성되며 쇼핑을 계속하거나, 주문을 취소하거나 또는 주문을 제출할 수 있습니다.

관리자로서, 사용자는 어플리케이션에 대한 보안 사용자 ID를 가지며 전개하는 동안 정의될 어플리케이션에 대한 보안 정책이 요구됩니다. 시작점은 주문을 열람하고 재고를 파악하며 도매 공급업체로부터 제품을 구입할 수 있는 로그인 화면입니다. 항목을 선택하고 최근 도매 가격을 점검한 후 원하는 크기와 수량을 주문할 수 있습니다. 어플리케이션은 도매업자가 사용자가 원하는 크기와 수량을 가지고 있는지 확인하며, 주문을 승인하거나 현재로서는 주문을 이행할 수 없음을 사용자에게 통지합니다.

어플리케이션 내에서는, 제품의 다양한 부분에서 여러 가지 조치가 발생합니다. 다음 테이블은 어플리케이션의 각 부분을 담당하는 제품의 프로세스 및 구성요소를 보여줍니다. 계속해서 각 작업을 수행하는 방법에 관한 자세한 정보는 다음과 같습니다.

표 1.

고객 어플리케이션 태스크	관리자 어플리케이션 태스크	기본 프로세스
제품 가격 표시		iSeries RPG 프로그램을 사용하여 웹 서비스를 작성하고 웹 개발 도구를 사용하여 가격을 보고 표시합니다.
상점에서 주문하기		Servlet 및 JSP 파일을 iSeries Java 개발 도구, iSeries 웹 개발 도구 및 IBM Toolbox for Java와 함께 사용하여 iSeries 서버의 재고를 액세스하고 파악하여 주문을 수행하고 주문 요약을 표시합니다.
	재고 보기	IBM WebFacing 도구를 사용하여 기존 RPG 프로그램을 웹 어플리케이션으로 변환하고 웹 개발 도구를 사용하여 웹 페이지를 사용자 정의합니다.
	상품 ID와 수량을 기입하여 도매 공급업체로부터 주문	구매 단추를 클릭할 때 호출되는 웹 서비스를 작성합니다.
상점에 대한 초기 웹 페이지 보기	재고를 주문하기 위한 초기 웹 페이지 보기	iSeries용 웹 개발 도구를 사용하여 두 개의 홈 페이지를 작성합니다.

프로젝트 기반 시나리오

iSeries 시나리오 어플리케이션은 SV000501, SV000514, SV000618, SV001585 및 SV001586이라는 다섯 개의 프로젝트로 구성됩니다. 이 안내서에는 사전 패키지 어플리케이션을 실행하는 방법에 관한 명령어가 들어 있습니다. 또한 안내서에는 단계별 모듈과 고급 모듈이 들어 있으며, 다양한 프로젝트를 사용자 스스로 빌드하는 방법을 보

여줍니다. 단계별 모듈은 어플리케이션 개발과 Development Studio Client에 상대적으로 익숙하지 않은 개발자용입니다. 고급 모듈은 어플리케이션 개발에 익숙하며 Development Studio Client에 보다 친숙한 개발자를 위해 설계되었습니다.

주: 다섯 개의 프로젝트가 있지만, SV001586 및 SV000514 프로젝트는 결합된 웹 서비스 모듈이므로 네 개의 모듈만이 존재합니다.

이 시나리오는 다음과 같은 다섯 가지 프로젝트 형태로 제공됩니다.

SV000501 프로젝트: 미해결 주문, 재고 및 제품 세부사항을 보기 위한 웹 프로젝트 작성 - 이 프로젝트는 iSeries 웹 개발 도구 및 IBM WebFacing 도구를 사용하여 작성되며, IBM WebFacing 도구를 사용하여 웹에 RPG 어플리케이션을 저장하기 위한 제한된 웹 어플리케이션 개발 지식을 가진 RPG 프로그래머를 위해 설계되었습니다.

SV001585 프로젝트: iSeries 서버에 클라이언트 주문을 배치하는 JSP 파일, HTML 코드 및 Servlet 작성 - 이 프로젝트는 IBM Toolbox for Java의 SQL 및 JDBC 클래스, iSeries Java 개발 도구의 RecordIOManager bean 및 iSeries 프로그램 호출 마법사를 사용합니다. 이들 요소는 iSeries 서버에 존재하는 데이터와 프로그램을 액세스하고 처리하는 다양한 방법을 보여줍니다. 이 프로젝트는 iSeries 데이터와 코드를 액세스하는 웹 페이지를 개발하려는 Java 프로그래머와 웹 어플리케이션 개발자를 위해 설계되었습니다. 추가로, iSeries 서버 관리 및 RPG 프로그래밍에 대한 작업 지식이 있어야 합니다.

SV000514 프로젝트: 제품 가격을 제공하는 iSeries 웹 서비스 작성 - 웹 서비스 마법사는 iSeries 프로그램 호출 마법사가 생성한 Java bean을 사용하여 iSeries 서버에 있는 하나 이상의 프로그램 프로시저를 호출하며, 정보를 다시 브라우저에 전달합니다. 이 프로젝트는 WWW(World Wide Web)를 통해 설명, 공개, 호출 그리고 찾을 수 있는 독립 언어 모듈 방식 어플리케이션을 작성하기 위해 웹 서비스를 사용하려는 RPG 프로그래머를 위해 설계되었습니다.

SV001586 프로젝트: iSeries 서버를 통해 도매 공급업체에 주문을 수행하기 위한 웹 서비스 작성 - 웹 서비스는 상품 ID와 필요한 수량을 승인한 다음, 도매 공급업체에 주문을 수행합니다. 이 프로젝트는 SV000514의 구성요소이며, 웹 서비스를 작성하려는 RPG 프로그래머를 위해 설계되었습니다.

SV000618 프로젝트: IBM WebFacing 도구에서 생성한 주문 양식, 재고 양식 및 구매 주문을 인터페이스하는 웹 프로젝트 작성 - 이 프로젝트에는 iSeries 웹 개발 도구가 필요하며, SV000514 및 SV001586에서 개발된 웹 서비스에 연결하며 사용하기 위한 HTML 및 JSP 파일 작성이 포함됩니다. 프로젝트는 웹 서비스에 대해 작업하려고 하고 RPG 및 Java 프로그래밍에 대한 지식이 있는 개발자를 위해 설계되었습니다.

제 2 장 기술 개념

시나리오 어플리케이션을 통해 작업하려면 여러 가지 기술 개념에 친숙해야 합니다(웹 어플리케이션 개발이 처음인 경우 특히). 다음은 어플리케이션을 통해 작업할 때 접하는 몇 가지 항목에 대한 간단한 목록입니다.

EAR(엔터프라이즈 아카이브 파일)

EAR 파일은 .ear 확장자를 갖는 표준 JAR(Java Archive) 파일입니다. 복수의 웹 프로젝트를 포함할 수 있으며, 이를 사용하여 웹 어플리케이션을 패키징하고 WAS(WebSphere Administrative Server)에 전개합니다. 참고: J2EE SDK 어플리케이션 전개 도구의 GUI 버전에서는 먼저 EAR 파일을 작성하고 JAR 및 WAR(웹 아카이브) 파일을 EAR 파일에 추가합니다. 그러나 명령행 패키지 도구를 사용할 경우, JAR 및 WAR 파일을 먼저 작성한 다음 EAR 파일을 작성하십시오.

IBM WebFacing 도구

IBM WebFacing 도구는 기존 5250 인터페이스를 브라우저 기반 그래픽 사용자 인터페이스로 변환합니다. 원본 iSeries 어플리케이션을 거의 또는 전혀 수정하지 않고, 프로그램의 사용을 인터넷 또는 인트라넷으로 확장할 수 있습니다.

JAR(Java 아카이브 파일)

JAR 파일은 .zip 파일과 유사한, 압축된 Java 파일 패키지입니다. 단일 파일에 수집되며 브라우저에 보다 빠르게 다운로드될 수 있도록 압축된 Java 애플릿용 클래스, 이미지 및 사운드 파일을 포함합니다.

JSP(Java Server Page)

JSP는 정적 HTML 페이지에 동적 내용을 표시하는 기능을 제공합니다. Java로 작성되므로, JSP는 서버 및 플랫폼에 독립적입니다. 웹 프리젠테이션을 웹 내용과 효과적으로 분리함으로써, JSP는 웹 페이지의 설계와 화면을 빠르게 변경해야 하는 개발자에게 도움이 될 수 있습니다.

프로그램 호출 bean

이들은 프로그램 호출 마법사에 의해 생성되는 Java bean입니다. 한 가지 유형은 Java 어플리케이션에서 사용하는 일반 Java bean이며, 다른 한 가지 유형은 웹 서비스 마법사가 웹 서비스를 작성할 때 사용할 수 있습니다.

프로그램 호출 마법사

프로그램 호출 마법사는 사용자가 iSeries 프로그램 또는 프로시저를 호출하는 데 필요한 Java bean과 연관된 PCML 파일을 작성하는 데 도움이 됩니다. 마법사는 사용자에게 프로그램 또는 서비스 프로그램 오브젝트와 오브젝트 매개변수에 관한 정보를 프롬프트하여, 원하는 Java bean(및 PCML 파일)을 작성합니다.

RPG(Report Program Generator)

iSeries 프로그래머가 사용하는 절차 프로그래밍 언어. RPG를 사용하여 청구 프로그램 및 주문 입력 프로그램과 같은 비즈니스 어플리케이션을 작성할 수 있습니다. 최신 버전인 ILE RPG IV는 이전 버전에 대한 프로그래머의 경험을 바탕으로, RPG 언어의 기능을 확장합니다.

Servlet

Java 사용 가능 서버 또는 IBM WebSphere Application Server와 같은 어플리케이션 서버에서 실행되는 Java로 작성된 서버측 프로그램. Servlet은 HTML 응답을 생성하여 요청에 대한 응답과 같은, 서버에서 지정한 타스크를 수행합니다. 예를 들어, 온라인 banking 어플리케이션에서는 servlet을 사용하여 서버에 데이터를 전송하면서 사용자에게 응답할 수 있습니다.

웹 구성요소

웹 구성요소를 사용하여 iSeries 서버 프로그램과 웹 페이지 간에 정보를 교환할 수 있는 누름 단추 및 데이터 입력 필드와 같은 iSeries 오브젝트를 정의할 수 있습니다. 개발자는 웹 구성요소를 사용하여 입력 필드 구문 점검 및 단추 클릭과 같은 사용자 이벤트를 캡처할 수 있습니다.

웹 상호작용 마법사

이 마법사는 iSeries 웹 개발 도구의 일부입니다. ILE 프로그램 및 웹 페이지 간의 상호작용을 작성하고 관리합니다. 마법사는 입력, 출력 및 오류 메시지가 표시되는 위치를 제어하며, 입력 및 출력 필드의 데이터를 ILE 프로그램으로 방향 지정합니다. 또한 웹 상호작용 마법사를 사용하여 사용자가 오류의 원인을 쉽게 식별할 수 있도록 오류가 발생한 영역에 오류 메시지를 맵핑할 수 있습니다.

웹 서비스

웹 서비스는 인터넷을 통해 사용하도록 설계 및 구현된 자체 포함 어플리케이션입니다. SOAP, WSDL 및 XML과 같은 개방 표준을 사용하여 작성됩니다. 클라이언트가 인터넷을 통해 재고 레벨을 점검할 수 있는 재고 관리 시스템 또는 공급업체로부터 제품 주문을 직접 추적하려는 경우를 포함하여, 웹 서비스를 사용할 수 있는 다양한 비즈니스 상황이 있을 수 있습니다.

WSDL(Web Services Definition Language)

WSDL은 웹 서비스 인터페이스를 정의하는 XML 기반 언어입니다. WSDL은 웹 서비스를 이해하며, 웹 서비스 및 서버 프로그램 간의 정보 플로우를 관리합니다. 예를 들어, 개발자는 WSDL을 사용하여 갱신된 주식 시세를 보여주는 웹 사이트용 인터페이스를 작성합니다.

WebSphere Studio Workbench

iSeries용 IBM WebSphere Development Studio Client는 IBM의 Eclipse 플랫폼의 구현, WebSphere Studio Workbench에 빌드됩니다. 확장 가능한, 유니버설 Workbench는 어플리케이션을 빌드하고 유지보수하는 데 필요한 모든 도구를 통합합니다. 개발자는 Development Studio Client를 사용하여 플러그인을 통해 새로운 오브젝트를 개발 환경에 통합하며 Java 파일, 그래픽, 비디오 등을 매끄럽게 추가할 수 있습니다.

제 3 장 시나리오 실행

Wholesale 및 *Retail store* 어플리케이션은 Development Studio Client Workbench 내부에서 또는 iSeries 플랫폼을 포함한 모든 플랫폼의 경우 WebSphere Application Server에서 실행할 수 있습니다. 어플리케이션 개요에 대해서는 1 페이지의 제 1 장 『iSeries 소매 상점 시나리오 어플리케이션 소개』를 참조하십시오.

이 장을 마칠 때쯤, 사용자는 다음을 수행할 수 있어야 합니다.

- iSeries 서버로 샘플 파일 및 Development Studio Client 오브젝트 복원
- 파일을 보유할 웹 프로젝트 및 WebFacing 프로젝트 작성
- iSeries 서버에서 복원된 파일을 Development Studio Client로 가져오기
- WAS(WebSphere Application Server) 구성
- WebFacing 프로젝트 구성
- WebSphere 테스트 환경에서 고객으로 어플리케이션 실행
- WebSphere 테스트 환경에서 관리자로 어플리케이션 실행
- WebSphere Application Server 구성
- 관리자 페이지 보호
- 외부 WAS 전개를 위한 EAR 파일 작성
- WAS에 어플리케이션 전개
- WAS에서 어플리케이션 실행

시작하기 전에

Workbench에서 어플리케이션을 테스트하려면 다음을 확인해야 합니다.

- V5R1 이상 iSeries 서버가 있음
- 최신 WebFacing PTF를 적용함. PTF에 대해서는 다음 지원 페이지를 참조하십시오.
<http://www.ibm.com/software/awdtools/wdt400/support/>
- iSeries 서버에 대한 NET USE 액세스가 있음

샘플 파일 설치

iSeries 시나리오 어플리케이션을 사용하려면 다음 파일에 대해 작업해야 합니다.

- Wholesale.savf
- Retailstor.savf
- Qdtssfl.savf
- SV000501.zip
- SV000514.zip
- SV000618.zip
- SV001585.zip
- SV001586.zip

.savf 파일에는 iSeries 데이터와 RPG 프로그램이 들어 있으며, .zip 파일에는 iSeries 데이터를 처리하기 위해 iSeries 프로그램과 상호작용하는 웹 어플리케이션이 들어 있습니다. 먼저 .savf 파일을 iSeries 서버로 복원해야 하며 .zip 파일을 Development Studio Client로 가져와서 Workbench에서 어플리케이션을 실행할 수 있습니다.

.savf 파일 복원

이 안내서의 샘플에 대해 작업하려면 WHOLESALE 및 RETAILSTOR 라이브러리를 iSeries 서버로 복원해야 합니다. 내용이 서로 다르므로, 제품의 이전 릴리스에 대한 라이브러리를 이미 복원한 경우에도 이를 수행해야 합니다.

주: 샘플 라이브러리 설치에 사용되는 .savf 파일은 V5R1 이후 iSeries 서버에서 사용 됩니다. 이 시나리오의 목적을 달성하기 위해 두 라이브러리는 동일한 iSeries 서버로 복원되지만, 실제 비즈니스용으로 이 어플리케이션을 개발 중인 경우 두 라이브러리를 두 개의 서로 다른 iSeries 서버로 복원합니다. WHOLESALE 라이브러리는 웹 서비스를 제공하는 iSeries 서버로 복원되며, RETAILSTOR 라이브러리는 소매 상점에 속하는 iSeries 서버로 복원됩니다.

Wholesale.savf 파일을 복원하려면 다음을 수행하십시오.

1. 그린 스크린 에뮬레이터를 통해 iSeries 서버에 로그인하십시오.
 - a. 저장 파일을 포함할 라이브러리를 작성하십시오. 에뮬레이터에 새로운 라이브러리를 작성하려면 CRTLIB를 입력하십시오.
 - b. 라이브러리의 이름을 SCENARIO로 지정하십시오.
 - c. 다음 행으로 탭하십시오. 라이브러리 유형으로 *TEST를 지정한 후 변경사항을 저장하려면 Enter를 누르십시오.
 - d. CRTSAVF 명령을 사용하여 두 개의 저장 파일을 작성하십시오. 두 행 사이에서 Enter를 누르십시오.

```
CRTSAVF FILE(SCENARIO/WHOLESALE)
CRTSAVF FILE(SCENARIO/RETAILSTOR)
```


이들 행은 Scenario 라이브러리에 저장 파일을 작성하고자 함을 나타냅니다.

2. 워크스테이션에서 명령 프롬프트 창을 여십시오.

- a. .savf 파일이 있는 디렉토리로 변경해야 합니다. 기본적으로,
cd c:\wdsc\wdscsampl을 입력해야 합니다. 다른 드라이브에 제품을 설치한 경우 또는 Development Studio Client의 홈 디렉토리로 "wdsc"를 사용하지 않도록 선택한 경우, 제품을 설치한 올바른 위치에 있는 wdscsampl 디렉토리로 변경하십시오.
- b. 명령행에서 ftp *hostname*을 입력하십시오. 여기서 *hostname*은 iSeries 서버 이름입니다(예를 들어, PROD400).
- c. iSeries 서버에 대한 사용자 ID와 암호를 입력하십시오.
- d. 명령행에서 cd /qsys.lib/scenario.lib를 입력하여 Scenario 라이브러리로 전환하십시오.
- e. 다음 행을 입력하십시오.

```
bin
put WHOLESale.savf WHOLESale.savf
put RETAILSTOR.savf RETAILSTOR.savf
quit
```

위의 행은 로컬 시스템에서 저장 파일을 가져와서 iSeries 서버에 저장하려는 것을 (종료 전에) 지정합니다.

3. iSeries 콘솔로 되돌아가서, Wholesale 라이브러리를 복원하십시오.

- a. RSTLIB를 입력한 다음 F4 키를 눌러서 라이브러리 복원 방법을 정의하십시오.
- b. 저장된 라이브러리 필드에서 WHOLESale을 입력한 다음 Tab 키를 누르십시오.
- c. 장치 필드에서 *savf을 입력한 다음 Tab 키를 누르십시오.
- d. 추가 값을 표시하려면 다음 필드에서 Enter를 누르고 저장 파일 필드로 Tab 키를 눌러 이동하십시오.
- e. 저장 파일 필드에 WHOLESale을 입력한 다음, Tab 키를 누르십시오.
- f. 라이브러리 필드에서 기존 값을 삭제한 다음 scenario를 입력하십시오.
- g. WHOLESale 라이브러리를 iSeries 서버로 복원하려면 Enter를 누르십시오.

4. Retailstor.savf 파일에 대해 다음 프로시저를 반복하십시오.

- a. RSTLIB를 입력한 다음 F4 키를 눌러서 라이브러리 복원 방법을 정의하십시오.
- b. 저장된 라이브러리 필드에서 RETAILSTOR을 입력한 다음 Tab 키를 누르십시오.
- c. 장치 필드에서 *savf을 입력한 다음 Tab 키를 누르십시오.
- d. 추가 값을 표시하려면 다음 필드에서 Enter를 누르고 저장 파일 필드로 Tab 키를 눌러 이동하십시오.
- e. 저장 파일 필드에 RETAILSTOR을 입력한 다음, Tab 키를 누르십시오.
- f. 라이브러리 필드에서 기존 값을 삭제한 다음 scenario를 입력하십시오.

- g. 조치를 저장하고 RETAILSTOR 라이브러리를 iSeries 서버로 복원하려면 Enter를 누르십시오.

라이브러리 목록 수정

RETAILSTOR 및 WHOLESALe 라이브러리를 iSeries 서버로 복원했지만, 이들이 사인 온할 때 사용하는 라이브러리 목록에 있는지 확인해야 합니다. 라이브러리 목록에 없으면, 웹 어플리케이션이 라이브러리를 찾지 못하게 될 수도 있으며 어플리케이션을 실행하거나 빌드할 때 문제점이 발생할 수 있습니다. 또한 QGPL 라이브러리가 라이브러리 목록에 있는지 확인해야 합니다.

이를 수행하려면, 라이브러리 목록에 대한 변경사항이 영구적이도록 작업 설명 및 사용자 프로파일을 갱신해야 합니다.

서로 다른 사용자가 다른 방법으로 라이브러리 목록을 수정할 수 있습니다. 방법을 모르는 경우 다음 예제를 참조하십시오.

1. iSeries 서버에서 dspllibl을 입력하고 Enter를 눌러서 라이브러리 목록을 표시하십시오.
2. RETAILSTOR, WHOLESALe 및 QGPL 라이브러리가 모두 라이브러리 목록에 있는지 확인하십시오. 있는 경우, 목록에서 나온 다음 애플레이터 창을 종료하십시오. 라이브러리 목록에 없을 경우에는 다음을 수행하십시오.
 - a. F12 키를 눌러서 현재 화면을 취소하십시오.
 - b. 작업 설명의 이름을 찾으십시오. 이를 수행하려면, dspusrprf *username*을 입력한 후 Enter를 눌러서 사용자 프로파일을 표시하십시오. 여기서 *username*은 iSeries 서버에 사인온하는 데 사용하는 ID입니다. 작업 설명은 사용자 프로파일 페이지 중 한 페이지에 있습니다(PgDn을 사용하십시오). 작업 설명을 찾으면, 이를 기록한 후 F12 키를 눌러서 화면을 종료하십시오.
 - c. 명령행에서 chgjobd를 입력한 다음 F4 키를 누르십시오.
 - d. 추가 매개변수를 참조하려면 작업 설명 필드에서 작업 설명 이름을 입력한 다음 F10 키를 누르십시오.
 - e. 초기 라이브러리 목록으로 PgDn 하십시오.
 - f. 초기 라이브러리 목록 아래 필드에 "+ 추가 값의 경우"가 표시됩니다. 더하기 기호를 입력한 다음, Enter 키를 누르십시오.
 - g. 추가 값 지정 페이지에서 존재하는 항목을 삭제한 후 RETAILSTOR, WHOLESALe 및 QGPL을 입력하십시오. 각각의 값 사이에 Tab 키를 입력하십시오. 어플리케이션이 RETAILSTOR 라이브러리를 먼저 찾아야 하므로, 반드시 이 순서대로 라이브러리를 입력해야 합니다. Enter를 누르십시오.
 - h. 변경사항을 저장하려면 Enter를 다시 누르십시오.
 - i. 명령행에서 chgusrprf를 입력한 다음 Enter를 누르십시오.

- j. 사용자 이름을 입력한 다음, Enter를 누르십시오.
- k. 추가 매개변수를 보려면 F10 키를 누르십시오.
- l. 작업 설명으로 PgDn 하십시오.
- m. 작업 설명의 이름이 dspusrprf 명령으로 찾은 작업 설명과 일치하지 않을 경우, 해당 명령으로 찾은 작업 설명의 이름을 입력한 후 Enter를 눌러서 변경 사항을 저장하십시오.

Development Studio Client에서 어플리케이션 가져오기 및 실행

Development Studio Client에 대하여 설치된 프로그램 파일에는 다섯 개의 zip 파일이 존재합니다(기본적으로, c:\wdsc\wdscsampl 디렉토리에 존재함). 이들 파일은 다음과 같습니다.

- SV000514.zip
- SV000618.zip
- SV001585.zip
- SV001586.zip
- SV000501.zip

각각의 zip 파일을 Development Studio Client Workbench로 별도로 가져와서 해당 EAR(Enterprise Archive) 파일을 각각 지정합니다. EAR 파일은 .ear 확장자를 갖는 표준 JAR(Java Archive) 파일입니다. 복수의 웹 프로젝트를 포함할 수 있으며, 이를 사용하여 웹 어플리케이션을 패키징하고 WAS(WebSphere Administrative Server)에 전개합니다.

처음 네 개 파일은 웹 프로젝트에 해당하며, 다섯 번째 파일은 WebFacing 프로젝트에 해당합니다. 먼저 웹 프로젝트를 작성한 다음, WebFacing 프로젝트를 작성합니다. 결국 zip 파일의 내용을 5개의 프로젝트로 가져옵니다.

로컬 시스템에 .zip 파일 추출

시나리오 .zip 파일의 내용을 Development Studio Client Workbench로 가져오려면, 먼저 .zip 파일이 아닌 파일 시스템으로 가져올 수 있도록 별도의 디렉토리로 압축을 풀어야 합니다.

주: WinZip 이외의 프로그램을 사용 중인 경우, 해당 프로그램에서 수행하는 방식으로 작업을 완료하십시오.

1. 로컬 시스템에서 c:\wdsc\wdscsampl로 이동하십시오. 다른 드라이브 또는 다른 기본 폴더에 설치하도록 선택한 경우 c:\wdsc 디렉토리가 다를 수 있습니다. 해당하는 경우, 디렉토리를 이동한 후 \wdscsampl을 찾으십시오.
2. SV000501.zip 파일을 두 번 클릭하십시오.

3. 압축풀기 단추를 클릭하여 zip 파일이 위치한 폴더로 이동하십시오(기본적으로, c:\wdsc\wdscsampl).
4. 모든 파일 라디오 단추를 선택한 다음, 압축풀기를 클릭하십시오.
5. wdscsampl 디렉토리를 검사하여 SV000501 폴더가 작성되었으며, 모든 필수 파일이 포함됨을 확인할 수 있습니다.
6. 모든 파일이 c:\scenario\SV000501로 풀리게 됨에 주의하십시오. 파일 시스템을 가져올 때 실수로 선택하지 않도록 SV000501.zip 파일을 삭제하십시오.
7. 나머지 네 개의 .zip 파일(SV000618.zip, SV001585.zip, SV001586.zip 및 SV000514.zip)에 대하여 이 섹션의 단계를 반복하십시오.

웹 프로젝트 작성

이 섹션에서는 SV000514, SV000618, SV001585 및 SV001586에 대한 웹 프로젝트를 작성합니다. 각각의 프로젝트를 작성하기 위한 단계가 별도로 요약됩니다. 단계들이 반복적으로 보이지만, 네 개의 프로젝트를 작성할 때 주의해야 할 몇 가지 중요한 차이점이 있다는 것에 주의하십시오(EAR 파일과 연관되는 프로젝트와 관련하여).

웹 프로젝트를 작성하려면 다음을 수행하십시오.

1. 시작 메뉴에서 iSeries용 IBM WebSphere Development Studio Client를 여십시오.
2. Development Studio Client에서 메뉴 표시줄의 창 > **Perspective** 열기 > 기타 > 웹을 클릭하여 웹 Perspective를 열고 확인을 클릭하십시오.
3. 파일 > 신규 > 동적 웹 프로젝트를 클릭하십시오.
 - a. 프로젝트 이름 필드에서 SV000514를 입력하십시오.
 - b. 고급 옵션 구성 선택란을 선택한 후 다음을 클릭하십시오.
 - c. **EAR** 프로젝트 필드 옆의 신규 단추를 클릭하십시오.
 - d. 프로젝트 이름 필드에서 SVWholeSaleEAR을 입력한 후 두 개 대화 상자에서 완료를 클릭하십시오.
4. 파일 > 신규 > 동적 웹 프로젝트를 클릭하십시오.
 - a. 프로젝트 이름 필드에서 SV001586을 입력한 다음 다음을 클릭하십시오(고급 옵션 구성 선택란이 자동으로 선택됩니다).
 - b. **EAR** 프로젝트 드롭다운 필드에서 SWholeSaleEAR이 선택되었는지 검증하십시오.
 - c. 완료를 클릭하십시오.
5. 파일 > 신규 > 동적 웹 프로젝트를 클릭하십시오.
 - a. 프로젝트 이름 필드에서 SV000618을 입력하고 다음을 클릭하십시오.
 - b. **EAR** 프로젝트 필드 옆의 신규 단추를 클릭하십시오.

- c. 새 프로젝트 이름 필드에서 SVStoreEAR을 입력한 후 두 개 대화 상자에서 완료를 클릭하십시오.
- 6. 파일 > 신규 > 동적 웹 프로젝트를 클릭하십시오.
 - a. 프로젝트 이름 필드에서 SV001585를 입력하고 다음을 클릭하십시오.
 - b. EAR 프로젝트 드롭다운 필드에서 SVStoreEAR이 선택되었는지 검증하십시오.
 - c. 완료를 클릭하십시오.

웹 프로젝트 파일 가져오기

파일 시스템을 작업공간으로 가져오기 위해, 다음 프로시저는 SV000514의 파일을 가져오는 방법을 요약합니다. 프로시저는 세 개의 다른 웹 프로젝트 파일 시스템 (SV001586, SV001585 및 SV000618)과 동일합니다. 이 섹션 뒤의 섹션은 WebFacing 프로젝트를 작성하고 가져오는 방법을 설명합니다.

파일을 가져오려면 다음을 수행하십시오.

1. 찾아보기를 클릭한 후 탐색하십시오.
2. **SV000514** 폴더를 마우스 오른쪽 단추로 클릭한 후 가져오기를 선택하십시오.
3. 가져오기 마법사에서 파일 시스템을 클릭한 후 다음을 클릭하십시오.
4. 시나리오 파일을 추출한 디렉토리를 찾아보십시오(기본적으로, c:\wdsc\wdscsampl).
5. **SV000514**를 클릭한 후 확인을 클릭하십시오.
6. 모든 SV000501 파일 시스템 구성요소를 선택하려면 모두 선택을 클릭하십시오.
7. 정보 폴더 필드에서 기본적으로 이미 입력되어 있지 않은 경우 **SV000514**를 입력하십시오.
8. 경고 없이 기존 자원 겹쳐쓰기 선택란을 선택하십시오.
9. 완료를 클릭한 후 문맥 루트 수정에 관하여 묻는 대화 상자가 표시되면 예를 클릭하십시오.
10. SV001586, SV001585 및 SV000618에 대해 2 - 9 단계를 반복하십시오. 단계를 완료할 때 반드시 서로 다른 프로젝트 이름을 사용하였는지 확인하십시오.

WebFacing 프로젝트 파일 가져오기 및 WebFacing 프로젝트 작성

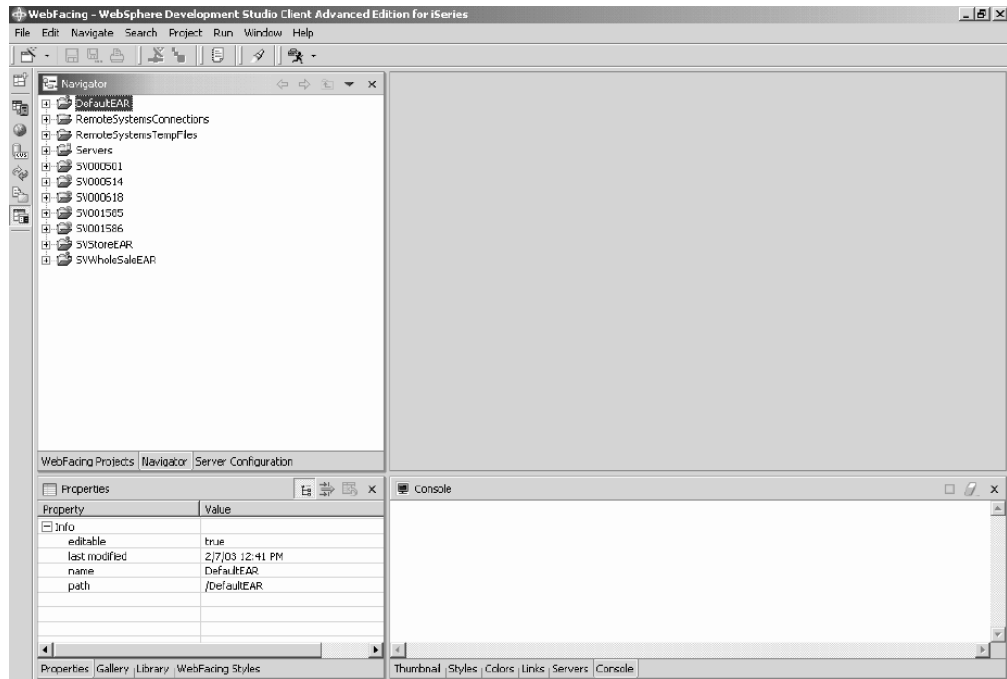
SV000501 프로젝트는 WebFacing 프로젝트이므로, 방금 작성한 네 개의 프로젝트와 다릅니다. WebFacing 프로젝트는 서로 다른 파일을 포함하며, iSeries 서버에서 DDS 파일을 처리하기 위한 특정 고유 구조를 갖기 때문에 다릅니다.

파일을 가져오고 WebFacing 프로젝트를 작성하려면 다음을 수행하십시오.

1. 현재 프로젝트 네비게이터 보기가 아닌 경우, 프로젝트 네비게이터 탭을 클릭하십시오.
2. 보기 내부를 마우스 오른쪽 단추로 클릭한 후 가져오기를 선택하십시오.

3. 가져오기 창에서 **WebFacing** 프로젝트를 선택한 후 다음을 클릭하십시오.
4. 시나리오 파일을 추출한 디렉토리를 찾아보십시오(기본적으로, c:\wdsc\wdscsampl).
5. **SV000501**을 선택한 후 확인을 클릭하십시오. 다음을 클릭하십시오.
6. 창의 찾은 **WebFacing** 프로젝트 영역에서 **SV000501** 선택란을 선택하십시오.
7. **EAR(엔터프라이즈 어플리케이션 프로젝트)** 필드에서 SVStoreEAR을 입력하십시오.
8. 완료를 클릭하십시오.

이제 모든 파일을 가져왔기 때문에, 다음과 유사한 작업공간이 표시됩니다.



가져온 프로젝트 옆에 오류가 표시될 경우(빨간색 x 마크로 표시됨), 각 프로젝트를 마우스 오른쪽 단추로 클릭하고 프로젝트 다시 빌드를 선택하십시오. 작업공간의 내용에 따라, 이를 수행하면 대부분의 오류가 정리되고 어플리케이션을 제대로 실행할 수 있게 됩니다. 오류가 여전히 표시될 경우 일부 코드가 Development Studio Client 5.1 릴리스로 완전히 이주되지 않았기 때문에 발생할 수 있습니다. 제품에 익숙해지려면 이 장의 연습문제를 계속해서 수행하거나 또는 어플리케이션 빌드를 시도해보기 위해 초안에서 다음 장으로 넘어갈 수 있습니다.

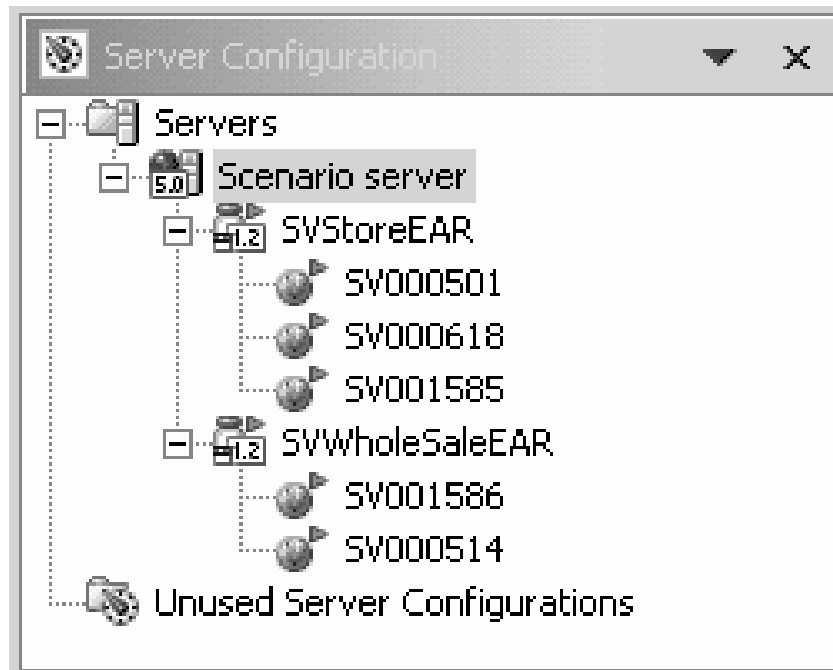
서버 구성

이제 모든 소스 파일을 가져왔으므로, SVWholesaleEAR 및 SVStoreEAR 어플리케이션을 인식할 수 있도록 WebSphere 테스트 환경 서버를 구성해야 합니다.

WebSphere 테스트 환경 서버를 구성하려면 다음을 수행하십시오.

1. 서버 Perspective로 전환하십시오. Workbench 메뉴 표시줄에서 창 > **Perspective 열기** > 기타 > 서버를 클릭한 다음 확인을 클릭하십시오.

2. 서버 구성 보기에서 서버를 마우스 오른쪽 단추로 클릭한 다음 신규 > 서버 및 서버 구성을 선택하십시오.
3. 서버 이름 필드에서 Scenario server를 입력하십시오.
4. 폴더 필드에서 Scenario folder를 입력하십시오.
5. 서버 유형 상자의 **WebSphere** 버전 **5.0**을 펼친 다음 기본적으로 아직 선택되지 않은 경우 테스트 환경을 클릭하십시오.
6. 완료를 클릭한 후 메시지가 표시되면 예를 클릭하십시오.
7. EAR 파일을 서버 구성에 추가하려면 다음을 수행하십시오.
 - a. 서버를 펼치십시오.
 - b. **Scenario server**를 마우스 오른쪽 단추로 클릭한 다음 추가 > **SVWholeSaleEAR**을 선택하십시오.
 - c. 서버를 펼치십시오.
 - d. **Scenario server**를 마우스 오른쪽 단추로 다시 클릭한 다음, 프로젝트 추가 및 제거를 선택하십시오.
 - e. 모두 추가 단추를 클릭한 후 SVStoreEAR과 SVWholeSaleEAR을 모두 창의 오른쪽으로 이동하십시오.
 - f. 완료를 클릭하십시오.
8. 서버 구성 보기에서 모든 나열된 어플리케이션을 표시하려면 **Scenario server**를 펼친 다음 **SVStoreEAR** 및 **SVWholeSaleEAR**을 펼치십시오.



서버가 프로젝트를 선택했는지 확인하려면 다음을 수행하십시오.

1. 서버 구성 보기 오른쪽에 있는 서버 보기에서 **Scenario server**를 마우스 오른쪽 단추로 클릭한 다음, 공개를 선택하십시오(서버 보기를 볼 수 없는 경우 서버 탭을 클릭하십시오). 공개 과정을 표시하는 공개 대화 상자가 표시됩니다.
2. 어플리케이션을 WebSphere Application Server 테스트 환경에 공개하려면 확인을 클릭하십시오.
3. 공개가 완료되면, 대화 상자에서 확인을 클릭하십시오.
4. 동일한 보기에서 **Scenario server**를 마우스 오른쪽 단추로 다시 클릭한 다음 시작을 클릭하십시오. 서버의 활동을 보여주는 콘솔 보기가 열립니다. 보기 맨 아래 "servername이(가) e-business를 위해 시작되었습니다." 메시지 행이 표시되면 서버가 시작됩니다. 모든 메시지를 보려면 아래로 화면이동하십시오.

WebFacing 프로젝트 구성

어플리케이션을 실행하거나 SV000501 모듈에서 작업하기 전에 먼저 실행할 어플리케이션에 대한 WebFacing 서버를 시작한 다음, 올바른 iSeries 서버를 사용할 수 있도록 프로젝트를 구성해야 합니다.

WebFacing 서버를 시작하려면 다음을 수행하십시오.

1. 그린 스크린 5250 에뮬레이터를 열고 사용자 ID와 암호를 사용하여 사인 온하십시오.
2. 명령행에서 `strtcpsvr *webfacing`를 입력하십시오.
3. 화면 맨 아래 "WEBFACING 서버 시작"이라는 메시지가 표시됩니다.

이제 WebFacing 특성을 변경해야 합니다. Development Studio Client Workbench로 돌아가십시오.

1. 웹 Perspective로 전환하십시오(화면 왼쪽에 클러스터된 아이콘을 클릭하여 Perspective를 전환할 수 있습니다).
2. 프로젝트 구조를 볼 수 있도록 프로젝트 네비게이터 탭을 클릭하십시오.
3. **SV000501**을 펼친 다음 웹 전개 설명자를 두 번 클릭하십시오.
4. 매개변수 탭을 클릭하십시오.
5. **WFDefaultHost**를 클릭하십시오.
6. 값 필드에서 기존 값을 지우고 iSeries 서버의 이름을 입력하십시오.
7. Workbench 메뉴 표시줄에서 저장 아이콘을 클릭하거나 파일 > 웹 전개 설명자 저장을 클릭하십시오. 메시지를 수신할 경우 확인을 클릭하십시오. 파일을 닫으십시오.

서버 정보 정의

이제 모든 5개의 프로젝트에 대해 iSeries 서버 정보를 정의해야 합니다. 이는 사용자 ID와 암호로 iSeries 서버에서 실행되도록 구성되게 해줍니다. 서버 정보를 정의하려면 다음을 수행하십시오.

1. 웹 Perspective에 있는지 확인하십시오.
2. 프로젝트 네비게이터 보기에서 **SV000501**을 마우스 오른쪽 단추로 클릭한 후 **iSeries 웹 도구 런타임 구성 지정**을 선택하십시오.
3. **iSeries** 서버 이름 필드에서 기존 값을 삭제한 다음, iSeries 서버의 이름을 입력하십시오.
4. 다른 기존 값을 삭제한 다음 사용자 ID와 암호를 입력하십시오.
5. 완료를 클릭하십시오.(필요한 경우 완료를 다시 클릭하십시오).
6. 다른 네 개의 프로젝트(SV000514, SV000618, SV001585 및 SV001586)에 대해 2 - 5 단계를 반복하십시오.

Workbench에서 어플리케이션 실행

이 때 Workbench의 WebSphere 테스트 환경에서 어플리케이션을 실행할 수 있습니다. 올바르게 실행 중인 경우, EAR 파일을 iSeries 서버로 내보낸 다음, iSeries WebSphere Application Server에 전개할 수 있습니다.

Workbench에서 고객으로 어플리케이션 실행

고객으로서 소매 상점에 대한 쇼핑 웹 페이지를 시작하고, 항목을 검색한 다음 수량과 크기를 주문합니다.

주: web.xml 파일이 다음 파일, `http://java.sun.com/j2ee/dtds/web-app_2_2.dtd`를 찾기 때문에 방화벽이 설치된 상태에서 어플리케이션을 실행하려고 시도할 경우 문제점이 발생할 수도 있습니다. 이러한 문제점을 해결하려면 어플리케이션을 실행하기 전에 먼저 모든 web.xml 파일에 있는 DOCTYPE 명령문을 다음과 같이 변경하십시오.

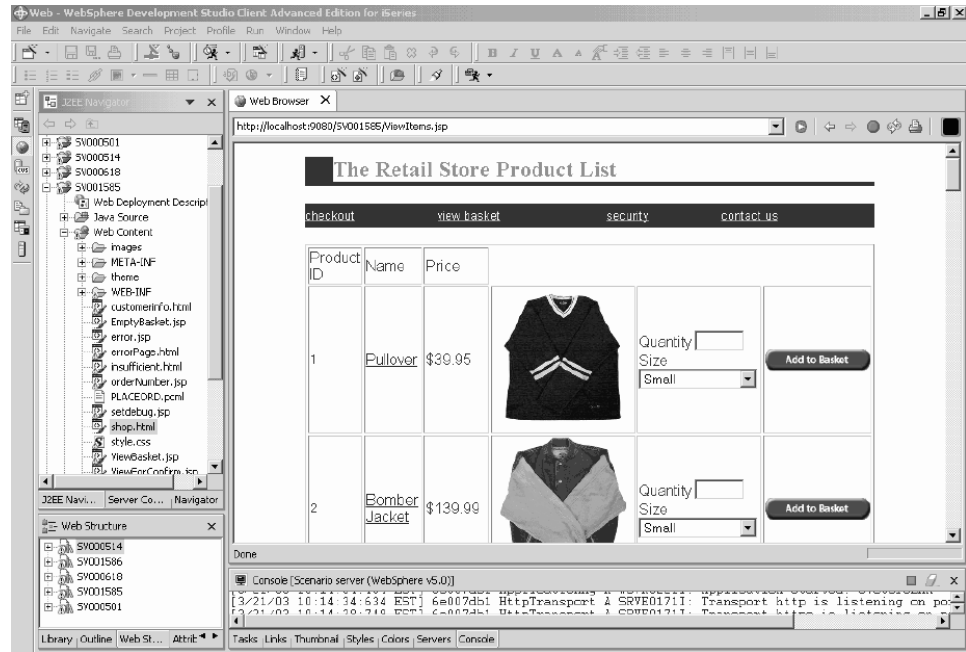
```
!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"x:/WDSC/eclipse/plugins/com.ibm.etools.j2ee/dtds/web-app_2_2.dtd"
```

여기서 `x:/wdsc`는 제품을 설치한 디렉토리입니다. 일반적으로 프로젝트 이름 > **Web Content** > **WEB-INF** 아래에서 web.xml 파일을 찾을 수 있습니다. 파일을 연 후, 소스 탭을 클릭하여 파일을 직접 편집하십시오.

Workbench에서 고객으로서 어플리케이션을 실행하려면 다음을 수행하십시오.

1. 웹 Perspective에서 **SV001585 > Web Content**를 펼치십시오.
2. **shop.html**을 마우스 오른쪽 단추로 클릭한 다음 서버에서 실행을 선택하십시오. Workbench 브라우저에서 어플리케이션이 실행됩니다.
3. 완료를 클릭하십시오.
4. T셔츠 이미지를 클릭하여 어플리케이션을 시작하십시오.

- 다음 페이지에 값을 입력하십시오. 폴오버 또는 보머 재킷을 주문 중인 고객인 것처럼 가정하여 크기와 수량을 선택한 다음 항목을 장바구니에 추가하십시오.



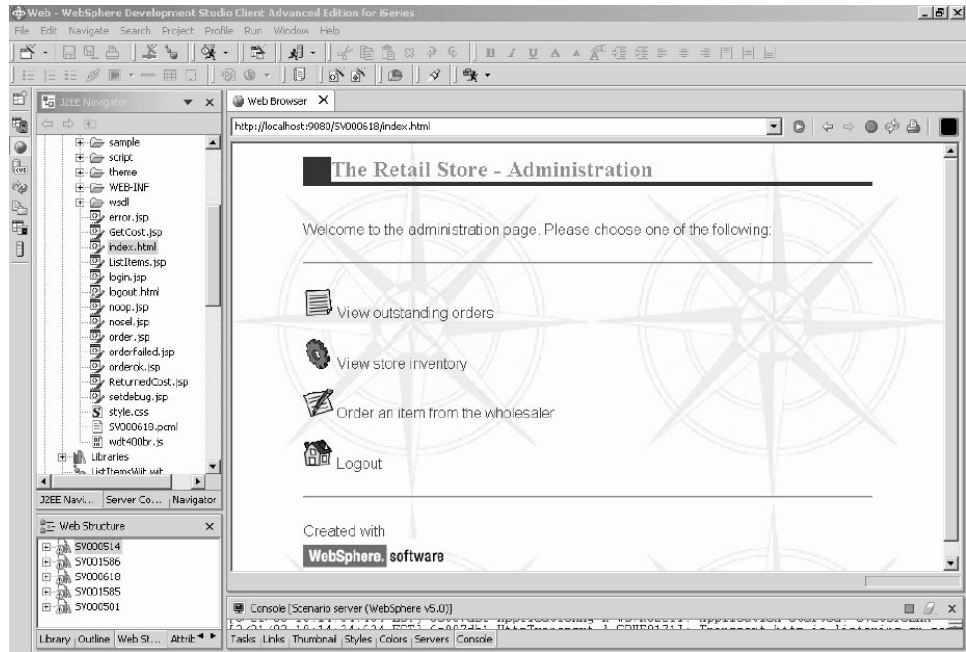
"주문이 처리되었음"이라는 텍스트를 표시하는 페이지가 샘플의 마지막 페이지입니다. 샘플에 만족할 경우, 브라우저를 닫을 수 있습니다.

Workbench에서 관리자로 어플리케이션 실행

관리자로서 소매 상점에 대한 관리 웹 페이지를 시작한 다음 주문을 또는 상점 재고를 점검합니다. Workbench에서 관리자로서 어플리케이션을 실행하려면 다음을 수행하십시오.

- 웹 Perspective에서 **SV000618 > Web Content**를 펼치십시오.
- index.html**을 마우스 오른쪽 단추로 클릭한 다음 서버에서 실행을 선택하십시오. Workbench 브라우저에서 어플리케이션이 실행됩니다.

- 상점 재고 보기 왼쪽에 있는 아이콘을 클릭하면 사용자가 상점에 대해 구입할 항목을 결정하는 관리자인 것처럼, 다음과 같은 페이지가 표시됩니다.



관리 페이지를 표시하려면 아이콘을 클릭하십시오.

WebSphere Application Server에 시나리오 어플리케이션 전개

이제 Workbench 테스트 환경에서 iSeries 시나리오 어플리케이션을 실행했으므로, 실제 세계에서 어플리케이션을 실행 중인 것처럼 어플리케이션을 WebSphere Application Server에 전개할 수 있습니다. 그러나 어플리케이션을 전개하기 전에, 다음 절에서 설명된 것처럼 관리자 페이지 보호 및 어플리케이션이 오른쪽 장소를 가리킬 수 있도록 웹 서비스 URL 변경과 같은, 몇 가지 조정을 수행해야 합니다.

주: WebSphere Application Server에 대한 전개는 선택적입니다. 계속해서 다음 장으로 이동하여 WebSphere Application Server에서 어플리케이션을 테스트하지 않고 모듈을 완료할 수 있습니다.

WebSphere Application Server 구성

이 장의 앞부분에서 19 페이지의 『Workbench에서 어플리케이션 실행』에 대해 설명했습니다. 이제 파일을 iSeries 서버에 전개했으므로, WebSphere Application Server를 사용하여 iSeries에서 어플리케이션을 실행할 수 있습니다.

어플리케이션을 WebSphere Application Server로 전개하려면(선택적) 다음을 확인해야 합니다.

- iSeries 서버에서 WebSphere Application Server 버전 4.0, 버전 5.0 또는 WebSphere Application Server Express 인스턴스를 작성했으며 액세스 권한이 있는지와 인스턴스가 실행 중인지(WebSphere Application Server에 대한 전개를 테스트하려는 경우에만).
- iSeries 서버의 HTTP 및 WebSphere Application Server 인스턴스에 대한 포트 번호를 알고 있는지.
- WebSphere 관리 콘솔 버전 4.0 또는 5.0을 워크스테이션에 설치했는지.

주: WAS 버전 5.0의 경우, 콘솔이 브라우저 기반이므로 워크스테이션에 콘솔이 있을 필요가 없습니다. 다양한 WAS 버전에 관한 추가 정보는 온라인 문서를 참조하십시오.

웹 사용 가능 iSeries 어플리케이션이 WebSphere Application Server를 사용하여 Java™ Servlet과 웹 사용자의 브라우저 및 iSeries 프로그램 또는 데이터 간에 통신하는 JSP(JavaServer Page)™를 실행합니다. 동일한 iSeries 시스템에서 HTML 페이지와 JSP 파일을 제공하려면 해당 시스템에 HTTP 서버가 또한 필요합니다. Apache에서 보강한 IBM HTTP Server를 사용하도록 권장합니다. 이 서버의 사용법에 관한 문서는 다음 사이트, http://publib.boulder.ibm.com/pubs/html/series_http/v5r1/index.htm에 있는 iSeries용 IBM HTTP Server Documentation Center에서 찾을 수 있습니다.

WebSphere Application Server는 다양한 프로세스에 대해 생성된 JavaServer Page, Java bean™ 및 Java Servlet을 실행합니다. iSeries용 IBM WebSphere Application Server용 기본 문서 자원 및 iSeries용 IBM WebSphere 관리 콘솔은 다음 웹 사이트에서 사용 가능합니다.

- iSeries용 IBM WebSphere Application Server 버전 4.0 Advanced Edition: <http://publib.boulder.ibm.com/was400/40/AE/english/docs/>
- iSeries용 IBM WebSphere Application Server 버전 4.0 Advanced Single Server Edition: <http://publib.boulder.ibm.com/was400/40/AEs/english/docs/>

IBM WebSphere® Application Server 문서에 익숙하게 되면, 특히 *J2EE* 모듈, *WebSphere Application Server 설치* 및 *WebSphere 관리 서버에 복수 인스턴스 설정* 섹션을 참조하십시오. 최소한 설치 링크 아래의 단계를 수행해야 합니다.

사이트 맵을 사용하여 WebSphere 관리 콘솔에 대한 필수 PTF를 설치, 구성 및 확보하는 방법에 관한 정보를 찾을 수 있습니다.

관리자 페이지 보호

관리자 페이지, `index.html`은 권한이 있는 사용자만이 액세스할 수 있어야 하므로 적절하게 보호되어야 합니다. 이는 웹 어플리케이션 논리에서 프로그램 방식으로 또는 WebSphere의 보안 기능을 사용하여 수행할 수 있습니다. 이 시나리오에서는 WebSphere

보안을 사용하여 페이지를 보호했습니다. WebSphere Application Server V4.0 Advanced Edition을 사용했습니다. 다음 웹 사이트에서 WebSphere Application Server에 관한 정보를 찾을 수 있습니다.

- iSeries용 IBM WebSphere Application Server 버전 4.0 Advanced Edition: <http://publib.boulder.ibm.com/was400/40/AE/english/docs/>
- iSeries용 IBM WebSphere Application Server 버전 4.0 Advanced Single Server Edition: <http://publib.boulder.ibm.com/was400/40/AEs/english/docs/>
- WebSphere용 RedBooks 페이지: WebSphere Application Server 버전 5.0 및 WebSphere Application Server Express Edition 버전 5.0에 대한 RedBooks를 <http://publib-b.boulder.ibm.com/redbooks.nsf/portals/WebSphere>에서 검색할 수 있습니다.

다른 WebSphere Application Server 버전을 사용 중인 경우, 웹 자원 보호에 관한 해당 버전의 문서를 참조하십시오.

Development Studio Client 내에서 또는 Application Assembly 도구에서 웹 페이지 및 Servlet과 같은, 웹 자원에 대한 보안을 구성할 수 있습니다. 이 시나리오의 경우, Development Studio Client를 사용합니다.

해당 웹 어플리케이션에 대한 보안 구성 및 특성을 검토하려면 다음을 수행하십시오.

1. Development Studio Client에서 웹 Perspective로 전환하십시오.
2. 네비게이터 보기에서 **SV000618 > Web Content > WEB-INF**를 펼치십시오.
3. **web.xml**을 두 번 클릭하여 web.xml 보기를 여십시오.
4. **보안** 탭을 클릭하십시오.
5. 관리 페이지인 index.html을 보호하기 위해, 보안 제한조건이 또한 정의되었습니다. 보기 맨 위에서 **보안 제한조건**을 클릭하십시오.
6. 목록에서 **SecurityConstraint**의 첫 번째 인스턴스를 클릭하십시오.
7. 오른쪽에서 **AdminPage**를 클릭하십시오.
8. 편집을 클릭하여 웹 자원 수집 대화 상자를 호출하십시오. index.html에 대한 GET 및 POST 메소드가 사전 선택되어 있습니다.
9. 확인을 클릭하십시오.

보안 역할 섹션에서 "Administrator"라는 정의된 보안 역할을 주의하십시오. 전개 시 개인이 해당 역할에 할당되므로 따라서 index.html 페이지에 대한 액세스가 부여됩니다. 권한 부여 역할 섹션에서 해당 보안 제한조건에 대해 Administrator 역할 액세스가 부여되었습니다. 이러한 보안이 발생할 경우, Administrator 역할에 할당된 사용자만이 사용자 ID 및 암호와 같은 적절한 신임이 제공된 후 index.html 페이지를 액세스할 수 있습니다. 자원이 보호된 경우, WebSphere Application Server는 먼저 사용자 인증을

시도합니다. 인증서를 사용하거나 또는 사용자에게 사용자 ID와 암호를 프롬프트하여 인증을 수행합니다. 프롬트는 기본 인증 대화 상자를 사용하거나 사용자 정의 양식을 사용하여 수행할 수 있습니다.

이 시나리오에서는 login.jsp라는 자체 로그인 페이지를 설계했습니다. 인증 프롬트를 구성하려면 web.xml 보기에서 페이지 탭을 선택하십시오. 로그인 섹션에서 양식이 인증 메소드로 사전 선택되어 있습니다. 또한 로그인 페이지의 이름이 login.jsp입니다. 로그인이 실패하면 오류 페이지가 표시됩니다. 이 때, 어플리케이션은 login.jsp 페이지를 다시 표시합니다.

iSeries WebSphere Application Server 전개용 EAR 파일 작성

어플리케이션을 iSeries WebSphere Application Server에 전개하려면 EAR 파일을 작성해야 합니다. EAR 파일은 .ear 확장자를 갖는 표준 JAR(Java Archive) 파일입니다. 복수의 웹 프로젝트를 포함할 수 있으며, 이를 사용하여 웹 어플리케이션을 패키징하고 WAS(WebSphere Administrative Server)에 전개합니다.

EAR 파일을 구성하려면 다음을 수행하십시오.

1. 웹 Perspective로 전환하십시오.
2. 네비게이터 보기에서 **SVStoreEAR**을 마우스 오른쪽 단추로 클릭한 후 내보내기를 선택하십시오.
3. 내보내기 창에서 **EAR** 파일을 클릭한 후 다음을 클릭하십시오.
4. 지원을 내보낼 위치 아래의 찾아보기를 클릭한 다음 EAR 파일을 저장할 수 있는 iSeries 통합 파일 시스템의 디렉토리로 이동하십시오. (네트워크 드라이브를 iSeries IFS로 맵핑했어야 합니다.)
5. 파일 이름 필드에서 SVRetailStorEAR.ear을 입력한 다음 열기를 클릭하십시오.
6. 완료를 클릭하십시오.
7. SVWholeSaleEAR에 대해 2 - 6 단계를 반복하십시오.

iSeries WebSphere Application Server에 EAR 파일 전개

이제 EAR 파일을 작성했으므로, WebSphere Application Server에 전개할 수 있습니다.

1. WebSphere 관리 콘솔을 여십시오.
2. 엔터프라이즈 어플리케이션을 마우스 오른쪽 단추로 클릭한 다음 엔터프라이즈 어플리케이션 설치를 선택하십시오.
3. 어플리케이션 설치(*ear) 라디오 단추를 선택하십시오.
4. 위쪽에 있는 찾아보기 단추를 클릭하십시오(아래쪽은 사용 불가능합니다).
5. EAR 파일을 내보낸 IFS 디렉토리로 이동하십시오.
6. **SVRetailStorEAR.ear**을 선택하십시오.

7. 어플리케이션명 필드에 "RETAILSTOR"을 입력하십시오.
8. 다음을 클릭하고 관리자의 역할에 대해 iSeries 서버 사용자 ID를 입력하십시오.
9. 웹 모듈용 가상 호스트 선택이라는 페이지가 표시될 때까지 다음을 반복적으로 클릭하십시오.
10. 세 가지 웹 모듈 모두에 대해 가상 호스트 선택을 클릭한 다음, 드롭다운 목록에서 선호하는 가상 호스트를 선택하십시오. (선택할 호스트가 확실하지 않으면, 기본값 또는 **default_host**를 사용하십시오.)
11. 다음을 클릭하십시오.
12. 세 가지 웹 모듈 모두에 대해 서버 선택을 클릭한 다음 사용하려는 서버를 선택하십시오. (선택할 서버가 확실하지 않으면, 기본 서버를 사용하십시오.)
13. 다음을 클릭하십시오.
14. 완료를 클릭한 후 대화 상자에서 확인을 클릭하십시오.
15. 엔터프라이즈 어플리케이션을 다시 마우스 오른쪽 단추로 클릭한 다음 엔터프라이즈 어플리케이션 설치를 선택하십시오.
16. 아래쪽에 있는 찾아보기 단추를 클릭하십시오(위쪽은 사용 불가능합니다).
17. EAR 파일을 배치한 IFS 디렉토리로 이동하십시오.
18. **SVWholesaleEAR.ear**을 선택하십시오.
19. 어플리케이션명 필드에 "WHOLESALE"을 입력하십시오.
20. 완료가 작동될 때까지 다음을 반복해서 클릭하십시오.
21. 완료를 클릭한 후 대화 상자에서 확인을 클릭하십시오.

주: 확인 메시지가 표시되려면 약간의 시간이 걸릴 수도 있습니다.

WebSphere Application Server에서 어플리케이션 실행

고객의 역할로 소매 상점 시작점을 실행하려면 다음 URL을 웹 브라우저에 입력하십시오.

`http://your iSeriesHostName:yourHTTPPortNumber/SV001585/shop.html`

관리자 역할로 WholeSale 시작점을 실행하려면 다음 URL을 웹 브라우저에 입력하십시오.

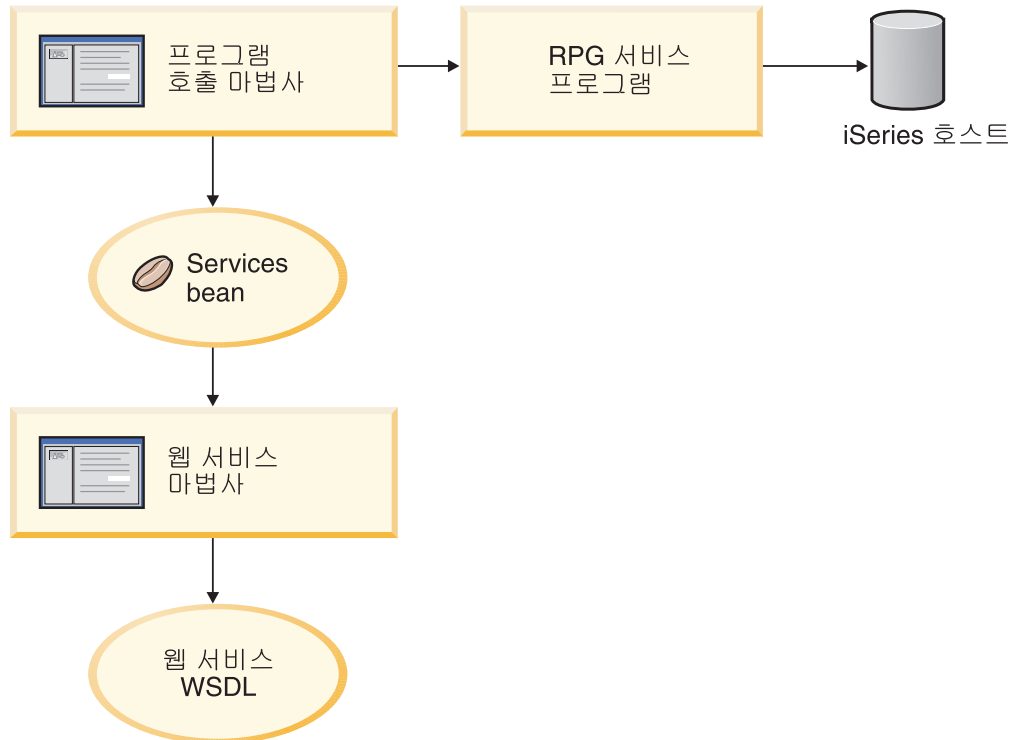
`http://your iSeriesHostName:yourHTTPPortNumber/SV000618/index.html`

HTTP 포트 번호를 모를 경우, WebSphere Application Server 관리자에게 문의하십시오.

제 4 장 단계별 모듈 1: 제품 가격을 리턴하는 웹 서비스 작성 (SV000514)

소개

이 모듈에서는 iSeries 서버에 존재하는 RPG 프로그램에서 웹 서비스를 작성하여 재고에 대한 제품 가격을 표시합니다. 먼저 항목 번호가 주어진 상태에서 iSeries 데이터베이스로부터 항목의 가격을 검색할 수 있는 프로시저를 사용하여 RPG 서비스 프로그램을 작성합니다. iSeries Java 개발 도구에서 프로그램 호출 마법사를 사용하여 RPG 프로그램을 호출하고 서비스 bean을 작성합니다. 그런 다음 웹 서비스 마법사를 사용하여 웹 서비스를 작성하고 생성된 샘플을 사용하여 웹 서비스를 확인합니다.



시작하기 전에

다음 전제조건이 일치할 경우에만 연습문제를 완료할 수 있습니다. 전제조건은 9 페이지의 제 3 장 『시나리오 실행』에서 보다 자세히 설명됩니다.

- iSeries 서버에 대한 TCP/IP 액세스가 있음.

- 최신 iSeries 서버 PTF를 적용했음. PTF 및 서비스 팩 정보는 다음 지원 페이지를 참조하십시오.
<http://www.ibm.com/software/awdtools/wdt400/support/>
- STRTCPSVR *ALL 명령을 사용하여 iSeries 서버를 시작했음
- STRTCPSVR *WEBFACING 명령을 사용하여 WebFacing 서버를 시작했음
- 9 페이지의 제 3 장 『시나리오 실행』의 모든 타스크를 완료했음(Workbench에서 어플리케이션을 테스트하는 데 필요하지 않은 선택적 WebSphere Application Server 타스크 제외).

새 웹 프로젝트 작성

이 웹 서비스를 작성하는 첫 번째 단계는 정보를 보유할 새로운 웹 프로젝트를 작성하는 것입니다.

1. Workbench IDE에서 웹 Perspective로 전환하거나 창 > **Perspective** 열기 > 기타 > 웹 > 확인을 클릭하여 웹 Perspective를 여십시오.
2. 파일 > 신규 > 동적 웹 프로젝트를 클릭하십시오.
3. 프로젝트 이름 필드에서 Project514를 입력하십시오.
4. 고급 옵션 구성 선택란을 선택한 후 다음을 클릭하십시오.
5. **EAR** 프로젝트 필드 옆의 신규 단추를 클릭하십시오.
6. 프로젝트 이름 필드에서 Project514EAR을 입력한 후 두 대화 상자에서 완료를 클릭하십시오.

이제 Project514와 Project514EAR 프로젝트가 프로젝트 네비게이터 보기의 작업공간에 추가된 것을 볼 수 있습니다.

iSeries 서버 정보 정의

웹 프로젝트를 작성한 후, 프로젝트가 정보를 확보하는 데 사용하는 iSeries 서버를 정의해야 합니다.

1. **Project514** 프로젝트를 마우스 오른쪽 단추로 클릭한 다음, **iSeries** 웹 도구 런타임 구성 지정을 선택하십시오.
2. 복원된 WHOLESALE 라이브러리가 존재하는 iSeries 서버의 이름을 입력하십시오 (예를 들어, PROD400).
3. 이 iSeries 서버에 대한 사용자 ID와 암호를 입력하십시오.
4. 라이브러리 필드에 Wholesale을 입력한 다음, 추가를 클릭하십시오.
5. 완료를 클릭하십시오.

RPG 서비스 프로그램 작성

항목 번호가 주어지면 어플리케이션이 항목의 가격을 검색하게 하고자 합니다. 이러한 작업은 QryProdCost라고 하는 프로시저를 포함하는 RPG 서비스 프로그램에 의해 처리됩니다. WHOLESALE 라이브러리는 CWWSSRV라는 RPG 서비스 프로그램을 포함합니다. 이 프로그램은 항목 번호를 입력으로 취하여, WHOLESALE 라이브러리 내에서 Inventory 파일을 열고, inventory 데이터베이스에서 가격을 검색한 다음 가격을 리턴할 수 있는 QryProdCost 프로시저를 포함합니다. 이 작업을 수용하기 위해, 인터페이스에는 두 개의 매개변수(항목 번호에 대하여 하나, 가격에 대하여 하나)가 있습니다. 항목 번호 또는 가격을 찾을 수 없을 경우, RPG 프로그램이 메시지를 인터페이스에 리턴합니다.

이 웹 서비스를 작성하려면 먼저 프로그램 호출 마법사를 사용하여 QryProdCost RPG 프로시저를 호출하는 Java bean을 작성합니다. 그런 다음 웹 서비스를 사용하여 Java bean을 통해 RPG 프로시저를 웹 서비스로 작동하게 합니다.

Java bean을 작성하려면 다음을 수행하십시오.

1. 웹 Perspective에서 **Project514**를 마우스 오른쪽 단추로 클릭한 다음 **신규 > 기타**를 선택하십시오.
2. 창 왼쪽 분할창에서 **iSeries > Java**를 선택한 다음, 창 오른쪽 분할창에 있는 프로그램 호출 **Bean**을 선택하십시오.
3. 프로그램 호출 마법사를 호출하려면 다음을 클릭하십시오.
4. 프로그램 추가 제목 바로 밑에 있는 **Java bean** 이름 필드에서 Inventory를 입력하십시오.
5. 프로그램 오브젝트 필드에서 RPG 서비스 프로그램의 이름인, CWWSSRV를 입력하십시오.
6. 라이브러리 필드에서 WHOLESALE을 입력하십시오.
7. 프로그램 유형 드롭다운 목록에서 ***SRVPGM**을 선택하십시오.
8. 시작점 필드에서 QryProdCost를 입력하십시오.
9. 프로그램 정의를 추가하려면 **확인**을 클릭하십시오.

매개변수 작성 및 Java bean 생성

이제 프로그램을 지정했으므로, 매개변수를 추가할 수 있습니다. CWWSSRV RPG 프로그램은 다음과 같은 두 개의 매개변수를 포함합니다.

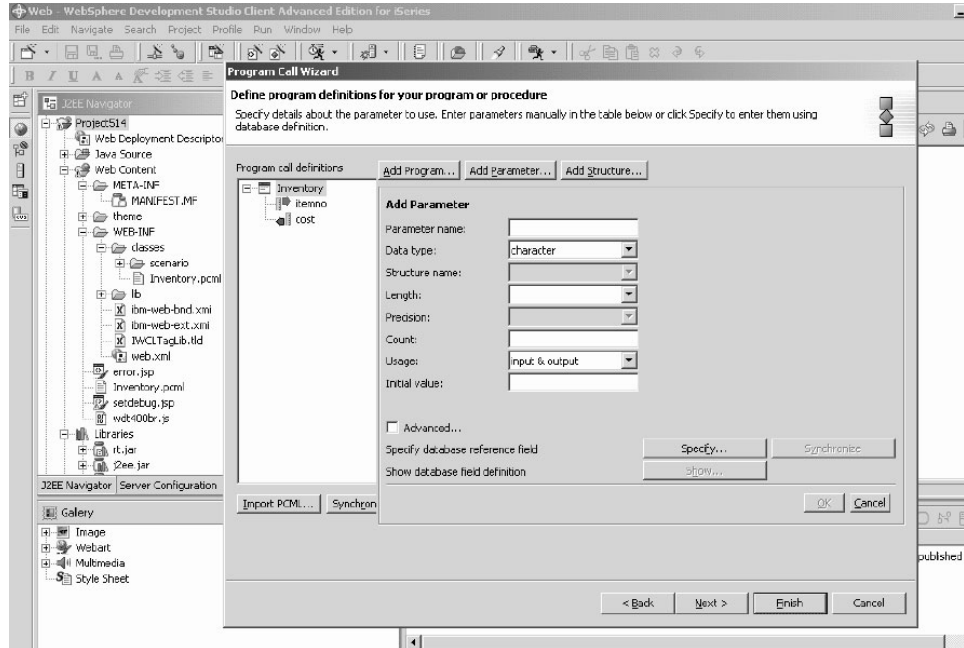
- 항목 번호 매개변수 - 프로그램은 이 번호를 사용하여 데이터베이스에서 항목을 찾습니다.
- 항목 비용 매개변수 - 프로그램은 이 번호를 사용하여 데이터베이스에서 항목의 비용을 찾습니다.

항목 번호 매개변수를 추가하려면 다음을 수행하십시오.

1. 프로그램 호출 마법사의 왼쪽 패널에서 **Inventory** 프로그램 호출 정의를 클릭하여 선택하십시오. 이 조치는 페이지 오른쪽에 있는 필드를 재입력합니다.
2. 매개변수 추가를 클릭하십시오.
3. 매개변수 이름 필드에서 `itemno`를 입력하십시오.
4. 데이터 유형 드롭다운 목록에서 **팩형 십진수**를 선택하십시오.
5. 길이 필드에서 `5`를 입력하십시오.
6. 정밀도 필드에서 `0`을 입력하십시오.
7. 사용 드롭다운 목록에서 **입력**을 선택하십시오.
8. 이 매개변수를 추가하려면 **확인**을 클릭하십시오. 왼쪽 분할창에 `itemno`가 **Inventory** 밑에 표시됩니다. 이제 두 번째 매개변수를 추가할 준비가 되었습니다.

항목 비용 매개변수를 입력하려면 다음을 수행하십시오.

1. 프로그램 호출 마법사의 왼쪽 패널에서 **Inventory** 프로그램 호출 정의를 다시 클릭하여 선택하십시오. 이 조치는 페이지 오른쪽에 있는 필드를 재입력합니다.
2. 매개변수 추가를 클릭하십시오.
3. 매개변수 이름 필드에서 `cost`를 입력하십시오.
4. 데이터 유형 드롭다운 목록에서 **팩형 십진수**를 선택하십시오.
5. 길이 필드에서 `7`을 입력하십시오.
6. 정밀도 필드에서 `2`를 입력하십시오.
7. 사용 드롭다운 목록에서 **출력**을 선택하십시오.
8. 이 매개변수를 추가하려면 **확인**을 클릭하십시오. 왼쪽 분할창에 `cost`가 **Inventory** 밑에 표시됩니다. 이 때, 마법사는 다음과 같이 표시되어야 합니다. 매개변수 왼쪽에 있는 아이콘은 유형이 입력, 입력 및 출력 또는 출력인지를 표시합니다.



이들 매개변수의 패키지 이름을 정의하려면 프로그램 호출 마법사에서 다음을 수행하십시오.

1. 다음을 클릭하십시오.
2. 패키지 필드에서 패키지 이름으로 **scenario**를 입력하십시오. 다른 필드는 기본값을 그대로 두십시오.
3. **Java** 어플리케이션 선택란을 지우십시오.

주: "다음과 같은 파일이 마법사에 의해 생성됩니다." 아래의 파일 목록을 검토하십시오. 생성되는 Java bean 이름은 `InventoryServices.java`입니다.

4. 파일을 생성하려면 **완료**를 클릭하십시오.

Java bean에서 웹 서비스 작성

RPG 프로그램을 호출하는 Java bean을 작성한 후, 다음 단계는 다른 프로그램이 인터넷을 통해 동일한 RPG 프로그램을 액세스할 수 있도록 bean을 웹 서비스로 변환하는 것입니다.

웹 서비스를 작성할 경우, 웹 서비스 마법사는 웹 서비스를 사용해야 하는 사용자에게 분배되는 WSDL 파일을 생성합니다. 서비스를 작성하려면 다음을 수행하십시오.

1. 웹 Perspective의 네비게이터 보기에서 **Project514 > Java 소스 > scenario**를 펼치십시오.
2. **InventoryServices.java**를 마우스 오른쪽 단추로 클릭한 다음 **신규 > 기타**를 선택하십시오.

3. 신규 창에서 웹 서비스를 클릭하고, 창 오른쪽에 있는 웹 서비스를 클릭한 후 다음을 클릭하십시오.
4. 프록시 생성 및 생성된 프록시 테스트 선택란을 선택하십시오.
5. 다음을 세 번 클릭하고 필요한 파일이 생성되었음을 확인하십시오. 웹 서비스 Java Bean ID 창이 표시되면, 완료를 클릭하십시오. 오류가 표시될 경우 완료를 두 번 클릭해야 할 수도 있습니다. 마법사를 완료할 수 없으면, 취소를 클릭하십시오. 어쨌든 대부분의 필수 파일이 작성됩니다.

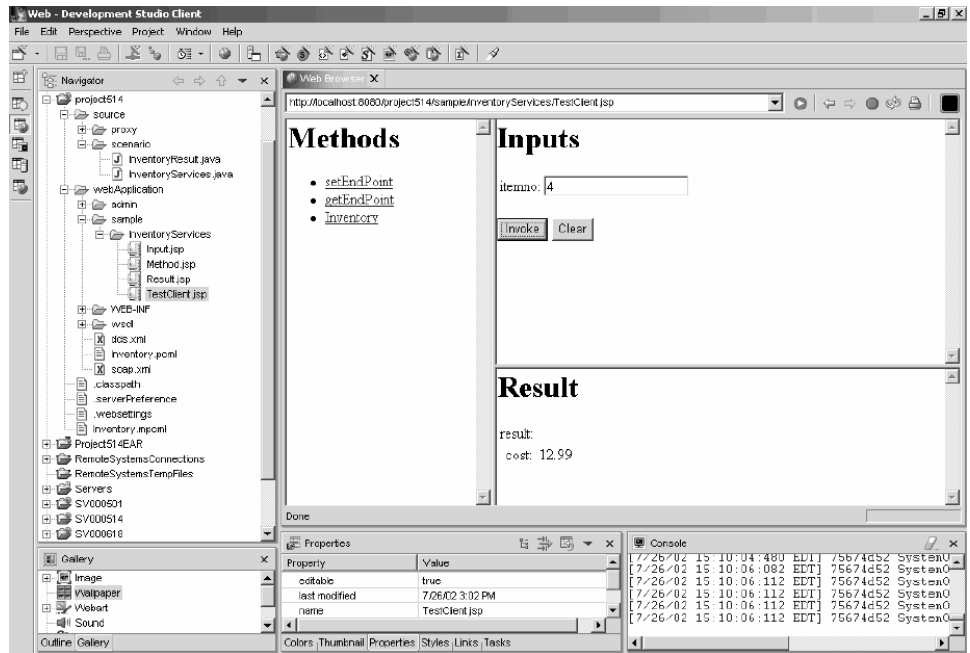
샘플 테스트

웹 서비스 작성 시, 명령어 중 하나는 샘플 생성을 요청하는 것입니다. 해당 옵션이 선택되었으므로, 웹 서비스 마법사는 웹 서비스를 테스트하는 데 사용할 수 있는 테스트 페이지를 사용하여 새로운 프로젝트를 작성했습니다. 네비게이터 보기에서, 새로운 프로젝트 이름은 **Project514Client**입니다. 또한 도구에서 자동으로 Workbench 오른쪽에 **TestClient.jsp**를 열었습니다. 이 파일은 **Project514Client > Web Content > sample > InventoryServices** 아래에 위치합니다.

샘플을 테스트하려면 다음을 수행하십시오.

1. 로드된 페이지에서 맨 아래 위치할 때까지 메소드 분할창을 아래로 화면이동하십시오. **Inventory (java.math.BigDecimal)** 메소드를 클릭하십시오.
2. **itemno** 필드에서 4를 입력한 다음 호출을 클릭하십시오.
3. 리턴되는 출력이 다음과 같은지 확인하십시오.

```
result:
cost: 12.99
```

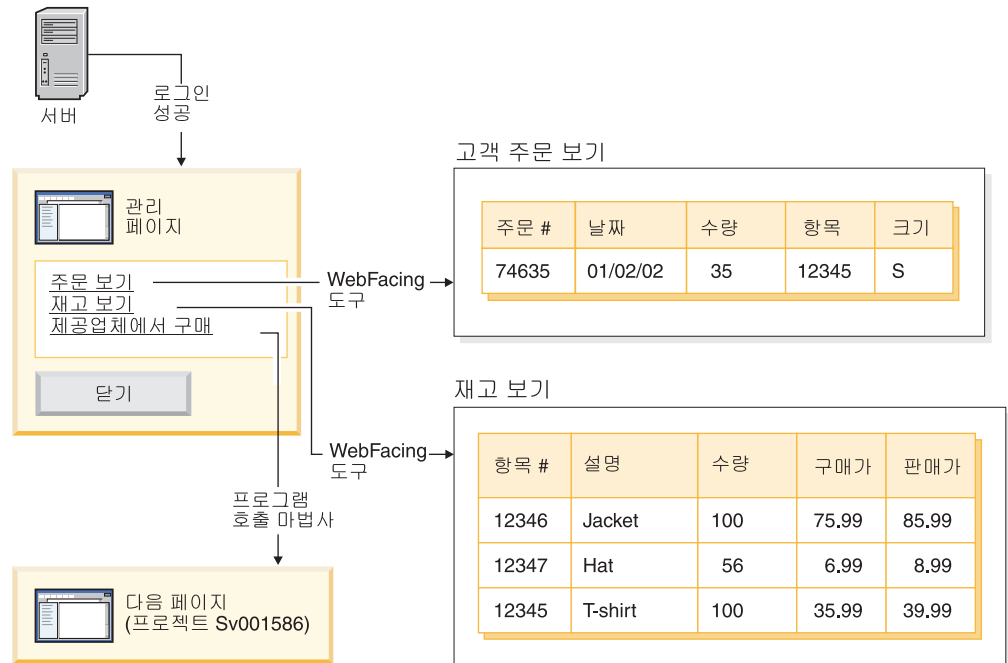


일단 결과를 확인하면, 모듈은 완료된 것입니다. 어플리케이션 인터페이스에 제품 가격을 리턴하는 웹 서비스가 작성되었습니다.

제 5 장 단계별 모듈 2: 재고를 열람하고 항목을 주문하기 위한 인터페이스 작성(SV000501)

소개

이 모듈에서는 IBM WebFacing 도구를 사용하여 관리자의 조작 옵션을 표시하는 인터페이스를 작성합니다. iSeries 서버에 성공적으로 로그인한 후, 인터페이스는 사용 가능한 재고를 표시하며 사용자는 기존 주문을 점검할 수 있습니다. 그런 다음 관리자로서, 도매 공급업체로부터 상품을 구입할 수 있습니다.



이 프로젝트에서는 iSeries 서버로 복원되는 두 개의 프로그램과 두 개의 화면 파일에 대해 작업합니다. 프로그램 이름은 ViewInventory 및 ViewOrder입니다. 프로그램은 ORDERDSP 및 QUERY와 같은 두 개의 화면 파일을 사용합니다. 이들 두 파일은 WebFacing 도구에서 생성한 JSP 파일에 대해 사용자 정의된 웹 설정을 포함하며, 이미지 및 하이퍼링크에 사용됩니다. 이미지 웹 설정은 필드 콘텐츠를 사용하여 이미지 파일 이름을 생성하고 JSP 파일에 이미지를 표시할 수 있게 해줍니다. 하이퍼링크 웹 설정은 JSP 파일에서 이미지를 클릭하여 또다른 웹 어플리케이션을 호출할 수 있게 해줍니다. CODE 편집기나 CODE 설계자를 사용하여 화면 파일 소스 코드를 점검함으로써 웹 설정 지정 방법을 판별할 수 있습니다.

시작하기 전에

다음 전제조건이 일치할 경우에만 연습문제를 완료할 수 있습니다. 전제조건은 9 페이지의 제 3 장 『시나리오 실행』에서 보다 자세히 설명됩니다.

- iSeries 서버에 대한 TCP/IP 액세스가 있음.
- STRTCPSVR *ALL 명령을 사용하여 iSeries 서버를 시작했음
- STRTCPSVR *WEBFACING 명령을 사용하여 WebFacing 서버를 시작했음
- 9 페이지의 제 3 장 『시나리오 실행』의 모든 작업을 완료했음(Workbench에서 어플리케이션을 테스트하는 데 필요하지 않은 선택적 WebSphere Application Server 작업 제외).

WebFacing 프로젝트 작성

첫 번째 수행해야 할 작업은 WebFacing 프로젝트를 작성하고 관련 CL 명령을 지정하는 것입니다. WebFacing 프로젝트를 작성하려면 다음을 수행하십시오.

1. Workbench에서 작업공간 왼쪽에 있는 Perspective 아이콘 중 하나를 클릭하거나 창 > Perspective 열기 > 기타 > WebFacing을 클릭한 다음 확인을 클릭하여 WebFacing Perspective로 전환하십시오.
2. 파일 > 신규 > WebFacing 프로젝트를 클릭하여 새로운 WebFacing 프로젝트를 작성하십시오.
3. Project501 프로젝트에 이름을 지정하십시오.
4. 엔터프라이즈 어플리케이션 프로젝트 옆의 기존 라디오 단추를 클릭하십시오.
5. 기존 프로젝트 이름 필드 값에 SVStoreEAR을 입력하십시오. 값은 대소문자를 구분하지 않습니다. 다음을 클릭하십시오.
6. J2EE 레벨 드롭다운 목록에 대해 1.3을 선택하십시오. 다음을 클릭하십시오.
7. 연결 필드에 iSeries 서버의 이름이 자동으로 입력되어야 합니다. 그렇지 않을 경우, 신규를 클릭하고 대화 상자에 iSeries 서버의 이름을 입력한 후 완료를 클릭하십시오.
8. DDS 목록 새로 고침을 클릭한 다음, 팝업 대화 상자에 암호와 사용자 ID를 입력하여 목록을 새로 고치십시오.
9. 생성된 라이브러리 목록에서 RETAILSTOR를 펼치십시오.
10. QDDSSRC를 클릭한 다음, 오른쪽 화살표(>>)를 클릭하여 파일을 이동하십시오.
11. CL 명령 지정 페이지에 도달할 때까지 다음을 두 번 클릭하십시오.
이제 관리자에 대한 명령 및 명령 레이블을 추가해야 합니다.
12. CL 명령 필드에서 call viewinvent 호출을 입력하십시오.
13. 명령 레이블 필드에서 기존 값을 삭제한 후 View inventory를 입력하십시오.
14. 지정된 값으로 사인 온 라디오 단추를 선택하십시오.

15. 추가를 클릭한 후 창 맨 아래 있는 목록에 추가된 항목을 주의하십시오. 이제 고객에 대한 두 번째 명령 및 명령 레이블을 추가해야 합니다.
16. **CL** 명령 필드에서 기존 값을 삭제한 후 call vieworder를 입력하십시오.
17. 명령 레이블 필드에서 기존 값을 삭제한 후 View orders를 입력하십시오.
18. 사인 온 프롬프트 라디오 단추를 선택하십시오.
19. 추가를 클릭한 후 창 맨 아래 있는 목록에 추가된 항목을 주의하십시오. 다음을 클릭하십시오.
20. 웹 스타일 선택 창에서 사용 가능한 스타일을 보려면 스타일을 화면이동하십시오. 앞으로도 해당 스타일에서 정보를 검색해야 하므로, 이 모듈에 대해 **avenue**를 선택하십시오. 다음을 클릭하십시오.
21. **아니오**. 지금 프로젝트만 작성하겠습니다. 선택란이 선택되었는지 확인한 다음 완료 버튼을 클릭하십시오.
22. 서버 구성 복구 메시지가 표시되면, 확인을 클릭하십시오. 이 메시지는 단지 SVStoreEAR 파일에 추가하려는 프로젝트를 확인하기 위한 것입니다.

Project501을 완전한 SV000501 프로젝트의 미러로 구성 중이므로, Project501이 제대로 표시되게 하려면 SV000501의 일부 이미지 파일을 Project501로 지금 복사해야 합니다. (그러나 원하는 경우, 인터페이스에 대한 자체 이미지를 추가할 수 있습니다.) 이미지를 복사하려면 다음을 수행하십시오.

1. 웹 Perspective로 전환하십시오.
2. 네비게이터 보기에서 **SV000501 > Web Content > images**를 펼치십시오.
3. (선택적)프로젝트로 가져오려는 모든 이미지를 확인하려면 **generated**를 펼치십시오.
4. **generated**를 마우스 오른쪽 단추로 클릭한 후 복사를 선택하십시오.
5. **Project501**이 다시 표시될 때까지 네비게이터 보기에서 위로 화면이동하십시오.
6. **Project501 > Web Content**를 펼치십시오.
7. **images** 폴더를 마우스 오른쪽 단추로 클릭한 다음 팝업 메뉴에서 붙여넣기를 선택하십시오.
8. 이제 Project501에 대한 **generated** 폴더가 생성됩니다. 추가한 이미지를 확인하려면 **generated**를 펼치십시오.

DDS 소스 변환

이제 프로젝트를 작성했으므로, DDS 화면 파일을 웹 페이지에 대한 JSP 파일로 변환할 수 있습니다. DDS 화면 파일을 변환할 경우, WebFacing 도구는 DDS 코드를 대체하며 웹 액세스를 가능하게 하는 JSP 및 XML 파일을 생성합니다. 생성된 파일은 레코드 형식에 대한 데이터를 보류하거나, 모양을 제어하거나, 화면의 웹 버전을 표시하거나, 데이터를 프롬프트하거나 또는 입력 오류를 처리합니다. 마법사는 또한 프로그램의 웹 작동 버전을 실행하기 위한 어플리케이션 홈 페이지를 생성합니다.

그러나 먼저, 올바른 iSeries 서버를 참조하도록 소스를 변경해야 합니다. 특히 어플리케이션에 대한 링크가 작동할 수 있도록 DDS 화면 파일 중 하나를 수정해야 합니다. 기본 이름 SV000501을 프로젝트 이름, Project501로 변경해야 합니다.

1. WebFacing Perspective로 되돌아가서 기본적으로 이미 열려있지 않은 경우 WebFacing 프로젝트 보기로 전환하십시오.
2. **Project501 > DDS**를 펼치십시오.
3. 두 번째 항목, <iSeriesserver>**RETAILSTOR/QDDSSRC(QUERY)**를 두 번 클릭하십시오.
4. 16행 정도로 화면이동하십시오. 다음과 같은 행이 표시됩니다.
`A*%%WB 12 FLD 1 next ('/SV000501/DetailPage.do?PRODNO=&{PRODNO}')`
5. SV000501을 삭제한 다음 Project501을 입력하십시오.
6. 파일을 저장한 후 닫으십시오.

주: 때때로 SV000501 프로젝트를 다시 실행하고자 할 경우, 해당 파일이 iSeries 서버에 존재하며 두 프로젝트 모두에 의해 액세스되므로 Project501 값을 다시 SV000501로 변경해야 합니다.

DDS 소스를 변환하려면 다음을 수행하십시오.

1. WebFacing Perspective에서 기본적으로 이미 열려있지 않은 경우 **WebFacing** 프로젝트 탭을 클릭하여 WebFacing 프로젝트 보기로 전환하십시오.
2. **Project501 > DDS**를 펼치십시오.
3. *iSeriesserver* > **RETAILSTOR/QDDSSRC(ORDERDSP)** 및 *iSeriesserver* > **RETAILSTOR/QDDSSRC(QUERY)**를 선택하십시오. 첫 번째 항목을 클릭한 다음, Shift 키를 누른 상태에서 두 번째 항목을 클릭하면 선택이 가능합니다.
4. 변환을 시작하려면 마우스 오른쪽 단추로 클릭한 다음 **변환**을 선택하십시오. 때때로 iSeries 서버와의 연결이 끊어진 경우 사용자 ID와 암호를 입력해야 할 수도 있습니다.

Workbench에서 UTF-8 지원 구성 — WAS 버전 4.0 사용자 전용

IBM WebFacing 도구 어플리케이션은 화면에서 복수 언어 표시를 지원합니다. 언어가 서로 다른 문자 세트를 지원하므로 브라우저 및 WebSphere Application Server 간의 데이터 스트림은 UTF-8로 인코딩됩니다. IBM WebFacing 도구가 제대로 작동하려면 Workbench에서 어플리케이션의 특성 파일에 UTF-8 지원을 구성해야 합니다.

주: 이 섹션은 WAS 4.0 사용자에게만 적용됩니다. WAS 버전 5.0은 이 작업을 자동으로 수행합니다.

UTF-8 지원을 구성하려면 다음을 수행하십시오.

1. 웹 Perspective로 전환하십시오. (화면 왼쪽에 클러스터된 아이콘을 클릭하여 Perspective를 전환할 수 있습니다.)
2. 프로젝트 구조를 볼 수 있도록 프로젝트 네비게이터 탭을 클릭하십시오.
3. **Project501**을 펼친 다음 웹 전개 설명자를 두 번 클릭하십시오.
4. 환경 탭을 클릭하십시오.
5. 추가 단추를 클릭하십시오.
6. 상위 왼쪽 작업 표시줄에 있는 아이콘을 클릭하여 서버 Perspective로 전환하거나, 메뉴 표시줄에서 **Perspective > 열기 > 기타 > 서버**를 클릭한 다음 확인을 클릭하십시오.
7. 네비게이터 보기에서 **Servers** 폴더를 펼치십시오.
8. **defaultInstance.wsi**를 두 번 클릭하여 기본 편집기에서 여십시오.
9. 환경 탭을 클릭한 다음 추가 단추를 클릭하십시오.
10. 편집 가능한 값은 "(New Variable)"로 표시됩니다. 이 기본 문자열을 삭제한 다음 `client.encoding.override`를 입력하십시오.
11. 값 필드에서 UTF-8을 입력하십시오.
12. 저장 아이콘을 클릭하거나 파일 > 웹 전개 설명자 저장을 클릭하십시오.

WebSphere Application Server용 UTF-8 지원 구성

(선택적)iSeries 어플리케이션을 WebSphere Application Server로 전개하려는 경우, WebSphere Application Server 및 Workbench에서 UTF-8 지원을 구성해야 합니다.

WebSphere Application Server 4.0 Advanced Edition에서 UTF-8 지원을 구성하려면

1. WebSphere 관리 콘솔을 시작하십시오.
2. 노드 아이콘을 펼친 다음 노드 이름 > 어플리케이션 서버 > **Default server**를 펼치십시오.
3. **JVM** 설정 탭을 선택한 다음 고급 **JVM** 설정 단추를 클릭하여 고급 **JVM** 설정 대화 상자를 여십시오.
4. 명령행 인수 필드에서 다음을 입력하십시오.
-Dclient.encoding.override=UTF-8
5. 확인을 클릭한 다음 **JVM** 설정 탭 아래 적용을 클릭하십시오.
6. WebSphere 어플리케이션에 대해 해당 변경사항이 유효하게 하려면 기본 서버를 중단한 다음 다시 시작하십시오. 서버를 중단하려면 **Default server**를 마우스 오른쪽 단추로 클릭한 후 중단을 선택하십시오. 이 프로세스가 완료된 후, **Default server**를 마우스 오른쪽 단추로 클릭한 다음 시작을 선택하십시오.

WebSphere Application Server 4.0 Advanced Single Server Edition에서 UTF-8 지원을 구성하려면

1. WebSphere 관리 콘솔을 시작하십시오.
2. 브라우저 기반 관리 콘솔에서 노드 아이콘을 펼친 다음 노드 이름 > 어플리케이션 서버 > **Default server** > 프로세스 정의 > **JVM** 설정을 펼치십시오.
3. JVM 설정 페이지의 고급 설정 섹션으로 화면이동한 다음 시스템 특성 링크를 클릭하십시오. 시스템 특성 페이지가 표시됩니다.
4. 새로운 시스템 특성을 추가하려면 신규를 클릭하십시오.
5. 이름 필드에서 `client.encoding.override`를 입력하십시오.
6. 값 필드에서 UTF-8을 입력하십시오.
7. 확인을 클릭하십시오. **JVM** 설정 페이지 상단의 링크와 함께 구성을 저장해야 함 메시지가 수신되면 링크를 클릭하여 구성 저장 페이지로 이동하십시오. 저장을 선택한 다음 확인을 클릭하십시오.
8. WebSphere 어플리케이션에 대해 해당 변경사항을 유효하게 하려면 어플리케이션 서버를 중단한 다음 다시 시작하십시오. 어플리케이션 서버를 중단하고 시작하는 방법은 WebSphere Application Server를 설치한 플랫폼에 따라 달라질 수 있습니다. 어플리케이션 서버 중단 및 시작에 관한 정보는 플랫폼에 대한 WebSphere Application Server 문서를 참조하십시오.

스타일 시트 작성

추가 페이지를 CSS(캐스케이딩 스타일 시트)와 통합하려는 경우, WebFacing 프로젝트의 스타일 또는 캐스케이딩 스타일 시트를 사용자 정의하여 유사하게 보이게 만들어야 합니다. 스타일 시트 사용자 정의를 완료한 후, 웹 상호작용 마법사에서 가격 및 색상과 같은, 상점 항목에 관한 정보를 표시하는 상세 웹 페이지를 스타일 시트를 사용하여 작성할 수 있습니다. 모듈의 목적을 위해 SV000501 프로젝트에서 `DetailPageResults.jsp` 스타일 시트를 통합합니다. 그러나 나중에 참조할 수 있도록 CSS 파일을 수동으로 사용자 정의할 수 있습니다(다음 섹션에서 요약됨).

수동으로 캐스케이딩 스타일 시트의 사용자 정의(선택적)

앞에서 언급한 대로, Project501을 위해 SV000501 프로젝트에서 `DetailPageResults.jsp` 스타일 시트를 통합하는 대신 자체 스타일 시트를 사용자 정의할 수 있습니다.

1. 프로젝트 네비게이터 보기로 전환한 다음, **Project501** > **Web Content** > **styles** > **apparea**를 펼치십시오.
2. **apparea.css**를 두 번 클릭하여 CSS 설계자에서 여십시오. 이 스타일 시트는 어플리케이션 영역의 모양을 제어합니다.
3. 파일을 수정한 후 저장하십시오.

4. 프로젝트 네비게이터 보기로 되돌아가서 **Project501 > Web Content > styles > chrome**으로 이동하십시오.
5. **avenue.css**를 두 번 클릭하여 CSS 설계자에서 여십시오. 이 스타일 시트는 전체 페이지의 모양을 제어합니다.
6. 파일을 수정한 후 저장하십시오.

웹 상호작용을 사용한 WebFacing 프로젝트 확장 및 향상

이 섹션에서는 WebFacing 프로젝트를 향상시킵니다. 항목의 목록이 표시될 경우, 항목 이미지를 클릭하여 해당 항목에 관한 세부사항을 찾고자 할 수도 있습니다. 이를 수행하려면 웹 상호작용 마법사를 사용하여 항목의 세부사항을 검색하고 또다른 웹 페이지에 이를 표시하는 RPG 프로그램을 호출합니다. 다음과 같은 단계를 수행하십시오.

- 서버 정보 정의
- 올바른 스타일 시트 복사
- 상호작용 작성
- 상호작용에 프로그램 및 매개변수 추가
- 매개변수 용도 변경

먼저, 다음 서버 정보를 정의해야 합니다.

1. 웹 Perspective로 전환하십시오.
2. 네비게이터 보기에서 **Project501**을 마우스 오른쪽 단추로 클릭한 후 **iSeries 웹 도구 런타임 구성 지정**을 선택하십시오.
3. iSeries 서버 이름, 사용자 ID 및 암호를 입력하십시오.
4. 라이브러리 필드에 **Retailstor**을 입력한 다음 **추가**를 클릭하십시오.
5. **완료**를 클릭하십시오. 필요한 경우 **완료**를 다시 클릭하십시오.

웹 상호작용을 작성하기 전에 먼저 Project501이 적절한 JSP 파일 형식을 표시할 수 있도록 올바른 스타일 및 스타일 시트를 복사해야 합니다. (40 페이지의 『수동으로 캐스캐이딩 스타일 시트의 사용자 정의(선택적)』를 완료한 경우 이 작업을 수행할 필요가 없습니다.) 먼저 스타일 폴더를 복사한 다음 **DetailPageResults.jsp**를 복사하십시오.

1. 네비게이터 보기에서 **SV000501 > Web Content**를 펼치십시오.
2. 스타일을 마우스 오른쪽 단추로 클릭한 다음 복사를 선택하십시오.
3. **Project501**이 다시 표시될 때까지 네비게이터 보기에서 위로 화면이동하십시오.
4. 프로젝트를 펼치고 **Web Content**를 마우스 오른쪽 단추로 클릭한 다음 팝업 메뉴에서 **붙여넣기**를 선택하십시오. 기존 스타일을 겹쳐쓰려면 **예**를 클릭하십시오.
5. **SV000501 > Web Content**를 향하여 다시 아래로 화면이동하십시오.
6. **DetailPageResults.jsp**를 마우스 오른쪽 단추로 클릭한 다음 복사를 선택하십시오.

7. **Project501**이 다시 표시될 때까지 다시 위로 화면이동하십시오.
8. 프로젝트를 펼치고 **Web Content**를 마우스 오른쪽 단추로 클릭한 다음 팝업 메뉴에서 **붙여넣기**를 선택하십시오.

새로운 요소가 Project501의 Web Content 폴더 아래에 추가되었습니다.

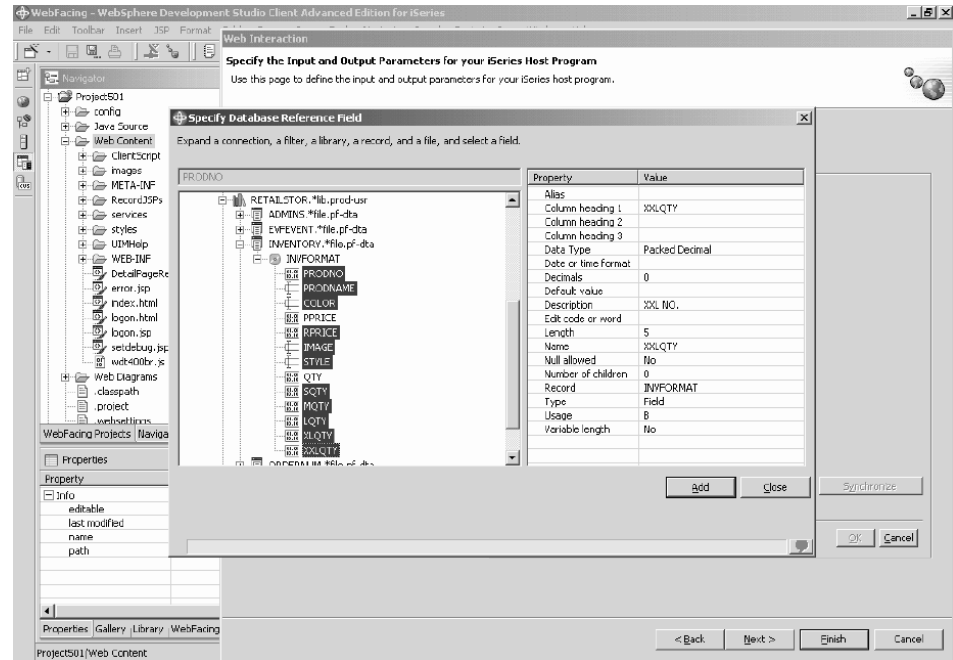
이제 다음과 같이 웹 상호작용을 작성할 수 있습니다.

1. 파일 > 신규 > 기타를 클릭하십시오.
2. 신규 대화 상자에서 왼쪽에 있는 웹을 클릭한 다음, 오른쪽에 있는 웹 상호작용을 클릭하십시오. 다음을 클릭하십시오.
3. 웹 상호작용 이름 필드에서 DetailPage를 입력한 후 다음을 클릭하십시오.
4. 입력 JSP 생성 라디오 단추를 선택하십시오.
5. (아직 선택하지 않은 경우)출력 페이지 사용 라디오 단추를 선택한 다음 추가를 클릭하십시오.
6. 출력 JSP 대화 상자에서 **Web Content**를 펼치고 사전 형식화된 출력 페이지인 **DetailPageResults.jsp**를 선택한 다음 확인을 클릭하십시오.
7. 다음을 클릭하십시오.

이제 프로그램과 매개변수를 상호작용에 추가할 수 있습니다. 11개의 매개변수를 동일한 프로그램에 추가해야 합니다. 개별 값을 사용하여 수동으로 각각의 매개변수를 추가하는 대신, 다음과 같은 방법을 사용하여 좀 더 빠르게 추가할 수 있습니다.

1. 아직 선택하지 않은 경우 **iSeries ILE 프로그램 사용**을 선택하십시오.
2. 프로그램 추가를 클릭하십시오.
3. 프로그램 별명 필드에서 DetailPage를 입력하십시오.
4. 프로그램 오브젝트 필드의 경우, 찾아보기를 클릭하십시오.
 - a. *iSeries*server > *LIBL > RETAILSTOR을 펼치십시오.
 - b. **DETAILPAGE.*pgm.rpgle**(RETAILSTOR 아래 첫 번째 파일)를 클릭한 다음 확인을 클릭하십시오.
5. 웹 상호작용 마법사에서 확인(오른쪽 아래)을 다시 클릭하십시오. DetailPage가 프로그램 호출 정의 아래, 마법사 왼쪽에 추가됩니다.
6. 마법사의 왼쪽 분할창에 있는 프로그램 호출 정의 섹션에서 **DetailPage**를 클릭하여 선택하십시오.
7. 매개변수 추가를 클릭하십시오.
8. 마법사 맨 아래 데이터베이스 참조 필드 지정 옆에 있는 지정을 클릭하십시오.
9. *iSeries*server > *LIBL > RETAILSTOR > **INVENTORY.*file.pf-dta > INVFORMAT**를 펼쳐서 13개의 매개변수 목록을 표시하십시오. 11개의 매개변수(PRODNO, PRODNAME, COLOR, RPRICE, IMAGE, STYLE, SQTY, MQTY, LQTY, XLQTY, XXLQTY) 각각을 클릭한 후 추가를 클릭하거나, CTRL

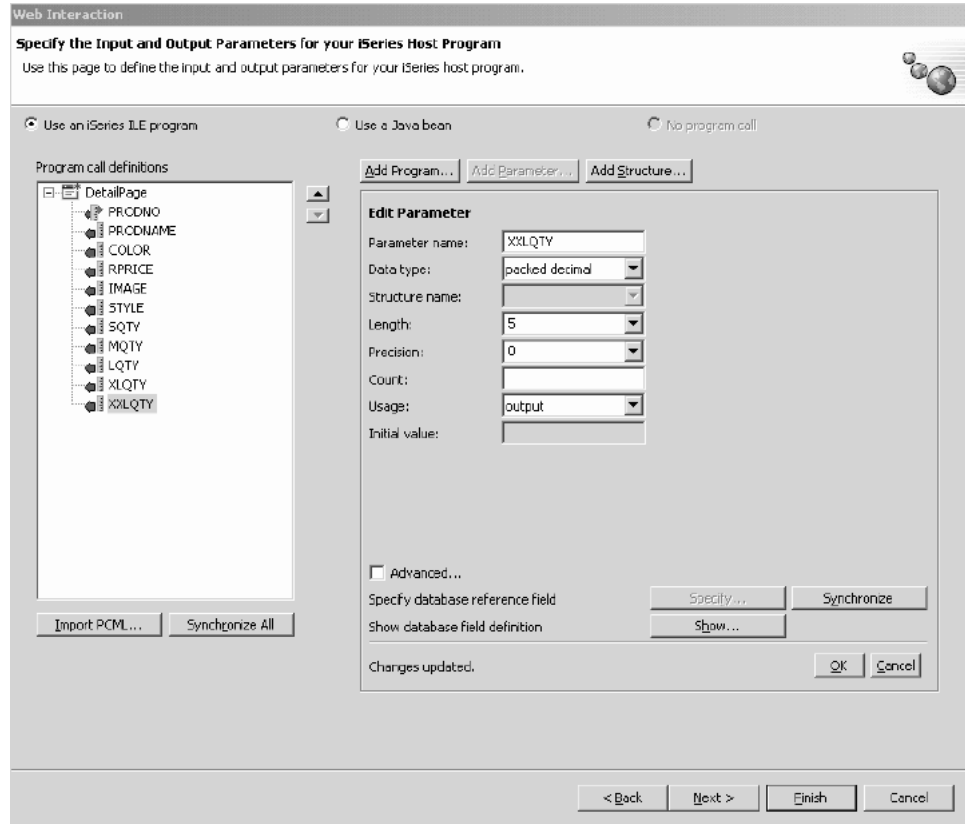
키를 누른 상태에서 각각을 클릭한 다음 추가를 클릭하여 추가해야 합니다. 기본적으로, PPRICE 및 QTY를 제외한 모든 필드를 선택합니다.



10. 닫기를 클릭하십시오.

이제 11개의 매개변수 중 10개에 대한 용도를 "출력"으로 변경해야 합니다(PRODNO를 제외한 모든 매개변수).

1. 계속해서 웹 상호작용 마법사에서 **PRODNAME**을 클릭하여 선택하십시오.
2. 용도 콤보 상자에서 선택된 값을 출력으로 전환한 다음 확인을 클릭하십시오. 첫 번째(PRODNO)를 제외한 모든 매개변수에 대해 이 단계(이전 단계와 함께)를 반복하십시오. 다음은 모든 매개변수를 조정할 경우 표시되는 작업공간입니다. 매개변수 왼쪽에 있는 아이콘은 유형이 입력, 입력 및 출력 또는 출력인지를 표시합니다.



3. 입력 양식 미리보기를 표시하려면 다음을 클릭하십시오.
4. 웹 상호작용을 작성하려면 완료를 클릭하십시오.
5. 메시지를 수신할 경우 모두 예 또는 확인을 선택한 다음 완료를 다시 선택하십시오.

웹 상호작용에 대한 프로젝트 링크

이제 입력 및 출력 매개변수를 사용하는 웹 상호작용 JSP 파일을 작성했습니다. WebFacing 구성요소에 대해서도 작동할 수 있도록 JSP 파일을 사용자 정의해야 합니다. 이러한 웹 상호작용을 호출하려면 WebFacing 어플리케이션에서 링크를 작성하는 코드를 입력해야 합니다. 이를 수행하려면 새로운 창에서 PRODNO 매개변수를 사용하여 DetailPageServlet servlet을 호출할 수 있도록 webface.js 파일에서 Javascript 기능을 추가해야 합니다.

링크를 작성하려면 다음을 수행하십시오.

1. 네비게이터 보기에서 **Project501 > Web Content > ClientScript**를 펼치십시오.
2. **webface.js**를 두 번 클릭하여 편집기에서 여십시오.
3. 파일 맨 아래로 화면이동한 후 다음 행을 입력하십시오.

```

var mywindow
function next(app)
{
mywindow = window.open(app,"Details","RESIZABLE=YES, HEIGHT=700, WIDTH=800");
}

```

4. 저장 아이콘을 클릭하거나 파일 > **webface.js** 저장을 클릭하십시오.

나중에 SV000501과 유사한 어플리케이션을 작성하려는 경우, DDS 소스에서 웹 설정을 변경하여 추가한 이미지를 작동하게 한 다음 JavaScript 기능에 대해 창 닫기 링크도 작동하게 해야 합니다.

이 어플리케이션에 포함된 RPG 코드는 변경사항을 표시하도록 변경되지만 나중에 어플리케이션을 수동으로 변경해야 합니다. 또한 웹 설정을 변경한 후 DDS 소스를 재변환해야 합니다.

결과를 복제하기 위한 코드를 검사할 수 있습니다.

DDS 소스를 보려면 다음을 수행하십시오.

1. WebFacing Perspective로 전환하십시오.
2. WebFacing 프로젝트 보기에서 **Project501 > DDS**를 펼치십시오.
3. **<iSeriesserver > RETAILSTOR/QDDSSRC(QUERY)**를 마우스 오른쪽 단추로 클릭한 후 열기 > **CODE** 설계자를 선택하십시오.
4. CODE 설계자가 열리면 **SCREEN1 > ITEMSUB**를 펼치십시오.
5. **IMAGESRC**를 클릭하여 선택하십시오.
6. 소스 탭을 클릭하십시오.
7. 창의 맨 아래 오른쪽에 있는 웹 설정 탭을 클릭하십시오.

주: 너비(픽셀 단위) 및 파일 이름과 같은 웹 설정 특성을 검사하십시오. 나중에 DDS 소스를 동일하게 변경한 후 소스를 재변환해야 합니다.

8. 소스의 다음 행을 주의하십시오.

```

A   PRODNO  R   0 5 6
A   PRODNAME R   0 5 16
A   IMAGESRC 19A 0 5 33
A*%%WB 13 FLD 100|100|&{IMAGESRC}
A*%%WB 12 FLD 1 javascript:next
      ('/Project501/DetailPageServlet?PRODNO=&{PRODNO}')

```

Project501이 마지막 행에 지정되어 있습니다. 앞에서 언급한 대로 SV000501 어플리케이션을 다시 실행하려는 경우, Project501 값을 다시 SV000501로 변경해야 합니다.

9. 저장 아이콘을 클릭하거나 메뉴 표시줄에서 파일 > 저장을 클릭하여 파일을 저장하십시오.
10. 파일을 닫은 다음 CODE 설계자를 닫으십시오.

DDS 소스에 대해 작업하는 방법에 관한 추가 정보는 Workbench의 도움말 Perspective 로 전환한 다음 IBM WebFacing 도구 문서를 참조하십시오.

파일 공개 및 서버 다시 시작

이 섹션에서 Project501 어플리케이션에 모든 변경사항이 적용될 수 있도록 파일을 공개한 후에 재시작합니다.

그러나 서버를 재시작하기 전에 먼저 Project501.war이 SVStoreEAR 파일에 추가되었는지 확인해야 합니다.

구성을 확인하려면 다음을 수행하십시오.

1. 네비게이터 보기에서 **Scenario** 폴더를 펼치십시오.
2. **Scenario server.wsi**를 두 번 클릭하십시오.

편집기가 프로젝트 항목을 자동으로 수정하게 할 것인지를 묻는 대화 상자가 표시될 수도 있습니다. 예를 클릭하십시오. 대화 상자가 표시되지 않을 경우에는 조치를 수행할 필요가 없습니다. 두 가지 방법 모두, 파일을 열면 자동으로 수정됩니다.

3. 파일을 저장한 후 닫으십시오.

서버를 재시작하려면 다음을 수행하십시오.

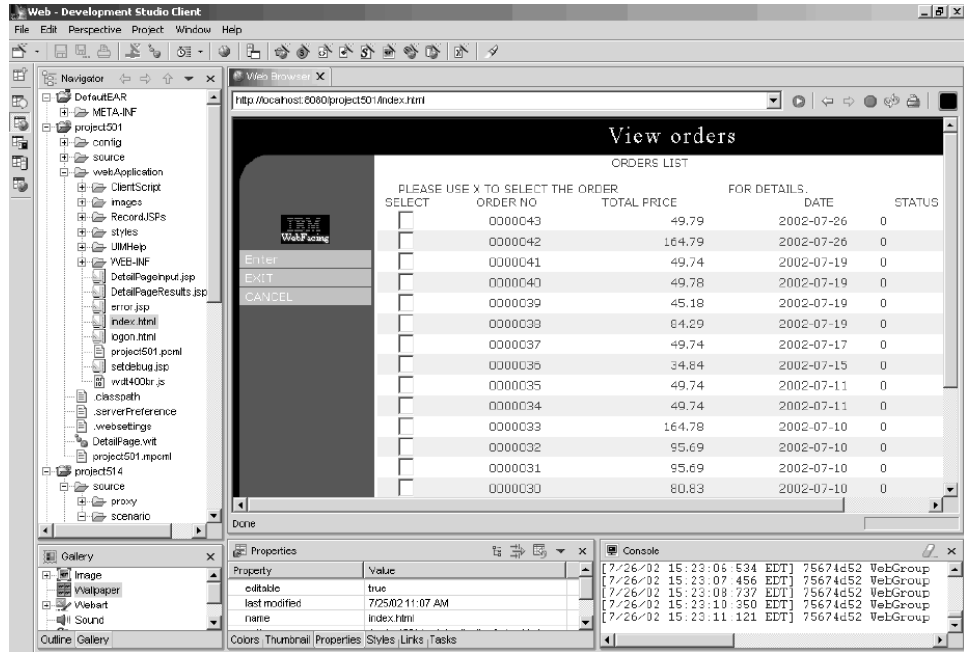
1. 서버 Perspective로 전환하십시오.
2. 서버 구성 보기에서 맨 아래 왼쪽에 있는 서버를 펼친 다음 **Scenario** 서버를 두 번 클릭하십시오.
3. 서버 보기에서 서버의 상태를 보려면 화면 맨 아래 오른쪽에 있는 서버 탭을 클릭하십시오.
4. **Scenario** 서버를 마우스 오른쪽 단추로 클릭한 후 공개를 선택하십시오. 공개가 완료되면 확인을 클릭하십시오.
5. 서버 보기에서 서버를 마우스 오른쪽 단추로 클릭한 후 시작 또는 재시작(둘 중 작동된 것)을 선택하십시오. 저장되지 않은 파일에 관한 프롬프트가 표시되면, 대화 상자를 취소하고 모든 열린 파일을 저장한 다음 다시 처리하십시오.
6. 콘솔 로그 정보(자동으로 열림)를 확인하십시오. 로그 맨 아래 다음과 같은 행이 표시되면 서버가 시작되었음을 알 수 있습니다. `server_name`이(가) e-business를 위해 시작되었습니다.

인터페이스 테스트

이제 주문을 검토하고 재고를 열람하기 위한 인터페이스를 작성하는 데 필요한 단계를 완료했습니다. 인터페이스를 테스트하려면 다음을 수행하십시오.

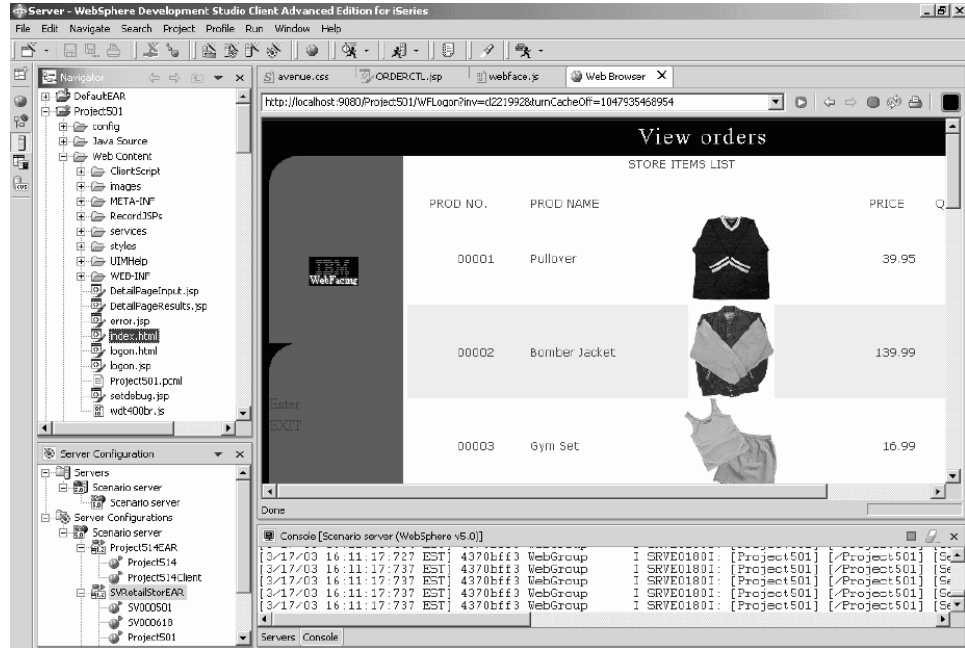
1. 웹 Perspective로 전환하십시오.
2. **Project501 > Web Content**를 펼치십시오.

3. **index.html**을 마우스 오른쪽 단추로 클릭한 다음 서버에서 실행을 선택하십시오.
4. 서버 선택사항 대화 상자가 열립니다. **Scenario** 서버가 기본적으로 선택되었는지 확인한 다음 **완료**를 클릭하십시오.
5. 관리자 어플리케이션을 호출하려면 주문 검토 - 기본 브라우저 창에서 실행을 클릭하십시오. iSeries 사용자 ID와 암호(해당 프로젝트 개발 시 사용한 iSeries 서버에 대한)를 사용하여 로그인하고 나면, 다음과 같은 페이지로 방향 지정됩니다.



해당 제품에 관한 세부사항을 보려면 제품 번호 옆에 X를 입력하십시오.

6. **index.html** 페이지로 돌아가려면 뒤로 화살표를 클릭하고, **재고 보기** - 기본 브라우저에서 실행 창을 클릭하여 고객 어플리케이션을 호출하면 다음과 같은 페이지로 방향 지정됩니다(사용자 ID와 암호를 사용하여 iSeries 서버에 로그인한 후).



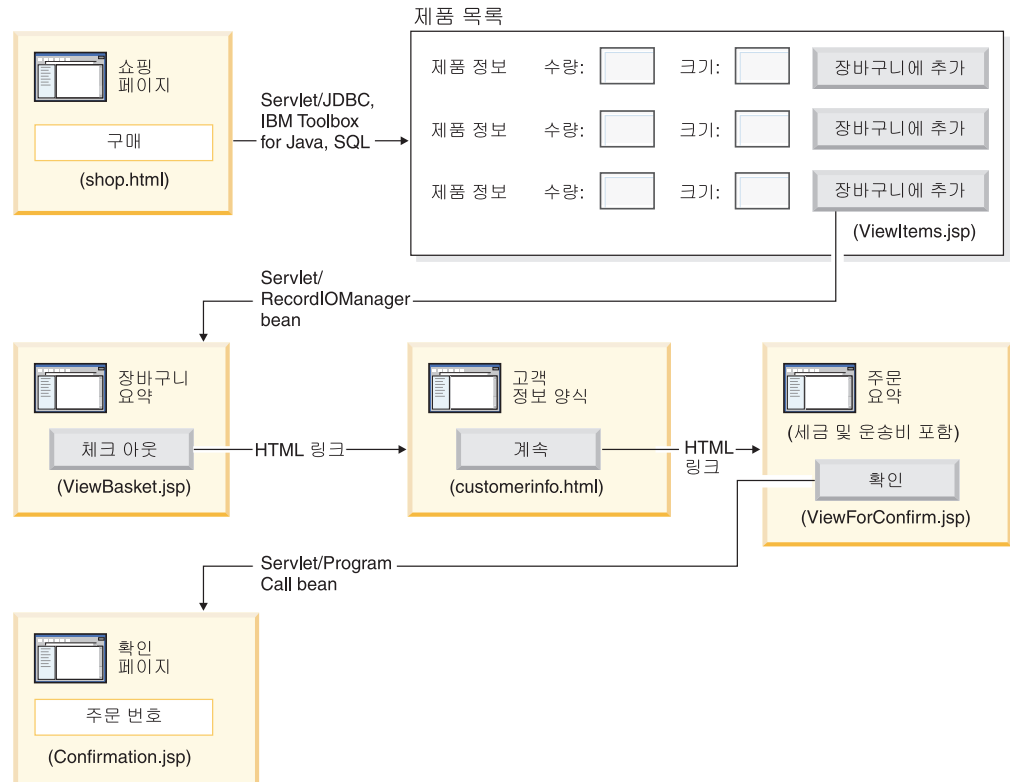
특정 항목에 관한 세부사항을 표시하려면 이미지를 클릭하십시오.

모듈을 완료했습니다. 재고를 열람하고 항목을 주문할 수 있는 사용자 인터페이스가 작성되었습니다.

제 6 장 고급 모듈 1: iSeries 서버에 고객 주문을 배치하는 HTML, servlet 및 JSP 파일 작성(SV001585)

소개

이 프로젝트는 풍부한 Java 프로그래밍 지식과 약간의 iSeries 데이터 관리 및 RPG 지식이 있는 사용자를 위해 설계되었습니다. 프로젝트는 IBM Toolbox for iSeries 데이터 액세스 클래스, RecordIOManager bean 및 프로그램 호출 bean에 대해 작업하여 iSeries 서버에 고객 주문을 배치하는 HTML 코드, servlet 및 JSP를 작성하는 방법을 보여줍니다. 사용자로서, 쇼핑 페이지에서 이동하면서 사용 가능한 제품을 열람하고 항목을 장바구니에 추가합니다. 선택한 모든 항목(장바구니 요약에 있는)에 만족할 경우, 체크 아웃 단추를 클릭하여 고객 정보 양식으로 이동합니다. 양식이 완료되면, 프로젝트는 확인 단추가 있는 주문 요약으로 리턴하며, 확인 단추를 클릭하면 주문 및 주문 번호를 표시하는 확인 페이지로 이동합니다.



상위 레벨 단계 요약

이 모듈은 고급 모듈이므로 명령어가 각각의 프로젝트 작성 단계를 일일이 안내하지는 않지만 프로젝트를 작성하기 위해 수행하는 iSeries 특정 개발 단계를 요약합니다. 상위 레벨 단계는 다음과 같습니다.

1. HTML 쇼핑 웹 페이지를 작성하십시오.
2. 고객이 구매할 수 있는 사용 가능한 항목으로 Java bean(JDBC 및 SQL 사용)을 입력하는 servlet을 작성하십시오.
3. 판매용 항목을 열람하고 고객이 원하는 항목의 수량 및 크기를 입력하며 장바구니에 추가 단추를 클릭하여 항목을 선택할 수 있게 해주는 JSP 파일을 작성하십시오.
4. RecordIOManager bean을 사용하여, 항목에 필요한 수량 및 크기를 빼고 "basket"이라는 Java bean에 해당 선택사항을 추가함으로써 iSeries INVENTORY 데이터베이스를 갱신하는 장바구니에 추가 단추를 클릭하여 호출된 servlet을 작성하십시오. servlet은 조작이 성공할 경우 ViewBasket.jsp로 응답을 재지정합니다. 조작이 실패할 경우, servlet은 오류 페이지를 표시합니다.
5. 고객이 개인용 정보를 입력할 HTML 양식을 작성하십시오.
6. 장바구니 내용, 세금과 운송 및 처리 비용을 표시하는 JSP 파일 구매 확인 페이지를 작성하십시오. 또한 페이지는 고객에 대한 확인 단추를 포함해야 합니다.
7. 고객이 확인 단추를 클릭할 때 호출되는 servlet을 작성하십시오. servlet은 iSeries 프로그램 호출 마법사를 사용하여 작성된 Java bean을 사용합니다. Java bean의 메소드 중 하나는 RPG 프로그램을 호출하여 고객의 장바구니 내용에 해당하는 새로운 주문을 iSeries 서버의 ORDERS 데이터베이스에 작성합니다. 그런 다음 servlet은 주문 번호를 리턴하며 웹 어플리케이션 세션에 Java bean을 배치한 다음 주문 번호를 포함하는 주문 확인 JSP 파일을 로드합니다.

시작하기 전에

다음 전제조건이 일치할 경우에만 연습문제를 완료할 수 있습니다. 전제조건은 9 페이지의 제 3 장 『시나리오 실행』에서 보다 자세히 설명됩니다.

- iSeries 서버에 대한 TCP/IP 액세스가 있음.
- STRTCPSVR *ALL 명령을 사용하여 iSeries 서버를 시작했음
- STRTCPSVR *WEBFACING 명령을 사용하여 WebFacing 서버를 시작했음
- WHOLESALE 및 RETAILSTOR 라이브러리를 iSeries 서버로 복원했음
- WHOLESALE, RETAILSTOR 및 QGPL 라이브러리가 라이브러리 목록에 있음
- 9 페이지의 제 3 장 『시나리오 실행』의 모든 타스크를 완료했음(Workbench에서 어플리케이션을 테스트하는 데 필요하지 않은 선택적 WebSphere Application Server 타스크 제외).

웹 페이지, servlet 및 JSP 파일 작성

SV001585 프로젝트의 구성요소를 구성하려면 다음을 수행하십시오.

1. 웹 프로젝트를 작성하십시오.
2. GetItems servlet을 호출하는 링크를 포함하는 Page Designer를 사용하여 shop.html 페이지를 작성하십시오.
3. iSeries Toolbox for Java 클래스용 jt400.jar 파일을 웹 프로젝트 lib 폴더로 가져오십시오. 이 jar 파일은 x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime에서 찾을 수 있습니다. 여기서 x는 Development Studio Client를 설치한 디렉토리입니다.

주: servlet 및 JSP에 대한 기본 iSeries Toolbox for Java JDBC 및 SQL 관련 부분을 보려면 SV001585 프로젝트의 GetItems.java 및 ViewItems.jsp를 참조하십시오. 웹 Perspective의 네비게이터 보기에서 **SV001585 > 소스**를 펼쳐서 GetItems.java를 찾을 수 있으며, **SV001585 > 웹 어플리케이션**을 펼쳐서 ViewItems.jsp를 찾을 수 있습니다.

4. 다음을 수행하는 iSeries Toolbox for Java JDBC 및 SQL을 사용하여 iSeriesINVENTORY 데이터베이스에서 의류 항목을 검색하는 GetItems servlet을 작성하십시오.
 - a. 세션에 SQL 조회 결과를 포함하는 ResultSet bean을 배치합니다.
 - b. 요청을 ViewItems.jsp로 재지정합니다.

GetItems.jsp에 대한 코드 샘플:

```
public void init() {
    .
    .
    .
    // Load the IBM Toolbox for Java JDBC driver.
    DriverManager.registerDriver(new com.ibm.as400.access.AS400JDBCdriver());
    // Note that we have retrieved the as400 name, userid, and password from
    // web.xml file using and xml parser.
    as400conn =
    DriverManager.getConnection(
        "jdbc:as400://" + as400 + ";naming=sql;errors=full",
        userid,
        password);

    dmd = as400conn.getMetaData();
    .
    .
    .
}

public void service(HttpServletRequest request, HttpServletResponse response){
    .
    .
    .
    Statement select =
    as400conn.createStatement(
        ResultSet.TYPE_SCROLL_INSENSITIVE,
        ResultSet.CONCUR_READ_ONLY);
```

```

ResultSet rs =
    select.executeQuery(
        "SELECT PRODNO, PRODNAME, RPRICE, IMAGE FROM "
        + retailLibrary
        + dmd.getCatalogSeparator()
        + inventoryFile);

HttpSession session = request.getSession(true);
session.setAttribute("resultset", rs);

response.sendRedirect("/ViewItems.jsp");

.
.
.
}

```

5. 테이블 형식으로 의류 항목을 표시하려면 이전 단계에서 확보한 ResultSet bean 으로부터 의류 항목을 검색하는 ViewItems JSP 파일을 작성하십시오. 또한 JSP 파일은 사용자가 크기 및 수량을 선택한 다음, 장바구니에 추가할 수 있도록 각각의 항목에 대한 양식을 포함해야 합니다. iSeries 웹 개발 도구에서 Page Designer 를 사용하여 JSP를 작성할 수 있습니다. 특히 설계 보기에 페이지를 배치한 다음, 소스 보기에서 적절한 코드를 추가할 수 있습니다. ViewItems.jsp에 대한 코드 샘플:

```

<!--Getting the ResultSet Object from the session--><%
    int columnCount = 0;
    ResultSet rs = (ResultSet)session.getAttribute("resultset");
    if(rs !=null)
%>
<%
{
    rs.beforeFirst();
    ResultSetMetaData rsmd = rs.getMetaData ();
    columnCount = rsmd.getColumnCount ();
%>
<TABLE border="1">
<TBODY>
<TR>
    <TD>Product ID</TD>
    <TD>Name</TD>
    <TD width="551">Price</TD>
    <TD colspan="2"></TD>
</TR>
<%while (rs.next ()) {
<TR>
<!--Creating a form for this row (or this item)-->
<FORM name="myform" action="/SV001585/AddtoBasket" onsubmit="return errorChecking(this);">

<!--Getting each column data from this row of ResultSet object-->
<!--Process data is a user defined method to modify the data for display if needed-->
        <%
            for (int i = 1; i <= columnCount; ++i){
                String value = rs.getString(i);
                if (rs.wasNull ())
                    value = "<null>";
                else{
                    if(i==1)
                        prodID=value;
                    value = processData(i,value);
                }
            }
        %>
        <TD><%=value%></TD>
    <%
    }
%>

```

```

<!--Creating quantity input field and size drop down menu-->
<!--Note that we are using product id as the name of the field-->
    <TD width="290">Quantity
    <INPUT size="5" type="text" name='<%=prodID+"Q"%>' ><BR>
    Size <SELECT name='<%=prodID+"S"%>'>
        <OPTION value="s" selected>Small</OPTION>
        <OPTION value="m" selected>Medium</OPTION>
        <OPTION value="l" selected>Large</OPTION>
        <OPTION value="XL" selected>Extra Large</OPTION>
        <OPTION value="XXL" selected>Extra Extra Large</OPTION>
    </SELECT>
</TD>
<TD><INPUT type="image" name="submit" src="images/Add_to_basket.gif"></TD>
</FORM>
</TR>
<%
}
%>
</TBODY>
</TABLE>
<%
}

```

6. iSeries Java 개발 도구 iseriesut.jar 파일을 웹 프로젝트용 lib 폴더로 가져오십시오. 이 JAR 파일은 x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime에서 찾을 수 있으며 여기서 x는 제품을 설치한 디렉토리입니다. 구현을 보려면 SV001585 프로젝트의 AddtoBasket.java 및 ViewBasket.jsp를 참조하십시오. 웹 Perspective의 네비게이터 보기에서 **SV001585** > 소스를 펼쳐서 AddtoBasket.java를 찾을 수 있으며, **SV001585** > 웹 어플리케이션을 펼쳐서 ViewBasket.jsp를 찾을 수 있습니다.
7. iSeries Java 개발 도구의 RecordIOManager bean을 사용하여 장비구니에 추가 단추에 의해 호출되는 AddtoBasket servlet을 작성합니다. 고객이 요청한 수량을 빼고 세션의 Basket Java bean에 항목을 추가하여 iSeries INVENTORY 데이터베이스를 갱신합니다. AddtoBasket.jsp 코드 샘플:

```

public class AddtoBasket extends HttpServlet {

    //Inner class of AddtoBasket
    public class MyRecordIOManager extends RecordIOManager {
        .
        .
        .
        public MyRecordIOManager(
            String hostInfo1,
            String hostInfo2,
            String hostInfo3,
            String file,
            String lib)throws Exception{
            super(hostInfo1, hostInfo2,hostInfo3,file,lib);
            setFileType(RecordIOManager.FILEACCESS_KEYED);
            setCommitLockLevel(RecordIOManager.COMMITLOCKLEVEL_ALL);
            //journal has the same name as the database file
            setJournal(file);
            //journal is in the same library as the database file
            setJournalLibrary(lib);
        }
        .
        .
        .
        public synchronized String updateDBFile(
            String id,

```

```

String size,
String quantity
) {
.
.
.
//opening the file
try {
if (openFile()) {
record = readRecord(key);
quantityAvailable = ((BigDecimal)
record.getValueAt(0,sizeColumn)).intValue();
totalQuantityAvailable = ((BigDecimal)
record.getValueAt(0, 8)).intValue();
if (quantityRequested <= quantityAvailable) {
newQuantity =
new BigDecimal(quantityAvailable - quantityRequested);
totalNewQuantity =
new BigDecimal
(totalQuantityAvailable - quantityRequested);
record.setValueAt(newQuantity, 0, sizeColumn);
record.setValueAt(totalNewQuantity, 0, 8);
// Note that we update the record but we don't commit
// in case the customer decides to
// empty the basket in which
// case we call the rollBack method
updateRecord(record);
status = success;
} else {
status = notEnough;
}
} else
status = accessError;
} catch (Exception e) {
e.printStackTrace();
status = accessError;
}

//closing the file and adding
try {
closeFile();
} catch (Exception e) {
//in case of error rollback
try {
rollback();
} catch (Exception e1) {
e1.printStackTrace();
}
status = accessError;
}

return status;
}
}

//init method of AddtoBasket servlet
public void init() {
hostInfo = GetItems.getHostInfo();
}

public void doGet(HttpServletRequest req, HttpServletResponse res) {
.
.
.
Basket basket = (Basket) session.getAttribute("basket");

```

```

MyRecordIOManager recIO = (MyRecordIOManager) session.getAttribute("recIO");
if (basket == null) {
    basket = new Basket();
    session.setAttribute("basket", basket);
}

if(recIO == null){
    if (recIO == null) {
        try {
            recIO =
                new MyRecordIOManager(
                    hostInfo[0],
                    hostInfo[1],
                    hostInfo[2],
                    GetItems.getInventoryFile(),
                    GetItems.getRetailLibrary());
        } catch (Exception e) {
            try {
                res.sendRedirect("errorPage.html");
                return;
            } catch (Exception e1) {
                e1.printStackTrace();
            }
        }
    }
}
id = req.getParameter("id");
size = req.getParameter(id + "S");
quantity = req.getParameter(id + "Q");

status = recIO.updateDBFile(id, size, quantity);
session.setAttribute("recIO", recIO);

if (status.equals("SUCCESS")) {
    basket.addItem(id, quantity, size);
    try {
        res.sendRedirect("ViewBasket.jsp");
        return;
    } catch (Exception e) {
        e.printStackTrace();
    }
} else {
    if (status.equals("NOT_ENOUGH")) {
        try {
            res.sendRedirect("insufficient.html");
            return;
        } catch (Exception e) {
            e.printStackTrace();
        }
    } else
        if (status.equals("ACCESS_ERROR")) {
            try {
                res.sendRedirect("errorPage.html");
                return;
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
}
}
}

```

8. 장바구니의 내용을 표시하는 ViewBasket JSP 파일을 작성하십시오.

9. ViewForConfirm.jsp를 호출하는 계속 단추를 포함하여 지불 정보를 입력하기 위한 customerinfo.html 양식을 작성하십시오.

10. 전체 장바구니의 내용과 총 밸런스를 표시하는 ViewForConfirm.jsp를 작성하십시오. 계산된 운반 비용과 iSeries ORDERS 데이터베이스에 주문을 제출하는 확인 단추를 추가하여 ViewBasket.jsp와 동일한 방법으로 ViewForConfirm.jsp를 개발할 수 있습니다.
11. iSeries Java 개발 도구의 iSeries 프로그램 호출 마법사를 사용하여 RETAILSTOR 라이브러리의 PLACEODR 서비스 프로그램을 액세스하는 PLACEORD.java bean을 작성하십시오. 마법사는 Java 어플리케이션 또는 웹 서비스 마법사가 iSeries ILE 프로그램을 액세스하는 데 사용할 bean을 작성합니다.
 - a. 마법사를 열려면 네비게이터 보기에서 SV001585를 마우스 오른쪽 단추로 클릭한 다음 신규 > 기타를 선택하십시오.
 - b. 신규 창에서 **iSeries > Java > 프로그램 호출 Bean**을 클릭하십시오.
 - c. 프로그램 호출 마법사에서 iSeries ILE 프로그램 이름, 라이브러리, 프로그램 유형, 입출력 매개변수에 관한 정보를 입력합니다.
 - d. 마법사의 마지막 창은 Java 어플리케이션, 웹 서비스 또는 둘 다에 대한 bean을 작성하기 위한 옵션을 제공합니다. 이 프로젝트에서는 Java 어플리케이션용 하나만을 작성해야 합니다.

주: PLACEODR 서비스 프로그램은 구조 배열을 취하여 각각의 요소를 ORDERS 데이터베이스의 한 레코드에 배치함으로써 각각의 배열에 대한 출력으로 하나의 주문 번호를 생성합니다.

12. ViewForConfirm.jsp 확인 단추에 의해 호출되는 PlaceOrder servlet을 작성하십시오.
 - servlet은 iSeries 프로그램 호출 마법사가 생성한 bean을 사용하여 iSeries 서버를 액세스하며 RETAILSTOR 라이브러리의 ORDERS 데이터베이스에 주문을 배치합니다.
 - 주문은 장바구니의 항목이며 ILE 프로그램에 구조 배열로 전송됩니다.
 - 이 배열의 각 구조가 장바구니의 한 항목입니다.
 - bean에서 호출하는 PLACEORD RPG 서비스 프로그램은 주문 번호를 출력 매개변수로 리턴하며 세션에 이를 배치합니다.

다음 코드 세그먼트는 PlaceOrder servlet이 PLACEORD bean을 사용하는 방법을 표시합니다.

```

.
.
.
public void init() throws ServletException {
    hostInfo = GetItems.getHostInfo();
    super.init();

    try {
        /* creating an instance of the PLACEORD bean created
           by iSeries Program Call Bean wizard */
        orderBean = new PLACEORD();
        orderBean.setConnectionData(hostInfo[0], hostInfo[1], hostInfo[2]);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

.
.
.

public void doPost(HttpServletRequest request, HttpServletResponse response) {

    ...

    PLACEORD.Orditems_Struct inputStruct = null;

    // retrieving the order items from the basket
    Basket basket = (Basket) request.getSession().getAttribute("basket");
    AddtoBasket.MyRecordIOManager recIO = (AddtoBasket.MyRecordIOManager)
        request.getSession().getAttribute("recIO");
    if (basket == null || basket.size() == 0 || recIO == null) {
        try {
            response.sendRedirect("errorPage.html");
        } catch (IOException e) {
            e.printStackTrace();
        }
    } else {
        items = basket.elements();
        // setting array of structure elements
        while (items.hasMoreElements()) {
            item = (String[]) items.nextElement();
            inputStruct = orderBean.getOrdItemAr(j);
            inputStruct.setItemNo(new BigDecimal(item[0]));
            inputStruct.setQuantity(new BigDecimal(item[1]));
            inputStruct.setSizeOrd(item[2]);
            j = j + 1;
        }
        // setting the rest of the array elements to dummy values
        for (int i = j - 1; i < 100; i++) {
            inputStruct = orderBean.getOrdItemAr(i);
            inputStruct.setItemNo(new BigDecimal(0));
            inputStruct.setQuantity(new BigDecimal(0));
            inputStruct.setSizeOrd("s");
        }

        // setting the other two input parameters of the bean
        orderBean.setNumOfItems(new BigDecimal(j));
        orderBean.setBalance((BigDecimal) request.getSession().getAttribute("balance"));
        try {
            // invoking the iSeries program
            orderBean.invoke();

            // retrieving the order number from PLACEORD bean
            orderNumber = (orderBean.getRetCode()).toString();
            request.getSession().setAttribute("orderNumber", orderNumber);
            basket.empty();
            // commit this order now
            recIO.commit();
            response.sendRedirect("orderNumber.jsp");
            return;
        }
    }
}

```

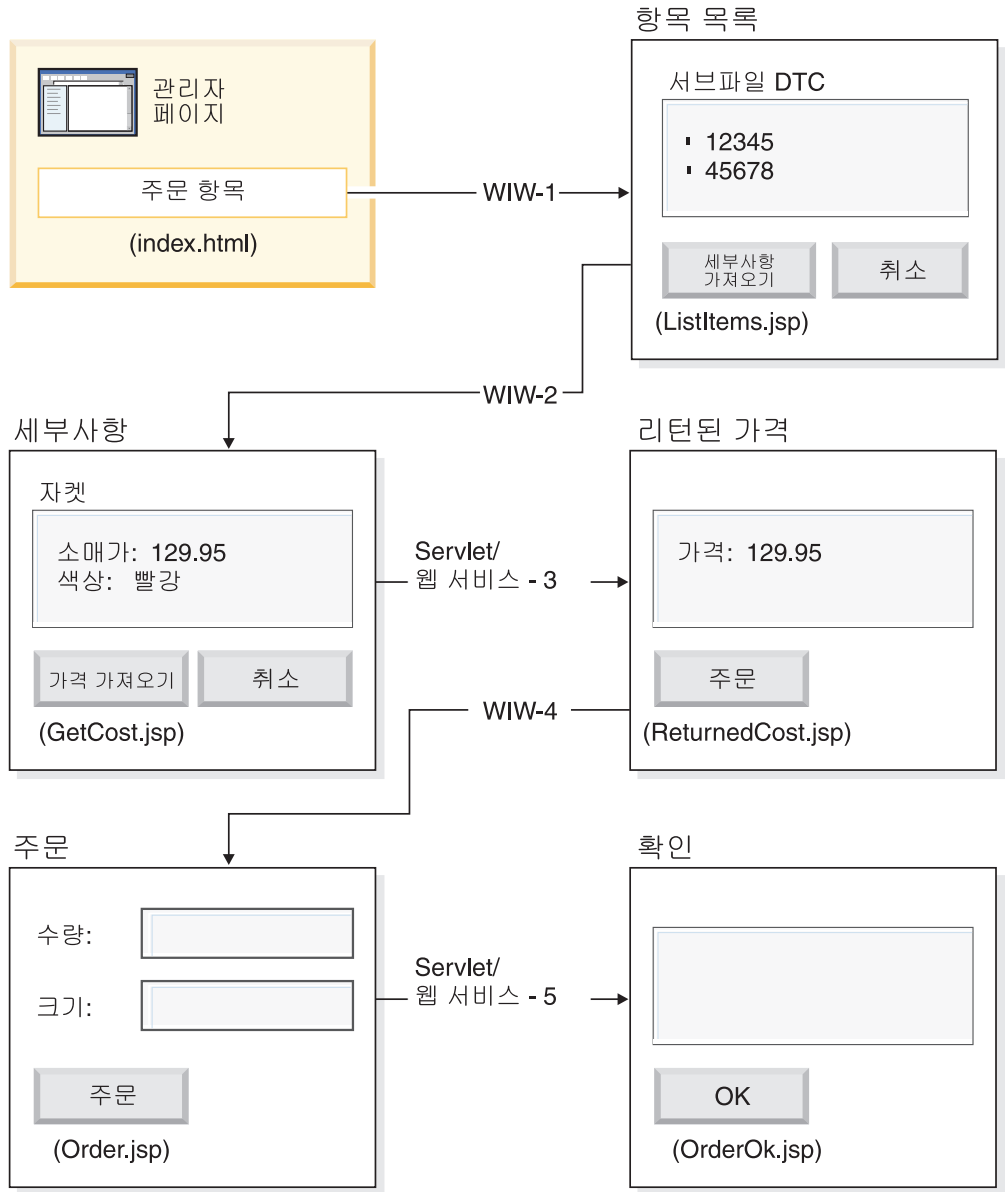
```
    } catch (Exception e) {  
        response.sendRedirect("errorPage.html");  
        e.printStackTrace();  
    }  
}  
}
```

13. 번호를 검색하여 고객에게 확인 메시지와 함께 이를 표시하는 OrderNumber servlet 을 작성하십시오. 고객이 장바구니에 항목을 추가하지 않은 경우, 대신 오류 페이지가 리턴되는지 확인하십시오.

제 7 장 고급 모듈 2: SV000514 및 SV001586 웹 서비스를 사용하는 웹 프로젝트 작성(SV000618)

소개

이 프로젝트는 RPG 프로그래밍 지식을 사용하여 iSeries 웹 서비스 및 RPG 프로그램에 대한 웹 클라이언트를 작성하는 방법을 보여줍니다. 이 프로젝트에서 사용자는 관리자의 역할을 담당하며, 일련의 웹 페이지를 이동하면서 재고 수량을 판별하고, 도매 공급업체로부터 소매 사업에 대한 추가 재고를 주문합니다. 항목 번호를 입력하고 항목의 세부사항을 열람하며, 수량과 크기를 주문하고 확인에 동의합니다.



(WIW = Web Interaction Wizard(웹 상호작용 마법사))

이 모듈은 고급 모듈이므로, 명령어가 각각의 프로젝트 작성 단계를 일일이 안내하지는 않지만 프로젝트를 작성하기 위해 수행하는 개발 단계를 요약합니다. 이 프로젝트에서는 다음과 같은 Development Studio Client 구성요소를 사용합니다.

- iSeries 웹 개발 도구 - Page Designer를 사용하여 웹 페이지를 작성. 웹 상호작용 마법사의 출력을 다양한 웹 구성요소와 통합함
- 원격 시스템 탐색기 - 항목 정보를 리턴하는 TNLSTITM RPG 서비스 프로그램 작성
- 웹 서비스 마법사 - 항목 가격을 찾아서 항목을 주문하는 servlet 프록시 코드 생성
- iSeries Java 개발 도구 - 필요한 servlet을 작성

- WebSphere 테스트 환경 - WebSphere Application Server를 통해 iSeries 서버로 전개하기 전 어플리케이션 검증

시작하기 전에

다음 전제조건이 일치할 경우에만 연습문제를 완료할 수 있습니다. 전제조건은 9 페이지의 제 3 장 『시나리오 실행』에서 보다 자세히 설명됩니다.

- iSeries 서버에 대한 TCP/IP 액세스가 있음.
- STRTCPSVR *ALL 명령을 사용하여 iSeries 서버를 시작했음
- STRTCPSVR *WEBFACING 명령을 사용하여 WebFacing 서버를 시작했음
- WHOLESALE 및 RETAILSTOR 라이브러리를 iSeries 서버로 복원했음
- WHOLESALE, RETAILSTOR 및 QGPL 라이브러리가 라이브러리 목록에 있음
- 9 페이지의 제 3 장 『시나리오 실행』의 모든 타스크를 완료했음(Workbench에서 어플리케이션을 테스트하는 데 필요하지 않은 선택적 WebSphere Application Server 타스크 제외).

웹 페이지, servlet, JSP 및 RPG 코드 작성

SV000618의 구성요소를 구성하려면 다음을 수행하십시오.

1. 작성할 모든 파일을 보유할 웹 프로젝트를 작성하십시오.
2. iSeries 재고 데이터베이스의 항목을 나열하는 ListItems JSP 파일을 작성하십시오. iSeries 웹 개발 도구에서 Page Designer를 사용하여 servlet을 작성할 수 있습니다. 특히 설계 보기에 페이지를 배치한 다음, 소스 보기에서 적절한 코드를 추가할 수 있습니다. 또한 데이터베이스 레코드를 사용하여 서버파일을 입력하는, TNLSTITM RPG 서비스 프로그램과 상호작용하는 서버파일 DTC(Design Time Control)를 삽입해야 합니다. DTC 제어 설정에서 서비스 프로그램을 지정할 수 있습니다.

다음으로, 웹 상호작용 마법사를 사용하여 입력 페이지를 작성하십시오.

- ListItems.jsp를 출력 페이지로 지정하여 재고 항목을 나열함으로써, 웹 상호작용 마법사가 ListItems.wit 파일을 작성하는지 확인하십시오.
- 서버파일 DTC에서 자동으로 TNLSTITM RPG 서비스 프로그램을 호출하기 때문에 웹 상호작용 마법사에 프로그램 호출을 지정하지 않았는지 확인하십시오. 마법사는 또한 ListItems.jsp 페이지를 호출하는 링크로 제공되는 ListItemsWitServlet을 생성합니다.
- 웹 상호작용 마법사가 생성한 ListItemsWit.wit 파일을 검토하려면 다음을 수행하십시오.
 1. **SV001618**을 펼치고 **ListItems.wit**를 두 번 클릭하여 파일에 대한 상호작용 마법사를 표시하십시오.

2. 마법사에서 다음을 클릭하여 상호작용에 지정된 값을 검토하십시오.

다음으로, `ListItems.jsp` 입력 페이지에서 입력을 가져오는 `GetCost JSP` 출력 페이지를 `Page Designer`를 사용하여 작성해야 합니다. 사용자가 `ListItems.jsp` 페이지에서 항목을 클릭하면, `GetCost.jsp` 페이지가 항목에 대한 세부사항을 표시합니다.

`GetCost.jsp` 페이지를 작성한 후, 웹 상호작용 마법사를 사용하여 `ListItems.jsp`(입력 페이지로 선택) 및 `GetCost.jsp`(출력 페이지로 선택) 간의 `WitOrder` 상호작용을 작성하십시오.

- 마법사의 프로그램 호출 페이지에서 `TNLSTITM RPG` 서비스 프로그램으로부터 `GetDetail` 프로시저 및 매개변수 호출을 지정하십시오.
- 프로시저에서 서버파일 `DTC API`가 통합되어 선택된 서버파일 레코드를 판별합니다. 프로시저는 이 정보를 사용하여 `INVENTORY` 데이터베이스로부터 선택된 레코드를 검색하며 `GetCost.jsp`에서 선택된 항목에 대한 이미지를 포함한, 세부사항을 표시합니다.
- 웹 상호작용 마법사에서 생성한 `WitOrderWit.wit` 파일을 검토하려면 다음을 수행하십시오.

1. **SV001618**을 펼치고 **WitOrder.wit**를 두 번 클릭하여 파일에 대한 상호작용 마법사를 표시하십시오.

2. 마법사에서 다음을 클릭하여 상호작용에 대해 지정된 값을 검토하십시오.

`flow` 매개변수가 출력 페이지에서 플로우 제어기로 지정되었는지 확인하십시오. 매개변수의 값이 적절한 `JSP` 파일을 표시하는지 확인할 수 있습니다.

다음으로, 관리자가 **비용 확보** 단추를 눌러서 도매 공급업체로부터 항목의 현재 비용을 검색할 수 있도록 `SV000514` 프로젝트에서 `WSDL(Web Services Definition Language)` 파일을 가져와야 합니다.

- 가격 가져오기 단추는 `QryProdCostServlet.jsp` 및 `SV000514` 프로젝트의 해당 웹 서비스를 호출합니다.
- 웹 서비스 마법사와 가져온 `WSDL` 파일을 사용하여 웹 서비스를 호출하는 데 필요한 `Java` 프록시 코드를 생성하십시오.
- `QryProdCostServlet.jsp`는 `GetCost.jsp` 페이지로부터 입력을 가져오며, `Java` 프록시 코드를 사용하여 선택된 항목의 가격을 찾기 위한 `SV000514` 웹 서비스를 호출하며 `ReturnedCost.jsp` 페이지에 가격을 표시합니다.
- `QryProdCostServicesProxy.java` 코드 및 `QryProdCostServlet.java`를 보려면 다음을 수행하십시오.

1. **SV001618** > 소스 > 프록시 > **soap**을 펼치십시오.

2. **QryProdCostServicesProxy.java**를 두 번 클릭하십시오.

3. QryProdCostServlet.java의 경우, **SV001618** > 소스에서

QryProdCostServlet.java를 두 번 클릭한 다음 Java 프록시 코드를 개시하는 방법을 주의하십시오.

다음으로, 웹 상호작용 마법사를 사용하여 관리자가 주문 단추를 클릭하여 도매 공급업체로부터 선택된 항목을 주문할 수 있도록 ReturnCost.jsp를 입력으로, Order.jsp를 출력으로 링크해야 합니다.

- 이 상호작용에서는 두 페이지의 링크가 올바른 정보를 표시하는 데 충분하므로, 프로그램 호출을 사용할 필요가 없습니다.
- WitPlaceOrder.wit를 보려면 다음을 수행하십시오.
 1. **SV001618**을 펼치십시오.
 2. **WitPlaceOrder.wit**를 두 번 클릭하여 상호작용을 여십시오.
 3. 마법사에서 다음을 클릭하여 지정된 값을 검토하십시오.

다음으로, 관리자가 주문된 항목의 크기와 수량을 지정할 수 있도록 SV001586 웹 서비스를 사용하십시오.

- SV001586 WSDL 파일을 해당 프로젝트로 가져와서 웹 서비스를 호출하는 Java 프록시 코드를 생성한 다음, 사용자가 Order.jsp에서 주문 단추를 눌렀을 때 호출되는 OrderSupplyServlet을 작성하십시오.
- servlet은 Order.jsp에서 정보를 수집하며 SV001586 웹 서비스를 호출하는 웹 서비스 Java 프록시 코드를 호출한 다음 항목을 주문합니다.
- Servlet은 주문이 성공할 경우 OrderOK.jsp를, 주문이 실패할 경우 오류 페이지를 표시합니다.
- SV001586 웹 서비스를 사용하기 위해 생성된 웹 서비스 프록시를 개시하고 호출하는 방법을 알려면 다음을 수행하십시오.
 1. **SV000618** > 소스를 펼치십시오.
 2. **OrderSupplyServlet.java**를 두 번 클릭한 다음 콘텐츠를 검사하십시오.

WebSphere Application Server에 전개하기 전에

어플리케이션을 WebSphere Application Server에 전개하기 전에 먼저 어플리케이션이 제대로 작동하려면 SV000514 및 SV001586 Java 클래스 파일 내에서 특정 URL을 변경해야 합니다.

SV000514의 경우,

- 웹 서비스 프록시 클래스인 QryProdCostServicesProxy에서 호출할 웹 서비스의 URL을 포함하는 변수가 정의됩니다.
- 프록시가 처음으로 작성될 때, 이 URL은 `http://localhost:9080/SV000514/servlet/rpcrouter`로 설정됩니다.

- 이 값으로 설정된 변수로, 19 페이지의 『Workbench에서 어플리케이션 실행』에서 설명한 대로 IDE에 있는 SV000514 프로젝트의 웹 서비스가 호출됩니다.
- 어플리케이션을 전개하기 전에 먼저 iSeries IFS 디렉토리의 EAR 파일인 SVWholeSale.ear을 전개한 위치를 가리키도록 이 URL 값을 변경해야 합니다.

SV001586 프로젝트의 경우,

- 웹 서비스 프록시 클래스인 OrderSupplyServicesProxy에서 호출할 웹 서비스의 URL을 포함하는 변수가 정의됩니다.
- 프록시가 처음으로 작성될 때, 이 URL은 `http://localhost:9080/SV001586/servlet/rpcrouter`로 설정됩니다.
- 이 값으로 설정된 변수로, 19 페이지의 『Workbench에서 어플리케이션 실행』에서 설명한 대로 IDE에 있는 SV001586 프로젝트의 웹 서비스가 호출됩니다.
- 어플리케이션을 전개하기 전에 먼저 iSeries IFS 디렉토리의 SVWholeSaleEAR.ear 파일을 전개한 위치를 가리키도록 이 URL 값을 변경해야 합니다.

제 8 장 주의사항

Note to U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. IBM은 다른 국가에서 이 자료에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산권을 침해하지 않는 한 기능상 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나 비IBM의 제품, 프로그램 또는 서비스 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
고객만족센터
전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을 현상 태대로 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 이 변경사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공되는 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함)간의 정보 교환 및
(ii) 교환된 정보의 상호 이용을 목적으로 정보를 원하는 프로그램 라이선스 사용자는 다음 주소로 문의하십시오.

135-270
서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
고객만족센터

이러한 정보는 해당 조항 및 조건(예를 들어, 사용료 지불 등)에 따라 사용할 수 있습니다.

이 정보에 기술된 라이선스가 있는 프로그램 및 이 프로그램에 대해 사용 가능한 모든 라이선스가 있는 자료는 IBM이 IBM 기본 계약 IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 비IBM 제품을 테스트하지 않았으므로, 이들 제품과 관련된 성능의 정확성, 호환성 또는 기타 주장에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위해 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며, 실제 기업에서 사용하는 이름 및 주소와 유사하더라도 전적으로 우연입니다.

저작권:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용없이 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이들 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 암시하지 않습니다. 귀하는 IBM의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용없이 이러한 샘플 응용프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 그 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

(c) (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp.의 샘플 프로그램에서 파생됩니다. (C) Copyright IBM Corp. 1992, 2002. All rights reserved.

프로그래밍 인터페이스 정보


프로그래밍 인터페이스 정보는 이 프로그램을 사용하여 어플리케이션 소프트웨어를 작성하는 데 도움을 주기 위한 것입니다.

일반 사용 프로그래밍 인터페이스를 사용하여 이 프로그램의 도구 서비스를 확보하는 어플리케이션 소프트웨어를 작성할 수 있습니다.

그러나 이 정보에는 진단, 수정 및 조정 정보도 포함될 수 있습니다. 어플리케이션 소프트웨어의 디버그를 돕기 위해 진단, 수정 및 조정 정보가 제공됩니다.

경고: 이 진단, 수정 및 조정 정보는 변경될 수 있으므로 프로그래밍 인터페이스로 사용하지 마십시오.

상표 및 서비스표

- 400
- AFP
- AIX
- AIX windows
- APPN
- Application System/400
- AS/400
- CUA
- DB2
- DB2 Extenders
- DB2 Universal Database
- 
- GDDM
- IBM
- OS/390
- OS/400
- POWER2
- PowerPC
- PROFS
- RPG/400
- RS/6000

- AS/400e
- BookManager
- C Set ++
- C/400
- CICS
- CICS/400
- CICS/ESA
- COBOL/2
- COBOL/400
- Common User Access
- IBMLink
- Integrated Language Environment
- iSeries
- Language Environment
- MQSeries
- Network Station
- Open Class
- Operating System/2
- Operating System/400
- OS/2
- S/390
- SAA
- SQL/400
- System/36
- System/38
- VisualAge
- VTAM
- WebSphere

InstallShield는 InstallShield Corporation의 상표입니다.

Intel 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Development Corporation의 상표입니다.

Java 및 모든 Java 기반 상표 및 로고는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표 또는 등록상표입니다.

Rational은 미국 또는 기타 국가에서 사용되는 IBM Corporation 및 Rational Software Corporation의 상표입니다.

Lotus, Lotus Notes 및 Domino는 미국 또는 기타 국가에서 사용되는 Lotus Development Corporation의 상표입니다.

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT, Win32, Win32s 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

Netscape Navigator는 Netscape Communications Corporation의 상표입니다.

UNIX는 The Open Group의 등록상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.



프로그램 번호: 5724-A81