



Aplicativo de Cenário da Loja de Varejo do iSeries

Versão 5.1 para Windows



Aplicativo de Cenário da Loja de Varejo do iSeries

Versão 5.1 para Windows

Índice

Capítulo 1. Introdução ao Aplicativo de Cenário da Loja de Varejo do iSeries . . .	1
Cenários com Base em Projetos	2
Capítulo 2. Conceitos Tecnológicos . . .	5
Capítulo 3. Executando o Cenário . . .	9
Antes de Começar	9
Instalando os Arquivos de Amostra.	9
Restaurando os Arquivos .savf	10
Modificando a Lista de Bibliotecas.	11
Importando e Executando o Aplicativo no Development Studio Client	12
Extraindo Arquivos .zip em seu Sistema Local.	12
Criando os Projetos da Web	13
Importando os Arquivos de Projeto da Web	13
Importando Arquivos de Projetos do WebFacing e Criando o Projeto do WebFacing.	14
Configuração do Servidor	15
Configurando o Projeto WebFacing	16
Definindo Informações sobre o Servidor.	17
Executando o Aplicativo no Workbench	17
Executando o Aplicativo como Cliente no Workbench	17
Executando o Aplicativo como Administrador no Workbench	18
Implementando o Aplicativo de Cenários no WebSphere Application Server	19
Configurando o WebSphere Application Server	19
Aplicando Segurança na Página do Administrador	20
Criando Arquivos EAR para Implementação do iSeries WebSphere Application Server.	21
Implementando os Arquivos EAR no iSeries WebSphere Application Server	22
Executando o Aplicativo no WebSphere Application Server	22
Capítulo 4. Módulo Passo a Passo 1: Criar um Serviço da Web para Retornar Preços de Produtos (SV000514)	25
Introdução.	25
Antes de Começar	25
Criando um Novo Projeto da Web.	26
Definindo as Informações do Servidor iSeries	26
Criando o Programa de Serviços RPG	26
Criando os Parâmetros e Gerando o Bean Java	27
Criando um Serviço da Web a partir do Bean Java	28
Testando a Amostra	29

Capítulo 5. Módulo Passo a Passo 2: Criar uma Interface para Visualizar Inventário e Itens de Pedidos (SV000501)	31
Introdução.	31
Antes de Começar	31
Criando um Projeto WebFacing.	32
Convertendo a Origem DDS.	33
Configurando Suporte a UTF-8 no Ambiente de Trabalho — Somente para Usuários do WAS Versão 4.0	34
Configurando Suporte a UTF-8 para o WebSphere Application Server	34
Criando uma Página de Estilo	35
(Opcional) Personalizando Manualmente uma Página de Estilo em Cascata.	35
Estendendo e Aprimorando seu Projeto WebFacing com uma Interação da Web	36
Vincular o Projeto a uma Interação com a Web	39
Publicando os Arquivos e Iniciando Novamente o Servidor	40
Testando a Interface	41
Capítulo 6. Módulo Avançado 1: Criar HTML, Servlets e Arquivos JSP que Fazem Pedidos do Cliente em um Servidor iSeries (SV001585)	43
Introdução.	43
Resumo das Etapas de Nível Superior	43
Antes de Começar	44
Criando a Página da Web, Servlets e Arquivos JSP	44
Capítulo 7. Módulo Avançado 2: Criar o Projeto da Web que Utiliza os Serviços da Web SV000514 e SV001586 (SV000618)	53
Introdução.	53
Antes de Começar	54
Criando as Páginas da Web, Servlets, JSPs e Código RPG.	54
Antes de Implementar no WebSphere Application Server	56
Capítulo 8. Avisos	59
LICENÇA DE COPYRIGHT:	60
Informações sobre a Interface de Programação.	61
Marcas Comerciais e Marcas de Serviço	61

Capítulo 1. Introdução ao Aplicativo de Cenário da Loja de Varejo do iSeries

O pacote de cenários é um aplicativo de amostra desenvolvido utilizando o IBM WebSphere Development Studio Client para iSeries, a tecnologia com base no Eclipse para o servidor iSeries. Os cenários foram criados para desenvolvedores que desejam utilizar um servidor iSeries para:

- Desenvolvimento Java
- Desenvolvimento para a Web
- Gerenciamento e transformação de código RPG em aplicativos da Web

O aplicativo consiste em dois arquivos .savf que são restaurados no servidor iSeries e em cinco arquivos .zip que são importados para o produto cliente. Eles formam uma série de páginas da Web com URLs para diversos pontos no processo.

Nota importante: Verifique a página de suporte para obter todos os pacotes de serviço, PTFs e outras informações:

ibm.com/software/awdtools/wdt400/support/.

Estes cenários o guiam por diversas partes do produto, com ênfase em componentes específicos do iSeries, como:

- A IBM WebFacing Tool
- Serviços Web
- Ferramentas de desenvolvimento da Web para iSeries (incluindo o assistente Web Interaction e o assistente de Chamada de Programa)
- Ferramentas de desenvolvimento Java para iSeries
- O IBM Toolbox para Java

Os cenários ilustram a situação de duas empresas, um fornecedor atacadista e uma loja varejista de roupas, que negociam entre si e utilizam servidores iSeries para lógica de negócios e dados. As duas empresas costumavam comunicar-se por e-mail, telefone e fax, para verificar o inventário, enviar pedidos e acompanhar o atendimento dos pedidos. Agora, pretendem utilizar a Web para executar suas transações de negócios comuns.

O varejista deseja um site na Web que:

- Os clientes possam utilizar para comprar produtos
- Os funcionários possam utilizar para solicitar inventário ao atacadista

O fornecedor atacadista deseja:

- Receber pedidos do varejista on-line, para acompanhamento
- Atender vários possíveis clientes

Neste cenário, você assumirá o papel de um consultor de programação para as duas empresas, ajudando-as a colocar seus negócios na Web.

O aplicativo tem dois pontos de entrada diferentes, com base no tipo de usuário. Como cliente, sua primeira etapa é visualizar os produtos que a loja tem para oferecer, neste caso, roupas casuais. Para fazer uma compra, você pode clicar em

um link para acessar uma tela de pedido. Depois de fazer o pedido, será gerada uma página de resumo e você poderá continuar comprando, cancelar o pedido ou enviar o pedido.

Como administrador, você tem um ID do usuário seguro para o aplicativo, exigindo a definição da política de segurança do aplicativo durante a implementação. Seu ponto de entrada é uma tela de login, na qual é possível ver seus pedidos e o inventário e comprar do fornecedor atacadista. É possível selecionar itens, verificar o preço de atacado atual e solicitar o tamanho e a quantidade desejados. O aplicativo verifica se o atacadista tem o tamanho e a quantidade desejados e confirma o pedido ou o avisa que não poderá atender o pedido agora.

Abaixo da superfície do aplicativo, estão ocorrendo várias ações nas diversas partes do produto. A tabela a seguir ilustra o processo e o componente do produto responsáveis por cada parte do aplicativo. Prossiga com as informações detalhadas sobre como executar cada tarefa.

Tabela 1.

Tarefas do Aplicativo do Cliente	Tarefas do Aplicativo do Administrador	Processo Subjacente
Exibir preços de produtos		Utilize um programa RPG do iSeries para criar um serviço da Web e utilize as ferramentas de desenvolvimento da Web para visualizar e exibir os preços.
Fazer um pedido da loja		Utilize servlets e arquivos JSP juntamente com as ferramentas de desenvolvimento Java do iSeries, ferramentas de desenvolvimento da Web do iSeries e o IBM Toolbox para Java para acessar e visualizar o inventário em um servidor iSeries e para fazer pedidos e mostrar o resumo de uma compra.
	Visualizar inventário	Utilize a IBM WebFacing Tool para converter um programa RPG existente em um aplicativo da Web e as ferramentas de desenvolvimento da Web para personalizar a página da Web.
	Fazer um pedido para o fornecedor atacadista com o ID e quantidade da mercadoria.	Crie um serviço da Web que seja chamado quando você clicar no botão Purchase .
Visualizar a página da Web da loja.	Visualizar a página da Web para inventário de pedidos.	Utilize ferramentas de desenvolvimento da Web para iSeries para criar as duas home pages.

Cenários com Base em Projetos

O aplicativo de cenários do iSeries é composto de cinco projetos, denominados SV000501, SV000514, SV000618, SV001585 e SV001586. Este guia contém instruções de como executar o aplicativo compactado. O guia também contém módulos passo a passo e módulos avançados, mostrando como você mesmo pode construir dos diversos projetos. Os módulos passo a passo são destinados a desenvolvedores com pouca experiência no desenvolvimento de aplicativos e Development Studio

Client. Os módulos avançados foram criados para desenvolvedores com experiência em desenvolvimento de aplicativos e mais familiarizados com o Development Studio Client.

Nota: Apesar de existirem cinco projetos, os módulos são apenas quatro, pois os projetos SV001586 e SV000514 são módulos de Serviços da Web combinados.

Este cenário está na forma de cinco projetos:

Projeto SV000501: Crie um projeto da Web para visualizar pedidos pendentes, inventário e detalhes do produto – Este projeto é criado com as ferramentas de desenvolvimento da Web do iSeries e a IBM WebFacing Tool e foi desenvolvido para programadores de RPG com conhecimento limitado de desenvolvimento de aplicativos da Web e desejam utilizar a IBM WebFacing Tool para colocar seus aplicativos de RPG na Web.

Projeto SV001585: Crie código HTML, servlets e arquivos JSP que registram pedidos de clientes em um servidor iSeries – Este projeto utiliza a IBM Toolbox para as classes Java SQL e JDBC, o bean RecordIOManager das ferramentas de desenvolvimento Java do iSeries e o assistente de chamada do Programa iSeries. Esses elementos mostram diversas formas de acessar e manipular dados e programas existentes no servidor do iSeries. Este projeto foi desenvolvido para programadores Java e desenvolvedores de aplicativos da Web que desejam desenvolver páginas da Web para acessar dados e códigos do iSeries. Além disso, é necessário ter conhecimentos práticos de gerenciamento do servidor iSeries e de programação RPG.

Projeto SV000514: Crie um serviço da Web do iSeries para fornecer preços de produtos – O assistente de Serviços da Web utiliza um bean Java gerado pelo assistente iSeries Program Call para chamar um ou mais procedimentos do programa no servidor iSeries e transmitir as informações de volta para um navegador. Este projeto foi desenvolvido para programadores de RPG que desejam utilizar os Serviços da Web para criar aplicativos modulares independentes que possam ser descritos, publicados, localizados e chamados na World Wide Web.

Projeto SV001586: Crie um serviço da Web para fazer pedidos ao fornecedor atacadista por meio de um servidor iSeries – O serviço da Web aceita um ID de mercadoria e a quantidade necessária e, em seguida, faz o pedido para o atacadista. Este projeto é um componente do SV000514, desenvolvido para programadores de RPG que desejam criar serviços da Web.

Projeto SV000618: Criar um projeto da Web para obter interface com o formulário de pedido, o formulário de inventário e a ordem de compra gerados pelo a IBM WebFacing Tool – Este projeto requer as ferramentas de desenvolvimento da Web do iSeries e envolve a criação de arquivos HTML e JSP para utilizar e conectar os serviços da Web desenvolvidos nos projetos SV000514 e SV001586. O projeto foi criado para desenvolvedores que desejam trabalhar com o Serviços da Web e têm conhecimentos de programação RPG e Java.

Capítulo 2. Conceitos Tecnológicos

Para trabalhar com o aplicativo de cenários, é necessário estar familiarizado com diversos conceitos tecnológicos, principalmente se não tiver experiência com desenvolvimento de aplicativos da Web. A seguir é fornecida uma pequena lista de alguns dos conceitos que serão encontrados no aplicativo.

Arquivo EAR (Enterprise Archive)

Um arquivo EAR é um arquivo JAR (Java Archive) padrão, com extensão .ear. Esses arquivos podem conter vários projetos da Web e podem ser utilizados para compactar e implementar aplicativos da Web no WAS (WebSphere Administrative Server). **Nota:** Na versão GUI da ferramenta de implementação de aplicativos J2EE SDK, primeiro você cria um arquivo EAR e depois adiciona os arquivos JAR e WAR (Web Archive) no arquivo EAR. No entanto, se você utilizar as ferramentas de compactação da linha de comando, criará primeiro os arquivos JAR e WAR e depois o arquivo EAR.

IBM WebFacing Tool

A WebFacing Tool da IBM converte as interfaces 5250 existentes em interfaces gráficas com o usuário com base no navegador. Com pouca ou nenhuma modificação em seus aplicativos originais do iSeries, você pode estender o uso de seus programas para a Internet ou uma intranet.

Arquivo JAR (Java Archive)

Um arquivo JAR é um pacote compactado de arquivos Java, semelhante a um arquivo .zip. Ele contém os arquivos de classe, de imagem e de som para um applet Java, reunidos em um mesmo arquivo e compactados para facilitar o download para o navegador.

JSP (Java Server Pages)

JSPs fornecem a capacidade de exibir conteúdo dinâmico em páginas HTML estáticas. Gravados em Java, os JSPs são independentes do servidor e da plataforma. Separando com eficiência a apresentação na Web do conteúdo da Web, JSPs podem ajudar desenvolvedores que precisem alterar rapidamente o design e a exibição de suas páginas da Web.

Bean de Chamada de Programa

Estes são os beans Java gerados pelo assistente de Chamada de Programa. Um tipo é um bean Java regular utilizado por aplicativos Java. O outro tipo também pode ser utilizado pelo assistente de serviços da Web para criar um serviço da Web.

Assistente de Chamada de Programa

O assistente de Chamada de Programa ajuda a criar os beans Java e o arquivo PCML associado, necessários para chamar um programa ou procedimento do iSeries. O assistente solicita as informações referentes a objetos do programa ou do programa de serviços, juntamente com os parâmetros dos objetos e, em seguida, cria os beans Java desejados (e o arquivo PCML).

RPG (Report Program Generator)

Uma linguagem de programação de procedimentos utilizada por programadores do iSeries. A linguagem RPG pode ser utilizada para criar aplicativos de negócios como programas de fatura e programas de entrada de pedidos. A versão mais recente, ILE RPG IV, expande os recursos da linguagem RPG, ao mesmo tempo que oferece suporte à experiência do programador com versões anteriores.

Servlet

Programas no lado do servidor, escritos em Java, executados em servidores ativados para Java ou servidores de aplicativos como IBM WebSphere Application Servers. Os servlets executam tarefas especificadas pelo servidor, como resposta a pedidos gerando uma resposta HTML. Por exemplo, os servlets podem ser utilizados em um aplicativo de banco on-line para responder ao usuário enquanto envia dados para o servidor.

Componentes da Web

Os Componentes da Web podem ser utilizados para definir objetos do iSeries como campos de entrada de dados e botões de comando, que podem trocar informações entre programas servidores do iSeries e a página da Web. Os desenvolvedores podem utilizar Componentes da Web para capturar eventos do usuário, como verificação da sintaxe de campos de entrada e cliques em botões.

Assistente de Interação com a Web

Esse assistente faz parte das ferramentas de desenvolvimento da Web do iSeries. Ele cria e gerencia as interações entre programas ILE e páginas da Web. O assistente controla onde as mensagens de entrada, de saída e de erro são exibidas e direciona os dados dos campos de entrada e de saída para os programas ILE. O assistente de Interação com a Web também pode ser utilizado para mapear mensagens de erro para a área na qual o erro ocorreu, para que o usuário possa identificar facilmente a origem do erro.

Serviços da Web

Os serviços da Web são aplicativos independentes, desenvolvidos e implementados para uso na Internet. Eles são criados com padrões abertos, como SOAP, WSDL e XML. Existem diversas situações de negócios com as quais é possível utilizar um serviço da Web, inclusive em um sistema de gerenciamento de inventário no qual os clientes podem verificar pela Internet o nível do inventário ou, para acompanhar um pedido diretamente do fornecedor.

WSDL (Web Services Definition Language)

WSDL é uma linguagem com base em XML que define a interface de um serviço da Web. A WSDL entende um serviço da Web e gerencia o fluxo de informações entre esse serviço e o programa do servidor. Por exemplo, um desenvolvedor utilizaria WSDL para criar uma interface para um site na Web que mostre cotações de estoque atualizadas.

WebSphere Studio Workbench

O IBM WebSphere Development Studio Client para iSeries é criado no WebSphere Studio Workbench, a implementação IBM da plataforma Eclipse. O Workbench

extensível e universal integra todas as ferramentas necessárias para construir e manter aplicativos. Os desenvolvedores podem utilizar o Development Studio Client para incorporar novos objetos no ambiente de desenvolvimento utilizando plug-ins e podem adicionar de forma integrada arquivos Java, gráficos, vídeo, etc.

Capítulo 3. Executando o Cenário

Os aplicativos *Wholesale* e *Retail store* podem ser executados dentro do Workbench do Development Studio Client ou no WebSphere Application Server para qualquer plataforma, inclusive a do iSeries. Consulte Capítulo 1, “Introdução ao Aplicativo de Cenário da Loja de Varejo do iSeries”, na página 1 para obter uma visão geral dos aplicativos.

Ao final deste capítulo, você terá aprendido a:

- Restaurar os arquivos de amostra e objetos Development Studio Client para o servidor iSeries
- Criar projetos da Web e um projeto WebFacing para conter os arquivos
- Importar os arquivos restaurados do servidor iSeries para o Development Studio Client
- Configurar o WAS (WebSphere Application Server)
- Configurar o projeto WebFacing
- Executar o aplicativo como cliente no Ambiente de Teste do WebSphere
- Executar o aplicativo como administrador no WebSphere Test Environment
- Configurar o WebSphere Application Server
- Aplicar segurança na página do administrador
- Criar arquivo EAR para implementação externa do WAS
- Implementar o aplicativo no WAS
- Executar o aplicativo no WAS

Antes de Começar

Para testar os aplicativos a partir do o Ambiente de Trabalho, é necessário assegurar que:

- Você tem uma V5R1 ou posterior no servidor iSeries
- Possuir os PTFs do WebFacing mais recentes aplicados. Consulte nossa página de suporte para PTFs:
<http://www.ibm.com/software/awdtools/wdt400/support/>
- Você tem acesso NET USE ao servidor iSeries

Instalando os Arquivos de Amostra

Para utilizar o aplicativo de cenários do iSeries, é necessário trabalhar com os seguintes arquivos:

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

Os arquivos .savf contêm dados e programas RPG do iSeries e os arquivos .zip contêm os aplicativos da Web que interagem com os programas do iSeries para manipular os dados do iSeries. Primeiramente, é necessário restaurar os arquivos .savf para o servidor iSeries e, em seguida, você poderá importar os arquivos .zip para o Development Studio Client e executar o aplicativo no workbench.

Restaurando os Arquivos .savf

Para trabalhar com as amostras deste guia, é necessário restaurar as bibliotecas WHOLESale e RETAILSTOR para o servidor iSeries. Você deve fazer isso mesmo se já tiver restaurado as bibliotecas para releases anteriores do produto, porque seus conteúdos são diferentes.

Nota: Os arquivos .savf utilizados para instalar a biblioteca de amostra são para uso com a V5R1 ou posterior do servidor iSeries. Para este cenário, as duas bibliotecas são restauradas para o mesmo servidor iSeries, mas se você estivesse desenvolvendo este aplicativo para uma empresa real, restauraria as duas bibliotecas para dois servidores iSeries diferentes. A biblioteca WHOLESale seria restaurada para o servidor iSeries que fornece os serviços da Web e a RETAILSTOR, para o servidor iSeries que pertence à loja varejista.

Para restaurar o arquivo Wholesale.savf:

1. Efetue logon no servidor iSeries por meio de um emulador de tela verde.
 - a. Crie uma biblioteca para conter os arquivos de gravação. Para criar uma nova biblioteca no emulador, insira CRTLIB.
 - b. Nomeie a biblioteca como SCENARIO.
 - c. Avance para a próxima linha com a tecla Tab, especifique *TEST como tipo de biblioteca e pressione Enter para salvar as alterações.
 - d. Crie dois arquivos de gravação utilizando o comando CRTSAVF, pressionando Enter entre as duas linhas:

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

Essas linhas especificam que você deseja criar arquivos de gravação na biblioteca Scenario.
2. Na estação de trabalho, abra uma janela do Prompt de Comandos.
 - a. É necessário alterar para o diretório no qual os arquivos .savf estão. Por padrão, você deve inserir `cd c:\wdsc\wdscsaml`. Se você instalou o produto em uma unidade diferente ou escolheu não utilizar "wdsc" como o diretório inicial para o Development Studio Client, altere o diretório `wdscsaml` no local correto em que o produto foi instalado.
 - b. Na linha de comandos, insira: `ftp hostname`, em que *hostname* é o nome do servidor iSeries, por exemplo, `PROD400`.
 - c. Insira o ID do usuário e senha para o servidor iSeries.
 - d. Na linha de comandos, insira `cd /qsys.lib/scenario.lib` para alternar para a biblioteca Scenario.
 - e. Insira as seguintes linhas:

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

Essas linhas (antes de sair) especificam que você deseja transferir os arquivos de gravação do sistema local para o servidor iSeries.

3. De volta ao console do iSeries, restaure a biblioteca Wholesale:
 - a. Insira RSTLIB e pressione F4 para definir como deseja restaurar a biblioteca.
 - b. No campo **Saved Library**, insira WHOLESale e pressione a tecla Tab.
 - c. No campo **Device**, insira *savf e pressione a tecla Tab.
 - d. Pressione Enter no próximo campo para exibir valores adicionais e avance para o campo **Save file** com a tecla Tab.
 - e. Insira WHOLESale no campo **Save file** e pressione a tecla Tab.
 - f. No campo **Library**, exclua o valor existente e insira cenário.
 - g. Pressione Enter para restaurar a biblioteca WHOLESale para o servidor iSeries.
4. Repita esse procedimento para o arquivo Retailstor.savf:
 - a. Insira RSTLIB e pressione F4 para definir como deseja restaurar a biblioteca.
 - b. No campo **Saved Library**, insira RETAILSTOR e pressione a tecla Tab.
 - c. No campo **Device**, insira *savf e pressione a tecla Tab.
 - d. Pressione Enter no próximo campo para exibir valores adicionais e avance para o campo **Save file** com a tecla Tab.
 - e. Insira RETAILSTOR no campo **Save file** e pressione a tecla Tab.
 - f. No campo **Library**, exclua o valor existente e insira cenário.
 - g. Pressione Enter para salvar sua ação e restaurar a biblioteca RETAILSTOR para o servidor iSeries.

Modificando a Lista de Bibliotecas

Apesar de você ter restaurado as bibliotecas RETAILSTOR e WHOLESale para o servidor iSeries, é necessário assegurar que elas estão na lista de bibliotecas utilizada ao se conectar. Caso contrário, os aplicativos da Web podem não conseguir localizar as bibliotecas e podem ocorrer problemas durante a execução ou compilação dos aplicativos. É necessário também certificar-se de que a biblioteca QGPL esteja na lista de bibliotecas.

Para fazer isso, é necessário atualizar a descrição do job e o perfil do usuário para que as alterações na lista de bibliotecas sejam permanentes.

Usuários diferentes têm formas diferentes de modificar sua lista de bibliotecas, mas se você não souber como, este é um exemplo:

1. No servidor iSeries, digite `dsplibl` e pressione Enter para exibir a lista de bibliotecas .
2. Verifique se as bibliotecas RETAILSTOR, WHOLESale e QGPL estão na lista. Se estiverem lá, é possível sair e encerrar sua janela do emulador. Se não estiverem, proceda da seguinte forma:
 - a. Pressione F12 para cancelar a tela atual.
 - b. Localize o nome de sua descrição do job. Para fazer isso, digite `dspusrprf username` e pressione Enter para exibir seu perfil de usuário, em que *username* é o ID utilizado para se conectar ao servidor iSeries. A descrição do job está em uma das páginas do perfil do usuário (utilize Page Down). AO localizar sua descrição de job, anote-a e pressione F12 para sair da tela.
 - c. Na linha de comandos, digite `chgjobd` e pressione F4.
 - d. No campo **Job Description**, digite o nome da descrição do job e pressione F10 para obter parâmetros adicionais.
 - e. Pressione page down até a **Initial library list**.

- f. No campo abaixo **Initial library list**, você verá "+ for more values". Insira um sinal de mais e pressione Enter.
- g. Na página **Specify more values**, exclua qualquer entrada existente e digite RETAILSTOR, WHOLESale, e QGPL, tabulando entre cada valor. É mandatário inserir as bibliotecas nessa ordem, pois os aplicativos precisam localizar primeiro a biblioteca RETAILSTOR. Pressione Enter.
- h. Pressione Enter novamente, para salvar as alterações.
- i. Na linha de comandos, digite: chgusrprf e pressione Enter.
- j. Digite o nome de usuário e pressione Enter.
- k. Pressione F10 para consultar os parâmetros adicionais.
- l. Pressione page down até a **Job description**.
- m. Se o nome de sua descrição do job não corresponder à descrição do job localizada com o comando dspusrprf, digite o nome da descrição do job localizada com esse comando e pressione Enter para salvar suas alterações.

Importando e Executando o Aplicativo no Development Studio Client

Cinco arquivos zip se encontram nos arquivos de programas instalados para o cliente do Delopment Studio, por padrão, no diretório c:\wdsc\wdscsampl. Esses arquivos são:

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

Cada um será importado separadamente para o workbench do Development Studio Client e a cada um será atribuído um arquivo EAR (Enterprise Archive) correspondente. Um arquivo EAR é um arquivo JAR (Java Archive) padrão, com extensão .ear. Eles podem conter vários projetos da Web e podem ser utilizados para compactar e implementar aplicativos da Web no WAS (WebSphere Administrative Server).

Os quatro primeiros arquivos correspondem aos projetos da Web e o quinto corresponde a um projeto WebFacing. Primeiramente, crie os projetos da Web e depois, o projeto WebFacing. Em seguida, importe o conteúdo dos arquivos zip para os cinco projetos.

Extraindo Arquivos .zip em seu Sistema Local

Antes que seja possível importar o conteúdo dos arquivos .zip de cenário para o workbench do Development Studio Client, é necessário extrai-los para diretórios separados, de forma que você possa importá-los como sistemas de arquivos, ao invés de como arquivos .zip.

Nota: Se você estiver utilizando um programa que não seja o WinZip, execute as tarefas da mesma forma que com esse programa.

1. Em seu sistema local, navegue para c:\wdsc\wdscsampl. O diretório c:\wdsc pode ser diferente se você escolher instalar em uma unidade diferente ou em uma pasta base diferente. Nesse caso, navegue até esse diretório e, em seguida, localize \wdscsampl.
2. Dê um clique duplo no arquivo .zip a seguir: SV000501.zip.

3. Clique no botão **Extract** e navegue para a mesma pasta em que os arquivos zip se encontram, por padrão, c:\wdsc\wdscsampl.
4. Selecione o botão de opções **All Files** e, em seguida, clique em **Extract**.
5. Se você observar no diretório wdscsampl, você verá que uma pasta SV000501 foi criada, com todos os arquivos necessários dentro.
6. Agora que todos os arquivos foram extraídos para c:\scenario\SV000501, exclua o SV000501.zip, de forma que você não o selecione acidentalmente ao importar seu sistema de arquivos.
7. Repita as etapas desta seção para os outros quatro arquivos .zip (SV000618.zip, SV001585.zip, SV001586.zip e SV000514.zip.)

Criando os Projetos da Web

Nesta seção, você criará projetos da Web para SV000514, SV000618, SV001585 e SV001586. As etapas para criar cada projeto são descritas separadamente. Apesar de parecerem repetitivas, lembre-se que existem algumas diferenças, pequenas porém importantes, que devem ser observadas na criação dos quatro projetos (relativas a quais arquivo EAR são associados a quais projetos).

Para criar os projetos da Web:

1. Abra o IBM WebSphere Development Studio Client para iSeries no menu **Iniciar**.
2. Em **Development Studio Client**, abra a perspectiva da Web com a barra de menus, clicando em **Window > Open Perspective > Other > Web** e clique em **OK**.
3. Clique em **File > New > Dynamic Web Project**.
 - a. No campo **Project name**, insira SV000514.
 - b. Marque a caixa de opções **Configure advanced options** e clique em **Next**.
 - c. Clique no botão **New** ao lado do campo **EAR project**.
 - d. No campo **Project name**, insira SVWholeSaleEAR e clique em **Finish** nas duas caixas de diálogo.
4. Clique em **File > New > Dynamic Web Project**.
 - a. No campo **Project name**, insira SV001586 e clique em **Next** (A caixa de opções **Configure advanced options** deve ser marcada automaticamente).
 - b. Verifique se SWholeSaleEAR está selecionado no campo drop down **EAR project**.
 - c. Clique em **Finish**.
5. Clique em **File > New > Dynamic Web Project**.
 - a. No campo **Project name**, insira SV000618 e clique em **Next**.
 - b. Clique no botão **New** ao lado do campo **EAR project**.
 - c. No campo **New project name**, insira SVStoreEAR e clique em **Finish** nas duas caixas de diálogo.
6. Clique em **File > New > Dynamic Web Project**.
 - a. No campo **Project name**, insira SV001585 e clique em **Next**.
 - b. Verifique se SVStoreEAR está selecionado no campo drop down **EAR project**.
 - c. Clique em **Finish**.

Importando os Arquivos de Projeto da Web

Para importar os sistemas de arquivos para o espaço de trabalho, o procedimento a seguir descreve como importar os arquivos em SV000514. O procedimento é igual

para os outros três sistemas de arquivos do projeto da Web: SV001586, SV001585 e SV000618. A seção que se segue a esta explica como criar e importar o projeto do WebFacing.

Para importar arquivos:

1. Clique em **Browse** e navegue para
2. Clique com o botão direito na pasta **SV000514** e selecione **Import**.
3. A partir do assistente para **Import**, clique em **File System** e, em seguida, clique em **Next**.
4. Procure o diretório no qual os arquivos do cenário foram extraídos; por padrão, o diretório deve ser c:\wdsc\wdscsampl.
5. Clique em **SV000514** e clique em **OK**.
6. Clique em **Select All** para assegurar que todos os componentes do sistema de arquivos SV000501 sejam selecionados.
7. No campo **Into Folder**, insira **SV000514**, se ainda não estiver preenchido por padrão.
8. Selecione a caixa de opções **Overwrite existing resources without warning**.
9. Clique em **Finish** e em **Yes** se receber algum diálogo perguntando sobre a modificação da raiz de contexto.
10. Repita as etapas 2 a 9 para o SV001586, SV001585 e SV000618, certificando-se de utilizar nomes de projetos diferentes conforme concluir as etapas.

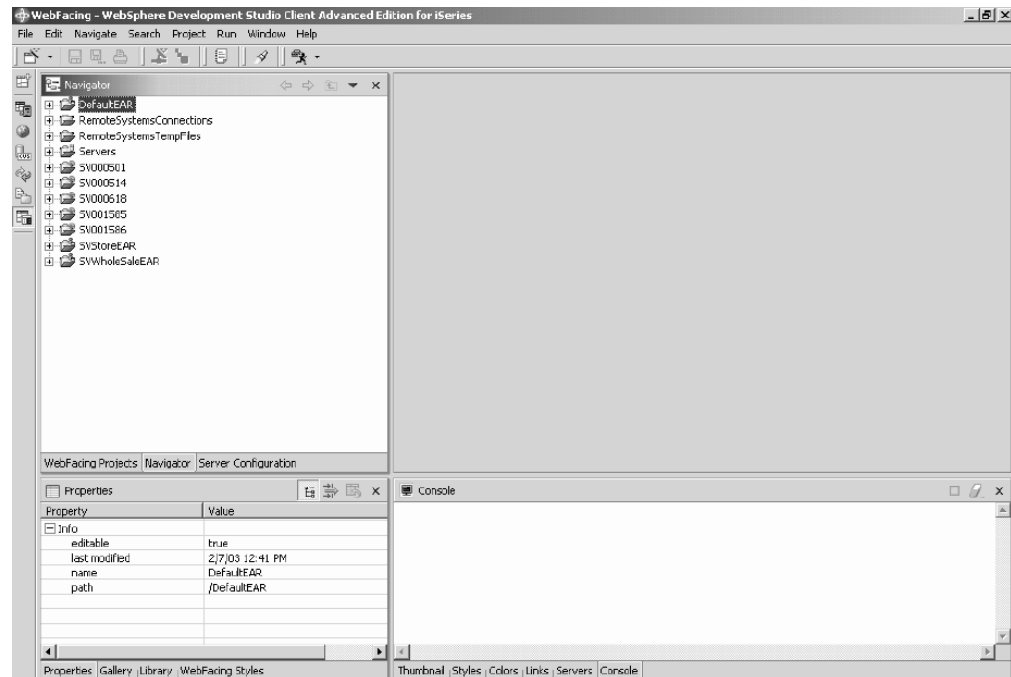
Importando Arquivos de Projetos do WebFacing e Criando o Projeto do WebFacing

O projeto SV000501 é diferente dos quatro projetos da Web que você acaba de criar, pois SV000501 é um projeto WebFacing. Os projetos WebFacing são diferentes porque incluem arquivos diferentes e têm uma estrutura exclusiva, específica para tratar arquivos DDS a partir do servidor iSeries.

Para importar os arquivos e criar um projeto WebFacing:

1. Se você não estiver na exibição Project Navigator, clique na guia **Project Navigator**.
2. Dê um clique duplo dentro da exibição e selecione **Import**.
3. Na janela Import, selecione **WebFacing Projects** e clique em **Next**.
4. Procure o diretório no qual os arquivos do cenário foram extraídos; por padrão, o diretório deve ser c:\wdsc\wdscsampl.
5. Selecione **SV000501** e clique em **OK**. Clique em **Next**.
6. Na área de janela **WebFacing projects found**, selecione a caixa de opções **SV000501**.
7. No campo **Enterprise Application Project (EAR)**, digite **SVStoreEAR**.
8. Clique em **Finish**.

Agora que todos os arquivos foram importados, o espaço de trabalho deve ter esta aparência:



Ao ver erros ao lado de seus projetos importados, (indicados por marcas de x vermelhas), clique com o botão direito em cada projeto e selecione **Rebuild Project**. Dependendo do que se encontra em seu espaço de trabalho, esse procedimento deve limpar a maioria dos erros, de forma que o aplicativo seja executado adequadamente. Qualquer erro ainda observado pode estar ali porque alguns códigos não foram completamente migrados para o release 5.1 do Development Studio Client. Você pode dar continuidade aos exercícios deste capítulo para se familiarizar com o produto, ou pode ir para o capítulo seguinte para tentar construir os aplicativos a partir do rascunho.

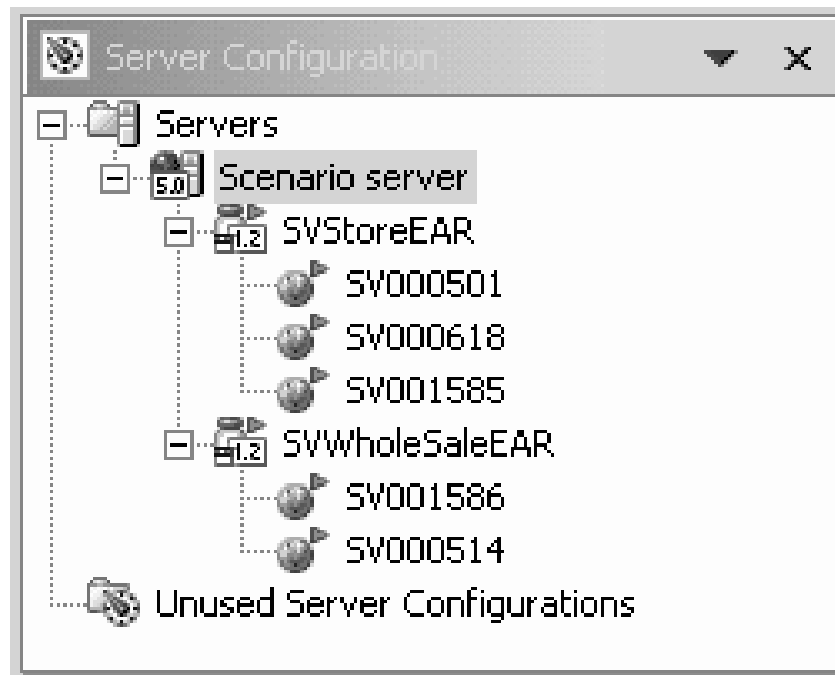
Configuração do Servidor

Agora que você importou todos os arquivos de origem, é necessário configurar o servidor do Ambiente de Teste do WebSphere para que ele reconheça os aplicativos SVWholesaleEAR e SVStoreEAR.

Para configurar o servidor do Ambiente de Teste do WebSphere:

1. Vá para a perspectiva Server. Na barra de menus do workbench, clique em **Window > Open Perspective > Other > Server** e clique em **OK**.
2. Na visualização Configuração do Servidor, clique com o botão direito do mouse em **Servers** e selecione **New > Server and Server Configuration**.
3. No campo **Server name**, insira Scenario server.
4. No campo **Folder**, insira Scenario folder.
5. Na caixa **Server type**, expanda o **WebSphere version 5.0** e clique em **Test Environment**, se ainda não estiver selecionado por padrão.
6. Clique em **Finish** e em **Yes** se receber alguma mensagem.
7. Para adicionar os arquivos EAR na configuração do servidor:
 - a. Expanda **Servers**.

- b. Clique com o botão direito do mouse em **Scenario server** e selecione **Add > SVWholeSaleEAR**.
 - c. Expanda **Servers**.
 - d. Clique com o botão direito do mouse em **Scenario server** novamente e selecione **Add and Remove Projects**.
 - e. Selecione/clique no botão **Add All** para mover SVStoreEAR e SVWholeSaleEAR para o lado direito da janela.
 - f. Clique em **Finish**.
8. Na visualização de configuração do servidor, expanda **Servidor do cenário** e, em seguida, expanda **SVStoreEAR** e **SVWholeSaleEAR** para ver todos os aplicativos relacionados:



Para se certificar de que o servidor obteve os projetos:

1. Na exibição Servers, à direita da exibição Server Configuration, (clique na guia **Servers** se não conseguir ver a exibição Servers), clique com o botão direito do mouse em **Scenario server** e selecione **Publish**. Uma caixa de diálogo **Publishing** será exibida, mostrando o andamento da publicação.
2. Clique em OK para publicar seus aplicativos no ambiente de testes do WebSphere Application Server.
3. Após a publicação, clique em **OK** na caixa de diálogo.
4. Na mesma visualização, clique com o botão direito do mouse em **Scenario server** novamente e clique em **Start**. A visualização Console será aberta, mostrando as atividades do servidor. O servidor estará iniciado quando a mensagem "Server *servername* open for e-business" aparecer na parte inferior da visualização. Role para baixo para ver todas as mensagens.

Configurando o Projeto WebFacing

Antes de executar o aplicativo ou de trabalhar com o módulo SV000501, é necessário iniciar o servidor WebFacing para execução pelo aplicativo e configurar o projeto para que ele utilize o servidor iSeries correto.

Para iniciar o servidor WebFacing:

1. Abra um emulador 5250 de tela verde e conecte-se com seu ID do usuário e senha.
2. Na linha de comandos, insira `strtcpsvr *webfacing`.
3. Na parte inferior da tela, será exibida a mensagem "WEBFACING server starting".

Agora é necessário alterar as propriedades do WebFacing, novamente no workbench do Development Studio Client:

1. Vá para a perspectiva da Web (para alternar entre perspectivas, clique nos ícones em cluster no lado esquerdo da tela).
2. Clique na guia **Project Navigator** para ver as estruturas de projeto.
3. Expanda **SV000501** e dê um clique duplo em **Web Deployment Descriptor**.
4. Clique na guia **Parameters**.
5. Clique em **WFDefaultHost**.
6. No campo **Value**, apague o valor existente e insira o nome do servidor iSeries.
7. Na barra de menus do workbench, clique no ícone Salvar ou em **File > Save Web Deployment Descriptor**. Se aparecer alguma mensagem, clique em **OK**. Feche o arquivo.

Definindo Informações sobre o Servidor

Agora é necessário definir as informações sobre o servidor iSeries para os cinco projetos, assegurando que estejam configurados para executar com o servidor iSeries com seu ID do usuário e senha. Para definir informações sobre o servidor:

1. Certifique-se de que está na perspectiva da Web.
2. Na visualização do Project Navigator, clique com o botão direito do mouse em **SV000501** e selecione **Specify iSeries Web Tools Run-time Configuration**.
3. No campo **iSeries server name**, exclua o valor existente e insira o nome do servidor iSeries.
4. Exclua os outros valores existentes e insira o ID do usuário e a senha.
5. Clique em **Finish** (e em **Finish** novamente, se necessário).
6. Repita as etapas de 2 a 5 para os outros quatro projetos: SV000514, SV000618, SV001585 e SV001586.

Executando o Aplicativo no Workbench

Neste ponto, é possível executar o aplicativo no Ambiente de Teste do WebSphere no workbench. Depois de verificar se ele está sendo executado corretamente, você pode exportar o arquivo EAR para o servidor iSeries e implementá-lo no WebSphere Application Server do iSeries.

Executando o Aplicativo como Cliente no Workbench

Como cliente, você começaria na página da Web de compras da loja varejista, consultaria os produtos e faria um pedido de uma determinada quantidade e tamanho.

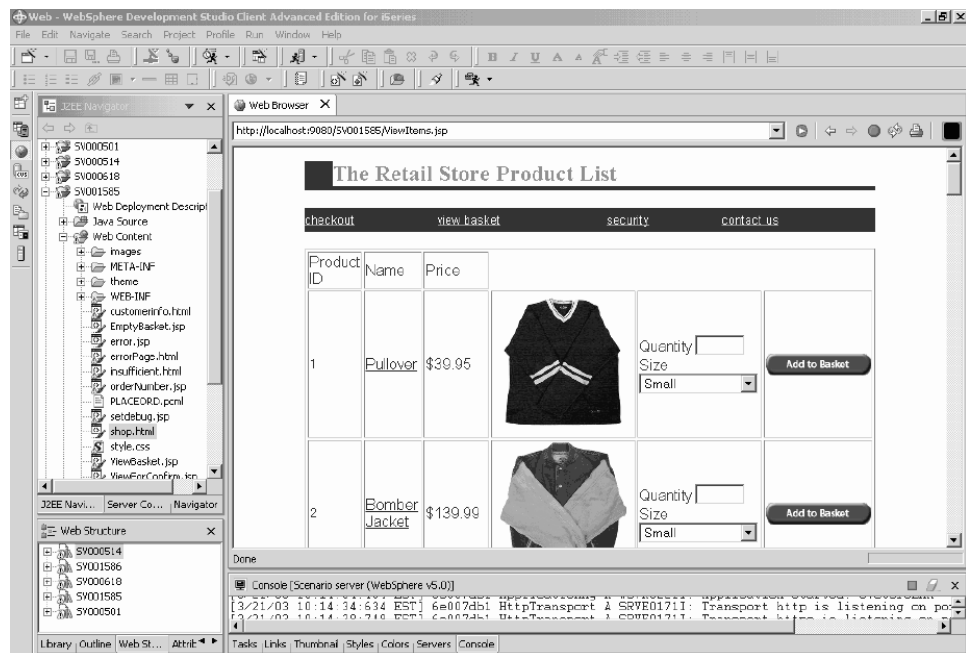
Nota: Podem ocorrer problemas se você tentar executar o aplicativo atrás de um firewall, pois o arquivo `web.xml` procura o seguinte arquivo: `http://java.sun.com/j2ee/dtds/web-app_2_2.dtd`. Para resolver esse problema, antes de executar o aplicativo, altere a instrução `DOCTYPE` em todos os arquivos `web.xml` para:

```
!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"
```

em que x:/wdsc é o diretório no qual o produto foi instalado. Geralmente, os arquivos web.xml podem ser localizados em *Project name* > **Web Content** > **WEB-INF**. Depois de abrir o arquivo, clique na guia **Source** para editar o arquivo diretamente.

Para executar o aplicativo como um cliente no Workbench:

1. Na perspectiva da Web, expanda **SV001585** > **Web Content**.
2. Clique com o botão direito do mouse em **shop.html** e selecione **Run on Server**. Essa ação ativa o aplicativo no navegador o Ambiente de Trabalho.
3. Clique em **Finish**.
4. Entre no aplicativo, clicando na imagem das camisetas.
5. Tente inserir valores na página seguinte, simulando que é um cliente fazendo o pedido de um Pulôver ou de uma Jaqueta, selecionando o tamanho e a quantidade e adicionando itens na cesta de compras:



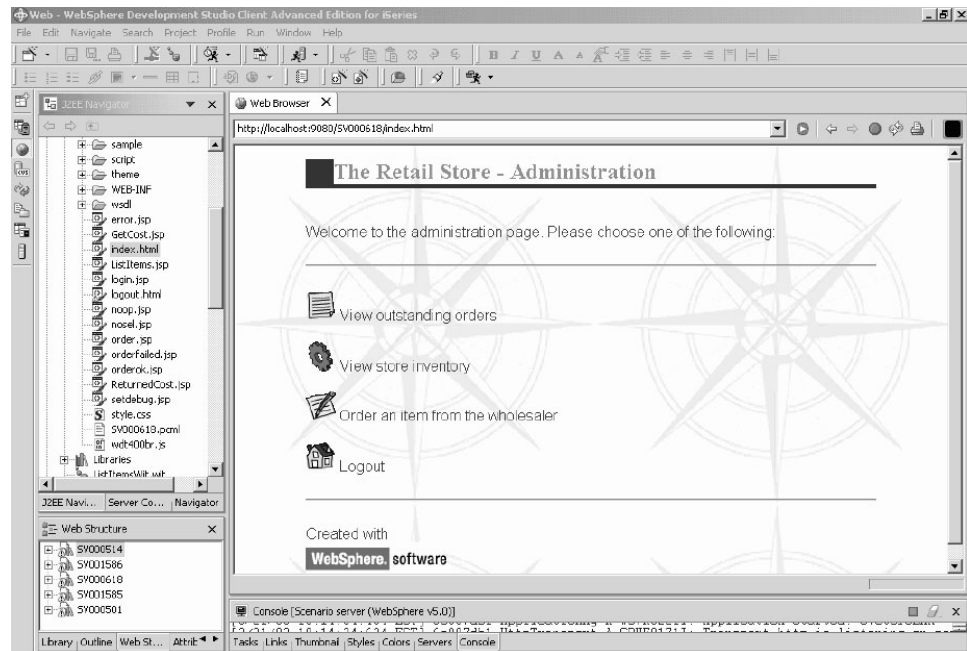
A página que exibe o texto "your order has been processed" é a última da amostra. Quando estiver satisfeito com a amostra, você poderá fechar o navegador.

Executando o Aplicativo como Administrador no Workbench

Como administrador, você começaria na página da Web de administração da loja varejista e verificaria os pedidos e o inventário da loja. Para executar o aplicativo como administrador no Workbench:

1. Na perspectiva da Web, expanda **SV000618** > **Web Content**.
2. Clique com o botão direito do mouse em **index.html** e selecione **Run on Server**. Essa ação ativa o aplicativo no navegador o Ambiente de Trabalho.
3. Clique no ícone à esquerda de **View store inventory** para exibir a página a seguir, como se você fosse um administrador decidindo o que comprar para a

loja:



Tente clicar nos ícones para exibir as páginas administrativas.

Implementando o Aplicativo de Cenários no WebSphere Application Server

Agora que você executou o aplicativo de cenários do iSeries no ambiente de teste do Ambiente de Trabalho, poderá implementar o aplicativo no WebSphere Application Server, como se estivesse executando o aplicativo real. Antes de implementar o aplicativo, é necessário fazer alguns ajustes, como aplicar segurança na página do administrador e alterar o URL de um serviço da Web para que o aplicativo indique o local correto, como explicado nas próximas seções.

Nota: A implementação no WebSphere Application Server é opcional. Você pode prosseguir com o próximo capítulo e concluir os módulos sem testar o aplicativo no WebSphere Application Server.

Configurando o WebSphere Application Server

Anteriormente no capítulo, discutimos o “Executando o Aplicativo no Workbench” na página 17. Agora que você implementou os arquivos no servidor iSeries, poderá executar o aplicativo no iSeries utilizando o WebSphere Application Server.

Para implementar o aplicativo no WebSphere Application Server (opcional), é necessário assegurar que:

- Você criou ou tem acesso a uma instância do WebSphere Application Server versão 4.0, 5.0 ou WebSphere Application Server Express no servidor iSeries e que a instância esteja em execução (somente se desejar testar a implementação no WebSphere Application Server).
- Você sabe os números de portas das instâncias do HTTP e do WebSphere Application Server no servidor iSeries.
- Você possui o WebSphere Administrative Console versão 4.0 ou 5.0 instalado na estação de trabalho.

Nota: Na versão 5.0 do WAS, o console se baseia no navegador, portanto, não é necessário ter um console na estação de trabalho. Consulte a documentação on-line para localizar informações adicionais sobre as diversas versões do WAS.

Seus aplicativos do iSeries ativados para Web utilizam WebSphere Application Server para executar os servlets Java e as JSPs (JavaServer Pages™) que fazem a comunicação entre o navegador da Web do usuário e os programas ou dados do iSeries. Para servir páginas HTML e arquivos JSP do mesmo Sistema iSeries, também é necessário ter um servidor HTTP nesse sistema. Recomenda-se a utilização do IBM HTTP Server powered by Apache. Você pode localizar a documentação sobre como utilizar esse servidor no: IBM HTTP Server for iSeries Documentation Center, em http://publib.boulder.ibm.com/pubs/html/iseries_http/v5r1/index.htm.

O WebSphere Application Server executa JavaServer pages, bean Java e servlets Java gerados por diversos processos. Os recursos básicos da documentação do IBM WebSphere Application Server para iSeries e do IBM WebSphere Administrative Console para iSeries estão disponíveis nos seguintes sites na Web:

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

É altamente recomendado familiarizar-se com a documentação do IBM WebSphere Application Server, principalmente as seções sobre *J2EE modules*, *Installing WebSphere Application Server* e *Setting up multiple instances of the WebSphere administrative server*. No mínimo, é necessário executar as etapas no link *Installation*.

Utilize o mapa do site para localizar informações sobre como instalar, configurar e obter os PTFs necessários para o WebSphere Administrative Console.

Aplicando Segurança na Página do Administrador

Como a página do administrador, `index.html`, deve ser acessível somente para pessoas autorizadas, é necessário aplicar segurança nela. Isto pode ser feito de maneira programática, na lógica do aplicativo da web, ou utilizando o recurso de segurança do WebSphere. Neste cenário, utilizamos a segurança do WebSphere para a página. Observe que utilizamos o WebSphere Application Server V4.0 Advanced Edition. Você localizará informações sobre o WebSphere Application Server em seus sites na Web:

- IBM WebSphere Application Server Versão 4.0 Advanced Edition for iSeries em <http://publib.boulder.ibm.com/was400/40/AE/english/docs/>
- IBM WebSphere Application Server Versão 4.0 Advanced Single Server Edition for iSeries em <http://publib.boulder.ibm.com/was400/40/AEs/english/docs/>
- Página dos RedBooks para WebSphere: Pesquise os RedBooks para WebSphere Application Server Versão 5.0 e WebSphere Application Server Express Edition Versão 5.0 em <http://publib-boulder.ibm.com/redbooks.nsf/portals/Websphere>

Se estiver utilizando uma versão diferente do WebSphere Application Server, consulte a documentação dessa versão a respeito de aplicação de segurança nos recursos da Web.

A segurança nos recursos da Web, como páginas e servlets, pode ser configurada dentro do Development Studio Client ou na Ferramenta de Montagem do Aplicativo. Neste cenário, utilizamos o Development Studio Client.

Para revisar a configuração e as propriedades de segurança deste aplicativo da Web:

1. No Development Studio Client, vá para a perspectiva da Web.
2. Na visualização do Navegador, expanda **SV000618 > Web Content > WEB-INF**.
3. Dê um clique duplo em **web.xml** para abrir a visualização **web.xml**.
4. Clique na guia **Security**.
5. Para aplicar segurança em **index.html**, na página de administração, uma restrição de segurança também foi definida. Na parte superior da visualização, clique em **Security Constraints**.
6. Clique na primeira instância de **SecurityConstraint** na lista.
7. À direita, clique em **AdminPage**.
8. Clique em **Edit** para chamar a caixa de diálogo **Web Resource Collections**. Observe que os métodos GET e POST para **index.html** estão pré-selecionados.
9. Clique em **OK**.

Na seção **Security roles**, observe a função de segurança definida, denominada "Administrator". Durante a implementação, as pessoas são atribuídas a essa função, obtendo acesso à página **index.html**. Na seção **Authorized roles**, observe que demos à função Administrator acesso a esta restrição de segurança. Com esta segurança configurada, somente os usuários designados com a função Administrator terão acesso à página **index.html**, depois de terem fornecido as credenciais corretas, como ID do usuário e senha. Quando houver segurança aplicada em um recurso, o WebSphere Application Server primeiro tenta autenticar o usuário. A autenticação é feita utilizando certificados ou solicitando ao usuário um ID do usuário e senha. A solicitação pode ser feita com o diálogo básico de autenticação ou com um formulário personalizado.

Neste cenário, desenvolvemos nossa própria página de logon, denominada **login.jsp**. Para configurar seu prompt de autenticação, selecione a guia **Pages** na visualização **web.xml**. Na seção **Login**, observe que **Form** está pré-selecionado como método de Autenticação. Observe também que o nome da página Login é **login.jsp**. A página Erro é exibida quando o logon não é bem-sucedido. Nesse caso, o aplicativo exibe novamente a página **login.jsp**.

Criando Arquivos EAR para Implementação do iSeries WebSphere Application Server

É necessário criar arquivos EAR para implementar seu aplicativo no iSeries WebSphere Application Server. Um arquivo EAR é um arquivo JAR (Java Archive) padrão, com extensão **.ear**. Esses arquivos podem conter vários projetos da Web e podem ser utilizados para compactar e implementar aplicativos da Web no WAS (WebSphere Administrative Server).

Para criar os arquivos EAR:

1. Vá para a perspectiva da Web.
2. Na visualização Navegador, clique com o botão direito do mouse em **SVStoreEAR** e selecione **Export**.
3. Na janela Export, clique em **EAR file** e em **Next**.

4. Em **Where do you want to export the resources to?**, clique em **Browse** e navegue para um diretório no IFS (Integrated File System) do iSeries no qual possa manter os arquivos EAR. (É necessário ter mapeado a unidade de rede para um IFS do iSeries).
5. Insira **SVRetailStorEAR.ear** no campo **File name** e clique em **Open**.
6. Clique em **Finish**.
7. Repita as etapas 2 a 6 para o **SVWholeSaleEAR**.

Implementando os Arquivos EAR no iSeries WebSphere Application Server

Agora que você criou os arquivos EAR, poderá implementá-los no WebSphere Application Server.

1. Abra o WebSphere Administrative Console.
2. Clique com o botão direito do mouse em **Enterprise Applications** e selecione **Install Enterprise Application**.
3. Selecione o botão de opções **Install Application (*.ear)**.
4. Clique no botão **Browse** superior (o inferior não estará disponível).
5. Navegue para o diretório IFS para o qual exportou os arquivos EAR.
6. Selecione **SVRetailStorEAR.ear**.
7. Insira "RETAILSTOR" no campo **Application name**.
8. Clique em **Next** e insira o ID do usuário do servidor iSeries para a função de administrador.
9. Clique em **Next** várias vezes, até chegar na página denominada **Selecting Virtual Hosts for Web Modules**.
10. Nos três módulos da Web, clique em **Select Virtual Host** e selecione o servidor virtual de sua preferência na lista drop down. (Se não souber qual selecionar, utilize **default** ou **default_host**).
11. Clique em **Next**.
12. Nos três módulos da Web, clique em **Select Server** e selecione o servidor que deseja utilizar. (Se não souber qual selecionar, utilize o **Default Server**).
13. Clique em **Next**.
14. Clique em **Finish** e em **OK** na caixa de diálogo.
15. Clique novamente com o botão direito do mouse em **Enterprise Applications** e selecione **Install Enterprise Application**.
16. Clique no botão **Browse** inferior (o superior não estará disponível).
17. Navegue para o diretório IFS no qual colocou os arquivos EAR.
18. Selecione **SVWholesaleEAR.ear**.
19. Insira "WHOLESALE" no campo **Application name**.
20. Clique em **Next** várias vezes, até que **Finish** esteja ativo.
21. Clique em **Finish** e em **OK** na caixa de diálogo.

Nota: Pode demorar alguns minutos para que apareça uma mensagem de confirmação.

Executando o Aplicativo no WebSphere Application Server

Para assumir a função do cliente e executar o ponto de entrada da loja varejista, insira o seguinte URL em um navegador da Web:

`http://your_iSeriesHostName:yourHTTPPortNumber/SV001585/shop.html`

Para assumir a função do administrador e executar o ponto de entrada WholeSale, insira o seguinte URL em um navegador da Web:

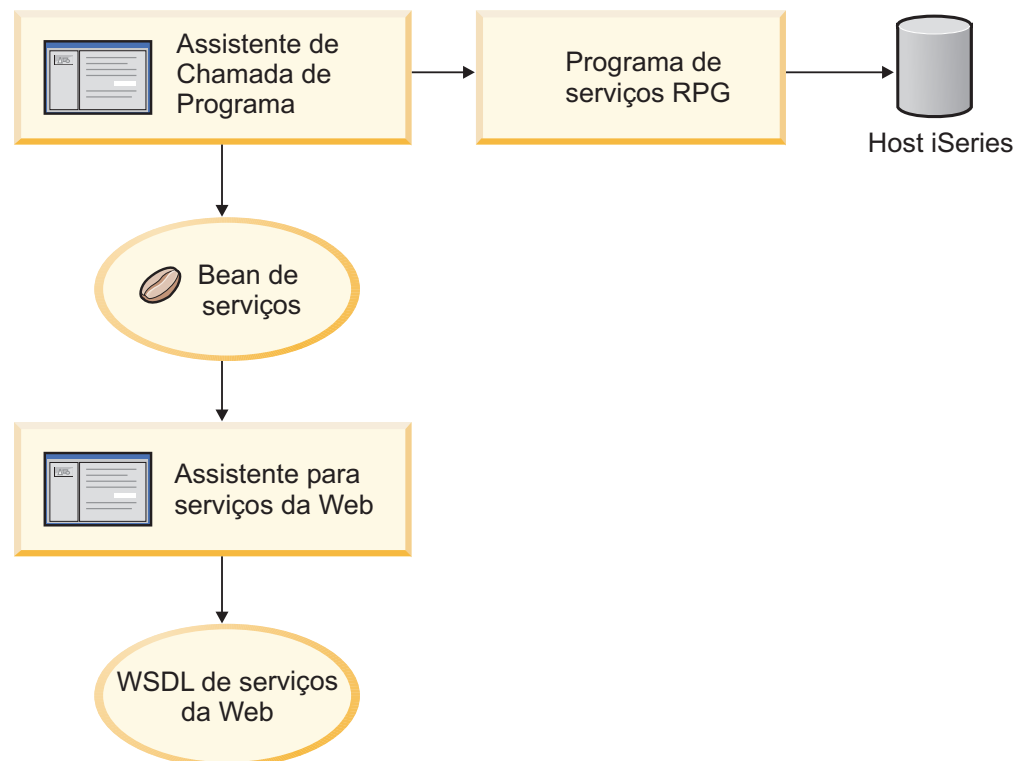
`http://your iSeriesHostName:yourHTTPPortNumber/SV000618/index.html`

Se não souber o número da porta HTTP, pergunte ao administrador do WebSphere Application Server.

Capítulo 4. Módulo Passo a Passo 1: Criar um Serviço da Web para Retornar Preços de Produtos (SV000514)

Introdução

Neste módulo, você criará um serviço da web a partir de um programa RPG existente no servidor iSeries para exibir preços de produtos para o inventário. Primeiramente, crie um programa de serviços RPG com um procedimento que possa recuperar o custo de um item do banco de dados do iSeries, fornecendo um número de item. Utilize o Assistente de Chamada do Programa das ferramentas de desenvolvimento Java do iSeries para chamar o programa RPG e criar um bean de serviços. Em seguida, utilize o assistente de Serviços da Web para criar um serviço da Web e utilize a amostra gerada para verificá-lo.



Antes de Começar

É possível concluir os exercícios somente se os pré-requisitos a seguir forem atendidos. Os pré-requisitos estão descritos em mais detalhes em Capítulo 3, "Executando o Cenário", na página 9.

- Você tem acesso TCP/IP a um servidor iSeries.
- Possuir os PTFs do servidor iSeries mais recentes aplicados. Consulte nossa página de suporte para PTF e as informações do Service Pack: <http://www.ibm.com/software/awdtools/wdt400/support/>
- Você iniciou os servidores do iSeries com o comando STRTCPSVR *ALL
- Você iniciou o servidor WebFacing com o comando STRTCPSVR *WEBFACING

- Você concluiu todas as tarefas do Capítulo 3, “Executando o Cenário”, na página 9, (exceto as tarefas opcionais do WebSphere Application Server, que não são obrigatórias para testar o aplicativo no o Ambiente de Trabalho).

Criando um Novo Projeto da Web

A primeira etapa na criação deste serviço da web é criar um novo projeto da Web para conter as informações.

1. A partir do IDE do Ambiente de Trabalho, vá para a perspectiva da Web ou abra a perspectiva, clicando em **Window > Open Perspective > Other > Web > OK**.
2. Clique em **File > New > Dynamic Web project**.
3. No campo **Project name**, insira Project514 .
4. Marque a caixa de opções **Configure advanced options** e clique em **Next**.
5. Clique no botão **New** ao lado do campo **EAR project**.
6. No campo **Project name**, insira Project514EAR e clique em **Finish** nas duas caixas de diálogo.

Agora, você poderá ver que os projetos Project514 e Project514EAR estão adicionados no espaço de trabalho na visualização Project Navigator.

Definindo as Informações do Servidor iSeries

Depois de criar o projeto da Web, é necessário definir qual servidor iSeries o projeto utiliza para obter informações.

1. Clique com o botão direito do mouse no projeto **Project514** e selecione **Specify iSeries Web Tools Run-time configuration**.
2. Insira o nome do servidor iSeries no qual a biblioteca WHOLESALE restaurada reside, por exemplo, PROD400.
3. Insira o ID do usuário e a senha deste servidor iSeries.
4. Digite Wholesale no campo **Library** e clique em **Add**.
5. Clique em **Finish**.

Criando o Programa de Serviços RPG

Você deseja que o aplicativo recupere o preço de um item, fornecendo o número do item. Isto é tratado por um programa de serviços do RPG que contém um procedimento denominado QryProdCost. A biblioteca WHOLESALE contém um programa de serviços do RPG denominado CWWSSRV. Esse programa contém um procedimento QryProdCost que pode considerar os números de itens como entrada, abrir o arquivo de Inventário dentro da biblioteca WHOLESALE, recuperar o preço do banco de dados de inventário e retornar esse preço. Para ajudar essa tarefa, a interface tem dois parâmetros, um para o número do item e outro para o preço. Se o número ou o preço do item não for localizado, o programa RPG retornará uma mensagem para a interface.

Para criar este serviço da Web, primeiro crie um bean Java com o assistente de Chamada do Programa, para chamar o procedimento RPG QryProdCost. Em seguida, utilize os serviços da Web para ativar o procedimento RPG como serviço da Web, por meio do bean Java.

Para criar o bean Java:

1. Na perspectiva da web, clique com o botão direito do mouse em **Project514** e selecione **New > Other**.
2. Selecione **iSeries > Java** a partir do painel esquerdo da janela e, em seguida, selecione **Program Call Bean** no painel direito.
3. Clique em **Next** para chamar o assistente de Chamada de Programa.
4. No campo **Java bean name**, debaixo do título **Add Program**, insira **Inventory**.
5. No campo **Program object**, insira **CWSSRV**, o nome do programa de serviços RPG.
6. No campo **Library**, insira **WHOLESALE**.
7. Na lista drop down **Program type**, selecione ***SRVPGM**.
8. No campo **Entry point**, insira **QryProdCost**.
9. Clique em **OK** para adicionar a definição do programa.

Criando os Parâmetros e Gerando o Bean Java

Agora que o programa foi especificado, você pode adicionar parâmetros. O programa RPG CWSSRV contém estes dois parâmetros:

- Parâmetro Número do item - O programa utiliza esse número para localizar o item no banco de dados.
- Parâmetro Custo do item - O programa utiliza esse número para localizar o custo do item no banco de dados.

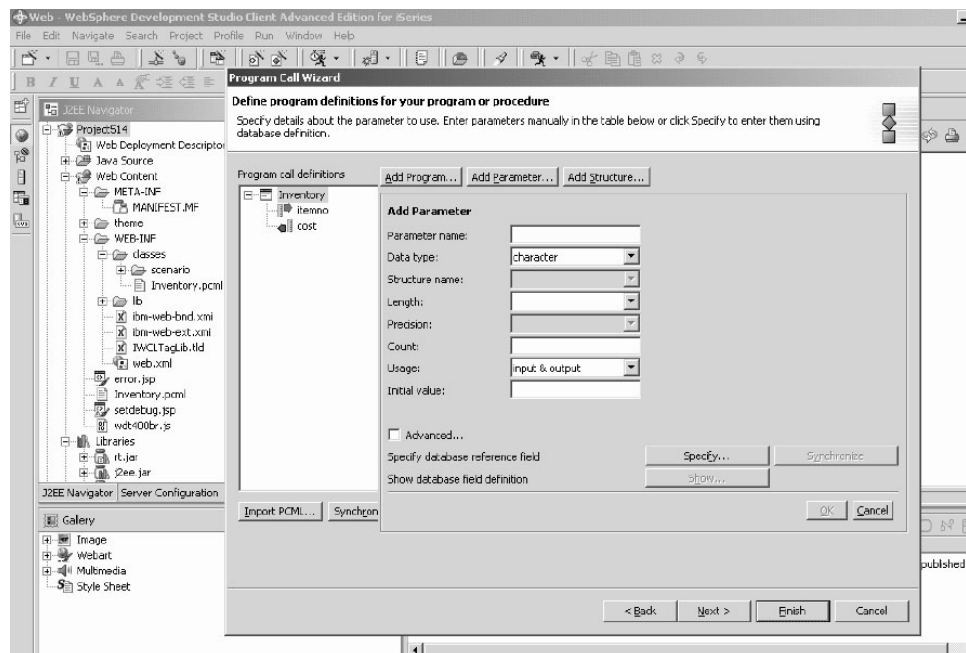
Para adicionar o parâmetro Custo do item:

1. No painel da esquerda do assistente de Chamada de Programa, clique na definição de chamada de programa **Inventory** para selecioná-la. Essa ação ocupa novamente os campos no lado direito da página.
2. Clique em **Add Parameter**.
3. No campo **Parameter name**, insira **itemno**.
4. Na lista drop down **Data type**, selecione **packed decimal**.
5. No campo **Length**, insira **5**.
6. No campo **Precision**, insira **0**.
7. Na lista drop down **Usage**, selecione **input**.
8. Clique em **OK** para adicionar esse parâmetro. No painel esquerdo, observe que **itemno** aparece abaixo de **Inventory**. Agora você está pronto para adicionar o segundo parâmetro.

Para inserir o parâmetro Custo do item:

1. No painel da esquerda do assistente de Chamada de Programa, clique novamente na definição de chamada de programa **Inventory** para selecioná-la. Essa ação ocupa novamente os campos no lado direito da página.
2. Clique em **Add Parameter**.
3. No campo **Parameter name**, insira **cost**.
4. Na lista drop down **Data type**, selecione **packed decimal**.
5. No campo **Length**, insira **7**.
6. No campo **Precision**, insira **2**.
7. Na lista drop down **Usage**, selecione **output**.
8. Clique em **OK** para adicionar esse parâmetro. No painel esquerdo, observe que **cost** aparece abaixo de **Inventory**. Neste ponto, o assistente terá esta aparência. Observe que o ícone à esquerda do parâmetro mostra se ele é do tipo entrada e

saída ou saída:



Para definir o nome do pacote para esses parâmetros, no assistente de Chamada de Programa:

1. Clique em **Next**.
2. No campo **Package**, insira **scenario** como nome do pacote. Deixe os padrões nos outros campos.
3. Limpe a caixa de entrada **Java Application**.

Nota: Revise a lista de arquivos em "These files will be generated by the wizard" e observe que o nome do bean Java gerado será **InventoryServices.java**.

4. Clique em **Finish** para gerar os arquivos.

Criando um Serviço da Web a partir do Bean Java

Depois de criar o bean Java que chama o programa RPG, a próxima etapa é converter o bean em serviço da Web para que outros programas possam acessar o mesmo programa RPG pela Internet.

Ao criar o serviço da Web, um assistente de Serviços da Web gera os arquivos WSDL que são distribuídos aos usuários que precisam utilizar o Serviço da Web. Para criar o serviço:

1. Na visualização do Navegador da perspectiva da Web, expanda **Project514 > Java Source > scenario**.
2. Clique com o botão direito do mouse em **InventoryServices.java** e selecione **New > Other**.
3. Na janela **New**, clique em **Web Services**, em **Web Service** no lado direito da janela e em **Next**.
4. Selecione as caixas de opções **Generate a proxy** e **Test the generated proxy**.
5. Clique em **Next** três vezes e observe a geração dos arquivos necessários. Ao surgir a janela **Web Service Java Bean Identity**, clique em **Finish**. Pode ser necessário clicar em **Finish** uma segunda vez se for observado algum erro. Se

você não conseguir concluir o assistente, apenas clique em **Cancel** e a maioria dos arquivos necessários será criada de qualquer forma.

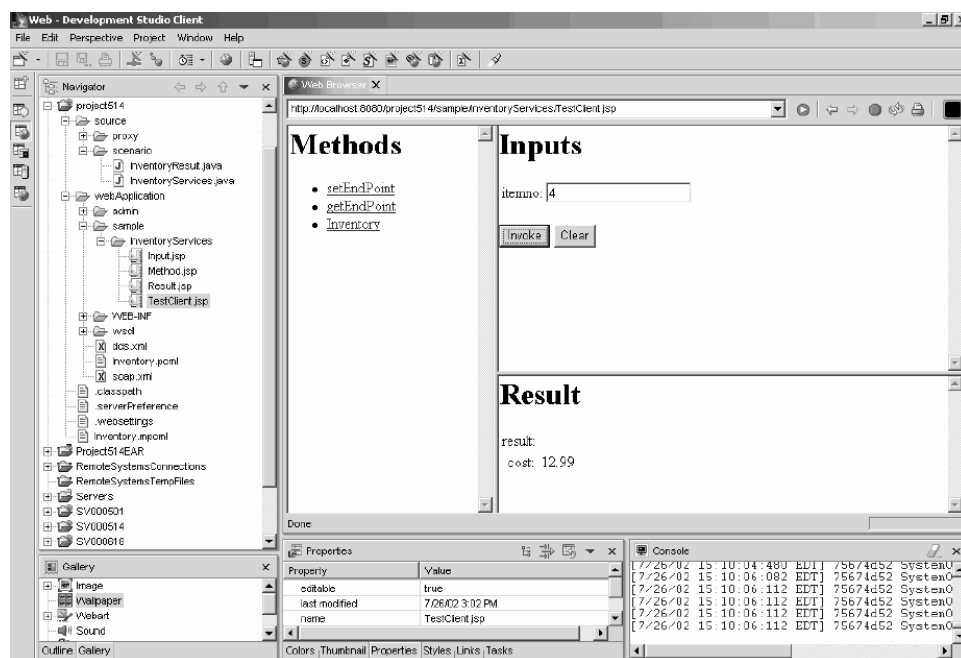
Testando a Amostra

Quando você criou o serviço da Web, uma das instruções foi solicitar a geração de uma amostra. Como essa opção foi selecionada, o assistente Serviços da Web criou um novo projeto com as páginas de teste, que pode ser utilizado para testar o serviço da Web. Na visualização **Navigator**, observe que o nome desse novo projeto é **Project514Client**. Observe também que a ferramenta abriu automaticamente **TestClient.jsp**, no lado direito do workbench. Esse arquivo está localizado em **Project514Client > Web Content > sample > InventoryServices**.

Para testar a amostra:

1. Na página carregada, role para baixo a área de janela de métodos até o fim. Clique no método **Inventory (java.math.BigDecimal)**.
2. No campo **itemno**, insira 4 e clique em **Invoke**.
3. Verifique se a saída retornada é:

result:
cost: 12.99

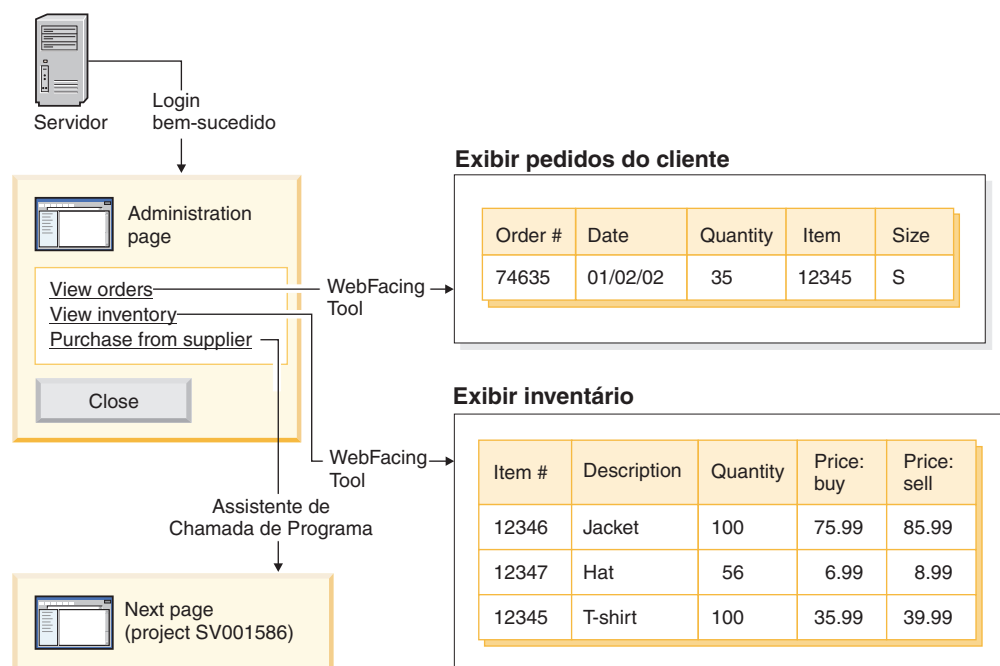


Quando o resultado for confirmado, você terá concluído o módulo. Você criou o serviço da Web que retorna preços de produtos na interface do aplicativo.

Capítulo 5. Módulo Passo a Passo 2: Criar uma Interface para Visualizar Inventário e Itens de Pedidos (SV000501)

Introdução

Neste módulo, você criará uma interface que exibe as opções de operação de um administrador, utilizando WebFacing Tool da IBM. Após um login com êxito no servidor iSeries, a interface exibe o inventário disponível e você poderá verificar pedidos existentes. Como administrador, você poderá comprar mercadorias do fornecedor atacadista.



Neste projeto, você trabalha com dois programas e dois arquivos de exibição restaurados para o servidor iSeries. Os nomes dos programas são ViewInventory e ViewOrder. Eles utilizam os seguintes arquivos de exibição: ORDERDSP e QUERY. Esses dois arquivos contêm definições da Web personalizadas para os arquivos JSP gerados pelo WebFacing Tool e são utilizados para imagens e hyperlinks. A definição da Web da imagem permite utilizar o conteúdo de um campo para gerar o nome do arquivo de imagem e exibi-la no arquivo JSP. A definição da Web do hyperlink permite chamar outro aplicativo da Web quando você clica na imagem no arquivo JSP. É possível utilizar o CODE Editor ou o CODE Designer para verificar o código fonte do arquivo de exibição, para determinar como especificar as definições da Web.

Antes de Começar

É possível concluir os exercícios somente se os pré-requisitos a seguir forem atendidos. Os pré-requisitos estão descritos em mais detalhes em Capítulo 3, "Executando o Cenário", na página 9.

- Você tem acesso TCP/IP a um servidor iSeries.

- Você iniciou os servidores do iSeries com o comando STRTCPSVR *ALL
- Você iniciou o servidor WebFacing com o comando STRTCPSVR *WEBFACING
- Você concluiu todas as tarefas do Capítulo 3, “Executando o Cenário”, na página 9, (exceto as tarefas opcionais do WebSphere Application Server, que não são obrigatórias para testar o aplicativo no o Ambiente de Trabalho).

Criando um Projeto WebFacing

A primeira coisa a ser feita é criar um projeto WebFacing e especificar os comandos CL relevantes. Para criar o projeto WebFacing:

1. No o Ambiente de Trabalho, alterne para a perspectiva WebFacing, clicando em um dos ícones de perspectiva no lado esquerdo do espaço de trabalho ou clicando em **Window > Open Perspective > Other > WebFacing** e em **OK**.
2. Crie um novo projeto WebFacing, clicando em **File > New > Project WebFacing**.
3. Dê ao projeto o nome Project501.
4. Clique no botão de opções **Existing** ao lado de **Enterprise Application Project**.
5. Insira SVStoreEAR no valor do campo **Existing project name**; não se o valor fizer distinção entre maiúsculas e minúsculas. Clique em **Next**.
6. Na lista drop down **J2EE level**, selecione **1.3**. Clique em **Next**.
7. No campo **Connection**, o nome de seu servidor iSeries deve ser preenchido automaticamente. Caso contrário, clique em **New** e insira o nome do servidor iSeries no diálogo e clique em **Finish**.
8. Clique em **Refresh DDS list** e insira a senha e o ID do usuário no diálogo popup, para atualizar a lista.
9. Na lista de bibliotecas gerada, expanda **RETAILSTOR**.
10. Clique em **QDDSSRC** e, em seguida, na seta para a direita (>), a fim de mover os arquivos.
11. Clique em **Next** duas vezes, até chegar à página **Specify CL commands**. Agora é necessário adicionar um comando e um rótulo de comando para o administrador:
 12. No campo **CL command**, insira `call viewinvent`.
 13. No campo **Command label**, exclua os valores existentes e insira `View inventory`.
 14. Selecione o botão de opções **Sign on with specified values**.
 15. Clique em **Add** e observe a adição na lista, na parte inferior da janela. Agora é necessário adicionar um segundo comando e um rótulo de comando para o cliente:
 16. No campo **CL command**, exclua o valor existente e insira `call vieworder`.
 17. No campo **Command label**, exclua os valores existentes e insira `View orders`.
 18. Selecione o botão de opções **Prompt for signon**.
 19. Clique em **Add** e observe a adição na lista, na parte inferior da janela. Clique em **Next**.
 20. Na janela **Choose a Web style**, role pelos estilos para ver o que está disponível. Como precisamos recuperar informações deste estilo depois, selecione **avenue** para este módulo. Clique em **Next**.
 21. Verifique se a caixa de opções **No. I only want to create the project now** está selecionada e clique em **Finish**.

22. Se aparecer uma mensagem Reparar configuração do servidor, clique em **OK**. Essa mensagem é apenas para confirmar se você deseja que o projeto seja adicionado no arquivo SVStoreEAR.

Como você está construindo o Project501 como espelho do projeto SV000501 completo, agora é necessário copiar alguns dos arquivos de imagem do SV000501 para o Project501, para que o Project501 seja exibido corretamente (Se desejar, você pode adicionar suas próprias imagens na interface). Para copiar as imagens:

1. Vá para a perspectiva da Web.
2. Na visualização do Navegador, expanda **SV000501 > Web Content > imagens**.
3. (Opcional) Expanda **generated** para anotar todas as imagens que estão sendo importadas para o projeto.
4. Clique com o botão direito do mouse em **generated** e selecione **Copy**.
5. Role para cima na visualização Navegador, até ver **Project501** novamente.
6. Expanda **Project501 > Web Content**.
7. Clique com o botão direito do mouse na pasta **images** e selecione **Paste** no menu pop-up.
8. Agora, a pasta **generated** existe para o Project501. Expanda **generated** para anotar as imagens adicionadas.

Convertendo a Origem DDS

Agora que você criou o projeto, poderá converter os arquivos de exibição DDS em arquivos JSP para a página da Web. Quando você converte os arquivos de exibição DDS, a WebFacing Tool gera JSPs e arquivos XML que substituem o código DDS e tornam possível o acesso à Web. Os arquivos gerados contêm os dados do formato de registro ou controlam sua aparência, exibição da versão da Web da tela, prompt de dados e tratamento de erros de entrada. O assistente também gera uma home page do aplicativo para ativar a versão do programa ativada para a Web.

No entanto, primeiro é necessário alterar a origem para fazer referência ao servidor iSeries correto. Mais especificamente, é necessário fazer uma modificação em um dos arquivos DDS de exibição, para que o link do aplicativo funcione. É necessário alterar o nome padrão, SV000501, para o nome do projeto, Project501:

1. Volte para a perspectiva WebFacing e alterne para a visualização WebFacing Projects se esta ainda não estiver aberta por padrão.
2. Expanda **Project501 > DDS**.
3. Dê um clique duplo na segunda entrada,
`<servidor_iSeries>RETAILSTOR/QDDSSRC(QUERY)`
4. Role para baixo até aproximadamente a linha 16, na qual poderá ver a linha:
`A*%WB 12 FLD 1 next ('/SV000501/DetailPage.do?PRODNO=&{PRODNO}')`
5. Exclua SV000501 e insira Project501.
6. Salve e feche o arquivo.

Nota: Se em algum ponto você desejar executar o projeto SV000501 novamente, deverá retornar o valor Project501 para SV000501, já que esse arquivo existe no servidor iSeries e é acessado pelos dois projetos.

Para converter a origem DDS:

1. Na perspectiva WebFacing, clique na guia **WebFacing Projects** para alternar para a visualização WebFacing Projects se esta já não estiver aberta por padrão.
2. Expanda **Project501 > DDS**.

3. Selecione *iSeriesserver* > **RETAILSTOR/QDDSSRC(ORDERDSP)** e *iSeriesserver* > **RETAILSTOR/QDDSSRC(QUERY)**, clicando no primeiro e mantendo a tecla Shift pressionada enquanto clica no segundo.
4. Clique com o botão direito e selecione **Convert** para iniciar a conversão. Pode ser necessário inserir o ID do usuário e a senha, se em algum ponto você for desconectado do servidor iSeries.

Configurando Suporte a UTF-8 no Ambiente de Trabalho — Somente para Usuários do WAS Versão 4.0

Os aplicativos WebFacing Tool da IBM oferecem suporte à exibição de vários idiomas em uma tela. Como os idiomas utilizam conjuntos de caracteres diferentes, os fluxos de dados entre o navegador e o WebSphere Application Server têm codificação UTF-8. Para que WebFacing Tool da IBM funcione corretamente, é necessário configurar o suporte a UTF-8 no arquivo de propriedades do aplicativo no o Ambiente de Trabalho.

Nota: Esta seção é válida somente para usuários do WAS 4.0. O WAS versão 5.0 executa esta tarefa automaticamente.

Para configurar suporte a UTF-8:

1. Vá para a perspectiva da Web (para alternar entre perspectivas, clique nos ícones em cluster no lado esquerdo da tela).
2. Clique na guia **Project Navigator** para ver as estruturas de projeto.
3. Expanda **Project501** e dê um clique duplo em **Web Deployment Descriptor**.
4. Clique na guia **Environment**.
5. Clique no botão **Add**.
- 6.
7. Vá para a perspectiva Servidor, clicando em seu ícone na barra de tarefas superior esquerda, ou clique em **Perspective > Open > Other > Server** na barra de menus e em **OK**.
8. Na visualização Navegador, expanda a pasta **Servers**.
9. Dê um clique duplo em **defaultInstance.wsi** para abri-lo no editor padrão.
10. Clique na guia **Environment** e no botão **Add**.
11. Será exibido um valor editável, "(New Variable)". Exclua essa cadeia padrão e insira: `client.encoding.override`.
12. No campo **Value**, insira UTF-8.
13. Clique no ícone Salvar ou em **File > Save Web Deployment Descriptor**.

Configurando Suporte a UTF-8 para o WebSphere Application Server

(Opcional) Para implementar o aplicativo iSeries no WebSphere Application Server, também é necessário configurar suporte a UTF-8 no WebSphere Application Server e no o Ambiente de Trabalho.

Para Configurar Suporte a UTF-8 no WebSphere Application Server 4.0 Advanced Edition

1. Inicie o WebSphere Administrative Console.
2. Expanda o ícone **Nodes** e **Node name > Application servers > Default server**.
3. Selecione a guia **JVM Settings** e clique no botão **Advanced JVM settings** para abrir o diálogo **Advanced JVM settings**.

4. No campo **Command line arguments**, insira:
`-Dclient.encoding.override=UTF-8`
5. Clique em **OK** e em **Apply** na guia **JVM settings**.
6. Para que esta alteração seja efetivada nos aplicativos WebSphere, pare o servidor padrão e, em seguida, inicie-o novamente. Para parar o servidor, clique com o botão direito do mouse em **Default server** e selecione **Stop**. Ao concluir esse processo, clique com o botão direito do mouse em **Default server** e selecione **Start**.

Para Configurar Suporte a UTF-8 no WebSphere Application Server 4.0 Advanced Single Server Edition

1. Inicie o WebSphere Administrative Console.
2. No Console Administrativo com base no navegador, expanda o ícone **Nodes** e **Node name > Application servers > Default server > Process definition > JVM settings**.
3. Role até a seção Definições Avançadas da página Definições de JVM e clique no link Propriedades do sistema. A página de Propriedades do Sistema é exibida.
4. Clique em **New** para adicionar uma nova Propriedade do Sistema.
5. No campo **Name**, insira `client.encoding.override`.
6. No campo **Value**, insira `UTF-8`.
7. Clique em **OK**. Se for exibida uma mensagem **Configuration needs to be saved** com um link na parte superior da página **JVM settings**, clique no link para ir para a página **Save configuration**. Selecione **Save** e clique em **OK**.
8. Para que esta alteração seja efetivada nos aplicativos do WebSphere, pare o servidor de aplicativos e, em seguida, inicie-o novamente. O método para parar e iniciar o servidor de aplicativos pode variar, dependendo da plataforma na qual o WebSphere Application Server foi instalado. Consulte a documentação do WebSphere Application Server para sua plataforma para obter informações sobre como parar e iniciar o servidor de aplicativos.

Criando uma Página de Estilo

Para integrar páginas adicionais com uma CSS (Cascading Style Sheet), é necessário personalizar o estilo do projeto WebFacing ou a CSS, para que fiquem parecidos. Depois de personalizar a página de estilo, você poderá utilizar o assistente de Interação com a Web para criar uma página da Web detalhada, utilizando a página de estilo, para exibir informações sobre itens da loja, como preço e cor. Neste módulo, você vai incorporar a página de estilo `DetailPageResults.jsp` do projeto `SV000501`. Para referência futura, você pode personalizar manualmente o arquivo CSS (descrito na próxima seção).

(Opcional) Personalizando Manualmente uma Página de Estilo em Cascata

Conforme mencionado anteriormente, em vez de incorporar a página de estilo `DetailPageResults.jsp` do projeto `SV000501` para o `Project501`, você pode personalizar suas próprias páginas de estilo:

1. Alterne para a visualização do Project Navigator e expanda **Project501 > Web Content > styles > apparea**.
2. Dê um clique duplo em **apparea.css** para abrir o arquivo no Designer CSS. Essa página de estilo controla a aparência da área do aplicativo.
3. Modifique e salve o arquivo.

4. De volta na visualização do Project Navigator, navegue para **Project501 > Web Content > styles > chrome**.
5. Dê um clique duplo em **avenue.css** para abrir o arquivo no Designer CSS. Essa página de estilo controla a aparência da página inteira.
6. Modifique e salve o arquivo.

Estendendo e Aprimorando seu Projeto WebFacing com uma Interação da Web

Nesta seção, você vai aperfeiçoar o projeto WebFacing. Quando a lista de itens for exibida, você tem que clicar em uma imagem de item para ver detalhes sobre esse item. Para isto, utilize o assistente de Interação com a Web para chamar um programa RPG para recuperar os detalhes do item e exibi-los em outra página da Web. Você executará as seguintes etapas:

- definir as informações do servidor
- copiar a página de estilo correta
- criar a interação
- adicionar programas e parâmetros na interação
- alterar o uso dos parâmetros

Primeiro, é necessário definir as informações sobre o servidor:

1. Vá para a perspectiva da Web.
2. Na visualização Navigator, clique com o botão direito do mouse em **Project501** e selecione **Specify iSeries Web Tools run-time configuration**.
3. Insira o nome do servidor iSeries, o ID do usuário e a senha.
4. Digite **Retailstor** no campo **Library** e clique em **Add**.
5. Clique em **Finish**. Clique em **Finish** novamente, se necessário.

Antes de criar a Interação com a Web, é necessário copiar os estilos e páginas de estilo corretos, para que o Project501 exiba o formato de arquivo JSP correto (se você tiver concluído o exercício “(Opcional) Personalizando Manualmente uma Página de Estilo em Cascata” na página 35, não será necessário executar esta tarefa). Primeiro, copie a pasta de estilos e, em seguida, copie o **DetailPageResults.jsp**:

1. Na visualização do Navegador, expanda **SV000501 > Web Content**.
2. Clique com o botão direito do mouse em **Styles** e selecione **Copy**.
3. Role para cima na visualização Navegador, até ver **Project501** novamente.
4. Expanda o projeto, clique com o botão direito do mouse em **Web Content** e selecione **Paste** no menu pop-up. Clique em **Yes** para sobrescrever os estilos existentes.
5. Role para baixo, até **SV000501 > Web Content**.
6. Clique com o botão direito do mouse em **DetailPageResults.jsp** e selecione **Paste**.
7. Role para cima, até ver **Project501** novamente.
8. Expanda o projeto, clique com o botão direito do mouse em **Web Content** e selecione **Paste** no menu pop-up.

Observe os novos elementos adicionados abaixo da pasta Conteúdo da Web do Project501.

Agora, você poderá criar a Interação com a Web:

1. Clique em **File > New > Other**
2. No diálogo **New**, clique em **Web** no lado esquerdo e em **Web Interaction** no lado direito. Clique em **Next**.
3. No campo **Web Interaction name**, insira **DetailPage** e clique em **Next**.
4. Selecione o botão de opções **Generate Input JSP**.
5. Selecione o botão de opções **Use output pages** (se ainda não estiver selecionado) e clique em **Add**.
6. No diálogo **Output JSP**, expanda **Web Content**, selecione **DetailPageResults.jsp**, a página de saída pré-formatada e clique em **OK**.
7. Clique em **Next**.

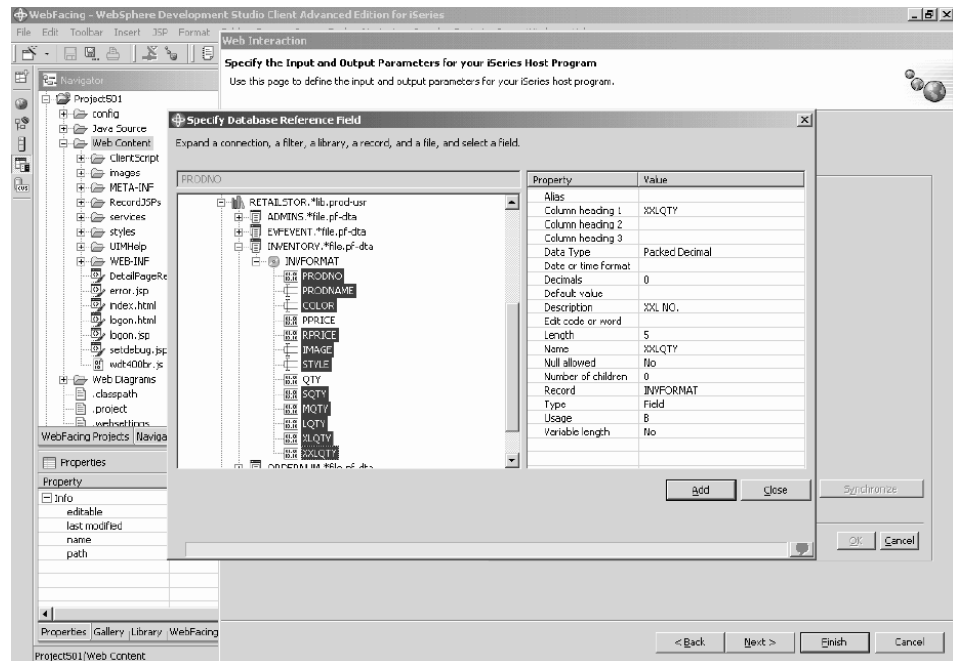
Agora, você poderá adicionar programas e parâmetros na interação. É necessário adicionar 11 parâmetros no mesmo programa. Em vez de adicionar cada parâmetro com seus valores individuais manualmente, adicione-os de uma forma um pouco mais rápida:

1. Selecione **Use an iSeries ILE program**, se ainda não estiver selecionado.
2. Clique em **Add Program**.
3. No campo **Program alias**, insira **DetailPage**.
4. No campo **Program object**, clique em **Browse**.
 - a. Expanda *iSeriesserver* > ***LIBL > RETAILSTOR**.
 - b. Clique em **DETAILPAGE.*pgm.rpgle** (o primeiro arquivo abaixo de RETAILSTOR) e clique em **OK**.
5. De volta para o assistente de Interação com a Web, clique em **OK** (no canto inferior direito).

Observe que **DetailPage** foi adicionado no lado esquerdo do assistente, abaixo de Definições de chamada do programa.

6. Na seção Definições de chamada do programa no painel esquerdo do assistente, clique em **DetailPage** para selecionar essa opção.
7. Clique em **Add Parameter**.
8. Ao lado de **Specify database reference field**, na parte inferior do assistente, clique em **Specify**.
9. Expanda *iSeriesserver* > ***LIBL > RETAILSTOR > INVENTORY.*file.pf-dta > INVFORMAT** para exibir uma lista de 13 parâmetros. É necessário adicionar os 11 parâmetros a seguir, clicando uma vez em cada um deles e em **Add**, ou mantendo a tecla CTRL pressionada enquanto clica em cada um deles e clicando em **Add**: **PRODNO, PRODNAME, COLOR, RPRICE, IMAGE, STYLE, SQTY, MQTY, LQTY, XLQTY, XXLQTY**. Basicamente, você selecionará todos os

campos, exceto PPRICE e QTY.

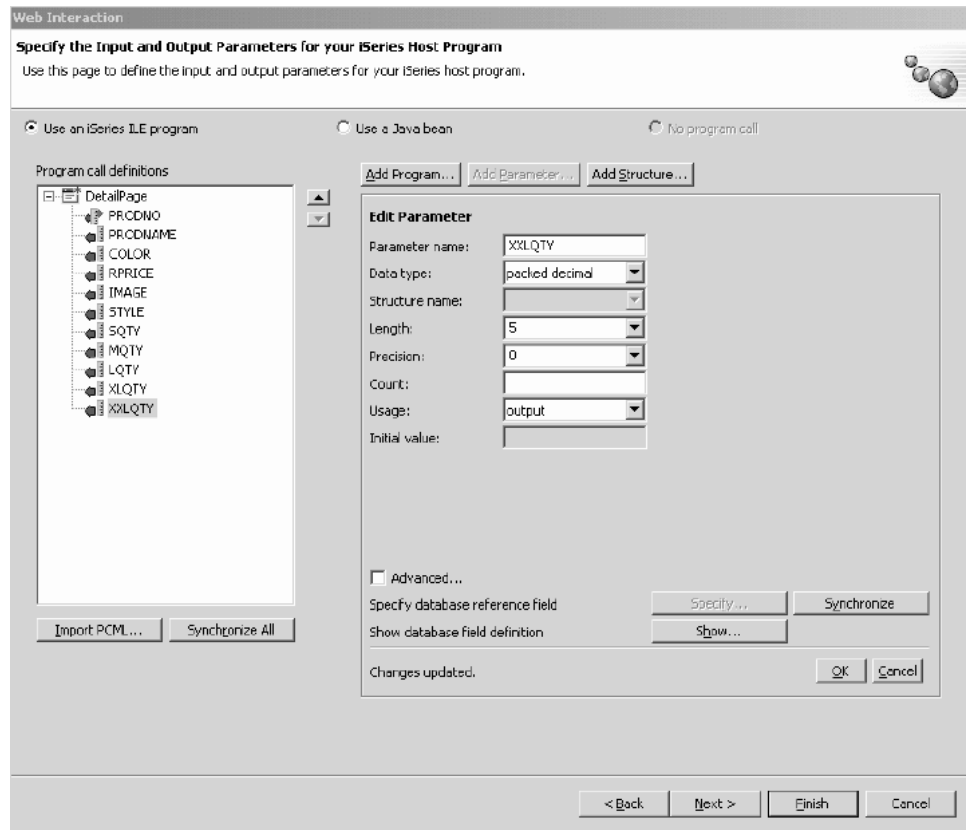


10. Clique em **Close**.

Agora, é necessário alterar o uso de 10 dos 11 parâmetros para "output" (todos os parâmetros, exceto PRODNO).

1. Ainda no assistente de Interação com a Web, clique em **PRODNAME** para selecionar essa opção.
2. Na caixa de combinação **Usage**, troque o valor selecionado para **output** e clique em **OK**. Repita essa etapa (com a etapa anterior) para todos os parâmetros, exceto o primeiro (PRODNO). Quando tiver ajustado todos os parâmetros, o espaço de trabalho terá a seguinte aparência. Observe que o ícone à esquerda

do parâmetro mostra se ele é do tipo entrada, entrada e saída ou saída.



3. Clique em **Next** para consultar a visualização **Input Form**.
4. Clique em **Finish** para criar a Interação com a Web.
5. Selecione **Yes to all** ou **OK** e **Finish** novamente, se for exibida alguma mensagem.

Vincular o Projeto a uma Interação com a Web

Agora que você criou um arquivo JSP de Interação com a Web que utiliza parâmetros de entrada e saída, deverá personalizar esse arquivo JSP para que ele também funcione com o componente WebFacing. É necessário inserir o código para criar um link a partir do aplicativo WebFacing para chamar esta Interação com a Web. Para isto, adicione uma função JavaScript no arquivo `webface.js`, para que possa chamar o servlet `DetailPageServlet` com o parâmetro `PRODNO` em uma nova janela.

Para criar o link:

1. Na visualização do Navegador, expanda **Project501 > Web Content > ClientScript**.
2. Dê um clique duplo em `webface.js` para abri-lo em um editor.
3. Role para a parte inferior do arquivo e insira as seguintes linhas:

```
var mywindow
function next(app)
{
mywindow = window.open(app,"Details","RESIZABLE=YES,
HEIGHT=700, WIDTH=800");
}
```
4. Clique no ícone Salvar ou em **File > Save webface.js**.

Para criar um aplicativo semelhante ao SV000501 no futuro, também será necessário alterar uma definição da Web na origem DDS para ativar a imagem que foi adicionada e ativar também o link para fechar a janela na função JavaScript.

O código RPG incluído neste aplicativo é alterado para mostrar a alteração, no entanto, será necessário fazer a alteração manualmente nas próximas aplicações. Além disso, depois de alterar as definições da Web, será necessário converter a origem DDS.

Você pode verificar o código para duplicar o resultado.

Para visualizar a origem DDS:

1. Abra a perspectiva WebFacing.
2. Na visualização de projetos WebFacing, expanda **Project501 > DDS**.
3. Clique com o botão direito do mouse em **<servidor_iSeries > RETAILSTOR/QDDSSRC(QUERY)** e selecione **Open With > CODE Designer**.
4. Depois de abrir o CODE Designer, expanda **SCREEN1 > ITEMSUB**.
5. Clique em **IMAGESRC** para selecionar essa opção.
6. Clique na guia **Source**.
7. Clique na guia **Web Settings**, na parte inferior direita da janela.

Nota: Verifique as propriedades da Definição da Web, como a largura em pixels e o nome do arquivo. Futuramente, será necessário fazer as mesmas alterações na origem DDS e reconvertê-la.

8. Observe as seguintes linhas na origem:

```
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}')
```

Observe principalmente que Project501 é especificado na última linha. Conforme mencionado anteriormente, para executar o aplicativo SV000501 novamente, é necessário retornar o valor de Project501 para SV000501.

9. Salve o arquivo, clicando no ícone Salvar ou em **File > Save** na barra de menus.
10. Feche o arquivo e o CODE Designer.

Para obter informações adicionais sobre como trabalhar com a Origem DDS, vá para a perspectiva de ajuda do Ambiente de Trabalho e consulte a documentação da IBM WebFacing Tool.

Publicando os Arquivos e Iniciando Novamente o Servidor

Nesta seção, você publicará seus arquivos e iniciará novamente, para que o aplicativo Project501 possa receber todas as alterações.

No entanto, antes de iniciar novamente o servidor, é necessário verificar se o Project501.war foi adicionado no arquivo SVStoreEAR.

Para verificar a configuração:

1. Na visualização do Navegador, expanda a **Scenario folder**.

2. Dê um clique duplo em **Scenario server.wsi**. Poderá ser exibido um diálogo perguntando se você deseja que o editor corrija as entradas do projeto automaticamente. Clique em **Yes**. Se não aparecer um diálogo, não será necessário executar nenhuma ação. De qualquer forma, o arquivo será modificado automaticamente quando for aberto.
3. Salve e feche o arquivo.

Para iniciar novamente o servidor:

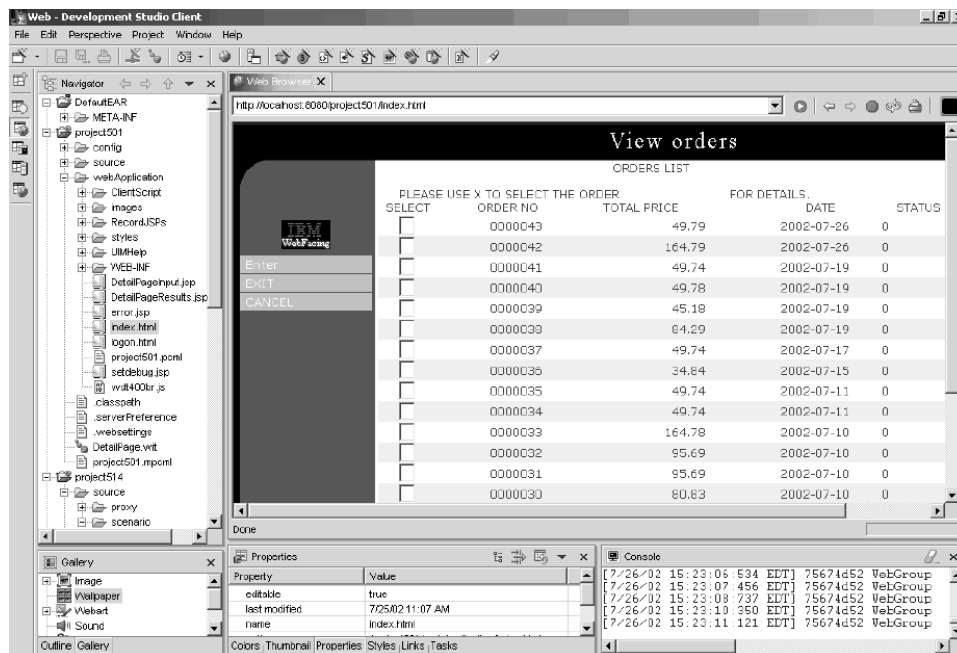
1. Alterne para a perspectiva Servidor.
2. Na visualização Configuração do Servidor, no canto inferior esquerdo, expanda **Servers** e dê um clique duplo em **Scenario server**.
3. Clique na guia **Servers** no canto inferior direito da tela para ver o status do servidor na visualização Servidores.
4. Clique com o botão direito do mouse em **Scenario server** e selecione **Publish**. Clique em **OK** quando a publicação for concluída.
5. Clique com o botão direito do mouse no servidor na visualização Servidores e selecione **Start** ou **Restart** (o que estiver ativado). Se aparecer algum aviso a respeito de arquivos não salvos, cancele o diálogo, salve os arquivos abertos e continue.
6. Verifique as informações de log do Console (aberto automaticamente). Você saberá que o servidor foi iniciado quando for exibida a seguinte linha na parte inferior do log: Servidor *server_name* aberto para e-business.

Testando a Interface

Você concluiu as etapas necessárias para criar uma interface de visualização de pedidos e de inventário. Para testar a interface:

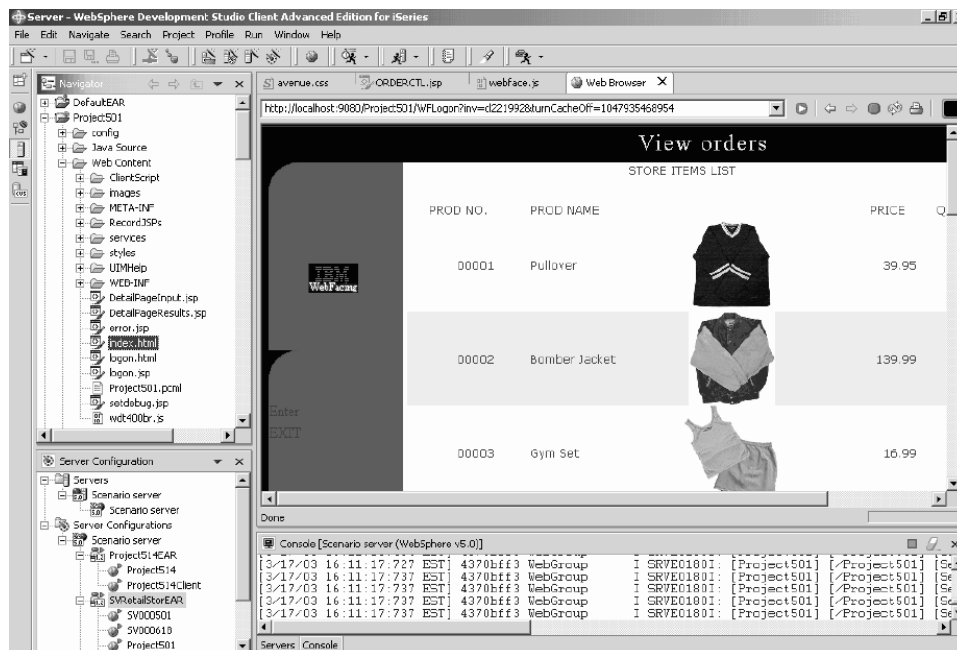
1. Vá para a perspectiva da Web.
2. Expanda **Project501 > Web Content**.
3. Clique com o botão direito do mouse em **index.html** e selecione **Run on Server**.
4. Um diálogo **Server Selection** será aberto. Verifique se **Scenario server** está selecionado por padrão e clique em **Finish**.
5. Clique em **View orders – Launch in main browser window** para chamar o aplicativo administrador. Depois de efetuar login com seu ID do usuário e senha do iSeries (para o servidor iSeries utilizado no desenvolvimento deste

projeto), você será direcionado para a seguinte página:



Tente inserir um X ao lado de um dos números de produto para ver detalhes sobre ele.

6. Clique na seta para trás para obter a página index.html e clique na janela **View inventory – Launch in main browser** para chamar o aplicativo cliente. Você será direcionado para a seguinte página (depois de efetuar logon no servidor iSeries com seu ID do usuário e senha):



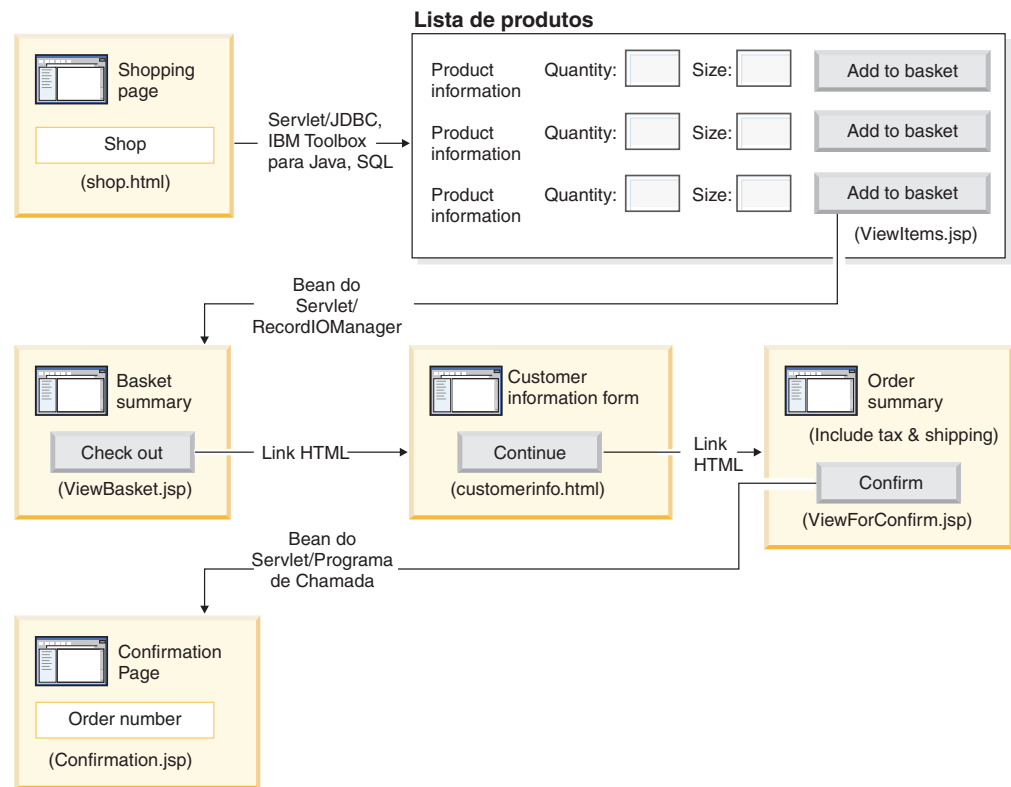
Tente clicar em uma imagem para exibir detalhes sobre um item específico.

Você concluiu o módulo! Você criou uma interface com o usuário para visualizar o inventário e itens do pedido.

Capítulo 6. Módulo Avançado 1: Criar HTML, Servlets e Arquivos JSP que Fazem Pedidos do Cliente em um Servidor iSeries (SV001585)

Introdução

Este projeto é destinado a usuários com bons conhecimentos em programação Java e algum conhecimento de gerenciamento de dados do iSeries e RPG. O projeto mostra como trabalhar com as classes de acesso a dados do IBM Toolbox para iSeries, o bean RecordIOManager e o bean de Chamada de programa para criar código HTML, servlets e JSPs que fazem pedidos do cliente em um servidor iSeries. Como usuário, você utiliza a página de compras para visualizar os produtos disponíveis e adicionar itens na cesta de compras. Quando estiver satisfeito com os itens selecionados (no resumo da Cesta), você clica em um botão de registro de saída para ser direcionado a um formulário de informações sobre o cliente. Depois de preencher o formulário, o projeto retorna um resumo do pedido com um botão de confirmação, que o leva para uma página de confirmação com seu pedido e o número do pedido.



Resumo das Etapas de Nível Superior

Como este é um módulo avançado, as instruções não o guiam por cada etapa de criação do projeto, mas descrevem as etapas de desenvolvimento específicas do iSeries executadas para criar esse projeto. Estas são as etapas de nível superior:

1. Gravar uma página da Web de compras em HTML.

2. Gravar um servlet que ocupa um bean Java (utilizando JDBC e SQL) com os itens disponíveis para compra pelo cliente.
3. Gravar um arquivo JSP para visualizar itens para venda e permitir que os clientes insiram a quantidade e o tamanho do item desejado e selecionem o item, clicando em um botão **Add to basket**.
4. Utilizando o bean RecordIOManager, gravar um servlet chamado com o clique no botão **Add to basket** que atualiza o banco de dados INVENTORY do iSeries, subtraindo a quantidade e o tamanho do item pedido e adicionando essa seleção em um bean Java denominado "basket". Em seguida, o servlet redireciona a resposta para ViewBasket.jsp, se a operação for bem-sucedida. Caso contrário, o servlet exibe uma página de erro.
5. Gravar um formulário HTML para que o cliente insira informações pessoais.
6. Gravar um arquivo JSP da página de confirmação de compra, que mostra o conteúdo da cesta, despesas de envio e impostos. A página também deverá conter um botão de confirmação para o cliente.
7. Gravar um servlet que será chamado quando o cliente clicar no botão de confirmação, que utiliza um bean Java criado com o assistente de Chamada de programa do iSeries. Uma das chamadas de método do bean Java chama um programa RPG para criar um novo pedido no banco de dados ORDERS no servidor iSeries correspondente ao conteúdo da cesta de compras do cliente. Em seguida, o servlet retorna um número e pedido, coloca o bean Java na sessão do aplicativo da Web e carrega um arquivo JSP de confirmação do pedido, contendo o número do pedido.

Antes de Começar

É possível concluir os exercícios somente se os pré-requisitos a seguir forem atendidos. Os pré-requisitos estão descritos em mais detalhes em Capítulo 3, "Executando o Cenário", na página 9.

- Você tem acesso TCP/IP a um servidor iSeries.
- Você iniciou os servidores do iSeries com o comando STRTCPSVR *ALL
- Você iniciou o servidor WebFacing com o comando STRTCPSVR *WEBFACING
- Você restaurou as bibliotecas WHOLESALE e RETAILSTOR para o servidor iSeries.
- As bibliotecas WHOLESALE, RETAILSTOR e QGPL estão na lista de bibliotecas.
- Você concluiu todas as tarefas do Capítulo 3, "Executando o Cenário", na página 9, (exceto as tarefas opcionais do WebSphere Application Server, que não são obrigatórias para testar o aplicativo no o Ambiente de Trabalho).

Criando a Página da Web, Servlets e Arquivos JSP

Para construir os componentes do projeto SV001585:

1. Crie um projeto da Web.
2. Grave uma página shop.html com o Page Designer, contendo um link que chama um servlet GetItems
3. Importe o arquivo jt400.jar para o iSeries Toolbox para classes Java na pasta *lib* do projeto da Web. Esse arquivo jar está em `x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime`, em que *x* é o diretório no qual o Development Studio Client foi instalado.

Nota: Consulte GetItems.java e ViewItems.jsp no projeto SV001585 para ver as partes relacionadas ao iSeries Toolbox principal para Java JDBC e

SQL para o servlet e o JSP. Na visualização Navegador da perspectiva da Web, você localizará `GetItems.java` expandindo **SV001585 > source** e `ViewItems.jsp`, expandindo **SV001585 > webApplication**.

4. Grave um servlet `GetItems` que utiliza a iSeries Toolbox para Java JDBC e SQL para recuperar itens de vestuário do banco de dados `INVENTORY` do iSeries e que:
 - a. Coloque o bean `ResultSet` contendo o resultado da consulta SQL na sessão
 - b. Redirecione o pedido para `ViewItems.jsp`

Amostra de código para `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. Grave um arquivo JSP `ViewItems` que recupere os itens de vestuário do bean `ResultSet` obtido na etapa anterior, para exibir os itens de vestuário no formato de tabela. O arquivo JSP também deve incluir um formulário para cada item que pode ser utilizado para selecionar o tamanho e a quantidade e adicionar o item na cesta de compras. Você pode utilizar o Page Designer nas ferramentas de desenvolvimento da Web do iSeries para gravar o JSP. Mais

especificamente, você pode mostrar a página na visualização Design e adicionar o código apropriado na visualização Origem. Amostra de código para 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. Importe o arquivo iseriesut.jar das ferramentas de desenvolvimento Java do iSeries para a pasta *lib* do projeto da Web. Esse arquivo JAR está em `x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime`, em que *x* é o diretório no qual o produto foi instalado. Consulte `AddtoBasket.java` e `ViewBasket.jsp` no projeto SV001585 para ver a implementação. Na visualização Navegador da perspectiva da Web, você localizará `AddtoBasket.java` expandindo **SV001585 > source** e `ViewBasket.jsp` expandindo **SV001585 > webApplication**.
7. Utilize o bean `RecordIOManager` das ferramentas de desenvolvimento Java do iSeries para gravar um servlet `AddtoBasket` chamado pelo botão **Add to basket**, que atualiza o banco de dados INVENTORY do iSeries, subtraindo a quantidade do pedido do cliente e adicionando os itens em um bean Java de Cesta na sessão. Amostra de código `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 {

```


- d. A última janela do assistente fornece a opção de criar um bean para um aplicativo Java, um serviço da Web ou ambos. Neste projeto, basta criar um para o aplicativo Java.

Nota: O programa de serviços PLACEODR obtém uma matriz de estruturas e coloca cada elemento de um registro de um banco de dados ORDERS, gerando o número do pedido como saída para cada matriz.

12. Grave um servlet PlaceOrder chamado pelo botão ViewForConfirm.jsp

Confirm.

- O servlet utiliza o bean gerado pelo assistente de Program Call do iSeries para acessar um servidor iSeries e registrar pedidos no banco de dados ORDERS da biblioteca RETAILSTOR.
- Os pedidos são os itens na cesta de compras, enviados para o programa ILE como uma matriz de estruturas.
- Cada estrutura dessa matriz é um item na cesta.
- O programa de serviços PLACEORD RPG chamado pelo bean retorna o número do pedido e um parâmetro de saída e o coloca na sessão.

O segmento de código a seguir mostra como o servlet PlaceOrder utiliza o 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();
    }
}
}
}

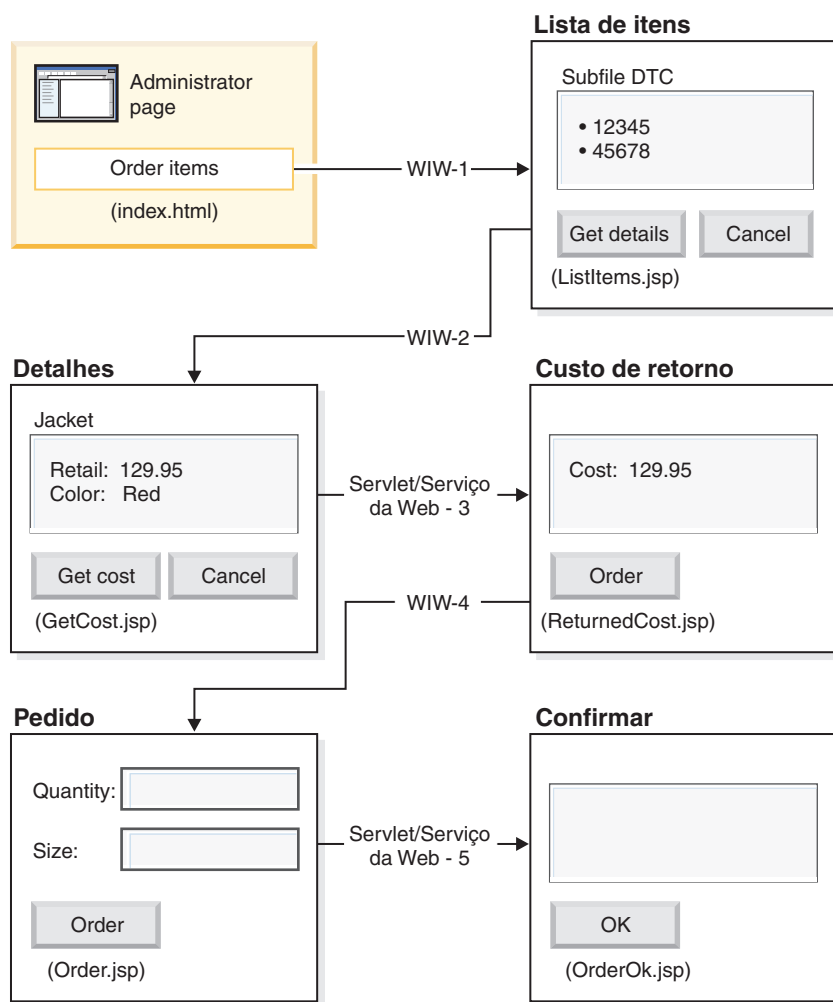
```

13. Grave um servlet OrderNumber que recupere o número e o exiba para o cliente, junto com uma mensagem de confirmação. Se o cliente não tiver adicionado itens em sua cesta de compras, certifique-se que seja retornada uma página de erro.

Capítulo 7. Módulo Avançado 2: Criar o Projeto da Web que Utiliza os Serviços da Web SV000514 e SV001586 (SV000618)

Introdução

Este projeto demonstra como utilizar seus conhecimentos em programação de RPG para criar clientes da Web para o iSeries Serviços de Web e programas RPG. Neste projeto, sua função é de administrador, passando por uma série de páginas da Web para determinar a quantidade em inventário e para fazer pedido de inventário adicional para a loja varejista ao fornecedor atacadista. Insira o número do item, visualize os detalhes do item, indique as quantidades e tamanhos do pedido e aceite a confirmação.



(WIW = Web Interaction Wizard)

Como este é um módulo avançado, as instruções não o guiam por cada etapa de criação do projeto, mas descrevem as etapas de desenvolvimento necessárias para criar esse projeto. O projeto utiliza os seguintes componentes do Development Studio Client:

- ferramentas de desenvolvimento da Web do iSeries para criar as páginas da Web com o Page Designer, incorporando a saída do assistente de Interação com a Web e diversos Componentes da Web
- O Remote System Explorer para criar um programa de serviços TNLSTITM RPG que retorna informações sobre o item
- O assistente Serviços de Web para gerar código de proxy do servlet, que localiza preços de itens e faz os pedidos
- ferramentas de desenvolvimento Java do iSeries para criar os servlets necessários
- O Ambiente de Teste do WebSphere para verificar o aplicativo antes da implementação no servidor iSeries através do WebSphere Application Server

Antes de Começar

É possível concluir os exercícios somente se os pré-requisitos a seguir forem atendidos. Os pré-requisitos estão descritos em mais detalhes em Capítulo 3, “Executando o Cenário”, na página 9.

- Você tem acesso TCP/IP a um servidor iSeries.
- Você iniciou os servidores do iSeries com o comando STRTCPSVR *ALL
- Você iniciou o servidor WebFacing com o comando STRTCPSVR *WEBFACING
- Você restaurou as bibliotecas WHOLESale e RETAILSTOR para o servidor iSeries.
- As bibliotecas WHOLESale, RETAILSTOR e QGPL estão na lista de bibliotecas.
- Você concluiu todas as tarefas do Capítulo 3, “Executando o Cenário”, na página 9, (exceto as tarefas opcionais do WebSphere Application Server, que não são obrigatórias para testar o aplicativo no o Ambiente de Trabalho).

Criando as Páginas da Web, Servlets, JSPs e Código RPG

Para construir os componentes do SV000618:

1. Crie um projeto da Web para conter todos os arquivos que serão criados.
2. Grave um arquivo JSP ListItems que relacione os itens no banco de dados do inventário do iSeries. Você pode utilizar o Page Designer nas ferramentas de desenvolvimento da Web do iSeries para gravar o servlet. Mais especificamente, você pode mostrar a página na visualização Design e adicionar o código apropriado na visualização Origem. Também é necessário inserir um subarquivo DTC (Design Time Control) para interagir com o programa de serviços TNLSTITM RPG, preenchendo o subarquivo com registros do banco de dados. Você pode especificar o programa de serviços nas definições de controle do DTC.

Em seguida, é necessário utilizar o assistente de Interação com a Web para criar a página de entrada:

- Especifique ListItems.jsp como página de saída para relacionar itens do inventário, o que garante que o assistente de Interação com a web crie um arquivo ListItems.wit.
- Certifique-se de não especificar chamadas de programa no assistente de Interação com a Web, já que o subarquivo DTC chama automaticamente o programa de serviços TNLSTITM RPG. O assistente também gera um ListItemsWitServlet que age como um link para chamar a página ListItems.jsp.
- Para rever o arquivo ListItemsWit.wit gerado pelo assistente de Interação com a Web:

1. Expanda **SV001618** e dê um clique duplo em **ListItems.wit** para exibir o assistente de interação do arquivo.
2. Clique em **Next** no assistente para revisar os valores especificados para a interação.

Em seguida, é necessário gravar uma página de saída JSP `GetCost` com o Page Designer, que utiliza a entrada da página `ListItems.jsp`. Quando um usuário clica em um item na página `ListItems.jsp`, uma página `GetCost.jsp` exibe detalhes do item.

Depois de criar a página `GetCost.jsp`, utilize o assistente de Interação com a Web para criar uma interação `WitOrder` entre `ListItems.jsp` (selecionado como página de entrada) e `GetCost.jsp` (selecionado como página de saída):

- Na página Chamada de Programa do assistente, especifique uma chamada do procedimento `GetDetail` e parâmetros do programa de serviços `TNLSTITM RPG`.
- No procedimento, as APIs do subarquivo `DTC` são incorporadas para determinar qual registro do subarquivo foi selecionado. O procedimento utiliza essas informações para recuperar o registro selecionado do banco de dados `INVENTORY` e exibe detalhes, inclusive a imagem, do item selecionado em `GetCost.jsp`.
- Para rever o arquivo `WitOrderWit.wit` gerado pelo assistente de Interação com a Web:
 1. Expanda **SV001618** e dê um clique duplo em **WitOrder.wit** para exibir o assistente de interação do arquivo.
 2. Clique em **Next** no assistente para revisar os valores especificados para a interação.

Observe que o parâmetro *flow* é especificado como controlador de fluxo na página de saída. Isto faz com que o valor do parâmetros assegure que os arquivos JSP apropriados sejam exibidos.

Em seguida, é necessário importar arquivos WSDL (Web Services Definition Language) do projeto `SV000514`, para que o administrador possa recuperar o custo atual de um item do fornecedor atacadista, pressionando o botão **Get cost**.

- O botão **Get cost** chama `QryProdCostServlet.jsp` e o serviço da Web correspondente do projeto `SV000514`.
- Utilize o assistente de Serviços da Web e os arquivos WSDL importados para gerar o código proxy Java necessário para chamar o serviço da Web.
- O `QryProdCostServlet.jsp` obtém a entrada da página `GetCost.jsp`, utiliza o código proxy Java para chamar o serviço da Web `SV000514` para localizar o custo do item selecionado e exibe o custo em uma página denominada `ReturnedCost.jsp`.
- Para visualizar o código `QryProdCostServicesProxy.java` e `QryProdCostServlet.java`:
 1. Expanda **SV001618** > **source** > **proxy** > **soap**.
 2. Dê um clique duplo em **QryProdCostServicesProxy.java**.
 3. No `QryProdCostServlet.java`, dê um clique duplo em **QryProdCostServlet.java**, em **SV001618** > **source** e observe como ele instancia o código proxy Java.

Em seguida, é necessário utilizar o assistente de Interação para a Web para vincular `ReturnCost.jsp` como entrada e `Order.jsp` como saída, para que o administrador possa clicar em um botão **Order** para pedir o item selecionado para o fornecedor atacadista.

- Com essa interação, não é necessário utilizar uma chamada de programa, pois o vínculo das duas páginas é suficiente para exibir as informações corretas.
- Para visualizar `WitPlaceOrder.wit`:
 1. Expanda **SV001618**.
 2. Dê um clique duplo em **WitPlaceOrder.wit** para abrir a interação.
 3. Clique em **Next** no assistente para revisar os valores especificados.

Em seguida, utilize o serviço da Web SV001586 para que o administrador possa especificar o tamanho e a quantidade dos itens pedidos.

- Importe os arquivos WSDL SV001586 para este projeto, gere código proxy Java para chamar o serviço da Web e grave um `OrderSupplyServlet` chamado quando o usuário pressiona o botão **Order** em `Order.jsp`.
- O servlet reúne informações de `Order.jsp`, chama o código proxy Java do serviço da Web, que chama o serviço da Web SV001586 e faz o pedido do item.
- O servlet exibe `OrderOK.jsp` se o pedido for bem sucedido. Caso contrário, exibe uma página de erro.
- Para ver como o proxy do serviço da Web gerado é instanciado e chamado para utilizar o serviço da Web SV001586:
 1. Expanda **SV000618 > source**.
 2. Dê um clique duplo em **OrderSupplyServlet.java** e verifique o conteúdo.

Antes de Implementar no WebSphere Application Server

Antes de implementar o aplicativo no WebSphere Application Server, é necessário alterar um URL específico dentro dos arquivos de classe Java SV000514 e SV001586 para que o aplicativo funcione corretamente.

No SV000514:

- Na classe de proxy dos serviços da Web, `QryProdCostServicesProxy`, é definida uma variável que contém o URL do serviço da Web a ser chamado.
- Na primeira vez que o proxy é criado, esse URL é definido como `http://localhost:9080/SV000514/servlet/rpcrouter`.
- Com a variável definida com esse valor, o serviço da Web no projeto SV000514 no IDE é chamado, como demonstrado em “Executando o Aplicativo no Workbench” na página 17.
- Antes de implementar este aplicativo, é necessário alterar esse valor de URL para que indique o local no qual o arquivo EAR foi implementado, `SVWholeSale.ear`, no diretório IFS do iSeries.

No projeto SV001586:

- Na classe de proxy dos serviços da Web, `OrderSupplyServicesProxy`, é definida uma variável que contém o URL do serviço da Web a ser chamado.
- Na primeira vez que o proxy é criado, esse URL é definido como `http://localhost:9080/SV001586/servlet/rpcrouter`.
- Com a variável definida com esse valor, o serviço da Web no projeto SV001586 no IDE é chamado, como demonstrado em “Executando o Aplicativo no Workbench” na página 17.

- Antes de implementar este aplicativo, é necessário alterar esse valor de URL para que indique o local no qual o arquivo SVWholeSaleEAR.ear foi implementado no diretório IFS do iSeries.

Capítulo 8. Avisos

Nota sobre Direitos Restritos para Usuários do Governo dos Estados Unidos - Uso, duplicação e divulgação restritos pelo documento GSA ADP Schedule Contract com a IBM Corporation.

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos. É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte o representante IBM local para obter informações sobre produtos e serviços atualmente disponíveis em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM, poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não-IBM são de inteira responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum direito sobre tais patentes. Pedidos de licença devem ser enviados, por escrito, à:

Gerência de Relações Comerciais e Industriais da IBM Brasil. Av. Pasteur, 138-146, Botafogo, Rio de Janeiro, RJ, CEP 22290-240.

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

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

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA" SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS OU CONDIÇÕES DE NÃO-VIOLAÇÃO, MERCADO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, esta disposição pode não se aplicar ao Cliente.

Esta publicação pode incluir imprecisões técnicas ou erros tipográficos. Periodicamente, são feitas alterações nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Referências nestas informações a Web sites não-IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses Web sites. Os materiais contidos nesses Web sites não fazem parte dos materiais deste produto IBM e a utilização desses Web sites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com o objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil. Av. Pasteur, 138-146, Botafogo, Rio de Janeiro, RJ, CEP 22290-240.

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito neste documento e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato de Licença do Programa Internacional IBM ou de qualquer outro contrato equivalente.

As informações relativas a produtos não-IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou outras fontes disponíveis publicamente. A IBM não testou esses produtos e não pode confirmar a exatidão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não-IBM. Dúvidas sobre os recursos de produtos não-IBM devem ser encaminhadas diretamente a seus respectivos fornecedores.

Essas informações contêm exemplos dos dados e relatórios utilizados nas operações comerciais diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos podem incluir nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT:

Essas informações contêm programas de aplicativos de exemplo na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. Você pode copiar, modificar e distribuir esses programas de exemplo sem necessidade de pagar à IBM, com o objetivo de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo da plataforma operacional para a qual os programas de exemplo são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, a manutenção ou a função desses programas. Você pode copiar, modificar e distribuir esses programas de exemplo de qualquer maneira, sem pagamento à IBM, com o objetivo de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com interfaces de programação de aplicativos da IBM.

Cada cópia ou qualquer parte desses programas de exemplo ou qualquer trabalho derivado deve incluir um aviso de copyright, com os dizeres:

(C) (nome da empresa) (ano). Partes desse código são derivados dos Programas de Exemplo da IBM Corp. (C) Copyright IBM Corp. 1992, 2002. Todos os direitos reservados.

Informações sobre a Interface de Programação


As informações sobre a interface de programação destinam-se a facilitar a criação de software aplicativo utilizando este programa.

As interfaces de programação de uso geral permitem que o cliente desenvolva um software aplicativo que obtém os serviços das ferramentas deste programa.

No entanto, essas informações também podem conter informações sobre diagnósticos, modificações e ajustes. As informações de diagnósticos, modificações e ajustes são fornecidas para ajudá-lo a depurar seu software aplicativo.

Aviso: Não utilize estas informações sobre diagnósticos, modificações e ajustes como uma interface de programação, pois elas estão sujeitas a alterações.

Marcas Comerciais e Marcas de Serviço

- 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
-  **eServer**
- GDDM
- IBM
- IBMLink
- 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 é uma marca comercial da InstallShield Corporation.

Intel e Pentium são marcas comerciais da Intel Corporation nos Estados Unidos e/ou em outros países.

Java e todas as marcas comerciais e logotipos baseados em Java são marcas comerciais ou marcas registradas da Sun Microsystems, Inc., nos Estados Unidos e/ou em outros países.

Rational é uma marca comercial da International Business Machines Corporation e Rational Software Corporation nos Estados Unidos e/ou em outros países.

Lotus, Lotus Notes e Domino são marcas comerciais da Lotus Development Corporation nos Estados Unidos e/ou em outros países.

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT, Win32, Win32s e o logotipo Windows são marcas comerciais da Microsoft Corporation nos Estados Unidos e/ou em outros países.

Netscape Navigator é uma marca da Netscape Communications Corporation.

UNIX é uma marca registrada do The Open Group.

Outros nomes de empresas, produtos e serviços podem ser marcas comerciais ou marcas de serviço de terceiros.



Número do Programa: 5724-A81

Impresso em Brazil