



Applicazione scenario iSeries - Retail Store

Versione 5.1 per Windows



Applicazione scenario iSeries - Retail Store

Versione 5.1 per Windows

Indice

Capitolo 1. Introduzione all'applicazione scenario del negozio a dettaglio iSeries . 1

Scenari basati sul progetto 2

Capitolo 2. Concetti tecnologici 5

Capitolo 3. Esecuzione dello scenario . . 9

Prima di cominciare 9

Installazione dei file di esempio 9

 Ripristino dei file .savf 10

 Modifica dell'elenco librerie 11

Importazione ed esecuzione dell'applicazione in Development Studio Client 12

 Estrazione dei file .zip sul proprio sistema locale . 12

 Creazione dei progetti Web 13

 Importazione dei file del progetto web 14

 Importazione dei file del progetto WebFacing e creazione del progetto WebFacing 14

 Configurazione del server 15

 Definizione delle informazioni sul server 17

Esecuzione dell'applicazione nel workbench 17

 Esecuzione dell'applicazione come cliente nel workbench 17

 Esecuzione dell'applicazione come amministratore nel workbench 18

Distribuzione dell'applicazione scenario al WAS (WebSphere Application Server) 19

 Configurazione di WebSphere Application Server . 19

 Protezione della pagina dell'amministratore . . . 20

 Creazione dei file EAR per la distribuzione su WebSphere Application Server iSeries. 21

 Distribuzione dei file EAR su WebSphere Application Server iSeries 22

 Esecuzione dell'applicazione in WebSphere Application Server 23

Capitolo 4. Modulo 1 dettagliato: Creare un servizio Web che restituisca i prezzi dei prodotti (SV000514) 25

Introduzione 25

Prima di cominciare 25

Creazione di un nuovo progetto Web. 26

Definizione delle informazioni del server iSeries . . 26

Creazione del programma di servizio RPG 26

 Creazione dei parametri e generazione del bean

 Java 27

 Creare un servizio Web dal bean Java 28

Verifica dell'esempio 29

Capitolo 5. Modulo 2 dettagliato: Creare un'interfaccia per visualizzare l'inventario e ordinare le voci (SV000501) 31

Introduzione 31

Prima di cominciare 31

Creazione di un progetto WebFacing 32

Conversione dell'origine DDS 33

 Configurazione del supporto UTF-8 in workbench — Solo per gli utenti di WAS

 Versione 4.0 34

 Configurazione del supporto UTF-8 per

 WebSphere Application Server 35

Creazione di un foglio di stile 35

 (Facoltativo) Personalizzazione manuale del foglio di stile a cascata. 36

Estensione e miglioramento del progetto WebFacing con una interazione web 36

Collegamento del progetto ad una Interazione web . 39

Pubblicazione dei file e riavvio del server 40

Verifica dell'interfaccia. 41

Capitolo 6. Modulo 1 avanzato: Creare HTML, servlet e file JSP che mettano gli ordini del cliente su un server iSeries (SV001585). 43

Introduzione 43

Riepilogo dei passi del livello avanzato 43

Prima di cominciare 44

Creazione di pagine web, servlet e file JSP 44

Capitolo 7. Modulo 2 avanzato: Creare il progetto Web che utilizza i servizi web SV000514 e SV001586 (SV000618) . 53

Introduzione 53

Prima di cominciare 54

Creazione pagine Web, servlet, JSP e codice RPG . . 54

Prima di effettuare la distribuzione a WebSphere

Application Server 56

Capitolo 8. Informazioni particolari. . . 59

LICENZA SOGGETTA ALLE LEGGI SUL DIRITTO

D'AUTORE: 60

Informazioni sull'interfaccia di programmazione . . 61

Marchi e marchi di servizio 61

Capitolo 1. Introduzione all'applicazione scenario del negozio a dettaglio iSeries

Il pacchetto scenario è un'applicazione di esempio sviluppata utilizzando IBM WebSphere Development Studio Client per iSeries, la tecnologia basata su Eclipse per il server iSeries. Lo scenario è rivolto a sviluppatori che desiderano utilizzare un server iSeries per:

- Sviluppo Java
- Sviluppo Web
- La gestione e la conversione del codice RPG in applicazioni Web

L'applicazione è composta da due file .savf che vengono ripristinati sul server iSeries e da cinque file .zip che vengono importati nel prodotto del client. Essi formano una serie di pagine Web con URL per diversi punti nel processo.

Nota importante: consultare la pagina relativa al supporto per tutti i service pack, le PTF e le altre informazioni: ibm.com/software/awdtools/wdt400/support/.

Tale scenario accompagna l'utente in varie parti del prodotto con un'enfasi sui componenti iSeries-specific, ad esempio:

- L'IBM WebFacing Tool
- I servizi Web
- Gli strumenti di sviluppo web per iSeries (incluso le procedure guidate Interazione Web e Chiamata programma)
- Gli strumenti di sviluppo Java per iSeries
- L'IBM Toolbox per Java

Lo scenario mostra la situazione di due società, una di vendita di abbigliamento all'ingrosso e una al dettaglio, che collaborano e che usano entrambi il loro server iSeries per dati e logica di lavoro. Precedentemente, le aziende comunicavano l'una con l'altra tramite posta elettronica, telefono e fax in modo da controllare l'inventario, inoltrare gli ordini e tenere traccia degli ordini per la relativa realizzazione. A questo punto, esse sperano di utilizzare il web per realizzare transazioni aziendali regolari.

La vendita al dettaglio desidera disporre di un sito web che:

- I clienti possano utilizzare per l'acquisto dei prodotti
- Gli impiegati possano utilizzare per effettuare l'ordine dell'inventario dalla vendita all'ingrosso

La vendita all'ingrosso desidera:

- Ricevere in linea ordini al dettaglio di cui tenere traccia
- Assistere più clienti potenziali

In questo scenario, l'utente assumerà il ruolo di un consulente di programmazione per entrambe le società, aiutando a spostare i loro lavori sul Web.

Quest'applicazione ha due diversi punti di immissione basati sul tipo di utente. In qualità di cliente, si inizia con il visualizzare i prodotti che il negozio può offrire, in questo caso abbigliamento casual. Se si desidera effettuare un acquisto, è

possibile fare clic su un collegamento per accedere ad una schermata dell'ordine. Una volta effettuato l'ordine, viene creata una pagina di riepilogo ed è possibile continuare con gli acquisti, annullare l'ordine o inoltrare l'ordine.

In qualità di amministratore, si dispone di un ID utente sicuro dell'applicazione, che richiede la normativa di sicurezza dell'applicazione da definire durante la distribuzione. Il punto di accesso è una schermata di collegamento dove è possibile visualizzare gli ordini, visualizzare l'inventario ed effettuare l'acquisto all'ingrosso. E' possibile selezionare articoli, controllare l'ultimo prezzo all'ingrosso e ordinare la dimensione e la quantità desiderate. L'applicazione verifica se il grossista dispone della quantità e della dimensione richieste, quindi conferma o notifica il mancato completamento dell'ordine alla data corrente.

Sotto la superficie dell'applicazione, si verificano molte azioni nelle diverse parti del prodotto. La seguente tabella mostra il processo ed il componente del prodotto che controllano le singole parti dell'applicazione. Continuare a consultare le informazioni dettagliate riguardo a come eseguire ciascuna attività.

Tabella 1.

Attività applicazione cliente	Attività applicazione amministratore	Processo di evidenziazione
Visualizzare prezzi dei prodotti		Utilizzare un programma RPG iSeries per creare un servizio Web ed utilizzare gli strumenti di sviluppo web in modo da visionare e visualizzare i prezzi.
Effettuare un ordine dal negozio		Utilizzare i file JSP e servlet insieme agli strumenti di sviluppo Java iSeries, gli strumenti di sviluppo web iSeries e l'IBM Toolbox per Java in modo da accedere e visualizzare l'inventario su un server iSeries così come effettuare ordini e mostrare il riepilogo di un acquisto.
	Visualizzare inventario	Utilizzare IBM WebFacing Tool per convertire un programma RPG esistente in un'applicazione Web e utilizzare gli strumenti di sviluppo web per personalizzare la pagina web.
	Effettuare l'ordine dalla vendita all'ingrosso con l'ID di merce e la quantità.	Creare un servizio web che venga richiamato quando si fa clic sul pulsante Acquista .
Visualizzare la pagina web iniziale del negozio.	Visualizzare la pagina web iniziale per l'ordine dell'inventario.	Utilizzare gli strumenti di sviluppo Web per iSeries per creare entrambe le pagine principali.

Scenari basati sul progetto

L'applicazione scenario iSeries è composta da cinque progetti, denominati SV000501, SV000514, SV000618, SV001585 e SV001586. Tale guida contiene le istruzioni su come eseguire l'applicazione precompressa. Essa contiene inoltre moduli dettagliati e moduli avanzati, che mostrano in che modo creare, in maniera autonoma, i diversi progetti. I moduli dettagliati sono rivolti a sviluppatori che sono relativamente nuovi allo sviluppo di applicazioni e a Development Studio

Client. I moduli avanzati sono rivolti a sviluppatori che hanno familiarità con lo sviluppo di applicazioni e ancora di più, con Development Studio Client.

Nota: Sebbene vi siano cinque progetti, esistono solo quattro moduli poiché i progetti SV001586 e SV000514 sono moduli Servizi Web combinati.

Questo scenario appare composto da cinque progetti:

Progetto SV000501: creare un progetto Web per visualizzare gli ordini in sospeso, l'inventario e i dettagli del prodotto – Questo progetto è creato con Strumenti di sviluppo web iSeries e IBM WebFacing Tool ed è rivolto a programmatori RPG che hanno molta familiarità con lo sviluppo di applicazioni Web, che desiderano utilizzare IBM WebFacing Tool per inserire le loro applicazioni RPG sul Web.

Progetto SV001585: Creare il codice HTML, i servlet e i file JSP che effettuano gli ordini del client su un server iSeries – Questo progetto utilizza le classi JDBC e SQL di IBM Toolbox per Java, il bean RecordIOManager di Strumenti di sviluppo Java iSeries e la procedura guidata Chiamata programma di iSeries. Tali elementi mostrano diversi modi di accesso e di gestione dei dati e programmi presenti sul server iSeries. Questo progetto è rivolto a programmatori Java e sviluppatori di applicazioni web che desiderano sviluppare pagine Web per accedere al codice e ai dati iSeries. Inoltre, occorrerebbe avere familiarità con il server iSeries e la programmazione RPG.

Progetto SV000514: Creare un servizio web iSeries per fornire i prezzi del prodotto – La procedura guidata Servizi Web utilizza un bean Java creato dalla procedura guidata Chiamata programma iSeries per richiamare una o più procedure sul server iSeries e ritrasferisce le informazioni al browser. Questo progetto è rivolto a programmatori RPG che desiderano utilizzare Servizi Web per creare applicazioni modulari autonome che possono essere descritte, pubblicate, ubicate e richiamate da World Wide Web.

Progetto SV001586: Creare un servizio Web per effettuare ordini all'ingrosso tramite un server iSeries – Il servizio web accetta un ID di merce, oltre alla quantità richiesta ed effettua un ordine utilizzando la vendita all'ingrosso. Questo progetto è un componente di SV000514 ed è progettato per i programmatori RPG che desiderano creare servizi Web.

Progetto SV000618: creare un progetto Web per interfacciarsi con il modulo di ordine, il modulo di inventario e acquistare l'ordine creato da IBM WebFacing Tool – Questo progetto richiede Strumenti di sviluppo web iSeries e include la creazione di file HTML e JSP per utilizzare e collegare servizi Web sviluppati in SV000514 e SV001586. Il progetto è rivolto a sviluppatori che desiderano utilizzare Servizi Web e che hanno familiarità con la programmazione in RPG e Java.

Capitolo 2. Concetti tecnologici

Per essere in grado di utilizzare l'applicazione scenario, occorre conoscere alcuni concetti relativi alla tecnologia, specialmente se non si è esperti nello sviluppo di applicazioni Web. Quello che segue è un breve elenco di termini che si potranno incontrare durante l'utilizzo dell'applicazione.

File EAR (Enterprise Archive)

Un file EAR è un file JAR (Java Archive) standard con un'estensione .ear. Tali file possono contenere più progetti Web ed è possibile utilizzarli per comprimere e distribuire le applicazioni Web in WAS (WebSphere Administrative Server). **Nota:** nella versione GUI dello strumento di distribuzione delle applicazioni J2EE SDK, occorre creare prima un file EAR, quindi aggiungere i file JAR e WAR (Web Archive) al file EAR. Se si utilizzano gli strumenti di compressione della riga comandi, comunque, occorre prima creare i file JAR e WAR, quindi il file EAR.

IBM WebFacing Tool

IBM WebFacing Tool converte le interfacce 5250 esistenti nelle GUI (graphical user interface) basate sul browser. E' possibile estendere l'uso dei programmi a Internet o ad una intranet modificando o meno in minima misura le applicazioni iSeries originali.

File JAR (Java Archive)

Un file JAR è un pacchetto compresso di file Java, simile a un file .zip. Il pacchetto contiene i file di classi, immagini e suoni per un applet Java, riuniti in un unico file e compressi per uno scaricamento più veloce nel browser.

JSP (Java Server Page)

Le JSP forniscono la possibilità di visualizzare il contenuto dinamico in pagine HTML statiche. Scritte in Java, le JSP sono indipendenti dalla piattaforma e dal server. Separando la presentazione Web dal contenuto Web, le JSP sono di ausilio per gli sviluppatori che necessitano di modificare velocemente il design e la visualizzazione delle pagine Web.

Bean chiamata programma

Si tratta di bean Java generati dal wizard Chiamata programma. Uno è un bean java regolare, utilizzato dalle applicazioni java. L'altro tipo può essere utilizzato dal wizard del servizio Web per creare un servizio Web.

Wizard Chiamata programma

Il wizard Chiamata programma è di ausilio nella creazione dei bean Java e del file PCML associato necessario per richiamare un programma o una procedura iSeries. Il wizard richiede all'utente le informazioni riguardanti il programma o gli oggetti del servizio programma e i parametri per gli oggetti, quindi crea i bean Java desiderati (e il file PCML).

RPG (Report Program Generator)

Un linguaggio di programmazione procedurale utilizzato dai programmatori iSeries. E' possibile utilizzare RPG per creare applicazioni aziendali come programmi per la fatturazione e per l'immissione di ordini. La versione più recente, ILE RPG IV, espande le capacità del linguaggio RPG, supportando l'esperienza che i programmatori hanno acquisito con le versioni precedenti.

Servlet

Programmi dal lato server, scritti in Java, che funzionano sui server abilitati a Java o sui server delle applicazioni come gli IBM WebSphere Application Server. I servlet eseguono attività specificate dal server, come soddisfare le richieste generando una risposta HTML. Ad esempio, è possibile utilizzare i servlet in un'applicazione di attività bancarie in linea per rispondere all'utente mentre invia i dati al server.

Componenti Web

E' possibile utilizzare i Componenti Web per definire gli oggetti iSeries come i campi di immissione dei dati e i pulsanti, che possono scambiare informazioni tra i programmi del server iSeries e la pagina Web. Gli sviluppatori possono utilizzare i Componenti Web per catturare gli eventi utente come il controllo della sintassi dei campi di immissione e la selezione dei pulsanti.

Wizard Interazione Web

Questo wizard fa parte di Strumenti di sviluppo web iSeries. Crea e gestisce le interazioni tra i programmi ILE e le pagine Web. Il wizard controlla dove vengono visualizzati i messaggi di errore, le immissioni e le emissioni e indirizza i dati dai campi di immissione e emissione ai programmi ILE. E' anche possibile utilizzare il wizard Interazione Web per mettere in corrispondenza i messaggi di errore con l'area in cui si è verificato l'errore in modo che l'utente ne possa identificare facilmente l'origine.

Servizi Web

I servizi Web sono applicazioni autonome progettate ed implementate per essere utilizzate su Internet. Vengono creati con standard aperti come SOAP, WSDL e XML. Esistono varie situazioni aziendali in cui è possibile utilizzare un servizio Web, compreso un sistema di gestione dell'inventario in cui i client possono controllare i propri livelli di inventario attraverso Internet o, se si desidera, effettuare l'ordine di un prodotto direttamente da un fornitore.

WSDL (Web services definition language)

WSDL è un linguaggio basato su XML che definisce l'interfaccia di un servizio Web. WSDL riconosce il servizio Web e gestisce il flusso di informazioni tra quest'ultimo e il programma server. Ad esempio, uno sviluppatore può utilizzare WSDL per creare un'interfaccia per un sito Web che visualizza le quote azionarie aggiornate.

WebSphere Studio Workbench

IBM WebSphere Development Studio Client per iSeries è costruito su WebSphere Studio Workbench, l'implementazione dell'IBM della piattaforma Eclipse. Il workbench universale, estensibile integra tutti gli strumenti necessari per creare e gestire le applicazioni. Gli sviluppatori possono utilizzare Development Studio

Client per incorporare nuovi oggetti nell'ambiente di sviluppo tramite l'utilizzo di plug-in e file Java aggiunti direttamente, grafici, video ecc.

Capitolo 3. Esecuzione dello scenario

E' possibile eseguire le applicazioni *Wholesale* e *Negozio al dettaglio* all'interno del workbench di Development Studio Client o su WebSphere Application Server per qualsiasi piattaforma incluso iSeries. Consultare Capitolo 1, "Introduzione all'applicazione scenario del negozio a dettaglio iSeries", a pagina 1 per una panoramica delle applicazioni.

Alla fine di questo capitolo, sarà possibile:

- Ripristinare i file di esempio e gli oggetti di Development Studio Client sul server iSeries
- Creare i progetti Web e un progetto WebFacing che contenga i file
- Importare i file ripristinati dal server iSeries in Development Studio Client
- Configurare il WAS (WebSphere Application Server)
- Configurare il progetto WebFacing
- Eseguire l'applicazione come cliente nell'ambiente di verifica WebSphere
- Eseguire l'applicazione come amministratore nell'ambiente di verifica WebSphere
- Configurare il proprio WAS (WebSphere Application Server)
- Proteggere la pagina dell'amministratore
- Creare i file EAR per la distribuzione WAS esterna
- Distribuire l'applicazione sul WAS
- Eseguire l'applicazione nel WAS

Prima di cominciare

Per verificare le applicazioni dal workbench, è necessario assicurarsi che:

- Si possiede un server iSeries V5R1 o successiva
- Sono state applicate le più recenti PTF WebFacing. Consultare la pagina di supporto per le PTF:
<http://www.ibm.com/software/awdtools/wdt400/support/>
- Si disponga dell'accesso NET USE all'Server iSeries

Installazione dei file di esempio

Per utilizzare l'Applicazione iSeries scenario è necessario lavorare con i seguenti file:

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

I file .savf contengono i dati iSeries e i programmi RPG, i file .zip contengono le applicazioni Web che interagiscono con i programmi iSeries per manipolare i dati

iSeries. E' necessario ripristinare prima i file .savf sul server iSeries e poi importare i file .zip in Development Studio Client ed eseguire l'applicazione nel workbench.

Ripristino dei file .savf

Per lavorare con gli esempi di questo manuale è necessario ripristinare le librerie WHOLESale e RETAILSTOR sull'Server iSeries. Questa operazione è necessaria anche se le librerie sono già state ripristinate per un release precedente del prodotto, poiché il loro contenuto è differente.

Nota: I file .savf utilizzati per installare la libreria di esempio vanno utilizzati con una versione V5R1 o successiva dell'Server iSeries. In questo scenario, sono state ripristinate entrambe le librerie sullo stesso Server iSeries, ma se si stesse sviluppando questa applicazione per un utilizzo reale sarebbe necessario ripristinare le due librerie su due Server iSeries differenti. E' necessario ripristinare la libreria WHOLESale sull'Server iSeries che sta fornendo i servizi Web e la libreria RETAILSTOR sull'Server iSeries che appartiene alla vendita al dettaglio.

Per ripristinare il file Wholesale.savf:

1. Collegarsi all'Server iSeries attraverso un'emulazione 5250.
 - a. Creare una libreria che contenga i file di salvataggio. Per creare una nuova libreria nell'emulazione, immettere CRTLIB.
 - b. Denominare la libreria SCENARIO.
 - c. Passare alla riga successiva; specificare *TEST come tipo di libreria e premere Invio per salvare le modifiche.
 - d. Creare due file di salvataggio utilizzando il comando CRTSAVF e premendo invio tra le due righe:

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

Queste righe specificano che si desidera creare i file di salvataggio nella libreria Scenario.

2. Sulla stazione di lavoro, aprire la finestra Richiesta comandi.
 - a. E' necessario passare alla directory dove sono presenti i file .savf. Per impostazione predefinita, è necessario immettere `c:\wdsc\wdscsampl`. Se è stato installato il prodotto su un drive diverso o se si è scelto di non utilizzare "wdsc" come directory principale per Development Studio Client, passare alla directory `wdscsampl` nel posto in cui è stato installato il prodotto.
 - b. Sulla riga comandi, immettere: `ftp hostname`, dove *hostname* è il nome del proprio Server iSeries, ad esempio, `PROD400`.
 - c. Immettere l'ID utente e la password per il server iSeries.
 - d. Sulla riga comandi, immettere `cd /qsys.lib/scenario.lib` per andare alla libreria Scenario.
 - e. Immettere le seguenti righe:

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

Queste righe (prima di quit) specificano che si desidera prendere i file di salvataggio dal sistema locale e inserirli nel server iSeries.

3. Dalla console iSeries, ripristinare la libreria Wholesale:

- a. Immettere RSTLIB e premere F4 per definire come si desidera ripristinare la libreria.
 - b. Nel campo **Libreria salvata** immettere WHOLESale e premere il tasto Tab.
 - c. Nel campo **Unità** immettere *savf e premere il tasto Tab.
 - d. Premere Invio nel campo successivo per visualizzare ulteriori valori e passare al campo **File di salvataggio** utilizzando il tasto Tab.
 - e. Immettere RETAILSTOR nel campo **File di salvataggio** e premere il tasto Tab.
 - f. Nel campo **Libreria**, cancellare il valore esistente ed immettere scenario.
 - g. Premere Invio per ripristinare la libreria WHOLESale sul proprio server iSeries.
4. Ripetere questa procedura per il file Retailstor.savf:
- a. Immettere RSTLIB e premere F4 per definire come si desidera ripristinare la libreria.
 - b. Nel campo **Libreria salvata** immettere RETAILSTOR e premere il tasto Tab.
 - c. Nel campo **Unità** immettere *savf e premere il tasto Tab.
 - d. Premere Invio nel campo successivo per visualizzare ulteriori valori e passare al campo **File di salvataggio** utilizzando il tasto Tab.
 - e. Immettere RETAILSTOR nel campo **File di salvataggio** e premere il tasto Tab.
 - f. Nel campo **Libreria**, cancellare il valore esistente ed immettere scenario.
 - g. Premere Invio per salvare l'azione e ripristinare la libreria RETAILSTOR sul server iSeries.

Modifica dell'elenco librerie

Sebbene siano state ripristinate le librerie RETAILSTOR e WHOLESale sul server iSeries, occorre assicurarsi che esse siano nell'elenco librerie utilizzato quando ci si collega. In caso contrario, le applicazioni Web potrebbero non essere in grado di rilevare le librerie e si potrebbero verificare dei problemi durante l'esecuzione o la creazione delle applicazioni. Occorre anche assicurarsi che la libreria QGPL sia presente nell'elenco librerie.

Per fare ciò, è anche necessario aggiornare la descrizione lavoro ed il profilo utente in modo che le modifiche all'elenco librerie siano permanenti.

Gli utenti hanno la possibilità di modificare in diversi modi le proprie librerie, l'elenco riportato di seguito ne descrive alcuni:

1. Sul server iSeries, immettere dsplibl e premere Invio per visualizzare il proprio elenco librerie.
2. Verificare che nell'elenco siano presenti le librerie RETAILSTOR, WHOLESale e QGPL. Se queste sono presenti, è possibile uscire e chiudere la finestra di emulazione. Se non sono presenti:
 - a. Premere F12 per uscire dal pannello corrente.
 - b. Individuare il nome della propria descrizione. Per fare ciò, immettere dspusrprf *username* e premere Invio per visualizzare il proprio profilo utente, laddove *username* è l'ID che si utilizza per collegarsi al server iSeries. La descrizione lavoro è su una delle pagine del proprio profilo utente (utilizzare Page Down). Una volta trovata la descrizione del lavoro, prenderne nota e premere F12 per uscire dallo schermo.
 - c. Sulla riga comandi, immettere chgjobd e premere F4.
 - d. Nel campo **Descrizione del lavoro**, immettere il nome della descrizione lavoro e premere F10 per vedere i parametri aggiuntivi.

- e. Scorrere la pagina fino a **Elenco librerie iniziale**.
- f. Nel campo sottostante **Elenco iniziale libreria**, verrà visualizzato "+ per altri valori". Immettere un segno più e premere Invio.
- g. Nella pagina **Specificare altri valori**, cancellare qualsiasi voce esistente ed immettere RETAILSTOR, WHOLESale, e il separatore QGPL, tra ogni valore. E' obbligatorio immettere le librerie in questo ordine in quanto le applicazioni devono prima trovare la libreria RETAILSTOR. Premere Invio.
- h. Premere nuovamente Invio per salvare le modifiche.
- i. Sulla riga comandi, immettere: chgusrprf e premere Invio.
- j. Immettere il nome utente e premere Invio.
- k. Premere F10 per vedere ulteriori parametri.
- l. Scorrere la pagina fino a **Descrizione lavoro**.
- m. Se il nome della propria descrizione del lavoro non coincide con la descrizione del lavoro, appare il comando dspusrprf, immettere il nome della descrizione del lavoro trovata con quel comando e premere Invio per salvare le modifiche.

Importazione ed esecuzione dell'applicazione in Development Studio Client

Esistono cinque file zip nei file di programma installati per il client Development Studio, per impostazione predefinita nella directory c:\wdsc\wdscsampl. I file sono:

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

Sarà necessario importare ogni file separatamente nel workbench di Development Studio Client e specificare anche un file EAR (Enterprise Archive) corrispondente per ciascuno. Un file EAR è un file JAR (Java Archive) standard con un'estensione .ear. Tali file possono contenere più progetti Web ed è possibile utilizzarli per comprimere e distribuire le applicazioni Web in WAS (WebSphere Administrative Server).

I primi quattro file corrispondono ai progetti Web e il quinto corrisponde a un progetto WebFacing. Occorre creare prima i progetti Web, poi il progetto WebFacing. In seguito, è possibile importare il contenuto dei file zip nei cinque progetti.

Estrazione dei file .zip sul proprio sistema locale

Prima di poter importare il contenuto dei file .zip dello scenario nel workbench di Development Studio Client, è necessario estrarli in directory separate in modo da poterli importare come file di sistema piuttosto che come file .zip.

Nota: Se si utilizza un programma diverso da WinZip, completare le attività secondo le procedure di quel programma.

1. Sul proprio sistema locale, esplorare c:\wdsc\wdscsampl. La directory c:\wdsc potrebbe essere diversa se si sceglie di installare un drive diverso o una cartella di base differente. In tal caso, esplorare quella directory, quindi trovare \wdscsampl.

2. Fare doppio clic con il tastino del mouse sul seguente file .zip: SV000501.zip.
3. Fare clic sul pulsante Estrai ed esplorare la stessa cartella in cui sono i file .zip, per impostazione predefinita, c:\wdsc\wdscsampl.
4. Selezionare il pallino **Tutti i file**, quindi fare clic su **Estrai**.
5. All'interno della directory wdscsampl, si vedrà che è stata creata una cartella SV000501 contenente tutti i file necessari.
6. Ora che tutti i file sono stati estratti in c:\scenario\SV000501, cancellare il file SV000501.zip per evitare di selezionarlo accidentalmente quando si importa il proprio file di sistema.
7. Ripetere i passi presentati in questa sezione per gli altri quattro file .zip (SV000618.zip, SV001585.zip, SV001586.zip e SV000514.zip.)

Creazione dei progetti Web

In questa sezione, verranno creati progetti Web per SV000514, SV000618, SV001585 e SV001586. I passi per la creazione di ciascun progetto sono illustrati separatamente. Sebbene i passi possano sembrare ripetitivi, tener presente che esistono delle piccole differenze di cui è necessario tenere conto durante la creazione dei quattro progetti (ad esempio, quale progetto è associato a un file EAR).

Per creare i progetti Web:

1. Aprire IBM WebSphere Development Studio Client per iSeries dal menu Start/Avvio.
2. In Development Studio Client, aprire la prospettiva Web dalla barra dei menu, facendo clic su **Finestra > Apri prospettiva > Altro > Web** e fare clic su **OK**.
3. Fare clic su **File > Nuovo > Progetto web dinamico**.
 - a. Nel campo **Nome progetto**, immettere SV000514.
 - b. Controllare la casella **Opzioni avanzate di configurazione** e fare clic su **Successivo**.
 - c. Fare clic sul pulsante **Nuovo** accanto al campo **Progetto EAR**.
 - d. Nel campo **Nome progetto**, immettere SWholeSaleEAR e fare clic **Fine** su entrambe le caselle di dialogo.
4. Fare clic su **File > Nuovo > Progetto web dinamico**.
 - a. Nel campo **Nome progetto**, immettere SV001586 e fare clic su **Successivo** (La casella di spunta **Opzioni avanzate di configurazione** potrebbe essere controllata automaticamente).
 - b. Verificare che SWholeSaleEAR sia selezionato nel campo a discesa **Progetto EAR**.
 - c. Fare clic su **Fine**.
5. Fare clic su **File > Nuovo > Progetto web dinamico**.
 - a. Nel campo **Nome progetto**, immettere SV000618 e fare clic su **Avanti**.
 - b. Fare clic sul pulsante **Nuovo** accanto al campo **Progetto EAR**.
 - c. Nel campo **Nuovo nome progetto**, immettere SVStoreEAR e fare clic su **Fine** su entrambe le caselle di dialogo.
6. Fare clic su **File > Nuovo > Progetto web dinamico**.
 - a. Nel campo **Nome progetto**, immettere SV001585 e fare clic su **Avanti**.
 - b. Verificare che SVStoreEAR sia selezionato nel campo a discesa **Progetto EAR**.
 - c. Fare clic su **Fine**.

Importazione dei file del progetto web

Per quanto riguarda l'importazione dei file di sistema nell'area di lavoro, la procedura che segue illustra la modalità di importazione dei file in SV000514. La procedura è la stessa per gli altri tre file di sistema del progetto web: SV001586, SV001585 e SV000618. La sezione successiva illustra la modalità di creazione ed importazione del progetto WebFacing.

Per importare i file:

1. Fare clic su **Sfoglia ed esplorare**
2. Fare clic col tasto destro sulla cartella **SV000514** e selezionare **Importa**.
3. Dalla procedura guidata **Importazione**, fare clic su **File di sistema**, quindi su **Avanti**.
4. Sfogliare la directory in cui sono stati estratti i file di scenario; per impostazione predefinita questa dovrebbe essere c:\wdsc\wdscsampl.
5. Fare clic su **SV000514**, quindi su **OK**.
6. Fare clic su **Seleziona tutto** per assicurarsi che siano selezionati tutti i componenti dei file di sistema SV000501.
7. Nel campo **Nella cartella**, immettere **SV000514** se il campo non è già riempito per impostazione predefinita.
8. Selezionare la casella di spunta **Sovrascrivi le risorse esistenti senza avvertire**.
9. Fare clic su **Fine**, quindi su **Sì** se viene visualizzata una finestra di dialogo che richiede di modificare la root contesto.
10. Ripetere i passi da 2 a 9 per SV001586, SV001585 e SV000618, assicurandosi di utilizzare i diversi nomi progetti.

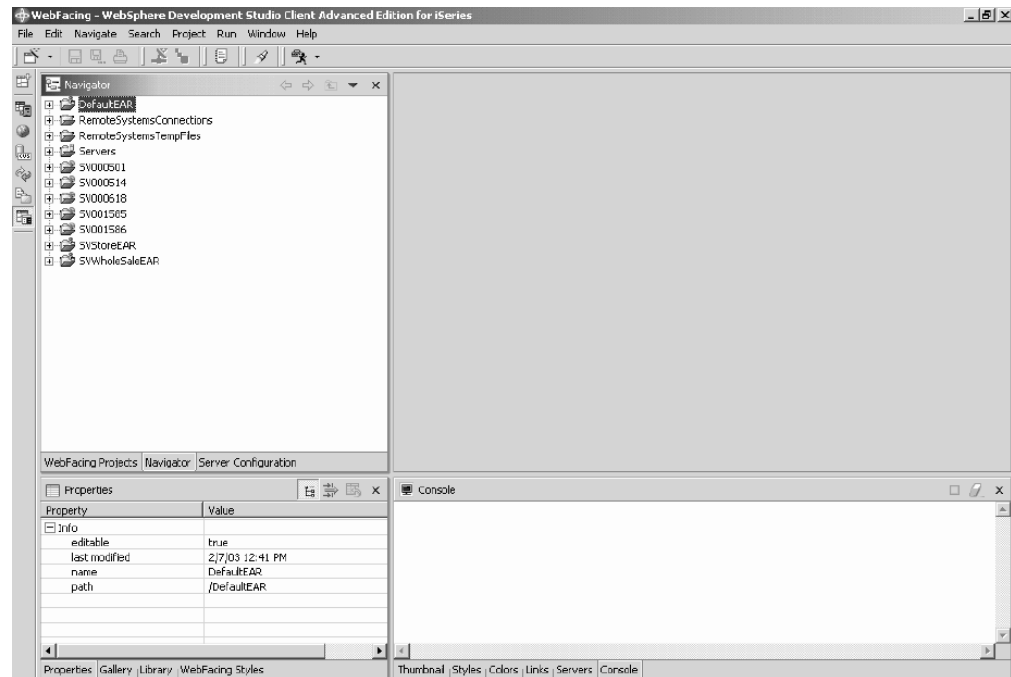
Importazione dei file del progetto WebFacing e creazione del progetto WebFacing

Il progetto SV000501 si differenzia dai quattro progetti Web appena creati, perché SV000501 è un progetto WebFacing. I progetti WebFacing sono diversi poiché includono file differenti e dispongono di una struttura unica, specifica per la gestione dei file DDS dal server iSeries.

Per importare i file e creare un progetto WebFacing:

1. Se non ci si trova nella vista Project Navigator, fare clic sul separatore **Project Navigator**.
2. Fare clic con il tasto destro nella vista e selezionare **Importa**.
3. Nella finestra **Importa**, selezionare **Progetti WebFacing** e fare clic su **Successivo**.
4. Sfogliare la directory in cui sono stati estratti i file di scenario; per impostazione predefinita questa dovrebbe essere c:\wdsc\wdscsampl.
5. Selezionare **SV000501** e fare clic su **OK**. Fare clic su **Avanti**.
6. Nell'area della finestra **Progetti WebFacing trovati**, selezionare la casella di spunta **SV000501**.
7. Nel campo **Enterprise Application Project (EAR)**, immettere **SVStoreEAR**.
8. Fare clic su **Fine**.

Dopo aver importato tutti i file, l'area di lavoro apparirà come segue:



Se si notano errori all'interno dei progetti importati, (indicati da segni x rossi), fare clic con il tastino destro su ciascun progetto e selezionare **Riesegui il build del Progetto**. In base al contenuto della propria area di lavoro, questo dovrebbe eliminare la maggior parte degli errori in modo da poter eseguire l'applicazione appropriatamente. Potrebbero esistere ancora errori causati dal fatto che parte del codice non è stato ancora migrato completamente al rilascio 5.1 di Development Studio Client. E' possibile continuare con gli esercizi in questo capitolo per essere introdotti al prodotto oppure è possibile procedere al capitolo successivo per cercare di effettuare il build delle applicazioni dalla bozza.

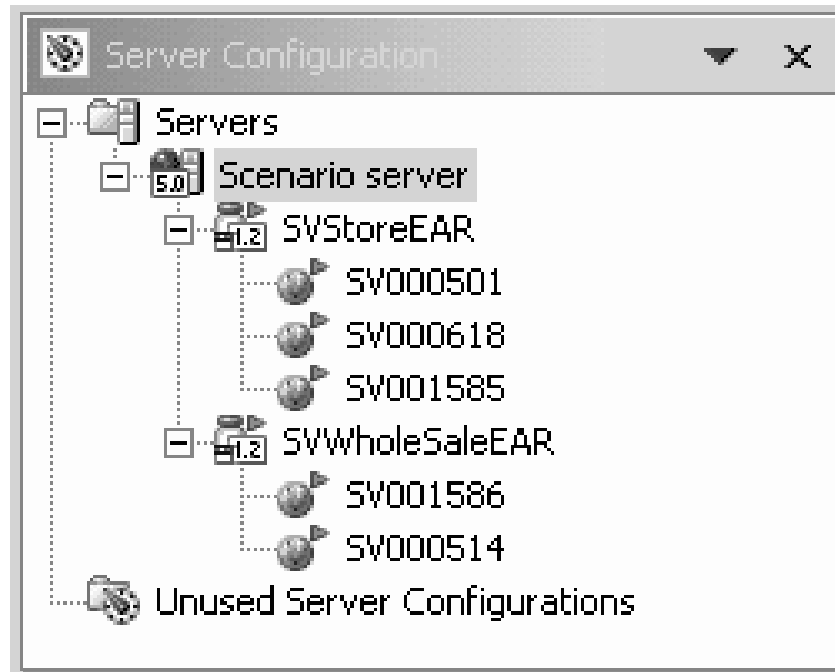
Configurazione del server

Dopo aver importato tutti i file origine, è necessario configurare il server dell'ambiente di verifica WebSphere in modo che riconosca le applicazioni SVWholesaleEAR e SVStoreEAR.

Per configurare il server dell'ambiente di verifica WebSphere:

1. Passare alla prospettiva Server. Dalla barra dei menu del workbench, fare clic su **Finestra > Apri prospettiva > Altro > Server**, quindi su **OK**.
2. Nella vista Configurazione server, fare clic col tasto destro del mouse su **Server** e selezionare **Nuovo > Server e Configurazione server**.
3. Nel campo **Nome server**, immettere Server scenario.
4. Nel campo **Cartella**, immettere Cartella scenario.
5. Nella casella **Tipo server**, espandere **WebSphere versione 5.0** e fare clic su **Ambiente di verifica**, se non è già selezionato per impostazione predefinita.
6. Fare clic su **Fine**, quindi su **Sì** se viene visualizzata qualsiasi messaggio.
7. Per aggiungere i file EAR alla configurazione del server:
 - a. Espandere **Server**.
 - b. Fare clic col tasto destro del mouse su **Server scenario** e selezionare **Aggiungi > SVWholeSaleEAR**.

- c. Espandere **Server**.
 - d. Fare nuovamente clic col tasto destro del mouse su **Server scenario** e selezionare **Aggiungi e Rimuovi progetti**.
 - e. Fare clic sul pulsante **Aggiungi tutti** per spostare SVStoreEAR e SVWholeSaleEAR sul lato destro della finestra.
 - f. Fare clic su **Fine**.
8. Nella vista Configurazione server, espandere **Server scenario**, quindi **SVStoreEAR** e **SVWholeSaleEAR** per visualizzare tutte le applicazioni elencate:



Per assicurarsi che il server abbia prelevato i progetti:

1. Nella vista Server, a destra della vista Configurazione server, (fare clic sul separatore **Server** se non si riesce a vedere la vista Server), fare clic col tasto destro del mouse su **Server scenario** e selezionare **Pubblica**. Viene visualizzata una casella di dialogo **Pubblicazione** che mostra la progressione della pubblicazione.
2. Fare clic su **OK** per pubblicare le applicazioni nell'ambiente di verifica di WebSphere Application Server.
3. Una volta terminata la pubblicazione, fare clic su **OK** nella casella di dialogo.
4. Nella stessa vista, fare nuovamente clic col tasto destro su **Server scenario** e fare clic su **Avvia**. Si apre la vista Console che mostra all'utente le attività del server. Il server è avviato quando viene visualizzato il messaggio "Server *nomeserver* aperto per e-business" alla base della vista. Scorrere la pagina per leggere tutti i messaggi.

Configurazione del progetto WebFacing

Prima di eseguire l'applicazione e di lavorare sul modulo SV000501, è necessario avviare il server WebFacing per eseguire l'applicazione e configurare il progetto in modo che possa utilizzare il server iSeries corretto.

Per avviare il server WebFacing:

1. Aprire un'emulazione 5250 e collegarsi con i propri ID utente e password.

2. Sulla riga comandi, immettere `strtcpsvr *webfacing`.
3. Alla base del pannello dovrebbe venire visualizzato il messaggio, "Server WEBFACING in fase di avvio".

Ora è necessario modificare le proprietà di WebFacing, di nuovo nel workbench di Development Studio Client.

1. Passare alla prospettiva Web (è possibile passare tra le prospettive facendo clic sulle icone raggruppate alla sinistra del video).
2. Fare clic sul separatore **Project Navigator** in modo tale che sia possibile esaminare le strutture del progetto.
3. Espandere **SV000501** e fare doppio clic su **Descrittore di distribuzione web**.
4. Fare clic sul separatore **Parametri**.
5. Fare clic su **WFDefaultHost**.
6. Nel campo **Valore**, cancellare il valore esistente ed immettere il nome del proprio server iSeries.
7. Dalla barra del menu del workbench, fare clic sull'icona Salva o su **File > Salva Descrittore di distribuzione Web**. Fare clic su **OK** se si riceve qualsiasi messaggio. Chiudere il file.

Definizione delle informazioni sul server

Ora è necessario definire le informazioni sull'Server iSeries per i cinque progetti, assicurandosi che essi siano configurati per essere eseguiti con l'Server iSeries con i propri ID utente e password. Per definire le informazioni sul server:

1. Assicurarsi di trovarsi nella prospettiva Web.
2. Nella vista Project Navigator, fare clic col tasto destro del mouse su **SV000501** e selezionare **Specifica configurazione runtime di iSeries Web Tools**.
3. Nel campo **Nome server iSeries**, cancellare il valore esistente ed immettere il nome dell'Server iSeries.
4. Cancellare gli altri valori esistenti ed immettere ID utente e password.
5. Fare clic su **Fine** (e fare nuovamente clic su **Fine** se necessario).
6. Ripetere i passi da 2 a 5 per gli altri quattro progetti: SV000514, SV000618, SV001585 e SV001586.

Esecuzione dell'applicazione nel workbench

A questo punto è possibile eseguire l'applicazione nell'ambiente di verifica di WebSphere nel workbench. Una volta verificata la funzionalità dell'applicazione, è possibile esportare il file EAR sul server iSeries e distribuirlo nel WebSphere Application Server iSeries.

Esecuzione dell'applicazione come cliente nel workbench

Come cliente, per quanto riguarda la vendita al dettaglio, si comincerà dalla pagina Web degli acquisti dove sarà possibile visionare gli articoli ed ordinare la quantità e la taglia.

Nota: è possibile che si verifichino problemi se si esegue l'applicazione dietro un firewall, perché il file `web.xml` ricercherà il seguente file:

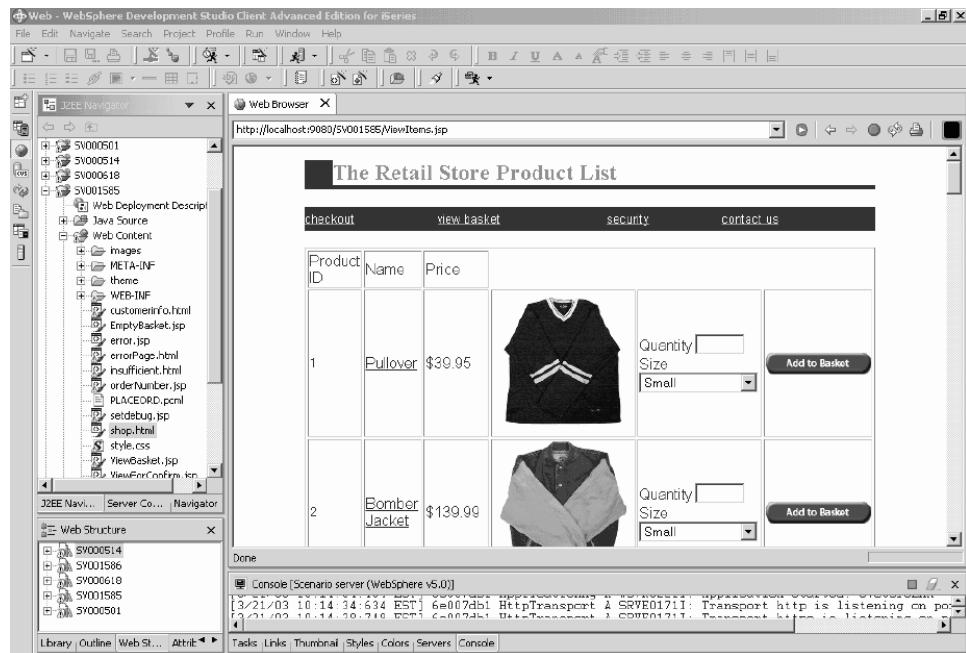
`http://java.sun.com/j2ee/dtds/web-app_2_2.dtd`. Per risolvere il problema, prima di eseguire l'applicazione, modificare l'istruzione DOCTYPE in tutti i file `web.xml` con:

```
!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"
```

dove x:/wdsc è la directory in cui è stato installato il prodotto. I file web.xml, generalmente, si trovano in *Nome progetto* > **Web Content** > **WEB-INF**. Una volta aperto il file, fare clic sul separatore **Origine** per modificare il file direttamente.

Per eseguire l'applicazione come cliente nel workbench:

1. Nella prospettiva Web, espandere **SV001585** > **Web Content**.
2. Fare clic col tasto destro su **shop.html** e selezionare **Esegui sul server**. Ciò avvia l'applicazione nel browser del workbench.
3. Fare clic su **Fine**.
4. Entrare nell'applicazione facendo clic sull'immagine che riproduce delle T-shirt.
5. Immettere i valori nella pagina successiva, fingendo di essere un cliente che ordina un Pullover o una Giacca Bomber, selezionando la taglia e la quantità, e aggiungendo gli articoli al proprio cestino:



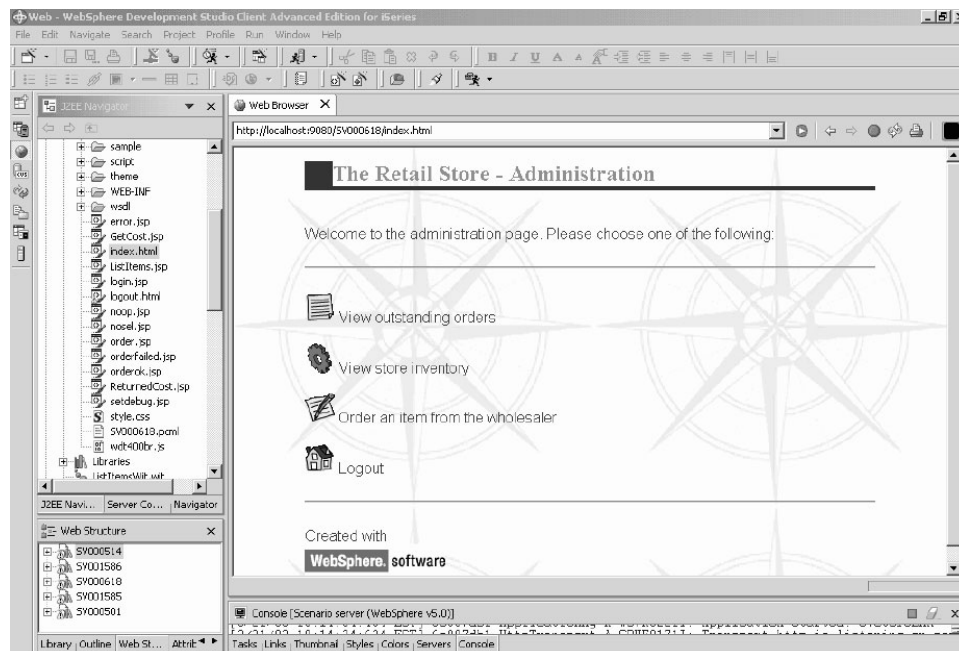
La pagina che visualizza il testo "l'ordine è stato elaborato" è l'ultima pagina dell'esempio. Se si è soddisfatti, è possibile chiudere il browser.

Esecuzione dell'applicazione come amministratore nel workbench

Come amministratore, per quanto riguarda la vendita al dettaglio, si comincerà dalla pagina Web dell'amministratore e si controlleranno gli ordini o l'inventario del magazzino. Per eseguire l'applicazione come amministratore nel workbench:

1. Nella prospettiva Web, espandere **SV000618** > **Web Content**.
2. Fare clic col tasto destro su **index.html** e selezionare **Esegui sul server**. Ciò avvia l'applicazione nel browser del workbench.
3. Fare clic sull'icona a sinistra di **Visualizza inventario magazzino** per visualizzare la pagina che segue, fingendo di essere un amministratore che

decide cosa comprare per il magazzino:



Fare clic sulle icone per visualizzare le pagine amministrative.

Distribuzione dell'applicazione scenario al WAS (WebSphere Application Server)

Una volta eseguita l'applicazione scenario iSeries nell'ambiente di verifica del workbench, è possibile distribuire l'applicazione al WebSphere Application Server, come se si stesse eseguendo l'applicazione in un situazione reale. Prima di distribuire l'applicazione, tuttavia, è necessario perfezionare alcuni particolari, come proteggere la pagina dell'amministratore e cambiare l'URL di un servizio Web in modo che l'applicazione punti alla destinazione giusta, come illustrato nelle sezioni che seguono.

Nota: La distribuzione su WebSphere Application Server è facoltativa; è comunque possibile continuare con il prossimo capitolo e completare i moduli senza verificare l'applicazione in WebSphere Application Server.

Configurazione di WebSphere Application Server

In una sezione precedente del capitolo si è discusso di "Esecuzione dell'applicazione nel workbench" a pagina 17. Dopo aver distribuito i file al server iSeries, è possibile eseguire l'applicazione sull'iSeries utilizzando WebSphere Application Server.

Per distribuire l'applicazione nel WebSphere Application Server (facoltativo) è necessario assicurarsi che:

- Sia stata creata o si abbia l'accesso a una istanza di WebSphere Application Server versione 4.0, versione 5.0 o WebSphere Application Server Express sull'Server iSeries e che l'istanza sia in esecuzione (solo se si desidera verificare la distribuzione su WebSphere Application Server).
- Si conoscano i numeri porta per le istanze HTTP e WebSphere Application Server sull'Server iSeries.

- Si disponga della console di gestione di WebSphere versione 4.0 o 5.0 installata sulla stazione di lavoro.

Nota: Per la versione 5.0 di WAS, la console si basa sul browser, quindi non è necessario disporre di una console sulla propria stazione di lavoro. Per ulteriori informazioni sulle differenti versioni di WAS, consultare la documentazione in linea.

Le applicazioni iSeries abilitate al web, utilizzano WebSphere Application Server per eseguire i servlet Java e le JSP (JavaServer Pages™) che comunicano tra il browser Web dell'utente e i programmi iSeries o i dati. Per rispondere alle richieste delle pagine HTML e dei file JSP dallo stesso sistema iSeries, è anche necessario un server HTTP su tale sistema. Si consiglia di utilizzare l'IBM HTTP Server supportato da Apache. E' possibile trovare la documentazione su come utilizzare questo server in: IBM HTTP Server per iSeries Documentation Center all'indirizzo http://publib.boulder.ibm.com/pubs/html/iseres_http/v5r1/index.htm.

WebSphere Application Server esegue le pagine JavaServer, i bean Java e i servlet Java generati per diversi processi. Le risorse della documentazione primaria per IBM WebSphere Application Server per iSeries e IBM WebSphere Administrative Console per iSeries sono disponibili sui seguenti siti Web:

- IBM WebSphere Application Server Versione 4.0 Advanced Edition per iSeries all'indirizzo <http://publib.boulder.ibm.com/was400/40/AE/english/docs/>
- IBM WebSphere Application Server Versione 4.0 Advanced Single Server Edition per iSeries all'indirizzo <http://publib.boulder.ibm.com/was400/40/AEs/english/docs/>

Si consiglia fortemente di leggere la documentazione per IBM WebSphere Application Server, in particolare, le sezioni *Moduli J2EE*, *Installazione di WebSphere Application Server* e *Impostazione di più istanze di WAS (WebSphere Administrative Server)*. E' necessario almeno eseguire i passi descritti nel collegamento *Installazione*.

Utilizzare la mappa del sito per trovare le informazioni su come installare, configurare e ottenere le PTF necessarie per la console di gestione di WebSphere.

Protezione della pagina dell'amministratore

Poiché la pagina dell'amministratore, `index.html`, dovrebbe essere accessibile solo da parte di persone autorizzate, è necessario proteggerla in modo appropriato. E' possibile effettuare ciò in modo programmatico, nella logica dell'applicazione Web, oppure utilizzando la funzione di sicurezza di WebSphere. In questo scenario, abbiamo utilizzato la sicurezza di WebSphere per proteggere la pagina. Notare che è stato utilizzato WebSphere Application Server V4.0 Advanced Edition. E' possibile trovare informazioni su WebSphere Application Server sui relativi siti Web:

- IBM WebSphere Application Server Versione 4.0 Advanced Edition per iSeries all'indirizzo <http://publib.boulder.ibm.com/was400/40/AE/english/docs/>
- IBM WebSphere Application Server Versione 4.0 Advanced Single Server Edition per iSeries all'indirizzo <http://publib.boulder.ibm.com/was400/40/AEs/english/docs/>
- Pagina RedBooks for WebSphere: Ricerca RedBooks for WebSphere Application Server Versione 5.0 e WebSphere Application Server Express Edition Versione 5.0 all'indirizzo <http://publib-b.boulder.ibm.com/redbooks.nsf/portals/WebSphere>

Se si sta utilizzando una versione differente di WebSphere Application Server, fare riferimento alla documentazione di quella versione per quanto riguarda la protezione delle risorse Web.

E' possibile configurare la sicurezza delle risorse Web, come le pagine Web e i servlet, all'interno di Development Studio Client o nell'AAT (Application Assembly Tool-Strumento di assemblaggio dell'applicazione). Per questo scenario verrà utilizzato Development Studio Client.

Per rivedere la configurazione della sicurezza e le proprietà per questa applicazione Web:

1. In Development Studio Client, passare alla prospettiva Web.
2. Nella vista Navigator, espandere **SV000618 > Web Content > WEB-INF**.
3. Fare doppio clic su **web.xml** per aprire la vista web.xml.
4. Fare clic sul separatore **Sicurezza**.
5. Per proteggere index.html, la pagina dell'amministratore, è stata definita anche una limitazione della Sicurezza. All'inizio della vista, fare clic su **Limitazioni di sicurezza**.
6. Fare clic sulla prima istanza di **SecurityConstraint** nell'elenco.
7. Sulla destra, fare clic su **AdminPage**.
8. Fare clic su **Modifica** per richiamare la casella di dialogo **Raccolte di risorse Web**. Notare che i metodi GET e POST per index.html sono già selezionati.
9. Fare clic su **OK**.

Nella sezione **Ruoli di sicurezza**, notare il ruolo di sicurezza definito denominato "Administrator". Durante la distribuzione, i singoli utenti vengono assegnati a tale ruolo, quindi, viene dato loro l'accesso alla pagina index.html. Nella sezione **Ruoli autorizzati**, notare che, al ruolo di Administrator, è stato fornito l'accesso a questa limitazione di sicurezza. Con questo tipo di sicurezza in atto, solo agli utenti assegnati al ruolo Administrator è concesso l'accesso alla pagina index.html, previa fornitura delle credenziali appropriate, cioè ID utente e password. Quando una risorsa è protetta, WebSphere Application Server tenta innanzitutto di autenticare l'utente. L'autenticazione viene effettuata tramite i certificati o tramite la richiesta di ID utente e password. La richiesta può avvenire tramite una tipica finestra di autenticazione o attraverso un modulo personalizzato.

In questo scenario, è stata progettata una pagina di collegamento personalizzata denominata login.jsp. Per configurare la richiesta di autenticazione, selezionare il separatore **Pagine** nella vista web.xml. Nella sezione **Collegamento**, notare che **Modulo** è già selezionato come metodo di Autenticazione. Notare anche che il nome della pagina di collegamento è login.jsp. Quando il collegamento ha esito negativo viene visualizzata la pagina Errore. In tal caso, l'applicazione visualizza nuovamente la pagina login.jsp.

Creazione dei file EAR per la distribuzione su WebSphere Application Server iSeries

E' necessario creare i file EAR per distribuire l'applicazione su WebSphere Application Server iSeries. Un file EAR è un file JAR (Java Archive) standard con un'estensione .ear. Tali file possono contenere più progetti Web ed è possibile utilizzarli per comprimere e distribuire le applicazioni Web in WAS (WebSphere Administrative Server).

Per creare i file EAR.

1. Passare alla prospettiva Web.
2. Nella vista Navigator, fare clic col tasto destro del mouse su **SVStoreEAR** e selezionare **Esporta**.
3. Nella finestra di esportazione, fare clic su **File EAR**, quindi su **Avanti**.
4. Alla richiesta **Dove si desidera esportare le risorse?**, fare clic su **Sfoggia** e andare alla directory dell'IFS iSeries in cui è possibile conservare i file EAR. (E' necessario che la propria unità di rete sia stata collegata a un IFS iSeries).
5. Immettere **SVRetailStorEAR.ear** nel campo **Nome file** e fare clic su **Apri**.
6. Fare clic su **Fine**.
7. Ripetere i passi da 2 a 6 per **SVWholeSaleEAR**.

Distribuzione dei file EAR su WebSphere Application Server iSeries

Una volta creati i file EAR, è possibile distribuirli su WebSphere Application Server.

1. Aprire la console di gestione di WebSphere.
2. Fare clic col tasto destro del mouse su **Applicazioni Enterprise** e selezionare **Installa applicazione enterprise**.
3. Selezionare il pallino **Installare applicazione (*ear)**.
4. Fare clic sul pulsante **Sfoggia** in alto (quello in basso non è disponibile).
5. Passare alla directory IFS in cui sono stati esportati i file EAR.
6. Selezionare **SVRetailStorEAR.ear**.
7. Immettere "RETAILSTOR" nel campo **Nome applicazione**.
8. Fare clic su **Avanti** ed immettere l'ID utente del server iSeries per il ruolo di amministratore.
9. Fare clic su **Avanti** ripetutamente, fino a quando non si arriva alla pagina intitolata **Selezione degli host virtuali per i moduli web**.
10. Per tutti e tre i moduli Web, fare clic su **Seleziona host virtuale** e selezionare il server virtuale preferito dall'elenco a discesa. (Se non si è sicuri della scelta da effettuare, utilizzare **predefinito** o **default_host**.)
11. Fare clic su **Avanti**.
12. Per tutti e tre i moduli Web, fare clic su **Seleziona server** e selezionare il server che si desidera utilizzare. (Se non si è sicuri della scelta da effettuare, utilizzare **Server predefinito**.)
13. Fare clic su **Avanti**.
14. Fare clic su **Fine**, quindi su **OK** nella casella di dialogo.
15. Fare nuovamente clic col tasto destro del mouse su **Applicazioni Enterprise** e selezionare **Installa applicazione enterprise**.
16. Fare clic sul pulsante **Sfoggia** in basso (quello in alto non è disponibile).
17. Passare alla directory IFS in cui sono stati collocati i file EAR.
18. Selezionare **SVWholeSaleEAR.ear**.
19. Immettere "WHOLESALE" nel campo **Nome applicazione**.
20. Fare clic su **Avanti** ripetutamente, fino a quando non viene abilitato il pulsante **Fine**.
21. Fare clic su **Fine**, quindi su **OK** nella casella di dialogo.

Nota: La visualizzazione del messaggio di conferma potrebbe richiedere alcuni minuti.

Esecuzione dell'applicazione in WebSphere Application Server

Continuando nel ruolo di cliente, per eseguire l'accesso alla vendita al dettaglio, immettere la seguente URL in un browser Web:

`http://NomeHostiSeries:NumeroPortaHTTP/SV001585/shop.html`

Continuando nel ruolo di amministratore, per eseguire l'accesso alla vendita all'ingrosso, immettere la seguente URL in un browser Web:

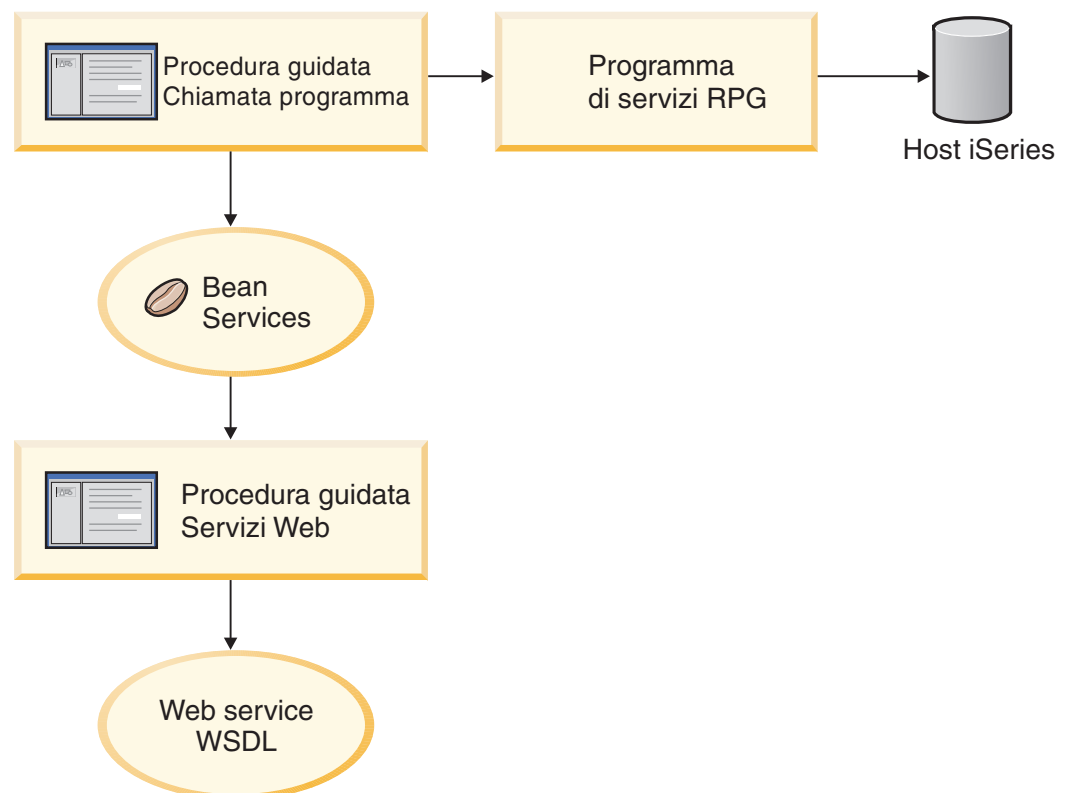
`http://NomeHostiSeries:NumeroPortaHTTP/SV000618/index.html`

Se non si conosce il numero della porta HTTP, chiederlo al proprio amministratore WebSphere Application Server.

Capitolo 4. Modulo 1 dettagliato: Creare un servizio Web che restituisca i prezzi dei prodotti (SV000514)

Introduzione

In questo modulo, è possibile creare un servizio Web da un programma RPG presente sul server iSeries, per visualizzare i prezzi dei prodotti per il proprio inventario. Creare, innanzitutto, un programma di servizio RPG con una procedura che può richiamare il costo di un articolo dal database iSeries, una volta fornito il numero dell'articolo. Utilizzare la procedura guidata Chiamata programma da Strumenti di sviluppo Java iSeries per richiamare il programma RPG e creare un bean di servizi. Utilizzare, poi, la procedura guidata Servizi Web per creare un servizio Web e utilizzare l'esempio generato per verificare il servizio Web.



Prima di cominciare

E' possibile completare l'esercitazione solo se si rispettano i prerequisiti che seguono. I prerequisiti vengono discussi in maggiori dettagli in Capitolo 3, "Esecuzione dello scenario", a pagina 9.

- L'utente ha accesso TCP/IP al server iSeries.
- Sono state applicate le più recenti PTF del server iSeries. Consultare la pagina di supporto per informazioni su PTF e Service pack:
<http://www.ibm.com/software/awdtools/wdt400/support/>
- Occorre aver avviato i server iSeries con il comando STRTCPSVR *ALL

- Occorre aver avviato il server WebFacing con il comando STRTCPSVR *WEBFACING
- Sono state completate tutte le attività presenti in Capitolo 3, "Esecuzione dello scenario", a pagina 9, (salvo le attività WebSphere Application Server facoltative, che non sono necessarie per verificare l'applicazione nel workbench).

Creazione di un nuovo progetto Web

Il primo passo nella creazione di questo servizio Web consiste nel creare un nuovo progetto Web che contenga le informazioni.

1. Dall'IDE workbench, passare alla prospettiva Web o aprire la prospettiva Web facendo clic su **Finestra > Apri prospettiva > Altro > Web > OK**.
2. Fare clic su **File > Nuovo > Progetto web dinamico**.
3. Nel campo **Nome progetto**, immettere Project514.
4. Controllare la casella **Opzioni avanzate di configurazione** e fare clic su **Successivo**.
5. Fare clic sul pulsante **Nuovo** accanto al campo **Progetto EAR**.
6. Nel campo **Nome progetto**, immettere Project514EAR e fare clic **Fine** su entrambe le caselle di dialogo.

Ora si noterà che i progetti Project514 e Project514EAR sono stati aggiunti all'area di lavoro nella vista Project Navigator.

Definizione delle informazioni del server iSeries

Una volta creato il progetto Web, è necessario definire quale server iSeries il progetto utilizzerà per ottenere le informazioni.

1. Fare clic col tasto destro del mouse sul progetto **Project514** e selezionare **Specifica configurazione runtime di iSeries Web Tools**.
2. Immettere il nome del server iSeries in cui si trova la libreria WHOLESALE ripristinata, ad esempio, PROD400.
3. Immettere ID utente e password per questo server iSeries.
4. Digitare Wholesale nel campo **Libreria** e fare clic su **Aggiungi**.
5. Fare clic su **Fine**.

Creazione del programma di servizio RPG

Si desidera che l'applicazione richiami il prezzo di un articolo, una volta fornito il relativo numero. Tale operazione viene gestita da un programma di servizio RPG che contiene una procedura chiamata QryProdCost. La libreria WHOLESALE contiene un programma di servizio RPG denominato CWWSSRV. Questo programma contiene una procedura QryProdCost che riconosce i numeri degli articoli come immissione, apre il file dell'Inventario all'interno della libreria WHOLESALE, richiama il prezzo dal database dell'inventario e lo restituisce. Per soddisfare questa attività, l'interfaccia dispone di due parametri, uno per il numero dell'articolo e l'altro per il prezzo. Se non viene rilevato il numero dell'articolo o il prezzo, il programma RPG restituisce un messaggio all'interfaccia.

Per creare questo servizio Web, occorre creare innanzitutto un bean Java con la procedura guidata Chiamata programma, per richiamare la procedura RPG QryProdCost. Utilizzare, quindi, i servizi Web per abilitare la procedura RPG come servizio Web tramite il bean Java.

Per creare il bean Java:

1. Nella prospettiva Web, fare clic col tasto destro del mouse su **Project514** e selezionare **Nuovo > Altro**.
2. Selezionare **iSeries > Java** dal pannello sinistro della finestra, quindi selezionare **Bean chiamata programma** sul pannello di destra.
3. Fare clic su **Avanti** per richiamare la procedura guidata Chiamata programma.
4. Nel campo **Nome bean Java**, sotto il titolo **Aggiungi programma**, immettere **Inventario**.
5. Nel campo **Oggetto programma**, immettere **CWWSSRV**, il nome del programma di servizio RPG.
6. Nel campo **Libreria**, immettere **WHOLESALE**.
7. Dall'elenco a discesa **Tipo programma**, selezionare ***SRVPGM**.
8. Nel campo **Punto di immissione**, immettere **QryProdCost**.
9. Fare clic su **OK** per aggiungere la definizione programma.

Creazione dei parametri e generazione del bean Java

Una volta specificato il programma, è possibile aggiungere i parametri. Il programma RPG CWWSSRV contiene i due parametri che seguono:

- Parametro Numero articolo - Il programma utilizza questo numero per trovare l'articolo nel database.
- Parametro Costo articolo - Il programma utilizza questo numero per trovare il costo dell'articolo nel database.

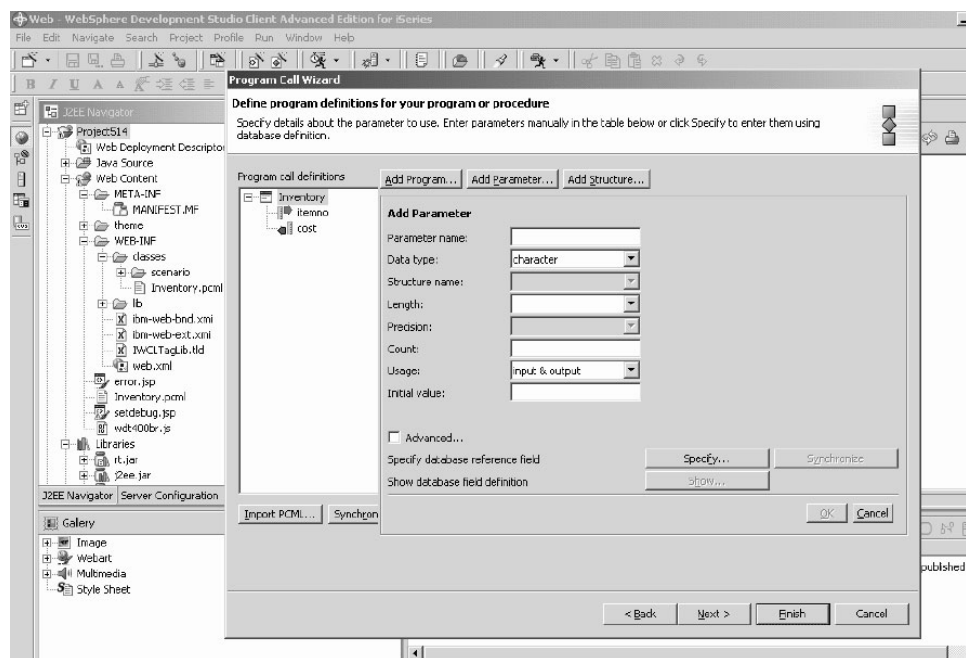
Per aggiungere il parametro relativo al numero dell'articolo:

1. Nel pannello di sinistra della procedura guidata Chiamata programma, fare clic sulla definizione della chiamata programma **Inventario** per selezionarla. Questa azione ripopola i campi sul lato destro della pagina.
2. Fare clic su **Aggiungi parametri**.
3. Nel campo **Nome parametro**, immettere **itemno**.
4. Dall'elenco a discesa **Tipo dati**, selezionare **decimale compresso**.
5. Nel campo **Lunghezza**, immettere **5**.
6. Nel campo **Precisione**, immettere **0**.
7. Dall'elenco a discesa **Utilizzo**, selezionare **input**.
8. Fare clic su **OK** per aggiungere questo parametro. Nel pannello di sinistra, verificare che **itemno** appaia sotto **Inventario**. Ora è possibile aggiungere il secondo parametro.

Per immettere il parametro relativo al costo dell'articolo:

1. Nel pannello di sinistra della procedura guidata Chiamata programma, fare clic sulla definizione della chiamata programma **Inventario** per selezionarla. Questa azione ripopola i campi sul lato destro della pagina.
2. Fare clic su **Aggiungi parametri**.
3. Nel campo **Nome parametro**, immettere **cost**.
4. Dall'elenco a discesa **Tipo dati**, selezionare **decimale compresso**.
5. Nel campo **Lunghezza**, immettere **7**.
6. Nel campo **Precisione**, immettere **2**.
7. Dall'elenco a discesa **Utilizzo**, selezionare **output**.
8. Fare clic su **OK** per aggiungere questo parametro. Nel pannello di sinistra, verificare che **cost** appaia sotto **Inventario**. A questo punto, la procedura guidata dovrebbe apparire come di seguito. Tener presente che l'icona a sinistra

del parametro indica se esso è di tipo immissione, immissione & emissione o emissione:



Per definire il nome pacchetto per questi parametri, nella procedura guidata Chiamata programma:

1. Fare clic su **Avanti**.
2. Nel campo **Pacchetto**, immettere scenario come nome del pacchetto. Negli altri campi lasciare i valori predefiniti.
3. Eliminare il contenuto della casella di spunta **Applicazione Java**.

Nota: Controllare l'elenco dei file sotto "Questi file verranno generati dalla procedura guidata" e verificare che il nome del bean Java generato sia InventoryServices.java.

4. Fare clic su **Fine** per generare i file.

Creare un servizio Web dal bean Java

Il passo successivo alla creazione del bean Java che richiama il programma RPG consiste nella conversione del bean in un servizio Web in modo tale che altri programmi possano accedere allo stesso programma RPG su Internet.

Una volta creato il servizio Web, la procedura guidata Servizi Web genera i file WSDL che vengono distribuiti agli utenti che necessitano di utilizzare il servizio Web. Per creare il servizio:

1. Nella vista Navigator della prospettiva Web, espandere **Project514 > Origine Java > scenario**.
2. Fare clic col tasto destro del mouse su **InventoryServices.java** e selezionare **Nuovo > Altro**.
3. Nella finestra Nuovo, fare clic su **Servizi Web**, fare clic su **Servizio Web** sul lato destro della finestra, quindi su **Avanti**.
4. Selezionare le caselle di spunta **Generare un proxy** e **Verificare il proxy generato**.

5. Fare clic su **Successivo** tre volte e notare la generazione dei file necessari. All'apparizione della finestra Identità di servizio Java Bean, fare clic su **Fine**. Potrebbe essere necessario fare clic su **Fine** una seconda volta se si notano errori. Se non si riesce a finire la procedura guidata, fare clic su **Cancella**, la maggior parte dei file necessari sarà comunque creata.

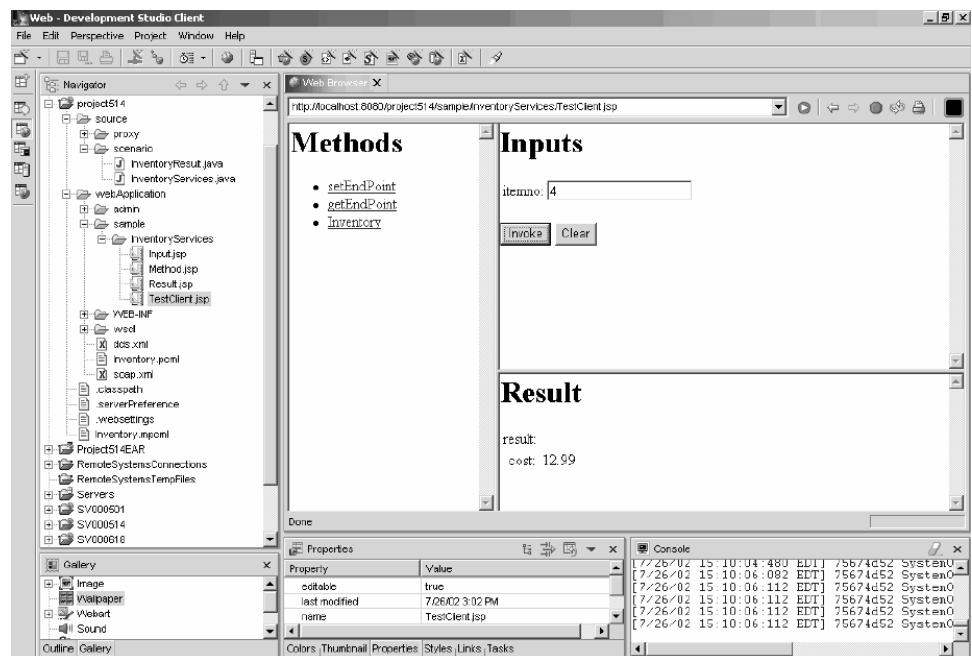
Verifica dell'esempio

Al momento della creazione del servizio Web, una delle istruzioni era di richiedere che venisse generato un esempio. Poiché tale opzione è stata selezionata, la procedura guidata Servizi Web ha creato un nuovo progetto con pagine di test da utilizzare per verificare il servizio Web. Nella vista **Navigator**, verificare che il nome di questo nuovo progetto sia **Project514Client**. Notare anche che lo strumento ha aperto automaticamente **TestClient.jsp** sul lato destro del workbench. Questo file si trova in **Project514Client > Web Content > sample > InventoryServices**.

Per verificare l'esempio:

1. Nella pagina caricata, scorrere il pannello dei metodi fino alla fine. Fare clic sul metodo **Inventory (java.math.BigDecimal)**.
2. Nel campo **itemno** immettere 4 e fare clic su **Richiama**.
3. Verificare che l'emissione restituita sia:

risultato:
costo: 12.99

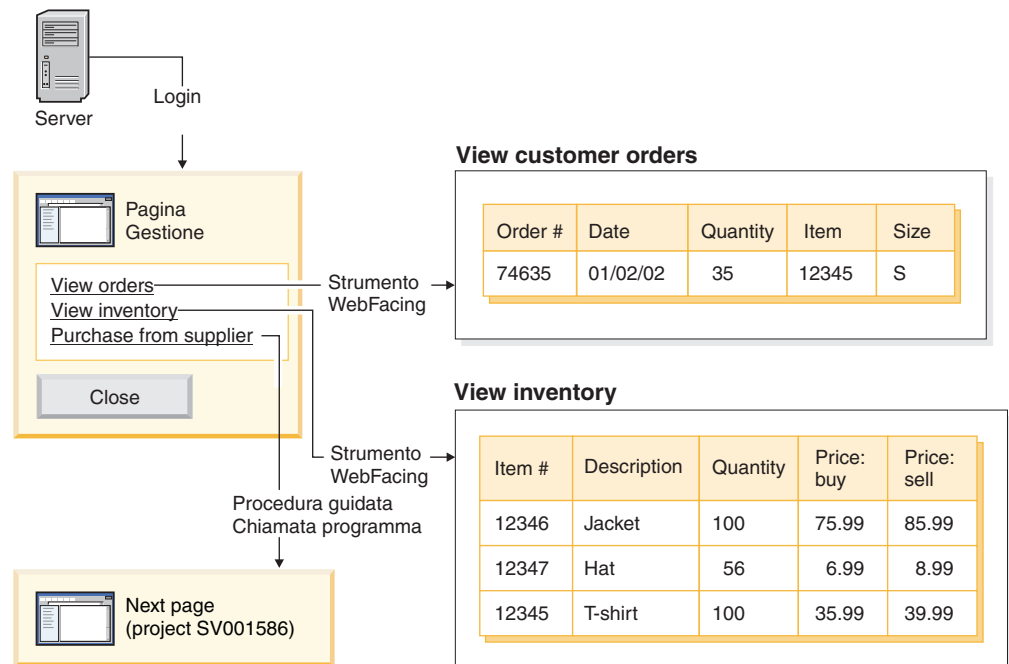


Una volta confermato il risultato, il modulo è completo. E' stato creato un servizio web che restituisce i prezzi nell'interfaccia dell'applicazione.

Capitolo 5. Modulo 2 dettagliato: Creare un'interfaccia per visualizzare l'inventario e ordinare le voci (SV000501)

Introduzione

In questo modulo, si crea un'interfaccia che visualizza le opzioni dell'operazione dell'amministratore utilizzando IBM WebFacing Tool. Dopo essersi collegato al server iSeries, l'interfaccia visualizza l'inventario disponibile ed è possibile controllare gli ordini esistenti. In qualità di amministratore, è possibile acquistare merci dal fornitore all'ingrosso.



In questo progetto, si lavora con due programmi e due file di visualizzazione ripristinati al proprio server iSeries. I nomi dei programmi sono ViewInventory e ViewOrder. I programmi utilizzano i due seguenti file di visualizzazione: ORDERDSP e QUERY. Questi due file contengono impostazioni web personalizzate per i file JSP generati da WebFacing Tool, e sono utilizzate per immagini ed hyperlink. L'impostazione dell'immagine Web consente di utilizzare il contenuto di un campo per generare nomi di file di immagine e visualizzare l'immagine nel file JSP. L'impostazione per l'hyperlink Web consente di richiamare un'altra applicazione web quando si fa clic sull'immagine nel file JSP. E' possibile utilizzare il CODE Editor o il CODE Designer per controllare il codice origine del file di visualizzazione, per determinare come specificare le impostazioni web.

Prima di cominciare

E' possibile completare l'esercitazione solo se si rispettano i prerequisiti che seguono. I prerequisiti vengono discussi in maggiori dettagli in Capitolo 3, "Esecuzione dello scenario", a pagina 9.

- L'utente ha accesso TCP/IP al server iSeries.

- Occorre aver avviato i server iSeries con il comando STRTCPSVR *ALL
- Occorre aver avviato il server WebFacing con il comando STRTCPSVR *WEBFACING
- Sono state completate tutte le attività presenti in Capitolo 3, "Esecuzione dello scenario", a pagina 9, (salvo le attività WebSphere Application Server facoltative, che non sono necessarie per verificare l'applicazione nel workbench).

Creazione di un progetto WebFacing

Innanzitutto, è necessario creare un progetto WebFacing e specificare i relativi comandi CL. Per creare il progetto WebFacing:

1. Nel workbench, passare alla prospettiva WebFacing facendo clic su una delle icone della prospettiva al di sotto del lato sinistro dell'area di lavoro o facendo clic su **Finestra > Apri Prospettiva > Altro > WebFacing**, quindi fare clic su **OK**.
2. Creare un nuovo progetto WebFacing facendo clic su **File > Nuovo progetto > WebFacing**.
3. Denominare il progetto Project501.
4. Selezionare il pallino **Esistente** accanto al **Progetto Applicazione Enterprise**.
5. Immettere SVStoreEAR nel valore all'interno del campo **Nome Progetto esistente**; da notare che il valore è sensibile al maiuscolo e al minuscolo. Fare clic su **Successivo**.
6. Per l'elenco a discesa **Livello J2EE**, selezionare **1.3**. Fare clic su **Successivo**.
7. Nel campo **Connessione**, il nome del server iSeries dovrebbe essere riempito automaticamente. In caso contrario, fare clic su **Nuovo** ed immettere il nome del proprio server iSeries nella casella di dialogo e fare clic su **Fine**.
8. Fare clic su **Aggiorna elenco DDS** ed immettere la password e l'ID utente nella casella a comparsa per aggiornare l'elenco.
9. Nell'elenco di librerie generato, espandere **RETAILSTOR**.
10. Fare clic su **QDDSSRC**, quindi fare clic sulla freccia destra (>>) per spostare i file.
11. Fare clic due volte su **Successivo**, fino ad arrivare alla pagina **Specifica comandi CL**.

Ora è necessario aggiungere un comando ed una etichetta comando per l'administrator:

12. Nel campo **Comando CL**, immettere la call `call viewinvent`.
13. Nel campo **Etichetta comando**, cancellare qualsiasi valore esistente ed immettere `Visualizza inventario`.
14. Selezionare il pallino **Collegati con i valori specificati**.
15. Fare clic su **Aggiungi** ed annotare l'aggiunta nell'elenco alla fine della finestra. Ora è necessario aggiungere un secondo comando ed una etichetta comando per il cliente:
16. Nel campo **Comando CL**, cancellare il valore esistente ed immettere `call vieworder`.
17. Nel campo **Etichetta comando**, cancellare il valore esistente ed immettere `Visualizza ordini`.
18. Selezionare il pallino **Richiesta di collegamento**.
19. Fare clic su **Aggiungi** ed annotare l'aggiunta nell'elenco alla fine della finestra. Fare clic su **Successivo**.

20. Sulla finestra **Scegli uno stile web**, scorrere gli stili per vedere quale è disponibile. Poiché è necessario recuperare informazioni da questo stile in avanti, per questo modulo selezionare **avenue**. Fare clic su **Successivo**.
21. Verificare che la casella di spunta **No. Creare il progetto ora** sia selezionata e fare clic su **Fine**.
22. Se compare il messaggio Riparazione Configurazione Server, fare clic su **OK**. Questo messaggio conferma che si desidera aggiungere il progetto al file SVStoreEAR.

Poiché si sta costruendo Project501 in modo speculare al progetto SV000501 completo, è ora necessario copiare alcuni dei file di immagine da SV000501 a Project501, in modo da consentire una corretta visualizzazione del Project501. (Tuttavia, se lo si desidera, è possibile aggiungere le proprie immagini per l'interfaccia.) Per copiare le immagini:

1. Passare alla prospettiva Web.
2. Nella vista Navigator, espandere **SV000501 > Web Content > immagini**.
3. (Facoltativo) Espandere **generato** per prendere nota di tutte le immagini che si sta importando nel progetto.
4. Fare clic con il tastino destro del mouse su **generato** e selezionare **Copia**.
5. Scorrere nella vista Navigator fino a vedere nuovamente **4Project501**.
6. Espandere **Project501 > Web Content**.
7. Fare clic con il tastino destro del mouse sulla cartella **immagini** e selezionare **Incolla** dal menu a comparsa.
8. Ora, esiste la cartella **generato** per Project501. Espandere **generato** per prendere nota delle immagini aggiunte.

Conversione dell'origine DDS

Una volta creato il progetto, è possibile convertire i file di visualizzazione DDS in file JSP per la propria pagina Web. Quando si convertono i file di visualizzazione DDS, WebFacing Tool genera file JSP e XML che sostituiscono il codice DDS e rendono possibile l'accesso al Web. I file generati conservano i dati per il formato di registrazione o controllano l'aspetto, visualizzano la versione web dello schermo, richiamano dati e gestiscono gli errori di immissione. Inoltre, la procedura guidata crea una home page per l'applicazione per avviare la versione abilitata a Web del programma.

E' innanzitutto necessario alterare l'origine per consultare il server iSeries corretto. In modo più specifico, è necessario effettuare una modifica in uno dei propri file di visualizzazione DDS per fare funzionare il collegamento all'applicazione. E' necessario modificare il nome predefinito, SV000501, nel nome del progetto, Project501:

1. Ritornare alla prospettiva WebFacing e alla vista Progetti WebFacing se non si è aperta per impostazione predefinita.
2. Espandere **Project501 > DDS**.
3. Fare doppio clic sulla seconda voce
`<iSeriesserver>RETAILSTOR/QDDSSRC(QUERY)`
4. Scorrere fino alla riga 16 circa, dove si vede la riga:
`A*%WB 12 FLD 1 next ('/SV000501/DetailPage.do?PRODNO=&{PRODNO}')`
5. Cancellare SV000501 ed immettere Project501.
6. Salvare e chiudere il file.

Nota: Se in un momento dato si desidera eseguire il progetto SV000501 di nuovo, è necessario riportare il valore Project501 a SV000501, poiché questo file esiste sul server iSeries ed è possibile accedervi da entrambi i progetti.

Per convertire l'origine DDS:

1. Nella prospettiva WebFacing, fare clic sul separatore **Progetti WebFacing** per ritornare alla vista Progetti WebFacing se non è già aperta per impostazione predefinita.
2. Espandere **Project501 > DDS**.
3. Selezionare *iSeriesserver* > **RETAILSTOR/QDDSSRC(ORDERDSP)** e *iSeriesserver* > **RETAILSTOR/QDDSSRC(QUERY)** facendo clic sul primo e poi tenendo premuto il tasto Shift mentre si fa clic sul secondo.
4. Fare clic e selezionare **Converti** per avviare la conversione. Potrebbe essere necessario immettere l'ID utente e la password se ci si era disconnessi dal server iSeries.

Configurazione del supporto UTF-8 in workbench — Solo per gli utenti di WAS Versione 4.0

Le applicazioni di IBM WebFacing Tool supportano la visualizzazione di più lingue sullo schermo. Dal momento che le lingue utilizzano diverse serie di caratteri, i flussi di dati fra il browser e WebSphere Application Server vengono codificati con UTF-8. Affinché IBM WebFacing Tool funzioni correttamente, è necessario configurare il supporto UTF-8 nel file di proprietà dell'applicazione nel workbench.

Nota: Questa sezione si applica solo agli utenti di WAS 4.0. WAS versione 5.0 esegue questa attività automaticamente.

Per configurare il supporto UTF-8:

1. Passare alla prospettiva Web (è possibile passare tra le prospettive facendo clic sulle icone raggruppate alla sinistra del video).
2. Fare clic sul separatore **Project Navigator** in modo tale che sia possibile esaminare le strutture del progetto.
3. Espandere **Project501** e fare doppio clic su **Descrittore di distribuzione web**.
4. Fare clic sul separatore **Ambiente**.
5. Fare clic sul pulsante **Aggiungi**.
- 6.
7. Passare alla prospettiva Server facendo clic sulla sua icona sulla barra delle attività in alto a sinistra o fare clic su **Prospettiva > Apri > Altro > Server** dalla barra del menu, quindi fare clic su **OK**.
8. Nella vista Navigator, espandere la cartella **Server**.
9. Fare doppio clic su **defaultInstance.wsi** per aprirlo nell'editor predefinito.
10. Fare clic sul separatore **Ambiente**, quindi sul pulsante **Aggiungi**.
11. Apparirà un valore modificabile, "(Nuova Variabile)". Cancellare questa stringa predefinita ed immettere: `client.encoding.override`.
12. Nel campo **Valore**, immettere UTF-8.
13. Fare clic sull'icona Salva oppure fare clic su **File > Salva descrittore di distribuzione Web**.

Configurazione del supporto UTF-8 per WebSphere Application Server

(Facoltativo) Se si desidera distribuire la propria applicazione iSeries a WebSphere Application Server, è necessario configurare il supporto UTF-8 su WebSphere Application Server come anche nel workbench.

Per configurare il supporto UTF-8 in WebSphere Application Server 4.0 Advanced Edition

1. Avviare la console amministrativa di WebSphere.
2. Espandere l'icona **Nodi** ed espandere **Nome nodo > Server dell'applicazione > Server predefinito**.
3. Selezionare il separatore **Impostazioni JVM** e fare clic sul pulsante **Impostazioni avanzate JVM** per aprire la casella di dialogo **Impostazioni avanzate JVM**.
4. Nel campo **Argomento riga comandi**, immettere:
`-Dclient.encoding.override=UTF-8`
5. Fare clic su **OK** e poi su **Applica** sotto il separatore **Impostazioni JVM**.
6. Per effettuare tale modifica nelle applicazioni WebSphere, arrestare il server predefinito e poi riavviarlo. Per arrestare il server, fare clic col tastino destro del mouse su **Server predefinito** e selezionare **Arresto**. Dopo aver completato questo processo, fare clic col tastino destro del mouse su **Server predefinito** e selezionare **Avvio**.

Per configurare il supporto UTF-8 in WebSphere Application Server 4.0 Advanced Single Server Edition

1. Avviare la console amministrativa di WebSphere.
2. Nella console amministrativa basata sul browser, espandere l'icona **Nodi** ed espandere **Nome nodo > Server dell'applicazione > Server predefinito > Definizione del processo > Impostazioni JVM**.
3. Passare alla sezione Impostazioni avanzate della pagina Impostazioni JVM e fare clic sul link Proprietà di sistema. Viene visualizzata la pagina Proprietà di sistema.
4. Fare clic su **Nuovo** per aggiungere una nuova Proprietà di sistema.
5. Nel campo **Nome**, immettere `client.encoding.override`.
6. Nel campo **Valore**, immettere UTF-8.
7. Fare clic su **OK**. Se si riceve un messaggio **Configurazione da salvare** con un collegamento nella parte superiore della pagina **Impostazioni JVM**, fare clic sul collegamento per passare alla pagina **Salva configurazione**. Selezionare **Salva**, quindi fare clic su **OK**.
8. Per effettuare tale modifica nelle applicazioni WebSphere, arrestare il server delle applicazioni, quindi riavviarlo. La modalità di arresto e riavvio dell'applicazione può variare dalla piattaforma su cui è stato installato WebSphere Application Server. Fare riferimento alla documentazione WebSphere Application Server per la piattaforma per le informazioni relative all'arresto e all'avvio del server delle applicazioni.

Creazione di un foglio di stile

Se si desidera integrare pagine aggiuntive con un foglio di stile a cascata (CSS), è necessario personalizzare lo stile del progetto WebFacing oppure il foglio di stile a cascata affinché siano uguali. Una volta personalizzato il foglio di stile, è possibile utilizzare la procedura guidata di interazione Web per creare una pagina web

dettagliata utilizzando il foglio di stile per visualizzare informazioni su voci di negozio quali prezzo e colore. Per quanto previsto in questo modulo, sarà possibile incorporare il foglio di stile `DetailPageResults.jsp` dal progetto `SV000501`. Tuttavia, per riferimenti futuri, è possibile personalizzare manualmente il file CSS (evidenziato nella sezione successiva).

(Facoltativo) Personalizzazione manuale del foglio di stile a cascata

Come precedentemente accennato, piuttosto che incorporare il foglio di stile `DetailPageResults.jsp` dal progetto `SV000501` per il `Project501`, è possibile personalizzare i propri fogli di stile:

1. Passare alla vista Project Navigator ed espandere **Project501 > Web Content > stili > apparea**.
2. Fare doppio clic su **apparea.css** per aprirlo nel CSS Designer. Questo foglio di stile controlla l'aspetto dell'area dell'applicazione.
3. Modificare e salvare il file.
4. Nella vista Project Navigator, navigare in **Project501 > Web Content > stili > croma**.
5. Fare doppio clic su **avenue.css** per aprirlo nel CSS Designer. Questo foglio di stile controlla l'aspetto dell'intera pagina.
6. Modificare e salvare il file.

Estensione e miglioramento del progetto WebFacing con una interazione web

In questa sezione, si migliora il progetto `WebFacing`. Una volta visualizzato l'elenco delle voci, è possibile fare clic sull'immagine della voce desiderata per trovare dettagli relativi a quella voce. Per fare ciò, si utilizza la procedura guidata di interazione `Web` per invocare un programma `RPG` per richiamare i dettagli della voce e visualizzarli su un'altra pagina web. Eseguire i seguenti passi:

- definire le informazioni del server
- copiare sul foglio di stile corretto
- creare l'interazione
- aggiungere programmi e parametri all'interazione
- modificare l'utilizzo dei parametri

Innanzitutto, è necessario definire le informazioni sul server:

1. Passare alla prospettiva `Web`.
2. Nella vista Navigator, fare clic col tasto destro del mouse su **Project501** e selezionare **Specifica configurazione runtime di iSeries Web Tools**.
3. Immettere il nome del proprio server `iSeries`, l'ID utente e la password.
4. Digitare `Retailstor` nel campo **Libreria**, quindi fare clic su **Aggiungi**.
5. Fare clic su **Fine**. Fare nuovamente clic su **Fine** se necessario.

Prima di creare l'interazione `Web`, è necessario copiare gli stili ed i fogli di stile corretti cosicché il `Project501` visualizzi l'appropriato formato del file `JSP`. (Non è necessario effettuare questo passo se è stato completato l'esercizio, "(Facoltativo) Personalizzazione manuale del foglio di stile a cascata"). Prima copiare la cartella degli stili, quindi copiare `DetailPageResults.jsp`:

1. Nella vista Navigator, espandere **SV000501 > Web Content**.

2. Fare clic con il tastino destro del mouse su **Stili** e selezionare **Copia**.
3. Scorrere nella vista Navigator fino a vedere nuovamente il proprio **Project501**.
4. Espandere il progetto, fare clic con il tastino destro del mouse su **Web Content** e selezionare **Incolla** dal menu a comparsa. Fare clic su **Sì** per sovrascrivere i file esistenti.
5. Scorrere fino a **SV000501 > Web Content**.
6. Fare clic con il tastino destro del mouse su **DetailPageResults.jsp** e selezionare **Copia**.
7. Scorrere fino a vedere nuovamente **Project501**.
8. Espandere il progetto, fare clic con il tastino destro del mouse su **Web Content** e selezionare **Incolla** dal menu a comparsa.

Prendere nota dei nuovi elementi aggiunti al di sotto della cartella Web Content de Project501.

Ora è possibile creare l'interazione web.

1. Fare clic su **File > Nuovo > Altro**.
2. Nella casella di dialogo **Nuovo**, fare clic su **Web** sul lato sinistro, quindi fare clic su **Interazione web** sul lato destro. Fare clic su **Successivo**.
3. Nel campo **nome Interazione web**, immettere **DetailPage** e fare clic su **Successivo**.
4. Selezionare il pallino **Genera Input JSP**.
5. Selezionare il pallino **Utilizza pagine di output** (se non è stato già selezionato) e fare clic su **Aggiungi**.
6. Dalla casella di dialogo **Output JSP**, espandere **Web Content**, selezionare la propria pagina di output pre-formattata **DetailPageResults.jsp** e fare clic su **OK**.
7. Fare clic su **Successivo**.

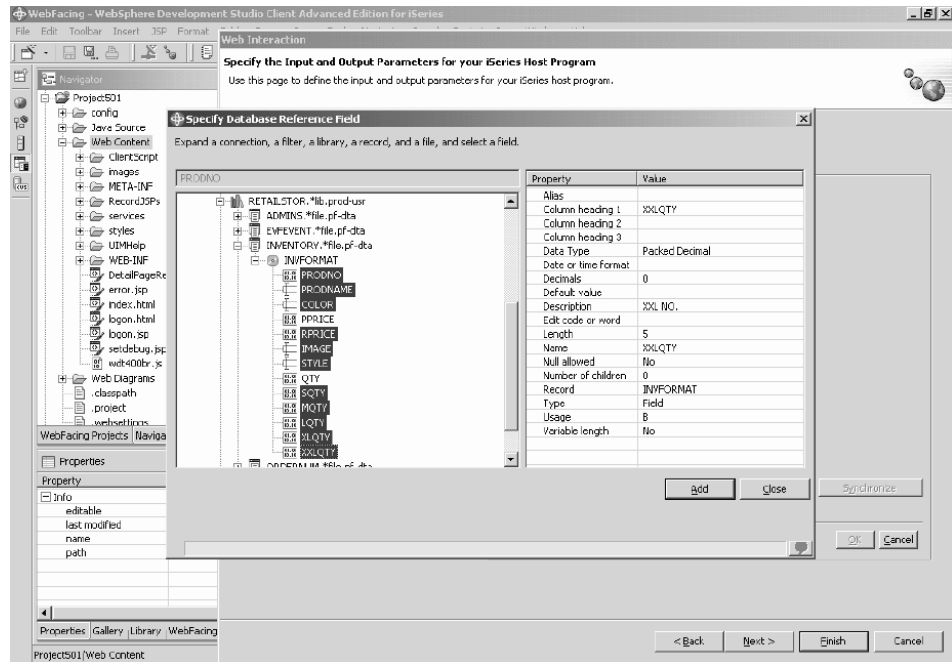
Ora, è possibile aggiungere programmi e parametri alla propria interazione. E' necessario aggiungere 11 parametri allo stesso programma. Invece di aggiungere ogni parametro con i propri valori manualmente, è possibile aggiungerli in un modo più rapido:

1. Selezionare **Utilizza un programma ILE iSeries** se non è stato già selezionato.
2. Fare clic su **Aggiungi Programma**.
3. Nel campo **Programma alias**, immettere **DetailPage**.
4. Per il campo **Oggetto programma**, fare clic su **Sfoggia**.
 - a. Espandere **iSeriesserver > *LIBL > RETAILSTOR**.
 - b. Fare clic su **DETAILPAGE.*pgm.rpgle** (il primo file sotto RETAILSTOR) e fare clic su **OK**.
5. All'interno della procedura guidata dell'Interazione web, fare clic su **OK** (in basso sulla destra).

Da notare che **DetailPage** viene aggiunto sul lato sinistro della procedura guidata sotto le definizioni call del Programma.

6. Nella sezione delle definizioni call del programma nel pannello sinistro della procedura guidata, fare clic su **DetailPage** per selezionarlo.
7. Fare clic su **Aggiungi parametri**.
8. Accanto a **Specifica campo di riferimento database** verso la fine della procedura guidata, fare clic su **Specifica**.

9. Espandere *iSeries*server > *LIBL > RETAILSTOR > INVENTORY.*file.pf-dta > INFORMAT per visualizzare un elenco di 13 parametri. E' necessario aggiungere i seguenti 11 facendo clic su ognuno una volta e facendo clic su **Aggiungi** oppure tenendo premuto il pulsante CTRL e facendo clic su ognuno, quindi su **Aggiungi**: PRODNO, PRODNAME, COLOR, RPRICE, IMAGE, STYLE, SQTY, MQTY, LQTY, XLQTY, XXLQTY. In sostanza, si selezionano tutti i campi ad eccezione di PPRICE e QTY.

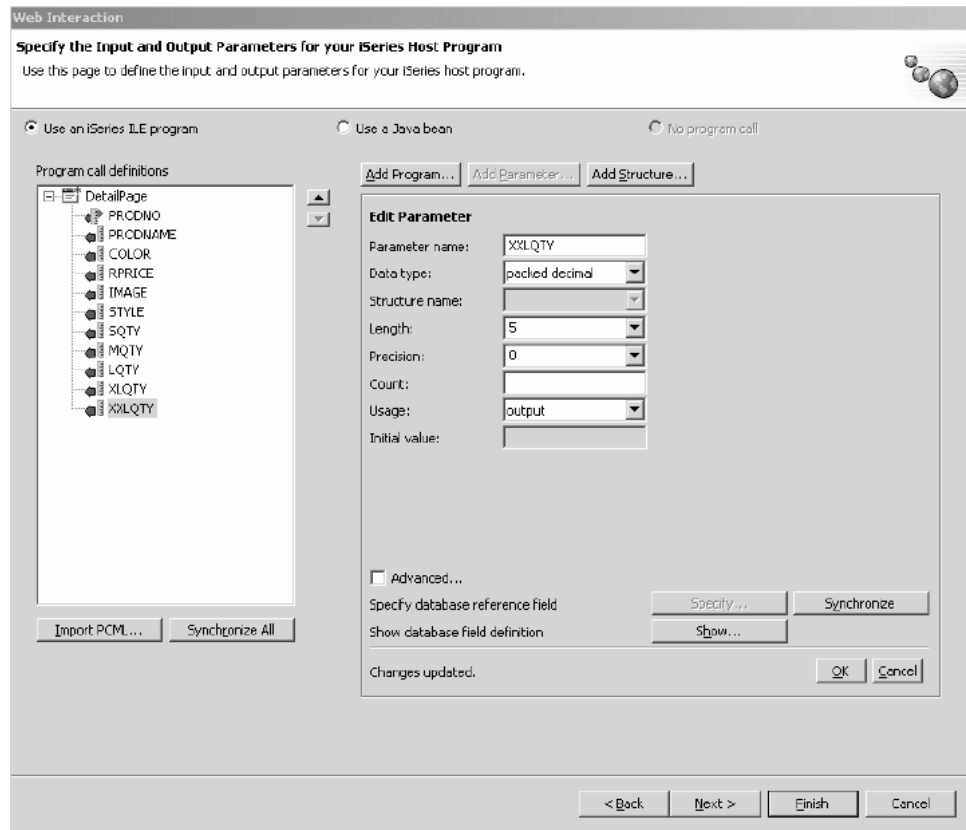


10. Fare clic su **Fine**.

E' ora necessario modificare l'utilizzo per 10 degli 11 parametri in "output" (tutti i parametri eccetto PRODNO).

1. Sempre all'interno della procedura guidata dell'Interazione web, fare clic su **PRODNAME** per selezionarlo.
2. Nella casella combo **Utilizzo**, cambiare il valore selezionato in **output** e fare clic su **OK**. Ripetere questo passo (con il passo precedente) per tutti i parametri ad eccezione del primo (PRODNO). L'area di lavoro si dovrebbe presentare in questo modo una volta aggiustati tutti i parametri. L'icona a sinistra del parametro indica se esso è di tipo immissione, immissione & emissione o

emissione



3. Fare clic su **Successivo** per vedere l'anteprima **Modulo di immissione**.
4. Fare clic su **Fine** per creare l'Interazione web.
5. Selezionare **Sì a tutti** o **OK**, quindi di nuovo **Fine** se si riceve qualsiasi messaggio.

Collegamento del progetto ad una Interazione web

Una volta creato un file JSP di interazione web che utilizza parametri di immissione ed emissione, è necessario personalizzare il file JSP in modo da farlo funzionare anche con il componente WebFacing. E' necessario immettere il codice per creare un collegamento dalla propria applicazione WebFacing per richiamare questa interazione web. Per fare ciò, è necessario aggiungere una funzione JavaScript nel file webface.js in modo da poter richiamare il servlet DetailPageServlet con il parametro PRODNO in una nuova finestra.

Per creare il collegamento:

1. Nella vista Navigator, espandere **Project501 > Web Content > ClientScript**.
2. Fare doppio clic su **webface.js** per aprirlo in un editor.
3. Scorrere alla fine del file ed immettere le seguenti righe:

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

4. Fare clic sull'icona Salva oppure fare clic su **File > Salva webface.js**.

Se si desidera creare un'applicazione simile a SV000501 in futuro, è anche necessario modificare una impostazione web nell'origine DDS per abilitare l'immagine aggiunta e il collegamento con finestra a comparsa per la funzione JavaScript.

Il codice RPG incluso in questa applicazione è alterato per mostrare la modifica, ad ogni modo, sarebbe necessario effettuare la modifica manualmente nelle applicazioni future. Inoltre, dopo aver modificato le impostazioni web, sarà necessario riconvertire l'origine DDS.

E' possibile controllare il codice per duplicare il risultato.

Per visualizzare l'origine DDS:

1. Passare alla prospettiva WebFacing.
2. Nella vista Progetti WebFacing, espandere **Project501 > DDS**.
3. Fare clic con il tastino destro del mouse su **<iSeriesserver > RETAILSTOR/QDDSSRC(QUERY)** e selezionare **Apri con > CODE Designer**.
4. Dopo l'apertura del CODE Designer, espandere **SCREEN1 > ITEMSUB**.
5. Fare clic su **IMAGESRC** per selezionarlo.
6. Fare clic sul separatore **Origine**.
7. Fare clic sul separatore **Impostazioni Web** nella parte inferiore a destra della finestra.

Nota: Controllare le proprietà delle impostazioni web quali l'ampiezza in pixel ed il nome del file. In futuro, sarà necessario effettuare le stesse modifiche all'origine DDS e poi riconvertire l'origine.

8. Tenere presenti le seguenti righe nell'origine:

```
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}')
```

Da notare in modo particolare che Project501 è specificato nell'ultima riga. Come già menzionato, se si desidera eseguire nuovamente l'applicazione SV000501, si dovrà riportare il valore del Project501 a SV000501.

9. Salvare il file facendo clic sull'icona Salva o su **File > Salva** dalla barra menu.
10. Chiudere il file e CODE Designer.

Per ulteriori informazioni su come lavorare con l'origine DDS, consultare la prospettiva Guida della workbench e la documentazione IBM WebFacing Tool.

Pubblicazione dei file e riavvio del server

In questa sezione, si pubblicano i file e si riavviano in modo che l'applicazione Project501 possa scegliere tutte le modifiche.

Tuttavia, prima di riavviare il server, è necessario verificare che Project501.war sia aggiunto al file SVStoreEAR.

Per verificare la configurazione:

1. Nella vista Navigator, espandere la **cartella Scenario**.

2. Fare doppio clic su **Scenario server.wsi**.

E' possibile visualizzare una finestra di dialogo per stabilire se si desidera che l'editor corregga le voci del progetto automaticamente. Fare clic su **Si**. Se non si vede questa finestra, non è necessario procedere in tal senso. Ad ogni modo, il file sarà automaticamente modificato quando viene aperto.

3. Salvare e chiudere il file.

Per riavviare il server:

1. Passare alla prospettiva Server.

2. Nella vista Configurazione Server, in fondo a sinistra, espandere **Server** e fare doppio clic su **Scenario server**.

3. Fare clic sul separatore **Server** in fondo allo schermo a destra per vedere lo stato del server nella vista Server.

4. Fare clic con il tastino destro del mouse su **Scenario server** e selezionare **Pubblica**. Fare clic su **OK** una volta terminata la pubblicazione.

5. Fare clic con il tastino destro del mouse sul server nella vista Server e selezionare **Avvio** o **Riavvio** (qualunque sia abilitato). In caso di richieste relative ai file non salvati, cancellare la finestra, salvare tutti i file aperti e procedere nuovamente.

6. Controllare la Console delle informazioni di collegamento (aperta automaticamente). Il server sarà avviato quando si viene visualizzato il seguente messaggio alla fine del collegamento: Server *server_name* aperto per e-business.

Verifica dell'interfaccia

E' stata completata la procedura necessaria per creare un'interfaccia per visualizzare ordini ed inventario. Per verificare l'interfaccia:

1. Passare alla prospettiva Web.

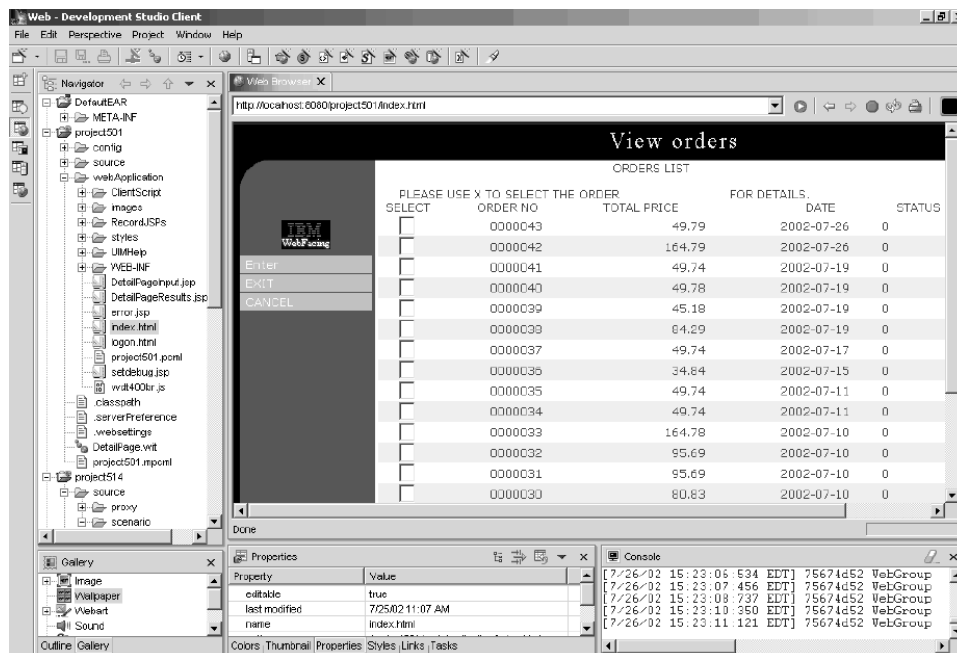
2. Espandere **Project501 > Web Content**.

3. Fare clic col tasto destro su **index.html** e selezionare **Esegui sul server**.

4. Viene aperta una finestra **Selezione Server**. Verificare che **Scenario server** sia selezionato come impostazione predefinita e fare clic su **Fine**.

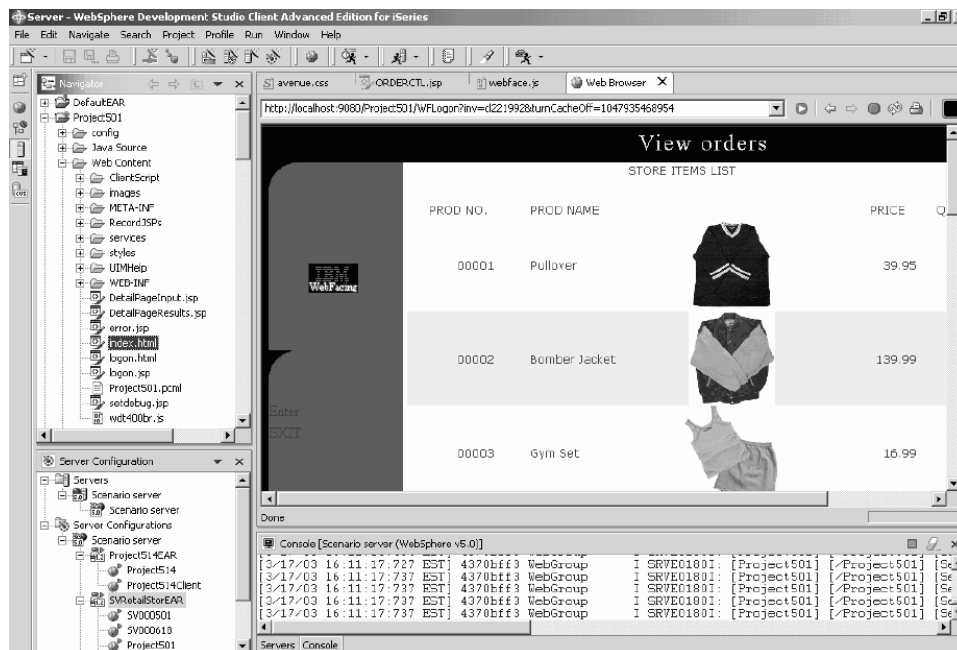
5. fare clic su **Visualizza ordini – Avvia nella principale finestra del browser** per richiamare l'applicazione dell'administrator. Dopo il collegamento con l'ID utente iSeries e la password (per il server iSeries utilizzato nello sviluppo del

progetto), si verrà indirizzati alla seguente pagina:



Provare ad immettere X accanto ad uno qualunque dei numeri del prodotto per vedere i dettagli relativi a quel prodotto.

6. Fare clic sulla freccia Indietro per andare alla pagina index.html, quindi fare clic sulla finestra **Visualizza inventario – Avvia nel browser principale** per richiamare l'applicazione del cliente e si verrà indirizzati alla seguente pagina (dopo il collegamento al server iSeries con l'ID utente e la password):



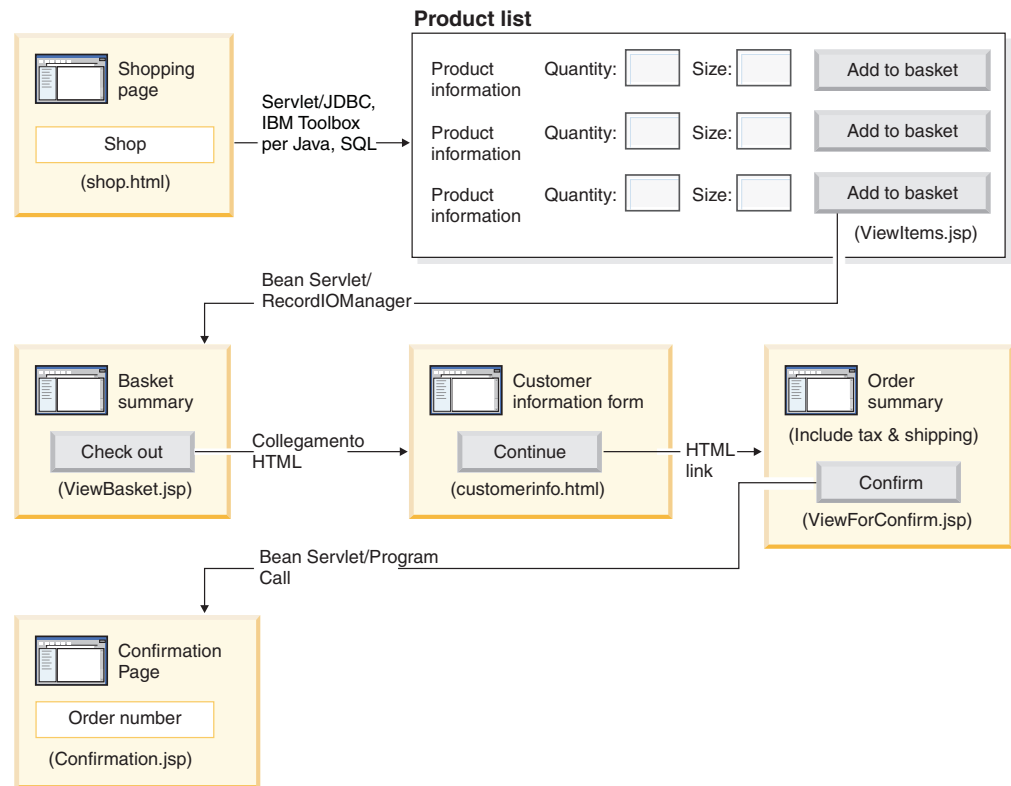
Provare a fare clic su un'immagine per visualizzare i dettagli su un particolare articolo.

Il modulo è ora completato! E' stata creata un'interfaccia per visualizzare voci di inventario ed ordini.

Capitolo 6. Modulo 1 avanzato: Creare HTML, servlet e file JSP che mettano gli ordini del cliente su un server iSeries (SV001585)

Introduzione

Questo progetto è inteso per utenti con una buona conoscenza di programmazione Java ed una conoscenza base della gestione dati di iSeries e di RPG. Il progetto illustra le modalità di lavoro con le classi di accesso ai dati di IBM Toolbox for iSeries, i bean di RecordIOManager ed i bean Chiamata Programma per creare codice HTML, servlet e file JSP che mettono gli ordini del cliente su un server iSeries. Come utente, dalla pagina di acquisto è possibile visualizzare i prodotti disponibili ed aggiungere articoli al carrello. Una volta stabiliti tutti gli articoli selezionati (nel riepilogo Carrello), è possibile fare clic su un pulsante di conferma ed essere indirizzati ad un modulo informativo per il cliente. Una volta completato il modulo, il progetto restituisce un riepilogo dell'ordine con un pulsante di conferma che porta ad una pagina di conferma che visualizza l'ordine effettuato ed il numero dell'ordine.



Riepilogo dei passi del livello avanzato

Poiché questo è un modulo avanzato, le istruzioni non indicano all'utente tutti i passi di creazione del progetto, ma sottolineano i passi di sviluppo specifici dell'iSeries- necessari per la creazione di un progetto di questo tipo. Questi sono i passi di livello avanzato:

1. Scrivere una pagina web di acquisto HTML.

2. Scrivere un servlet che popoli un bean Java (utilizzando JDBC e SQL) con gli articoli disponibili acquistabili dal cliente.
3. Scrivere un file JSP per visualizzare gli articoli in vendita e consentire ai clienti di immettere la quantità e la dimensione dell'articolo desiderato e di selezionare l'articolo facendo clic su un pulsante **Aggiungi al carrello**
4. Utilizzando il bean RecordIOManager, scrivere un servlet invocato facendo clic sul pulsante **Aggiungi al carrello** che aggiunge il database INVENTARIO iSeries sottraendo la quantità e la dimensione richiesta per l'articolo ed aggiungendo questa selezione ad un bean Java denominato "carrello". Poi il servlet reindirizza la risposta a ViewBasket.jsp se il funzionamento è corretto. In caso di funzionamento non corretto, il servlet visualizza una pagina di errore.
5. Scrivere un modulo HTML in cui il cliente possa immettere informazioni personali.
6. Scrivere una pagina di conferma di acquisto del file JSP che visualizzi il contenuto del carrello, le tasse e le spese di consegna e spedizione. La pagina deve contenere anche un pulsante di conferma per il cliente.
7. Scrivere un servlet che viene chiamato quando il cliente fa clic sul pulsante di conferma che utilizza un bean Java creato con la procedura guidata Chiamata Programma iSeries. Uno dei metodi bean Java richiama un programma RPG per creare un nuovo ordine nel database ORDINI sul server iSeries corrispondente al contenuto del carrello del cliente. Quindi il servlet restituisce un numero d'ordine, posiziona il bean Java sulla sessione dell'applicazione web e carica un file JSP di conferma dell'ordine che contiene il numero d'ordine.

Prima di cominciare

E' possibile completare l'esercitazione solo se si rispettano i prerequisiti che seguono. I prerequisiti vengono discussi in maggiori dettagli in Capitolo 3, "Esecuzione dello scenario", a pagina 9.

- L'utente ha accesso TCP/IP al server iSeries.
- Occorre aver avviato i server iSeries con il comando STRTCPSVR *ALL
- Occorre aver avviato il server WebFacing con il comando STRTCPSVR *WEBFACING
- Occorre aver ripristinato le librerie WHOLESale e RETAILSTOR sul proprio server iSeries.
- Occorre che le librerie WHOLESale, RETAILSTOR e QGPL siano nell'elenco librerie.
- Sono state completate tutte le attività presenti in Capitolo 3, "Esecuzione dello scenario", a pagina 9, (salvo le attività WebSphere Application Server facoltative, che non sono necessarie per verificare l'applicazione nel workbench).

Creazione di pagine web, servlet e file JSP

Per creare i componenti del progetto SV001585:

1. Creare un progetto web.
2. Scrivere una pagina shop.html con Page Designer che contiene un collegamento che invoca un servlet GetItems
3. Importare il file jt400.jar per il Toolbox iSeries per le classi Java nella cartella *lib* del progetto web. E' possibile trovare questo file jar in `x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime` laddove *x* è la directory in cui è stato installato Development Studio Client.

Nota: Consultare `GetItems.java` e `ViewItems.jsp` nel progetto `SV001585` per vedere il principale Toolbox iSeries per le parti relative a Java JDBC e SQL, per il servlet e JSP. Nella vista Navigator della prospettiva web, è possibile trovare `GetItems.java` espandendo **SV001585 > origine** e `ViewItems.jsp` espandendo **SV001585 > webApplication**.

4. Scrivere un servlet `GetItems` che utilizzi il Toolbox iSeries per Java JDBC e SQL per recuperare voci di abbigliamento dal database `INVENTARIO` iSeries che:
 - a. Posizionare il bean `ResultSet` contenente il risultato della ricerca SQL sulla sessione
 - b. Reindirizzare la richiesta a `ViewItems.jsp`

Esempio di codice per `GetItems.jsp`:

```
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. Scrivere un file JSP `ViewItems` che richiami le voci di abbigliamento dal bean `ResultSet` ottenuto nel passo precedente per visualizzare le voci di abbigliamento in formato di tabella. Il file JSP dovrebbe anche includere un modulo per ogni voce che è possibile utilizzare per selezionare la dimensione e la quantità, quindi aggiungere la voce al carrello. E' possibile utilizzare `Page`

Designer in Strumenti di sviluppo web iSeries per scrivere il JSP. In modo più specifico, è possibile vedere la pagina nella vista Progettazione ed aggiungere il codice nella vista Origine. Esempio di codice per ViewItems.jsp:

```

<!--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. Importare il file Strumenti di sviluppo Java iSeries iseriesut.jar nella cartella *lib* per il proprio progetto web. E' possibile trovare questo file JAR in `x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime`, laddove *x* è la directory in cui è stato installato il prodotto. Consultare `AddtoBasket.java` e `ViewBasket.jsp` nel progetto `SV001585` per visualizzare l'implementazione. Nella vista Navigator della prospettiva *Web*, è possibile trovare `AddtoBasket.java` espandendo **SV001585 > origine** e `ViewBasket.jsp` espandendo **SV001585 > webApplication**.
7. Utilizzare il bean `RecordIOManager` da Strumenti di sviluppo Java iSeries per scrivere un servlet `AddtoBasket` richiamato dal pulsante **Aggiungi al carrello** che aggiorna il database `INVENTARIO` iSeries sottraendo la quantità richiesta dal cliente ed aggiungendo gli articoli al bean Java `Carrello` nella sessione. Esempio di codice `AddtoBasket.jsp`:

```

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. Scrivere un file JSP ViewBasket che visualizza il contenuto del carrello.
9. Scrivere un modulo customerinfo.html per l'immissione di informazioni sul pagamento, con un pulsante **Continua** che invoca ViewForConfirm.jsp.
10. Scrivere un ViewForConfirm.jsp che mostra il contenuto dell'intero carrello più il bilancio totale. E' possibile sviluppare ViewForConfirm.jsp nello stesso modo di ViewBasket.jsp, con l'aggiunta di un canone di spedizione calcolato ed un pulsante **Conferma** che posiziona un ordine nel database ORDINI iSeries.
11. Utilizzare la procedura guidata Chiamata programma iSeries di Strumenti di sviluppo Java iSeries per creare un bean PLACEORD.java che accede al programma di servizio PLACEODR nella libreria RETAILSTOR. La procedura guidata crea bean che possono essere utilizzati da applicazioni Java o la procedura guidata dei Servizi web per accedere ai programmi ILE iSeries.
 - a. Per aprire la procedura guidata, fare clic con il tastino destro del mouse su SV001585 nella vista Navigator e selezionare **Nuovo > Altro**.
 - b. Nella finestra **Nuovo**, fare clic su **iSeries > Java > Bean Chiamata Programma**.

- c. Nella procedura guidata Chiamata Programma, immettere le informazioni sul nome programma ILE iSeries, sulla libreria, il tipo di programma, i parametri di immissione ed emissione.
- d. L'ultima finestra della procedura guidata consente di creare un bean per un'applicazione Java, un servizio web o entrambi. In questo progetto, è solo necessario crearne uno per un'applicazione Java.

Nota: Il programma di servizio PLACEODR porta un array di strutture e posiziona ogni elemento in un record di un database ORDINI, generando un numero d'ordine come output per ogni array.

12. Scrivere un servlet PlaceOrder invocato dal pulsante ViewForConfirm.jsp
Conferma.

- Il servlet utilizza il bean generato dalla procedura guidata Chiamata Programma iSeries per accedere ad un server iSeries e posizionare ordini nel database ORDINI della libreria RETAILSTOR.
- Gli ordini sono gli articoli nel carrello, inviati al programma ILE come array di strutture.
- Ogni struttura di questo array è un articolo nel carrello.
- Il programma di servizio RPG PLACEORD richiamato dal bean restituisce il numero d'ordine come parametro di emissione e lo posiziona sulla sessione.

Il seguente segmento di codice mostra come il servlet PlaceOrder utilizza il bean PLACEORD:

```

.
.
.
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();
    }
}
}
}

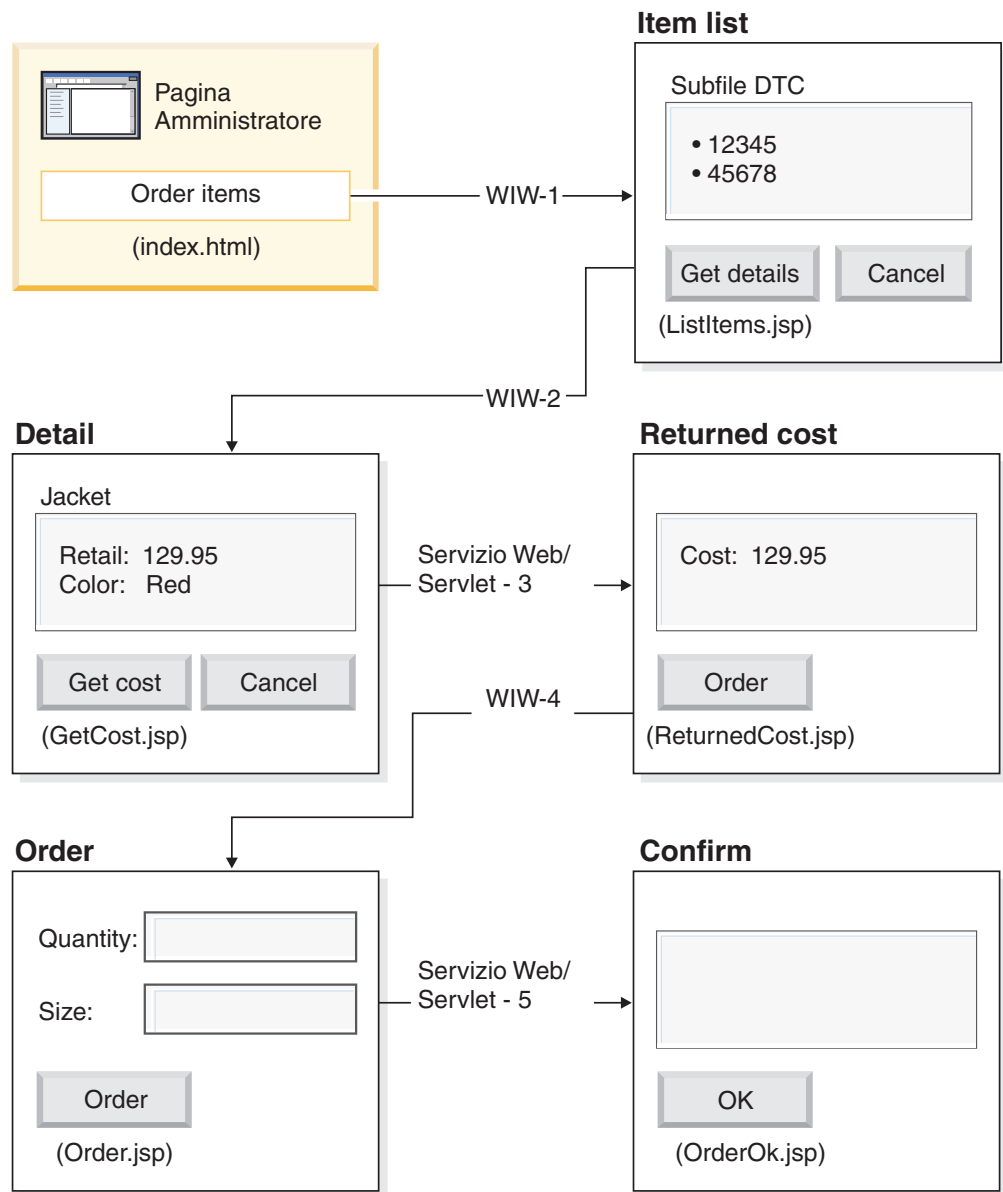
```

13. Scrivere un servlet OrderNumber che richiami il numero e lo visualizzi per il cliente insieme ad un messaggio di conferma. Se il cliente non ha aggiunto alcun articolo al carrello, assicurarsi che venga presentata una pagina di errore.

Capitolo 7. Modulo 2 avanzato: Creare il progetto Web che utilizza i servizi web SV000514 e SV001586 (SV000618)

Introduzione

Questo progetto mostra in che modo è possibile utilizzare la conoscenza della programmazione RPG per la creazione di client Web per programmi iSeries Servizi Web e RPG. In questo progetto, l'utente assume il ruolo di amministratore, passando per una serie di pagine Web per stabilire la quantità dell'inventario e per ordinare un inventario aggiuntivo per la vendita al dettaglio dalla vendita all'ingrosso. L'utente può immettere il numero dell'articolo, visualizzare i dettagli dell'articolo, ne ordina la quantità e la dimensione, accettare e confermare.



(WIW = Web Interaction Wizard)

Poiché questo è un modulo avanzato, le istruzioni non indicano all'utente tutti i passi di creazione del progetto, ma sottolineano i passi di sviluppo necessari per la creazione di un progetto di questo tipo. Questo progetto utilizza i seguenti componenti di Development Studio Client:

- Strumenti di sviluppo web iSeries per creare le pagine Web con Page Designer, che comprendono l'emissione dal wizard Interazione Web e da diversi componenti Web
- Remote System Explorer per creare un programma di servizio RPG TNLSTITM che riporta le informazioni sull'articolo.
- Il wizard Servizi Web per generare il codice proxy servlet, che rileva i prezzi dell'articolo ed effettua gli ordini degli articoli.
- Strumenti di sviluppo Java iSeries per creare il servlet richiesti
- L'Ambiente di verifica WebSphere per verificare l'applicazione prima della distribuzione al server iSeries tramite WebSphere Application Server

Prima di cominciare

E' possibile completare l'esercitazione solo se si rispettano i prerequisiti che seguono. I prerequisiti vengono discussi in maggiori dettagli in Capitolo 3, "Esecuzione dello scenario", a pagina 9.

- L'utente ha accesso TCP/IP al server iSeries.
- Occorre aver avviato i server iSeries con il comando `STRTCPSVR *ALL`
- Occorre aver avviato il server WebFacing con il comando `STRTCPSVR *WEBFACING`
- Occorre aver ripristinato le librerie `WHOLESALE` e `RETAILSTOR` sul proprio server iSeries.
- Occorre che le librerie `WHOLESALE`, `RETAILSTOR` e `QGPL` siano nell'elenco librerie.
- Sono state completate tutte le attività presenti in Capitolo 3, "Esecuzione dello scenario", a pagina 9, (salvo le attività WebSphere Application Server facoltative, che non sono necessarie per verificare l'applicazione nel workbench).

Creazione pagine Web, servlet, JSP e codice RPG

Per creare i componenti di SV000618:

1. Creare un progetto Web che conservi tutti i file che verranno creati dall'utente.
2. Scrivere un file JSP `ListItems` che elenca gli articoli nel database di inventario iSeries. E' possibile utilizzare Page Designer in Strumenti di sviluppo web iSeries per scrivere il servlet. In modo più specifico, è possibile vedere la pagina nella vista Progettazione ed aggiungere il codice nella vista Origine. Occorre inoltre inserire DTC (Design Time Control) del file secondario che interagisca con il programma di servizio TNLSTITM RPG, riempiendo il file secondario con i record di database. E' possibile specificare il programma di servizio nelle impostazioni di controllo DTC.

Successivamente, occorre utilizzare il wizard Interazione Web per creare la pagina di immissione:

- Specificare `ListItems.jsp` come pagina di emissione per elencare gli articoli dell'inventario, che assicura che il wizard Interazione Web crea un file `ListItems.wit`.
- Assicurarsi di non specificare nel wizard Interazione Web alcuna chiamata al programma, poiché il DTC del file secondario richiama automaticamente il

programma di servizio TNLSTITM RPG. Il wizard crea inoltre un ListItemsWitServlet che funge da collegamento per richiamare la pagina ListItems.jsp.

- Per visualizzare nuovamente il file ListItemsWit.wit creato dal wizard Interazione Web:
 1. Espandere **SV001618** e fare doppio clic su **ListItems.wit** per visualizzare il wizard di interazione del file.
 2. Fare clic su **Successivo** nel wizard per rivisualizzare i valori specificati per l'interazione.

Successivamente, è necessario scrivere una pagina di emissione GetCost JSP utilizzando il Page Designer, che prende l'immissione dalla pagina di immissione ListItems.jsp. Quando un utente fa clic su una voce nella pagina ListItems.jsp, una pagina GetCost.jsp visualizza i dettagli relativi a quella voce.

Al termine della creazione della pagina GetCost.jsp, utilizzare il wizard Interazione web per creare un'interazione WitOrder tra ListItems.jsp (che viene selezionata come pagina di immissione) e GetCost.jsp (che viene selezionata come pagina di emissione):

- Sulla pagina Chiamata programma del wizard, specificare un richiamo del parametro e della procedura GetDetail dal programma di servizio TNLSTITM RPG.
- Nella procedura, le API DTC del file secondario vengono integrate in modo da stabilire quale record del file secondario è stato selezionato. La procedura utilizza tali informazioni per richiamare il record selezionato dal database INVENTORY e visualizza i dettagli, inclusa l'immagine della vice selezionata su GetCost.jsp.
- Per visualizzare nuovamente il file WitOrderWit.wit generato dal wizard Interazione web:
 1. Espandere **SV001618** e fare doppio clic su **WitOrder.wit** per visualizzare il wizard di interazione relativo al file.
 2. Fare clic su **Successivo** nel wizard per rivisualizzare i valori specificati per l'interazione.

Notare che il parametro *flow* è specificato sulla pagina di emissione come comando flow. Ciò assicura al valore del parametro che vengano visualizzati i file JSP appropriati.

Successivamente, è necessario importare i file WSDL (Web Services Definition Language) dal progetto SV000514, in modo che l'amministratore possa recuperare l'attuale costo di un articolo dalla vendita all'ingrosso premendo il pulsante **Get cost**.

- Il pulsante **Get cost** richiama QryProdCostServlet.jsp e il servizio web corrispondente dal progetto SV000514.
- Utilizzare il wizard Servizi web e i file WSDL importati per generare il codice proxy Java necessario per richiamare il Servizio web.
- La pagina QryProdCostServlet.jsp prende l'immissione dalla pagina GetCost.jsp, utilizza il codice proxy Java per richiamare il servizio web SV000514 in modo da rilevare il costo dell'articolo selezionato e visualizza il costo in una pagina denominata ReturnedCost.jsp.
- Per visualizzare il codice QryProdCostServicesProxy.java e QryProdCostServlet.java:
 1. Espandere **SV001618** > **source** > **proxy** > **soap**.

2. Fare doppio clic su **QryProdCostServicesProxy.java**.
3. Per **QryProdCostServlet.java**, fare doppio clic su **QryProdCostServlet.java**, in **SV001618 > source**, e notare in che modo viene istanziato il codice proxy Java.

Successivamente, è necessario utilizzare il wizard Interazione Web per collegare **ReturnCost.jsp** come pagina di immissione e **Order.jsp** come quella di emissione, in modo che l'amministratore possa fare clic su un pulsante **Order** per effettuare l'ordine dell'articolo selezionato dal grossista.

- Con tale interazione, non è necessario utilizzare una chiamata programma, poiché il collegamento delle due pagine è sufficiente per visualizzare le informazioni valide.
- Per visualizzare **WitPlaceOrder.wit**:
 1. Espandere **SV001618**.
 2. Fare doppio clic su **WitPlaceOrder.wit** per aprire l'interazione.
 3. Fare clic su **Successivo** nel wizard per visualizzare nuovamente i valori specificati.

Successivamente, utilizzare il Servizio web **SV001586**, in modo che l'amministratore possa specificare la dimensione e la quantità degli articoli ordinati.

- Importare i file **SV001586 WSDL** nel progetto, generare il codice proxy Java per richiamare il Servizio web e scrivere un **OrderSupplyServlet** richiamato quando l'utente preme il pulsante **Order** da **Order.jsp**.
- Il servlet raccoglie informazioni dalla pagina **Order.jsp**, richiama il codice proxy Java del servizio web, che richiama il servizio web **SV001586** e ordina l'articolo.
- Il servlet visualizza **OrderOK.jsp** se l'ordine va a buon fine e una pagina di errore in caso non andasse a buon fine.
- Per visualizzare in che modo viene istanziato e richiamato il proxy del servizio Web generato per l'utilizzo del servizio web **SV001586**:
 1. Espandere **SV000618 > source**.
 2. Fare doppio clic su **OrderSupplyServlet.java** ed esaminarne il contenuto.

Prima di effettuare la distribuzione a WebSphere Application Server

Prima di distribuire l'applicazione a WebSphere Application Server, è necessario modificare un URL specifico nei file di classe Java **SV000514** e **SV001586** in modo che l'applicazione funzioni correttamente.

Per **SV000514**:

- Nella classe proxy dei servizi Web **QryProdCostServicesProxy**, viene definita una variabile che contiene l'URL del servizio web da richiamare.
- Quando il proxy viene creato per la prima volta, questo URL viene impostato su **http://localhost:9080/SV000514/servlet/rpcrouter**.
- Con la variabile impostata su questo valore, viene richiamato il servizio nel progetto web **SV000514** nell'IDE, come indicato in "Esecuzione dell'applicazione nel workbench" a pagina 17.
- Prima di distribuire questa applicazione, è necessario modificare questo valore URL in modo da indicare dove è stato distribuito il file EAR, **SVWholeSale.ear**, sulla directory IFS **iSeries**.

Per il progetto **SV001586**:

- Nella classe proxy dei servizi Web OrderSupplyServicesProxy, viene definita una variabile che contiene l'URL del servizio web da richiamare.
- Quando il proxy viene creato per la prima volta, questo URL viene impostato su `http://localhost:9080/SV001586/servlet/rpcrouter`.
- Con la variabile impostata su questo valore, viene richiamato il servizio web nel progetto SV001586 nell'IDE, come indicato in "Esecuzione dell'applicazione nel workbench" a pagina 17.
- Prima di distribuire questa applicazione, è necessario modificare questo valore URL in modo da indicare dove è stato distribuito il file SVWholeSaleEAR sulla directory IFS iSeries.

Capitolo 8. Informazioni particolari

Nota relativa a diritti limitati per gli utenti del governo degli Stati Uniti - L'uso, la duplicazione o la divulgazione sono limitati dal GSA ADP Schedule Contract con IBM Corp.

Queste informazioni sono state sviluppate per i prodotti e i servizi offerti negli U.S.A. L'IBM non può offrire i prodotti, i servizi o i dispositivi presentati in questo documento in altri paesi. Consultare il rappresentante IBM locale per informazioni sui prodotti e sui servizi attualmente disponibili nel proprio paese. Ogni riferimento relativo a prodotti, programmi o servizi IBM non implica che solo quei prodotti, programmi o servizi IBM possano essere utilizzati. In sostituzione a quelli forniti dall'IBM, possono essere utilizzati prodotti, programmi o servizi funzionalmente equivalenti che non comportino violazione dei diritti di proprietà intellettuale o di altri diritti dell'IBM. E' comunque responsabilità dell'utente valutare e verificare la possibilità di utilizzare altri prodotti, programmi o servizi non IBM.

L'IBM può avere brevetti o domande di brevetti in corso relativi a quanto trattato in questa pubblicazione. La fornitura di questo documento non implica la concessione di alcuna licenza su di essi. Chi desiderasse ricevere informazioni relative alla licenza, può rivolgersi per iscritto a:

IBM Director of Commercial Relations IBM Europe Schoenaicher Str. 220 D-7030 Boeblingen Deutschland

Per domande sulla licenza riguardanti informazioni double-byte (DBCS), contattare l'IBM Intellectual Property Department nel proprio paese oppure scrivere a:

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

Le informazioni contenute nel seguente paragrafo non si applicano al Regno Unito o a qualsiasi altro paese in cui tali disposizioni siano incompatibili con la legge locale: LA IBM (INTERNATIONAL BUSINESS MACHINES CORPORATION) FORNISCE LA PRESENTE PUBBLICAZIONE NELLO "STATO IN CUI SI TROVA", SENZA GARANZIE DI ALCUN TIPO, NE' ESPRESSE NE' IMPLICITE, IVI INCLUSE, A TITOLO ESEMPLIFICATIVO, GARANZIE IMPLICITE DI NON VIOLAZIONE, DI COMMERCIALIZZABILITÀ E IDONEITÀ PER UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia delle garanzie espresse o implicite in alcune transazioni, pertanto, questa affermazione potrebbe non essere a voi applicabile.

Queste informazioni possono contenere imprecisioni tecniche o errori tipografici. Alle informazioni di seguito riportate periodicamente vengono apportate delle modifiche; tali modifiche saranno incluse nelle nuove edizioni della presente pubblicazione. La IBM si riserva di apportare senza preavviso e in qualsiasi momento miglioramenti e/o modifiche al/i prodotto/i e/o al/i programma/i descritto/i in questa pubblicazione.

I riferimenti ai siti web non IBM contenuti in queste informazioni vengono forniti esclusivamente a scopo di convenienza e non costituiscono in alcun modo un avallo a tali siti. I materiali disponibili in questi siti non fanno parte del prodotto e l'utilizzo di questo è a discrezione dell'utente.

La IBM può utilizzare o divulgare ogni informazione fornita in qualunque modo ritenga appropriato senza alcun obbligo da parte sua.

Coloro che detengono la licenza su questo programma e desiderano avere informazioni allo scopo di consentire: (i) uno scambio di informazioni tra programmi indipendenti e altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

Lab Director IBM Canada Ltd. Laboratory 8200 Warden Avenue Markham, Ontario, Canada L6G 1C7

Tali informazioni possono essere disponibili, in base ad appropriate clausole e condizioni, includendo in alcuni casi, il pagamento di una tassa.

Il programma e tutto il materiale disponibile concessi in licenza descritti nella presente pubblicazione sono forniti dalla IBM in base ai termini dell'IBM Customer Agreement dell'IBM IPLA (International Program License Agreement), o a qualsiasi altro accordo equivalente.

Informazioni relativi a prodotti non IBM provengono dai fornitori di tali prodotti, dagli annunci pubblicati o da altre fonti pubblicamente disponibili. L'IBM non ha sottoposto a verifica tali prodotti e non può garantire l'accuratezza dell'esecuzione, la compatibilità o qualsiasi altra pretesa relativa a prodotti non IBM. Eventuali domande sulle prestazioni di prodotti non IBM vanno rivolte ai fornitori di tali prodotti.

Queste informazioni possono contenere esempi relativi a dati e prospetti utilizzati in operazioni commerciali ordinarie. Per illustrarle nel modo più completo possibile, gli esempi includono nomi di persone, società, marchi e prodotti. Tutti questi nomi sono fittizi e qualsiasi somiglianza con nomi e indirizzi utilizzati da reali organizzazioni aziendali è puramente casuale.

LICENZA SOGGETTA ALLE LEGGI SUL DIRITTO D'AUTORE:

Queste informazioni contengono esempi di programmi applicativi in linguaggio sorgente che illustrano tecniche di programmazione su varie piattaforme operative. Potete copiare, modificare e distribuire questi esempi di programmi sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi conformemente alle API (Application Programming Interface) a seconda della piattaforma operativa per cui tale esempi di programma sono stati scritti. Questi esempi non sono stati testati approfonditamente tenendo conto di tutte le condizioni possibili. La IBM, quindi, non può garantire o sottintendere l'affidabilità, la funzionalità o il funzionamento di questi programmi. Potete copiare, modificare e distribuire questi esempi di programma sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in modo conforme alle API (Application Programming Interface) della IBM.

Ogni copia, parte di questi programmi di esempio o lavoro derivato, devono includere un avviso sul copyright come ad esempio:

(C) (nome società) (anno). Le parti di questo codice provengono da IBM Corp. Sample Programs. (C) Copyright IBM Corp. 1992, 2002. Tutti i diritti riservati.

Informazioni sull'interfaccia di programmazione


Le informazioni sull'interfaccia di programmazione sono di ausilio per la creazione del software delle applicazioni utilizzando tale programma.

Le interfacce di programmazione ad uso generale, consentono di scrivere il software dell'applicazione, ottenendo i servizi degli strumenti di questo programma.

Tuttavia, queste informazioni possono inoltre contenere informazioni sulla diagnostica, sulle modifiche e sulla messa a punto. Le informazioni appena menzionate vengono fornite come ausilio per esecuzione di debug del software delle applicazioni.

Attenzione: non utilizzare le informazioni presenti come interfaccia di programmazione in quanto sono soggette a modifiche.

Marchi e marchi di servizio

- 400
- AFP
- AIX
- AIX windows
- APPN
- Application System/400
- AS/400
- AS/400e
- BookManager
- C Set ++
- C/400
- CICS
- CICS/400
- CICS/ESA
- COBOL/2
- COBOL/400
- Common User Access
- CUA
- DB2
- DB2 Extenders
- DB2 Universal Database
- 
- GDDM
- IBM
- IBMLink
- ile (Integrated Language Environment)
- iSeries
- Language Environment
- MQSeries
- Network Station
- Open Class
- Operating System/2
- Operating System/400
- OS/2
- OS/390
- OS/400
- POWER2
- PowerPC
- PROFS
- RPG/400
- RS/6000
- S/390
- SAA
- SQL/400
- System/36
- System/38
- VisualAge
- VTAM
- WebSphere

InstallShield è un marchio della InstallShield Corporation.

Intel e Pentium sono marchi della Intel Corporation negli Stati Uniti e/o in altri paesi.

Java e tutti i marchi e i logo Java sono marchi della Sun Microsystems, Inc. negli Stati Uniti e/o in altri paesi.

Rational è un marchio della International Business Machines Corporation e della Rational Software Corporation, negli Stati Uniti e/o in altri paesi.

Lotus, Lotus Notes e Domino sono marchi di Lotus Development Corporation negli Stati Uniti e/o in altri paesi.

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT, Win32, e il logo Windows sono marchi di Microsoft Corporation negli Stati Uniti e/o in altri paesi.

Netscape Navigator è un marchio di Netscape Communications Corporation.

UNIX è un marchio di The Open Group.

Altri nomi di società, prodotti e servizi possono essere marchi di altre società.



Numero programma: 5724-A81

Stampato in Italia