Getting Started with WebEnterprise Designer

iPlanet™ Unified Development Server

Version 5.0

August 2001

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.

Contents

List of Figures

List of Procedures

Preface

Getting Started with WebEnterprise Designer provides complete information on the iPlanet UDS model-based HTML application development tool, WebEnterprise Designer. This manual describes how to use the WebEnterprise Designer HTML Application Model Workshop.

This preface contains the following sections:

Product Name Change

Forte 4GL has been renamed the iPlanet Unified Development Server. You will see full references to this name, as well as the abbreviations iPlanet UDS and UDS.

Audience for This Guide

This manual is intended for application developers. We assume that you:

- have programming experience
- are familiar with SQL and your particular database management system
- understand the basic concepts of object-oriented programming as described in *A Guide to the iPlanet UDS Workshops*
- have used the iPlanet UDS Repository Workshop

For instructions on installing WebEnterprise Designer and for information about certified platforms, versions, and products, see the *WebEnterprise Installation Guide*.

Organization of This Guide

The following table briefly describes the contents of each chapter:

Chapter	Description
Chapter 1, "WebEnterprise Designer Concepts"	Defines the basic concepts of WebEnterprise Designer.
Chapter 2, "WebEnterprise Designer Tutorial"	Contains a tutorial that walks you step by step through the process of creating, testing, partitioning, and deploying a WebEnterprise Designer application.
Appendix A, "WebEnterprise Designer Example Applications"	Describes how to install and use the WebEnterprise Designer examples.
Appendix B, "Starting WebEnterprise Designer"	Describes WebEnterprise Designer components, directories, and environment variables, and provides instructions for starting WebEnterprise Designer.

Text Conventions

This section provides information about the conventions used in this document.

Format	Description
italics	Italicized text is used to designate a document title, for emphasis, or for a word or phrase being introduced.
`monospace`	Monospace text represents example code, commands that you enter on the command line, directory, file, or path names, error message text, class names, method names (including all elements in the signature), package names, reserved words, and URLs.
ALL CAPS	Text in all capitals represents environment variables (FORTE_ROOT) or acronyms (UDS, JSP, iMQ).
	Uppercase text can also represent a constant. Type uppercase text exactly as shown.
Key+Key	Simultaneous keystrokes are joined with a plus sign: Ctrl+A means press both keys simultaneously.
Key-Key	Consecutive keystrokes are joined with a hyphen: Esc-S means press the Esc key, release it, then press the S key.

Other Documentation Resources

In addition to this guide, there are additional documentation resources, which are listed in the following sections. The documentation for all iPlanet UDS products (including Express, WebEnterprise, and WebEnterprise Designer) can be found on the iPlanet UDS Documentation CD. Be sure to read "Viewing and Searching PDF Files" on page 15 to learn how to view and search the documentation on the iPlanet UDS Documentation CD.

iPlanet UDS documentation can also be found online at http://docs.iplanet.com/docs/manuals/uds.html.

The titles of the iPlanet UDS documentation are listed in the following sections.

iPlanet UDS Documentation

- *A Guide to the iPlanet UDS Workshops*
- *Accessing Databases*
- *Building International Applications*
- *Escript and System Agent Reference Guide*
- *Fscript Reference Guide*
- *Getting Started With iPlanet UDS*
- *Integrating with External Systems*
- *iPlanet UDS Java Interoperability Guide*
- *iPlanet UDS Programming Guide*
- *iPlanet UDS System Installation Guide*
- *iPlanet UDS System Management Guide*
- *Programming with System Agents*
- *TOOL Reference Guide*
- *Using iPlanet UDS for OS/390*

Express Documentation

- *A Guide to Express*
- *Customizing Express Applications*
- *Express Installation Guide*

WebEnterprise and WebEnterprise Designer Documentation

- *A Guide to WebEnterprise*
- *Customizing WebEnterprise Designer Applications*
- *Getting Started with WebEnterprise Designer*
- *WebEnterprise Installation Guide*

Online Help

When you are using an iPlanet UDS development application, press the F1 key or use the Help menu to display online help. The help files are also available at the following location in your iPlanet UDS distribution: `FORTE_ROOT/userapp/forte/cln/*.hlp`.

When you are using a script utility, such as Fscript or Escript, type help from the script shell for a description of all commands, or help `<command>` for help on a specific command.

iPlanet UDS Example Programs

A set of example programs is shipped with the iPlanet UDS product. The examples are located in subdirectories under `$FORTE_ROOT/install/examples`. The files containing the examples have a `.pex` suffix. You can search for TOOL commands or anything of special interest with operating system commands. The `.pex` files are text files, so it is safe to edit them, though you should only change private copies of the files.

Viewing and Searching PDF Files

You can view and search iPlanet UDS documentation PDF files directly from the documentation CD-ROM, store them locally on your computer, or store them on a server for multiuser network access.

NOTE You need Acrobat Reader 4.0+ to view and print the files. Acrobat Reader with Search is recommended and is available as a free download from http://www.adobe.com. If you do not use Acrobat Reader with Search, you can only view and print files; you cannot search across the collection of files.

➤ To copy the documentation to a client or server

1. Copy the `doc` directory and its contents from the CD-ROM to the client or server hard disk.

   You can specify any convenient location for the `doc` directory; the location is not dependent on the iPlanet UDS distribution.

2. Set up a directory structure that keeps the `udsdoc.pdf` and the `uds` directory in the same relative location.

   The directory structure must be preserved to use the Acrobat search feature.

NOTE To uninstall the documentation, delete the `doc` directory.

➤ To view and search the documentation

1. Open the file `udsdoc.pdf`, located in the `doc` directory.
2. Click the Search button at the bottom of the page or select Edit > Search > Query.
3. Enter the word or text string you are looking for in the Find Results Containing Text field of the Adobe Acrobat Search dialog box, and click Search.

   A Search Results window displays the documents that contain the desired text. If more than one document from the collection contains the desired text, they are ranked for relevancy.

   NOTE For details on how to expand or limit a search query using wild-card characters and operators, see the Adobe Acrobat Help.

4. Click the document title with the highest relevance (usually the first one in the list or with a solid-filled icon) to display the document.

   All occurrences of the word or phrase on a page are highlighted.

5. Click the buttons on the Acrobat Reader toolbar or use shortcut keys to navigate through the search results, as shown in the following table:

Toolbar Button	Keyboard Command
Next Highlight	Ctrl+]
Previous Highlight	Ctrl+[
Next Document	Ctrl+Shift+]

   To return to the `udsdoc.pdf` file, click the Homepage bookmark at the top of the bookmarks list.

6. To revisit the query results, click the Results button at the bottom of the `udsdoc.pdf` home page or select Edit > Search > Results.

Chapter 1

WebEnterprise Designer Concepts

Welcome to WebEnterprise Designer, the iPlanet UDS Web application generation tool. WebEnterprise Designer provides a wizard-based graphical development tool for modeling both the business end of your application and the Web pages of its user interface. WebEnterprise Designer automatically generates an application based on these models.You can then use standard iPlanet UDS workshops and HTML editors to further refine and customize your application.

This chapter contains a high-level discussion of the basic concepts of WebEnterprise Designer. Topics covered include:

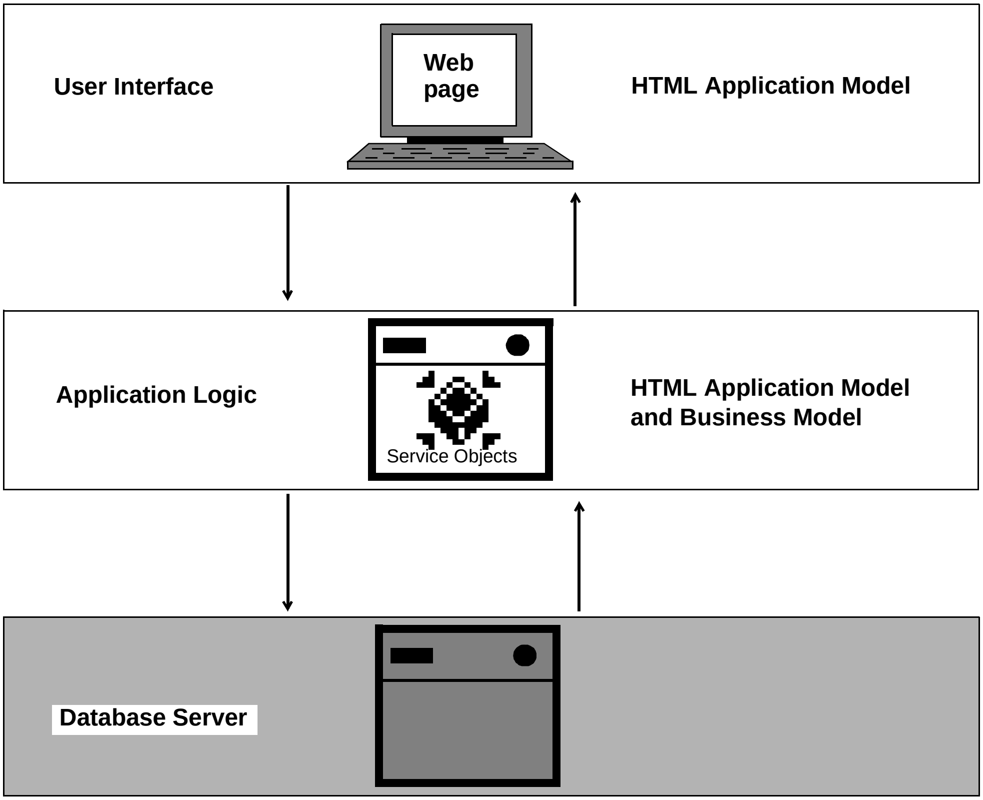
- overview of the WebEnterprise/Express elements of WebEnterprise Designer architecture
- overview of Express business modeling and the Express Business Model Workshop
- overview of HTML application models and the HTML Application Model Workshop
- overview of partitioning and deployment of an HTML application

Overview

WebEnterprise Designer is a model-based application development system that lets you create database query-update HTML applications without writing either TOOL or HTML code. To create an HTML application, you use two workshops—the Business Model Workshop and the HTML Application Model Workshop—to draw graphical representations of your application.

WebEnterprise Designer supports a three-tier architecture for your database applications. In a three-tier architecture, the user interface, application services, and database processing are divided into separate modules, as illustrated in Figure 1-1:

Figure 1-1 Three-Tier Architecture



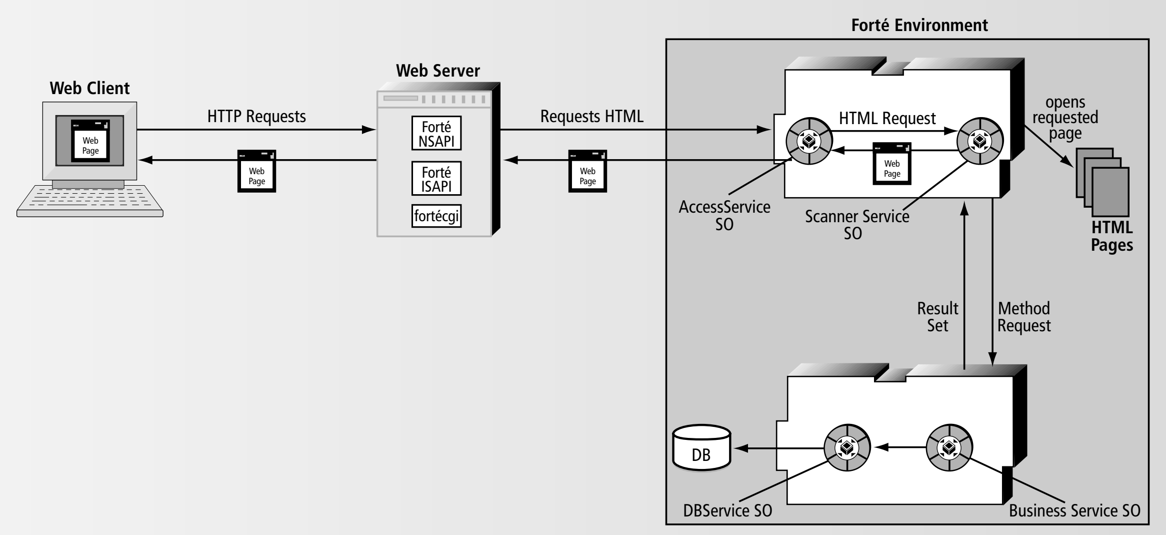
You build the first and second tiers of your application with WebEnterprise Designer. You use the HTML Application Model Workshop to create the user interface and user interface services; you use the Business Model Workshop to create the application business services, which provide access to database information.

The advantage of this architecture is that you can partition your application so that the business and application services run on different nodes from the database management system. In addition, iPlanet UDS enables you to provide failover and load balancing.

WebEnterprise Designer Runtime Architecture

Figure 1-2 illustrates the interactions of the parts of a WebEnterprise Designer application during runtime.

Figure 1-2 Anatomy of a WebEnterprise Designer Application at Runtime



In Figure 1-2, the Web Client at the left represents a client PC using a Web browser to view a WebEnterprise Designer application. When the user requests new information, the browser passes a URL to the Web server, requesting a Web page.

On the Web server, the fortecgi program or an iPlanet UDS Web server plug-in directs the URL to the WebEnterprise Designer AccessService service object, which passes the page request to the ScannerService service object, along with any parameter values relevant to the page.

The ScannerService service object opens the requested HTML page template and processes the iPlanet UDS tags to create the page. When the service object encounters certain tags in the HTML template, it invokes the `HandleTag` and `HandleCondition` methods of the handler class associated with the template. The handler class uses the Business Service service object to access business services (which access the database), and returns any data needed in a result set. The handler class caches the result set in the AccessService for future use.

After the ScannerService service object populates the page template with the result set, it returns the requested page back to the AccessService service object, which delivers it to the Web server. The Web server in turn passes the page to the client Web browser, which displays the page.

The AccessService service object maintains state for all the Web clients it serves. Its dialog duration is session. Unlike normal session duration service objects, you may load balance the AccessService service object. When you do, each new browser request is routed to the next available replicate. Subsequent requests from the same browser session will be directed back to the replicate that handled the first request.

Designing WebEnterprise Designer Applications

To define the objects and business services in your application, you create a *business model*. The business model is a graphical representation of the data in your application. To create the user interface for your application, and the services for the user interface, you create an *HTML application model*. The HTML application model is a graphical representation of the Web pages in your user interface and the control flow of your application.

Creating Business Models and Business Classes

The foundation of a WebEnterprise Designer HTML application is a business model. A business model is a graphical representation of the data in the system you are building. Business models are comprised of business classes that correspond to the objects in your system, and the relationships between them. For example, if you are building an order-entry system, some business classes in your model might be orders, customers, line items, and parts. You can have relationships between orders and customers, orders and line items, and so on.

The business classes in the business model map directly to database tables. Access to the database management system is automatic.

You construct business models in the Business Model Workshop. The Business Model Workshop is a graphical tool that supports a subset of the Universal Modeling Language (UML) notation for modeling a business system (described below). Once you define the business classes, you draw the relationships between them.

For more information about the Business Model Workshop, refer to *A Guide to Express*. For information about using the Business Model Workshop with WebEnterprise Designer, see "Building the Tutorial Business Model" on page 35.

The Universal Modeling Language

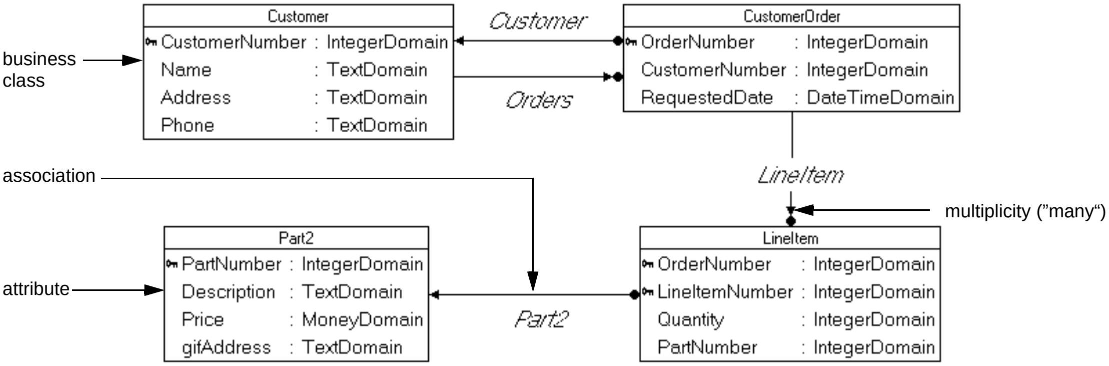
The Universal Modeling Language (UML) is a graphical notation for representing object-oriented concepts. For more information about UML, see *Object-Oriented Modeling and Design* (by Rumbaugh, Blaha, Premerlani, Eddy, and Lorensen, published by Prentice Hall). You use the tools in the Business Model Workshop to create UML symbols that represent classes and their relationships.

The Business Model Workshop supports the following UML concepts and graphical notations:

- class
- attribute
- association
- generalization
- aggregation
- inheritance
- multiplicity

Figure 1-3 illustrates the graphical elements of a business model. A brief description of each element follows the figure.

Figure 1-3 Business Model Elements



Business Class A business class represents an object in your business system. Each class corresponds to a database table and to one or more TOOL classes. A business class is represented by a box containing a separate region for attributes.

Attribute An attribute is a data value held by the objects in a class. Attributes appear in the lower part of a class, below the class name. Every attribute corresponds to a column in a database table.

Association An association is a relationship, or connection, between two classes. Associations in the Business Model Workshop are uni-directional; but, you can draw multiple associations (in different directions) between two classes.

Multiplicity Multiplicity describes how many instances of one class may relate to a single instance of another class. For example, one customer can place many orders—this is a “one-to-many” relationship. Line terminators specify the type of relationship: a solid dot indicates zero or more—“many.” No dot means one—a line without multiplicity symbols means a one-to-one association.

Aggregation An aggregation (not shown) is a special type of association that describes a relationship between the objects representing the components of a larger assembly and the object representing the entire assembly. Component objects cannot exist without the assembly object. A small diamond indicates the assembly end of the relationship.

Generalization/Inheritance A generalization is a relationship between a class and one or more specific versions of the class. The more general class is the superclass, the more specific class its subclass. Inheritance is the mechanism through which subclasses share the features (attributes, methods) of their superclasses. A subclass “inherits” the features of its superclass. A generalization is indicated by a triangle on the line connecting the classes.

Association Symbols

The following table displays the symbols that define types of associations:

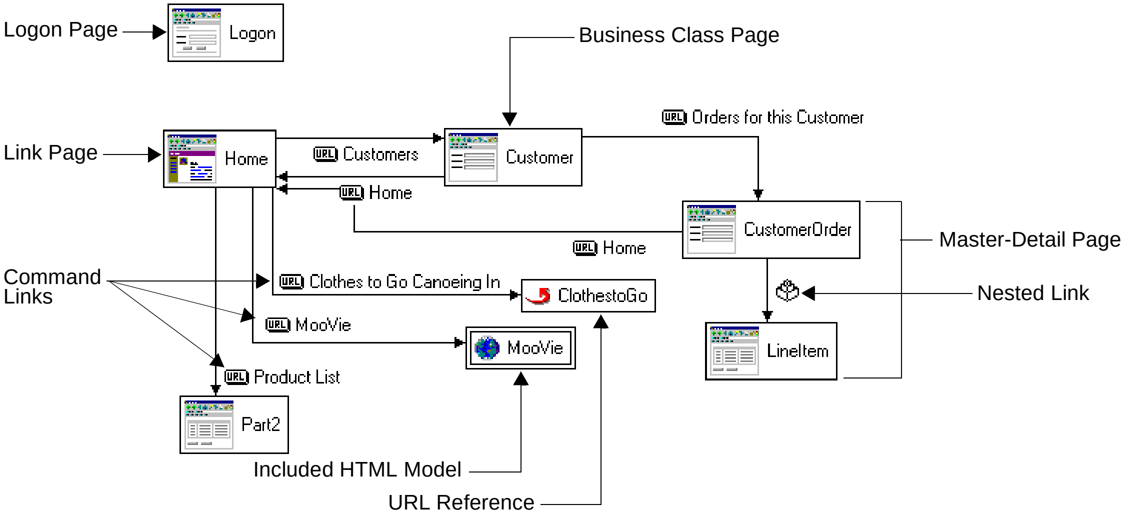
Symbol	**Description**
◂—— ——▸	Direction of association. The class at the end with the arrow is referred to as the ‘to’ class. The class at the end without the arrow is the ‘from’ class.
●	Zero or more (many).
no symbol	One

Symbol	Description
◇	Aggregation
△	Generalization

Creating HTML Application Models and Business Class Pages

Once you have described your business system in the Business Model Workshop, you can define the pages that will comprise the user interface of your HTML application and specify the flow of control between pages, how pages communicate with each other, how they are invoked, and how data is passed between them. Figure 1-4 illustrates the elements of an HTML application model.

Figure 1-4 HTML Application Model Elements



You build HTML application models in the HTML Application Model Workshop. An HTML application model is comprised of business class pages and links between them. You can also create links to external Web pages or to pages in other HTML application models. Business class pages are based on business classes in a business model. There are other page types that are not based on business classes. These include “link” pages, which serve to connect to other pages with hypertext links, “logon” pages, which provide a secure gateway to the application, and “URL reference objects,” which link to the address of a browser page. Pages can be nested, which allows the user to manipulate data from multiple tables in a single HTML page.

You create pages using a Page Wizard that guides you through a series of property dialogs, where you specify page layout, the type of database command interface, and page behavior.

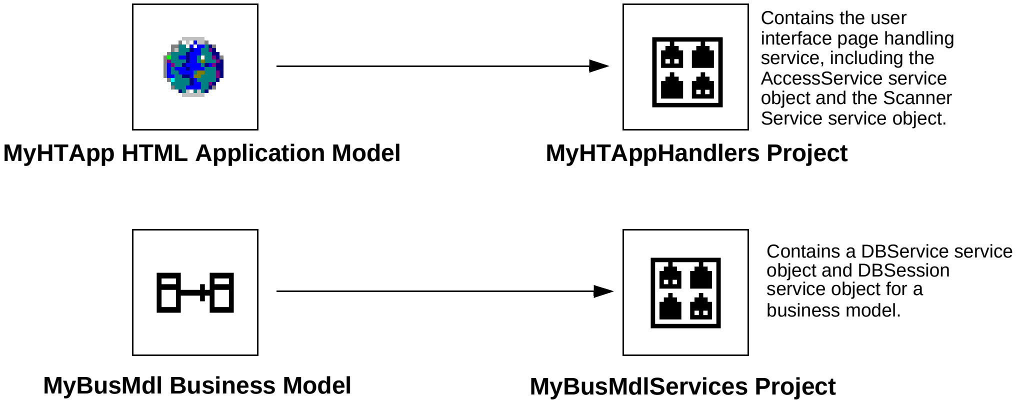
Generating Code

WebEnterprise Designer automatically generates HTML files and Web application server code based on the models you create in the Business Model and HTML Application Model workshops.

Generated TOOL Code

From the business model, WebEnterprise Designer generates the service objects and classes that handle the data in your application. From the HTML application model, WebEnterprise Designer generates the HTML pages that provide your user interface. Figure 1-5 shows the projects that are created from your business and HTML application models:

Figure 1-5 Generated Projects



Contains the user interface page handling service, including the AccessService service object and the Scanner Service service object.

Contains a DBService service object and DBSession service object for a business model.

The “Handlers” project generated from the HTML application model is the *main project* for the application. This project is responsible for dynamically generating the data content of the HTML page, based on user requests. The “Services” project generated from the business model functions as a *supplier project* to the main project. This project is responsible for providing access to the database.

Once these iPlanet UDS projects are generated, you can modify them, if necessary, as you would any iPlanet UDS projects by using the standard iPlanet UDS workshops. If you change parts of the model and rerun the Generate Code command, iPlanet UDS Express automatically regenerates the code to accommodate the changes, but maintains any custom changes you have made.

Generated HTML Pages

For each logical page in the HTML application model, WebEnterprise Designer generates a set of related HTML templates to support it. These pages are displayed in a web browser as the user interface to the application.

Testing, Partitioning, and Deploying an Application

Testing the project locally When you test run the application in WebEnterprise Designer, iPlanet UDS partitions the application so that as much of it as possible will run on your local machine. Any service objects that the local node cannot support are assigned to a suitable node in your development environment, allowing you to test the application independently of its deployment environments.

Default partitioning WebEnterprise Designer automatically creates all the service objects needed for your application. When you partition the application, iPlanet UDS creates logical server partitions based on the visibility and replication options of the service objects, and a logical client partition that starts the browser. If visibility and replication of the service objects are all the same, it creates two server partitions: one for the HTML page-handling service objects and one for the database handling service objects.

Partition Workshop Use the iPlanet UDS Partition Workshop when you are ready to deploy the application or to test it in a distributed environment. When partitioning or testing, you treat a WebEnterprise Designer application like any other iPlanet UDS application. You partition the application for each deployment environment, and create the distributions for each configuration you wish to deploy.

You can then use the standard iPlanet UDS services to create the distribution, including compiling the distributed partitions for optimal performance. In a final step, you make the location of the generated Web pages of the user interface available to the user interface services.

Managing Repository Elements

The business models and HTML application models that you create with WebEnterprise Designer are stored in your development repository the same way as iPlanet UDS projects you create with the Project Workshop—you must use the Repository Workshop to manage them. In the Repository Workshop, you can create new models, include existing models in your workspace or exclude them, and delete existing models from the repository. To get write access to individual models, you use Checkout and Branch commands.

In addition, all work that you perform in WebEnterprise Designer takes effect in the current workspace. You must be sure to save, update, and integrate your workspace as appropriate.

See *A Guide to the iPlanet UDS Workshops* for general information about using iPlanet UDS workspaces.

Chapter 2

WebEnterprise Designer Tutorial

This chapter walks you through the process of creating a Web application with WebEnterprise Designer. Along the way, you learn the basic concepts behind WebEnterprise Designer and experiment with all its basic features.

In this tutorial, you will learn how to:

- use the HTML Application Model Workshop
- build a multi-page HTML application model
- test run your Web application
- partition and deploy your Web application

This tutorial resembles the Express tutorial found in *A Guide to Express* for Express. The Express tutorial builds an application model from a business model; the WebEnterprise Designer tutorial builds an HTML application model from a similar business model.

For complete information on Express business models, see the Express documentation.

Introduction

In this tutorial, you will use WebEnterprise Designer to create an intranet application for the Order Entry department of the *Tip-a-Canoe and Paddles, Too* canoe-and-paddle store. While you create this application, you will learn all the basic techniques needed to develop simple Web applications.

This chapter consists of the following parts:

- "Building the Tutorial Business Model" on page 35

  In this section, you create a business model as the basis of your HTML application. You import a sample database schema from which you create the model's business classes, and then create relationships between them.

- "Starting the Tutorial HTML Application" on page 46

  In this section, you learn how to use the HTML Application Model Workshop. You learn about business class pages, their modes, designs, and styles.You create and run a single business class page.

- "Finishing the Tutorial HTML Application" on page 76

  In this section, you add pages of various types to your application, and connect them with a variety of links. You test run the application at several points to check your progress and experience how the model acts at runtime.

- "Partitioning and Deploying the Tutorial Application" on page 106

  In this section, you partition, deploy, and then test run the tutorial application.

- "Enhancing the Tutorial Application" on page 117

  This section describes several convenient features you could add to the tutorial application, including field validation, field formatting, and a URL reference page.

Using iPlanet UDS Online Help

iPlanet UDS online Help is available from any iPlanet UDS workshop or dialog window. iPlanet UDS Help follows Windows Help standards. If you are not familiar with Windows Help, or you just want tips on iPlanet UDS Help features, press the Help key when any Help window is active. This action displays "Help on Help," which is a Help system on using the Help system.

There are two kinds of Help:

- a general Help function, available through the Help menu

  Choosing Help > Help Topics on any workshop or dialog window displays a window containing three folders: a Contents folder with Help topics organized in a table of contents format, an Index folder for searching by index entries, and a Find folder for searching by words or phrases.

- context-sensitive Help, available by pressing the Help key

  Context-sensitive Help is available for almost all iPlanet UDS windows and dialogs. Simply press the Help key when a window is active to display its Help. When context-sensitive Help is not available, pressing the Help key displays the main Help contents page.

Two significant WebEnterprise subjects are available *only* through the Help system:

- WebEnterprise Designer dialog descriptions
- WebEnterprise projects and libraries

For information on how to copy selected text to the clipboard or to print a Help topic or file, see "Help on Help."

Setting Up Your Environment for the Tutorial

To prepare for the Tutorial, do the following:

- Create and populate the tables for the Tutorial in an existing database—see "Installing the Tutorial Database," which follows.
- Specify the database manager for the Tutorial database as a resource manager—see "Specifying a Resource Manager" on page 33.
- Verify the existence of the Tutorial image files—see "Tutorial Image Files" on page 34.

Installing the Tutorial Database

Use the following scripts to load and (optionally) unload the sample database tables. These scripts are specifically for Oracle, Microsoft SQLServer, and Sybase versions of SQL, but you can modify them for use with other SQL versions. These files are located in `FORTE_ROOT/install/examples/web/wedtut/install`.

Script name	Description
makehtut.ora	Creates and populates tables used by the HTML Tutorial in Oracle SQL format.
drophtut.ora	Drops the tables created by makehtut.ora.
makehtut.syb	Creates and populates tables used by the HTML Tutorial in Sybase or SQLServer SQL format.
drophtut.syb	Drops the tables created by makehtut.syb.

Use your database SQL interpreter to run the appropriate script to install the sample tables into an existing database.

For Oracle: Use SQL*Plus to load the makehtut.ora script.

For Microsoft SQLServer or Sybase: The first line of both `makehtut.syb` and `drophtut.syb` indicate the database name:

```
use cgtest
```

Change “cgtest” to the name of your database, then use the standard mechanism for redirecting the `makehtut` file to load the data into your database.

The makehtut scripts create the database schema described in Table 2-1.

Table 2-1 Sample Table Descriptions

Table Name	Columns	Primary Key
Customer	CustomerNumber	3
	Name	
	Address	
	Phone	
CustomerOrder	OrderNumber	3
	CustomerNumber	
	RequestedDate	
LineItem	OrderNumber	3
	LineItemNumber	3
	Quantity	
	PartNumber	
Part2	PartNumbe	3
	Description	
	Price	
	GifAddress	

Specifying a Resource Manager

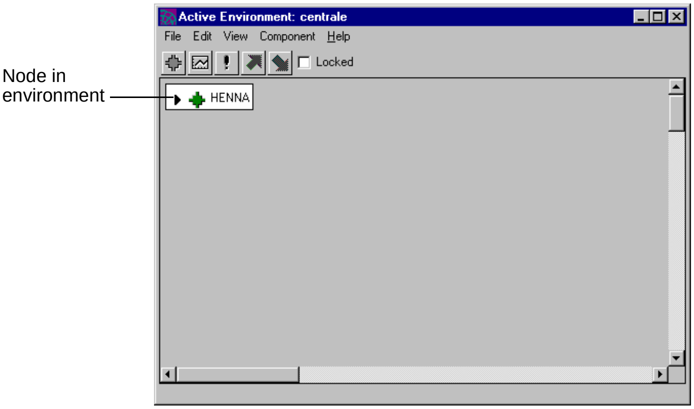
Now that you have the tables loaded into your database, you must make the database available to iPlanet UDS. You do this by using the Environment Console to create a resource manager. A resource manager is simply a database management system that can be accessed from iPlanet UDS.

If you already have a resource manager defined for your database type (Oracle or SQLServer), the following procedure is unnecessary.

➤ To specify a resource manager

1. Start the iPlanet UDS Environment Manager.
2. Choose Programs > Forte > Environment Console from the Start menu.

   The Environment Console window appears.



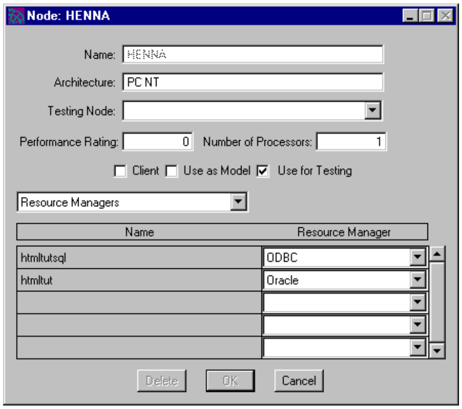
   All nodes in your environment are represented by icons. In the figure above, only one node is in the environment.

3. Enable the Locked toggle.
4. Select your node’s icon.
5. Choose Component > Properties.

   The property dialog for the selected node appears.

6. In the lower section (Name-Resource Manager section), select the database type from the Resource Manager drop list and type in a name in the Name column.

   For example, the following figure shows a resource manager for an Oracle database (htmltut) and one for a Microsoft SQL Server database (htmltutsql). ("ODBC" designates a Microsoft SQL Server database.)



Node: HENNA

Name: HENNA

Architecture: PC NT

Testing Node:

Performance Rating: 0 Number of Processors: 1

□ Client □ Use as Model ☑ Use for Testing

Resource Managers

Name	Resource Manager
htmltutsql	ODBC
htmltut	Oracle

7. Click OK to dismiss the node's property dialog.

8. Disable the Locked toggle, save, and exit the Environment Manager.

Tutorial Image Files

The Tutorial shows you how to display images on HTML pages. The four graphic files used in the Tutorial should have been installed on your Web server node by the WebEnterprise installer. These must be located in the following directory:

web_server_webenterprise_dir/examples/wedtut/images

The `web_server_webenterprise_dir` directory is where your the WebEnterprise Designer files for the Web server are installed. For example:

- c:\Netscape\Suitespot\forte—for Netscape Enterprise Server on NT
- /usr/netscape/suitespot/forte—for Netscape Enterprise Server on UNIX
- c:\InetPub\forte—for Microsoft IIS

See your System Administrator (or the person who installed WebEnterprise) for the exact location of your WebEnterprise Web server files, if you do not know where they are.

The four files are `canoea.gif`, `canoef.gif`, `paddle1.gif`, and `paddle2.gif`. A fifth file, `2paddles.gif`, is provided to be used in an optional enhancement described in "Auto-Formatting a Field" on page 119.

Building the Tutorial Business Model

This tutorial is based on the database tables you previously installed (see "Installing the Tutorial Database" on page 31). The first step in developing your application is to import the tables you wish to use into the Business Model Workshop. Each table you import becomes a business class in your business model. Any columns in the tables become attributes in their respective business classes.

Your first task is to create the business model.

Creating a Business Model

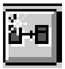
The business model generates the business service objects in the application logic tier of your three-tier application.

➤ **To create the HTMLtutorial business model**

1. Start WebEnterprise Designer.

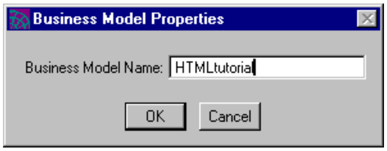
   For information, see "Starting WebEnterprise Designer" on page 141.

2. In the Repository Workshop, click the New Business Model button.

   Alternatively, you can choose Plan > New Business Model.

   The Business Model Properties dialog appears. Use this dialog to name the business model.

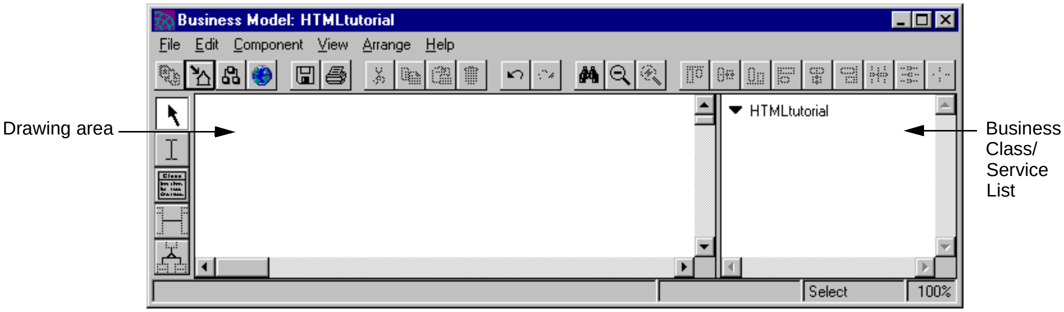
   

   Business Model Properties

   Business Model Name: HTMLtutorial

3. Type HTMLtutorial in the Business Model Name field and click OK.

   The Business Model Workshop appears.


Business Model: HTMLtutorial
File Edit Component View Arrange Help
HTMLtutorial
Select
100%
Drawing area
Business Class/ Service List

The next step is to import the tables you will need for the tutorial.

Importing Database Schema

The Business Model Workshop provides a utility for importing database schema. Before you import your schema, however, you must tell iPlanet UDS which resource manager to use for the database service of the business model.

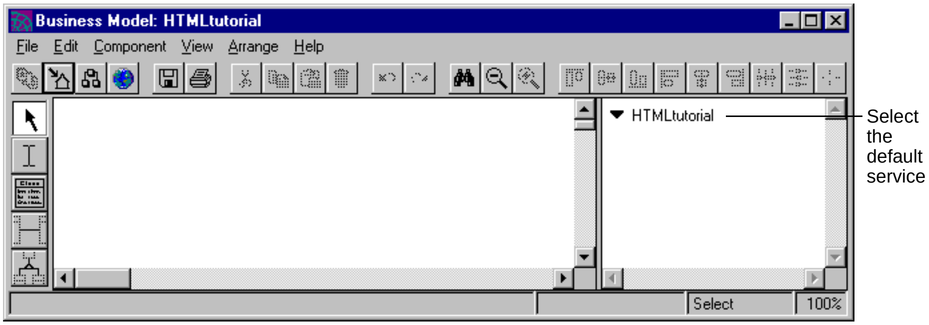
For every business model, WebEnterprise Designer creates a service (called an “Express service”) that serves as a central point of access and control for its business classes. Express services are intended to be used as shared application services for multiple applications. An iPlanet UDS service object capable of load balancing and failover is generated for each Express service.

For more information on Express services, see iPlanet UDS online Help or *A Guide to Express*.

You specify the resource manager as a property of the Express service.

➤ To specify a resource manager for your business model's Express service

1. In the Business Model Workshop for the HTMLtutorial model, select the HTMLtutorial service in the Business Class/Service list.


Business Model: HTMLtutorial
File Edit Component View Arrange Help
HTMLtutorial
Select
100%

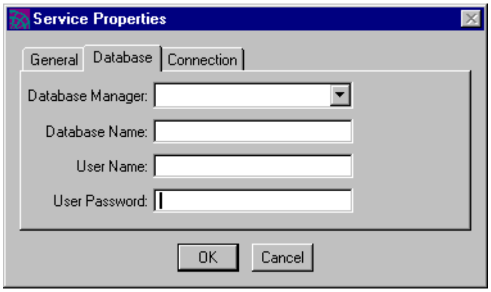
Select the default service

2. Choose the Component > Properties command.

   Alternatively, you can just double-click on the HTMLtutorial service.

   The Service Properties dialog appears.

3. Click the Database tab to display the Database properties.

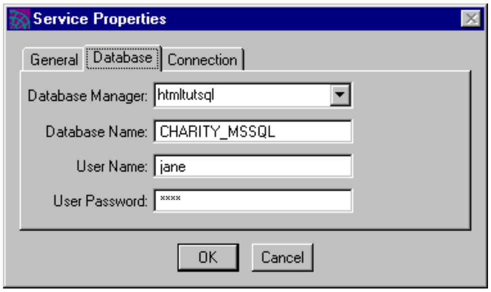


4. Select the name of your resource manager in the Database Manager drop list.

   If your resource manager is not on the list, review the section "Specifying a Resource Manager" on page 33.

5. Fill in the fields with appropriate data.

   For example, if the resource manager is named htmltutsql in "Specifying a Resource Manager" on page 33, on SQLServer database CHARITY_MSSQL with user jane, the specification would be like this:



Service Properties

Database Manager: htmltutsql

Database Name: CHARITY_MSSQL

User Name: jane

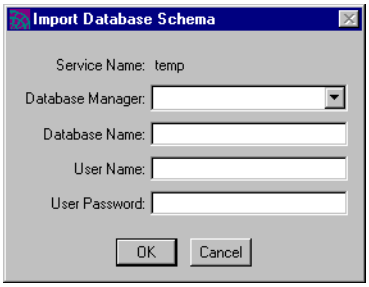
User Password: ****

   Now you are ready to import the database schema.

➤ To import database schema

1. In the Business Model Workshop, choose the File > Import Database Schema command.

   The Import Database Schema dialog appears, as shown below.



Import Database Schema

Service Name: temp

Database Manager:

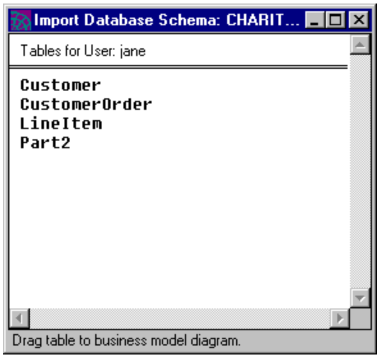
Database Name:

User Name:

User Password:

2. Fill in the fields exactly as you did for the Express service properties and click OK.

   The Import Database Schema window appears, as shown below. This dialog contains a list of tables found in the database schema of the database specified in the Service Properties dialog.



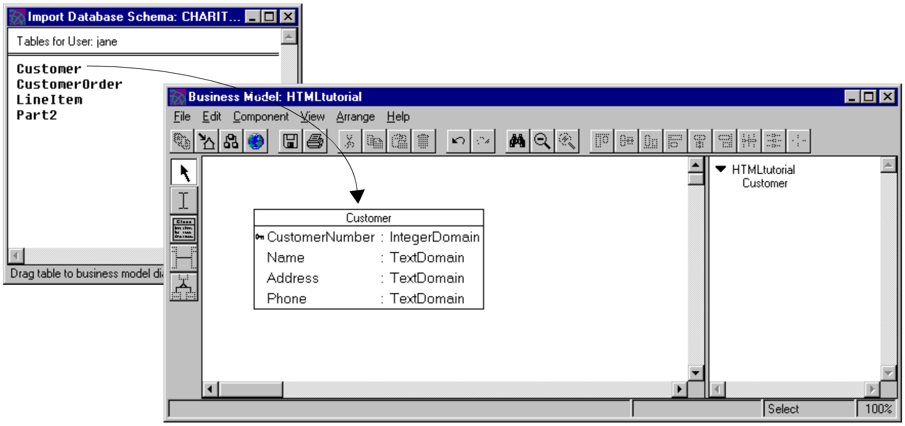
Tables for User: jane

Customer
CustomerOrder
LineItem
Part2

Drag table to business model diagram.

3. Select the table name "Customer," and drag it into the Business Model Workshop.

   A business class named Customer appears—with all its attributes automatically defined—as shown below. Express creates an attribute for each column in the Customer table.



Tables for User: jane

Customer
CustomerOrder
LineItem
Part2

Drag table to business model di

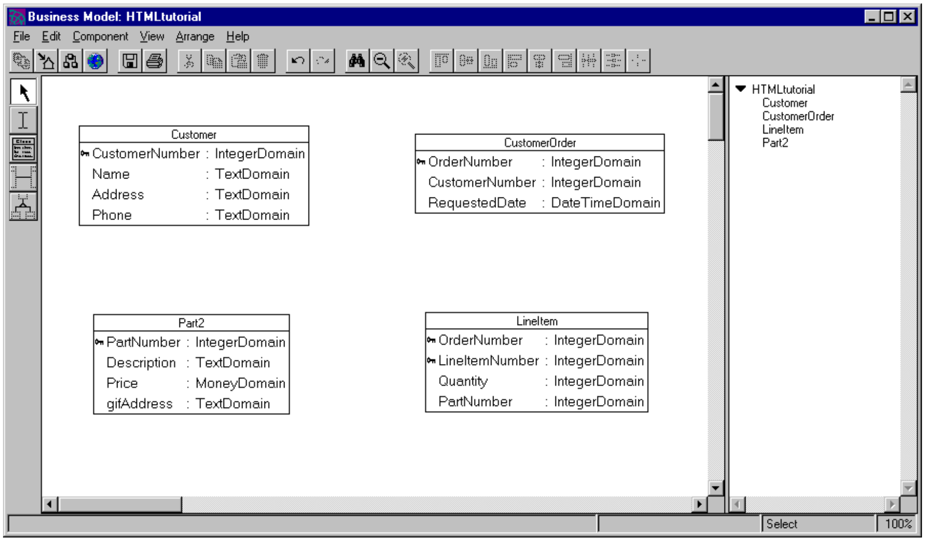
Customer

CustomerNumber	: IntegerDomain
Name	: TextDomain
Address	: TextDomain
Phone	: TextDomain

4. Repeat the process for the CustomerOrder, LineItem, and Part2 tables.

5. Close the Import Database Schema dialog.

   Move the classes around the drawing area to resemble the screen below.



Customer		
🔑 CustomerNumber	:	IntegerDomain
Name	:	TextDomain
Address	:	TextDomain
Phone	:	TextDomain

CustomerOrder		
🔑 OrderNumber	:	IntegerDomain
CustomerNumber	:	IntegerDomain
RequestedDate	:	DateTimeDomain

Part2		
🔑 PartNumber	:	IntegerDomain
Description	:	TextDomain
Price	:	MoneyDomain
gifAddress	:	TextDomain

LineItem		
🔑 OrderNumber	:	IntegerDomain
🔑 LineItemNumber	:	IntegerDomain
Quantity	:	IntegerDomain
PartNumber	:	IntegerDomain

6. Click the Save All button to save your work.

   Alternatively, you can choose the File > Save All command.

About Importing Business Classes

As you can see, Express represents each table you imported as a business class. Each column in the table has a corresponding attribute in the business class.

If you import a table from a database that supports primary keys in the data definition language (DDL), then Express will maintain and represent the primary key in the business class (as is indicated by the key icon). There is a primary key in each business class—refer to "Sample Table Descriptions" on page 32 to verify which attributes are primary keys. If your database does not support primary keys in the DDL, then you must specify a primary key attribute for each business class in your business model. For information about creating primary keys, see *A Guide to Express*.

The Tutorial example uses Microsoft SQLServer, and mentions Sybase and Oracle. All these databases support primary keys.

After you create your business classes, the next step is to create associations between them.

About Editing Class Names

Tables imported from an Oracle database will have all uppercase class and attribute names. These names will appear on your HTML pages in uppercase, which you may not consider attractive. In the HTML Application Model Workshop, you can edit attribute labels, but not class names. Therefore, to create more attractive class labels, edit all your class names (for example, with initial capitals, such as “Customer” and “Part”). Remember, however, that class and attribute names cannot have spaces in them.

Other databases (such as Microsoft SQLServer) can have mixed-case table and column names, which are imported as mixed-case class and attribute labels. Thus you can create the desired labels as database table names and avoid the uppercase issue. Of course, you can still edit them after you import them into the Business Model Workshop, if you prefer different labels.

Creating Associations

You create relationships, or associations, between business classes using the New Association tool. An association describes how one business object relates to another. The way you draw associations affects how the underlying iPlanet UDS classes are generated.

In the generated iPlanet UDS classes, an association from one class to another class is represented by an attribute in the from-class that is used to hold a reference to an object of the to-class. The direction of the association specifies which class contains the reference to the other class. The multiplicity of an association determines if the attribute references a single object or an array of objects, or if many attributes can reference the same object. Generated classes are discussed in detail in *Customizing Express Applications*.

Express assigns default values for association properties—in this tutorial, the defaults do not need to be changed. For more information about association properties, see *A Guide to Express* or online help.

➤ To create the associations of your model

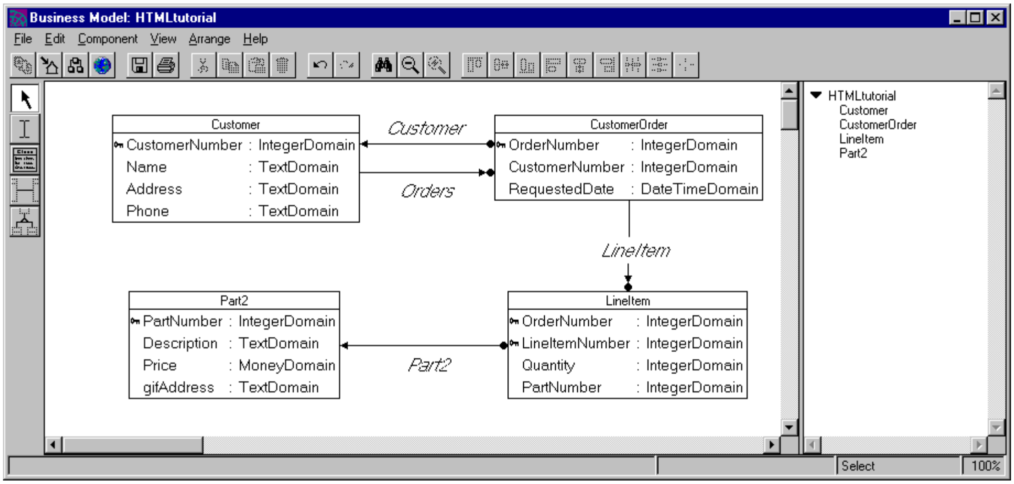
1. Click the New Association button.

   Alternatively, you can choose Component > New > Association.

2. Click on the `Customer` class, drag to the `CustomerOrder` class, and release the mouse button.

   When drawing associations, it does not matter which class you select first: the default direction is based on the primary and foreign keys of the business classes, not on drawing direction. (Step 5 on page 43, shows how to specify a non-default direction.) Notice that the name of the association is "Customer." Conventions for naming associations are described in *Customizing Express Applications*.

3. Create other associations, so that your business model looks like the following figure.



Customer	
CustomerNumber	: IntegerDomain
Name	: TextDomain
Address	: TextDomain
Phone	: TextDomain

CustomerOrder	
OrderNumber	: IntegerDomain
CustomerNumber	: IntegerDomain
RequestedDate	: DateTimeDomain

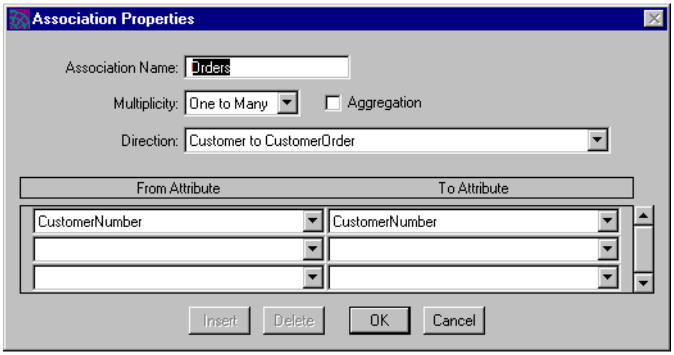
Part2	
PartNumber	: IntegerDomain
Description	: TextDomain
Price	: MoneyDomain
gifAddress	: TextDomain

LineItem	
OrderNumber	: IntegerDomain
LineItemNumber	: IntegerDomain
Quantity	: IntegerDomain
PartNumber	: IntegerDomain

4. To make the Orders link between the `Customer` and `CustomerOrder` classes, create an association and double-click on it to display the Association Properties dialog.

   The direction of an association in the business model affects link options in the HTML application model. In our HTML model (see "Linking the Master-Detail Pages to the Application" on page 99), we will need an association directed from the `Customer` class to the `CustomerOrder` class.

5. Specify the fields as shown in the following figure:



Association Name: Orders

Multiplicity: One to Many ☐ Aggregation

Direction: Customer to CustomerOrder

From Attribute	To Attribute
CustomerNumber	CustomerNumber

6. Save your work.

You now have a complete business model. Once you have established your business classes and their relationships, you can generate the TOOL classes.

Generating the Server Classes

The business classes you create represent the key business objects manipulated by the application. In Express, business objects are managed by services. These services provide the second tier of a three-tier application architecture. Express automatically creates the TOOL classes corresponding to the business classes as well as the service objects providing the services used to access the business classes. The services contain all the code required to access and update the database, as well as concurrency management capabilities required by a scalable application.

Deferring code generation

You can go directly to the HTML Application Model Workshop without generating server code at this point. When it comes time to generate code for the HTML model, iPlanet UDS checks first for the presence of server code and automatically generates it first, if it does not exist.

➤ To generate server classes

1. Click the Generate Server Code button.

   Alternatively, you can choose File > Generate Server Code.

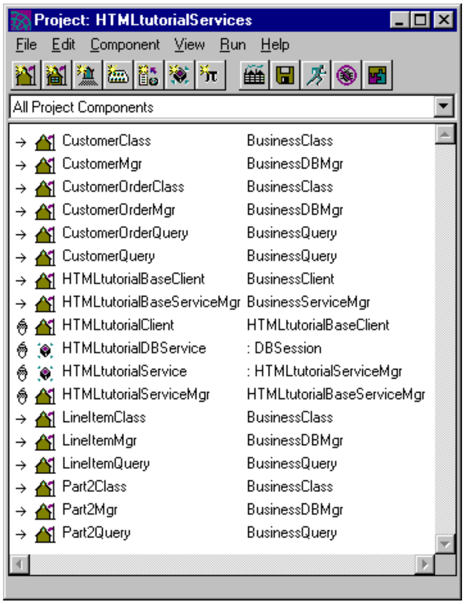
   A status box appears, reporting the progress of the generation. Server code is generated in the *business_model*Services project.

Viewing Generated Server Code

To view the generated server code, open the HTMLtutorialServices project from the Repository Workshop.

The HTMLtutorialServices project includes the following classes:

Figure 2-1 Generated Server Code for HTMLtutorial



CustomerClass	BusinessClass
CustomerMgr	BusinessDBMgr
CustomerOrderClass	BusinessClass
CustomerOrderMgr	BusinessDBMgr
CustomerOrderQuery	BusinessQuery
CustomerQuery	BusinessQuery
HTMLtutorialBaseClient	BusinessClient
HTMLtutorialBaseServiceMgr	BusinessServiceMgr
HTMLtutorialClient	HTMLtutorialBaseClient
HTMLtutorialDBService	: DBSession
HTMLtutorialService	: HTMLtutorialServiceMgr
HTMLtutorialServiceMgr	HTMLtutorialBaseServiceMgr
LineItemClass	BusinessClass
LineItemMgr	BusinessDBMgr
LineItemQuery	BusinessQuery
Part2Class	BusinessClass
Part2Mgr	BusinessDBMgr
Part2Query	BusinessQuery

Generated server classes are subclasses of the classes in the ExpressServices project. The generated classes are described in the following table:

Generated Class	Description
*business_class*Class	Based on the business class defined in the business model. Each object stores a record of data from the table on which the business class is based.
*business_class*Query	Represents the information used to retrieve the data for the *business_class*`Class` from the underlying database; also represents information used to update the underlying database. This class defines constants, used as attribute indexes, for each of the attributes defined for the *business_class*`Class`.
*service_name*BaseClient	Provides the interface between the application client and the *service_name*`Mgr`. Lets the client select and update *business_class*`Class` objects, which are managed by the corresponding *service_name*ServiceMgr.
*service_name*Client	Contains your customizations for *service_name*`BaseClient`. This class is not regenerated when you regenerate your business model code, preserving your customizations.
*business_class*Mgr	Manages select and update operations for the corresponding *business_class*`Class` on the underlying database server.
*service_name*BaseServiceMgr	Accepts select and update requests from BusinessClient objects and sends them to the appropriate *business_class*`Mgr`.
*service_name*ServiceMgr	Contains your customizations for *service_name*`BaseServiceMgr`.

In addition, two service objects are generated:

Generated Service Object	Description
*service_name*DBService	Provides the database session.
*service_name*Service	Manages database interaction.

For information about partitioning and using these service objects, see "About HTML Application Projects and Service Objects" on page 107.

Your next step is to build your HTML model.

Starting the Tutorial HTML Application

This tutorial HTML application is based on the business model you previously created (see "Building the Tutorial Business Model" on page 35). The business model defines the objects and business services of the application. Now, you will create the user interface for your application and the services for the interface. You do this by creating an HTML application model.

This section shows you how to use the HTML Application Model Workshop and wizards to create the HTML model for the client side of an application. First, however, we will discuss the elements of the model you will be creating.

About HTML Application Models

You create an HTML application model with the HTML Application Model Workshop. You will learn how to start and use the workshop in the tutorial.

An HTML application model is a graphical representation of the user interface of your application. HTML application models consist of pages connected by links. The pages represent HTML pages to be displayed in a Web browser. The links specify the flow of control between the pages, how they communicate with each other, how they are invoked, and how data is passed between them.

Figure 2-2 shows an HTML application model and identifies the types of items you can create in a model. This model is similar to the one you will create for the tutorial.

Figure 2-2 An HTML Application Model

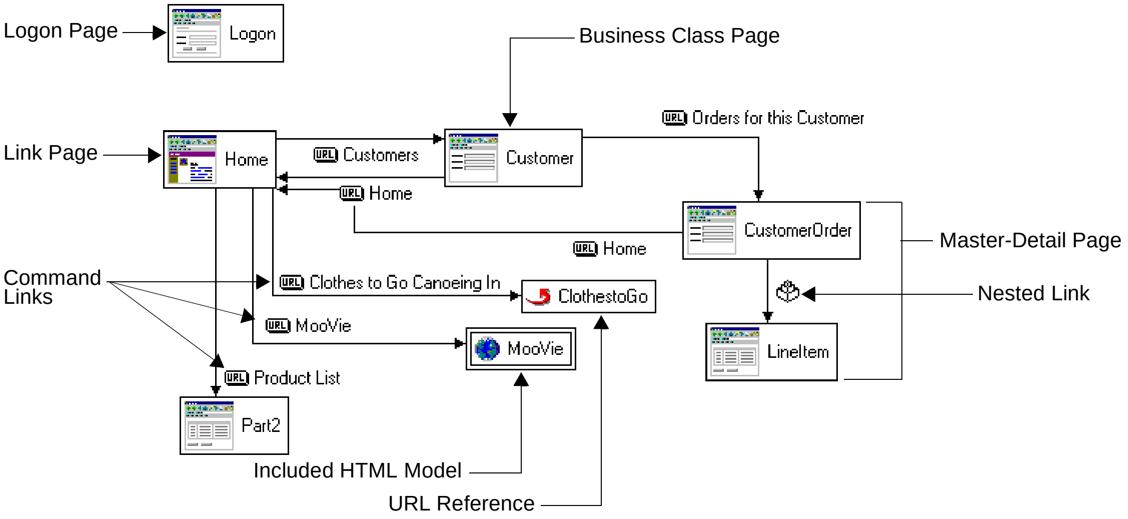


Table 2-2 describes the different elements of the HTML model shown in Figure 2-2.

Table 2-2 The Elements of an HTML Application Model

Item	Description	For more information, see
Business class page	A page based on a business class in a business model that is a supplier plan to the HTML application model. You can create multiple pages for each business class. Business class pages can be formatted as form or list pages.	"Creating a Business Class Page" on page 57
Link page	A page that displays links to other pages. You can link to business class pages, URL reference pages, or pages in other HTML application models.	"Creating a Link Page" on page 77
Logon page	A page that prompts for user name, password, or other authentication criteria. The developer must supply the custom code for user validation.	The chapter on security in *Customizing WebEnterprise Designer Applications*.

Table 2-2 The Elements of an HTML Application Model *(Continued)*

Item	Description	For more information, see
Master-detail page	A page that displays one or more called pages within a calling page. This type of pages is constructed of business class pages connected by a nested or folder link. See Nested link, below.	"Creating a Master-Detail Page" on page 93
Command link	The most common type of link. The user invokes the called page by clicking a hypertext area on the calling page. The hypertext area can be the link's label, a field's label, or a field's data.	"Creating a Command Link" on page 79
Nested link, Folder link	A link that displays the called page nested within the calling page. Any two classes can be nested, but this is most common where the two underlying business classes have a one-to-many relationship. If the relationship is an aggregate relationship, a nested link is the only kind of link allowed. A folder link is a nested link that displays multiple called pages within the calling page.	"Creating a Nested Link" on page 98
Included HTML model	A link to a page in another HTML application model.	iPlanet UDS online Help
URL reference	A URL reference to another page. You can either specify the URL explicitly or specify a class attribute that specifies the URL.	"Creating a URL Reference" on page 121

About Business Class Pages

A *business class page* is a page based on a business class in a business model that is a supplier to the HTML application model. When you assign a business class to a page, WebEnterprise Designer automatically creates this type of page. The page in the model represents a set of actual HTML pages in a generated application. The page set includes a page for each page mode. The attributes of the underlying business class appear as fields on the corresponding HTML page at runtime. Once you generate the business class page's HTML code, you can view and modify the business class page in your HTML editor, just like any other HTML page.

About Page Designs and Styles

Page designs and styles determine what your pages will look like in the browser. You specify the default design or style as a property of the HTML application. You can then specify a different design or style for an individual page as a property of that page.

Page designs Page designs are boilerplate layouts for the pages of your application. Designs determine how elements are placed on the page in the browser. WebEnterprise Designer provides a number of designs. These are defined by a set of HTML files found in the `FORTE_ROOT/userapp/express/clx/designs` directory. You can customize these files or create your own designs. For information on customizing designs or creating your own, see *Customizing WebEnterprise Designer Applications*.

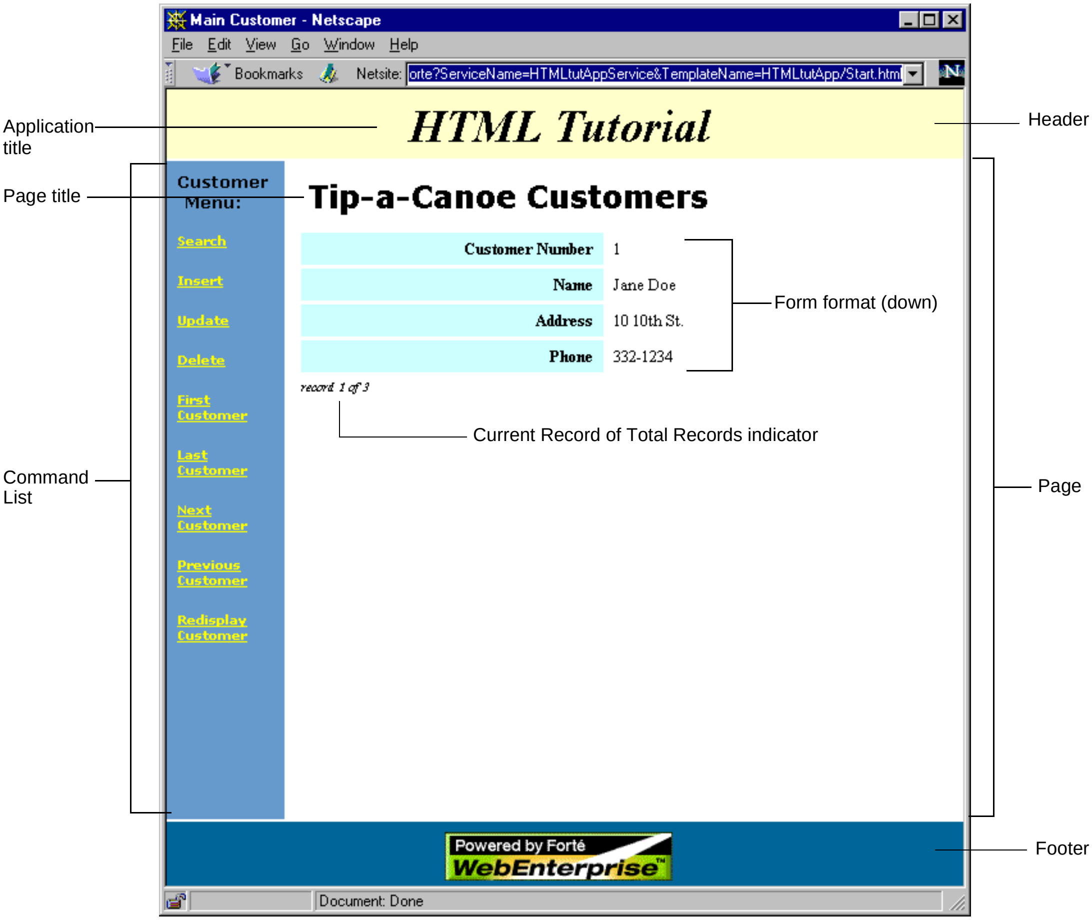
Page styles Page styles are boilerplate specifications of font and color characteristics. WebEnterprise Designer provides two styles, Cool and Steel. Page styles are defined in Cascading Style Sheet files (.css extension), found in the `/forte/styles` directory under the Web server directory. Other page style files are in the `FORTE_ROOT/userapp/express/clx/styles` directory, which include the definition text and example graphic that display in the Styles dialog when you select a style.

You can customize these files or create your own styles. For information on customizing page styles, see *Customizing WebEnterprise Designer Applications*.

About Your First Page

Now you are ready to create the first page of your HTML application model. This page will display customer information, so it will be based on the Customer business class page. Figure 2-3 shows you what the page will look like in your browser:

Figure 2-3 Anatomy of a Runtime HTML Page



Customer Number	1
Name	Jane Doe
Address	10 10th St.
Phone	332-1234

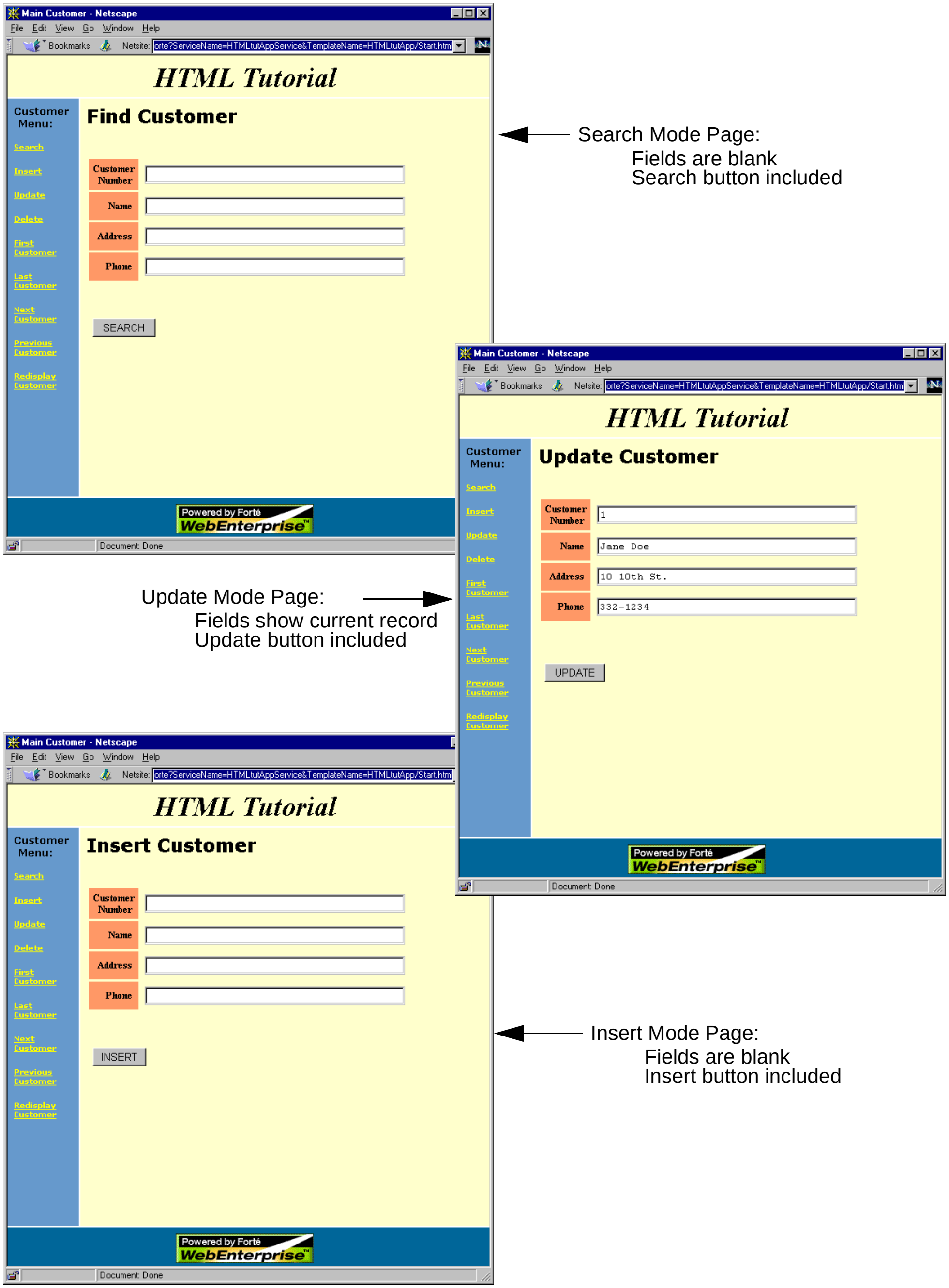
record 1 of 3

You will create the Customer business class page as a *form* page. A form page displays data one record at a time. The way the record is displayed is configurable, either down the page (the default) or across the page.

Page modes

This page, and every business class page, has several *modes*. Modes are different formats the page is displayed in under different conditions. For example, when you click the `Insert` command (in the command list), the page is displayed in Insert mode. Similarly, the `Search` and `Update` commands display the page in Search and Update modes. The different modes of the page in Figure 2-3 are shown in Figure 2-4.

Figure 2-4 Update, Search, and Insert Modes of a Form Page



HTML Tutorial

Find Customer

Customer Number	
Name	
Address	
Phone	

Search Mode Page:
Fields are blank
Search button included

HTML Tutorial

Update Customer

Customer Number	1
Name	Jane Doe
Address	10 10th St.
Phone	332-1234

Update Mode Page:
Fields show current record
Update button included

HTML Tutorial

Insert Customer

Customer Number	
Name	
Address	
Phone	

Insert Mode Page:
Fields are blank
Insert button included

You will see some of these modes in action when you finish the first part of the tutorial.

Now, your first task is to create a new HTML application model.

Creating an HTML Application Model

To create an HTML application model means to open the HTML Application Model Workshop and define the basic properties of the model you wish to create. The HTML Application Wizard helps you through this process.

You can open the HTML Application Model Workshop from the Business Model Workshop or the Repository Workshop. If you open it from the Business Model Workshop, the current business model automatically becomes a supplier plan to the new HTML application model. If you open it from the Repository Workshop, you have to specify supplier plans manually.

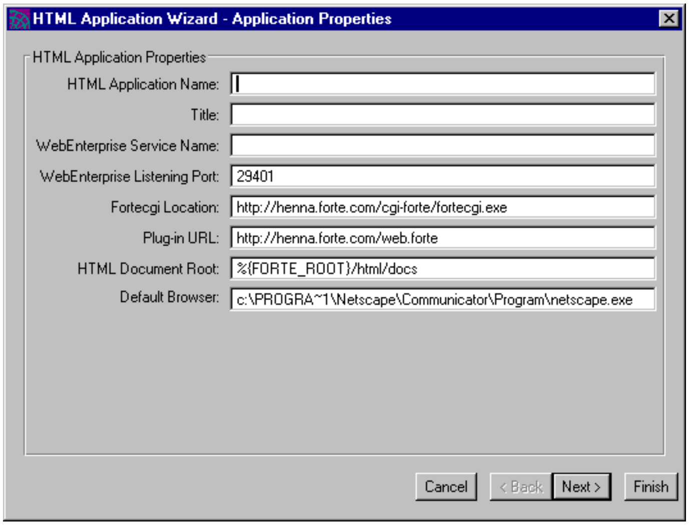
For complete information on all pages of this wizard, see the iPlanet UDS online Help system. Either search the index for “HTML Application Wizard,” or press your Help key when the Wizard is the active window on your machine.

➤ **To create an HTML application**

1. In the Business Model Workshop (displaying the HTMLtutorial business model), click the New HTML Model button.

   Or, you can choose File > New HTML Model. This button and command are also available in the Repository Workshop (the command is off the Plan menu).

   The HTML Application Wizard appears. The first page of this Wizard is used to define the name of the application and other related information.


HTML Application Wizard - Application Properties
HTML Application Properties
Cancel
< Back
Next >
Finish

HTML Application Name:

Title:

WebEnterprise Service Name:

WebEnterprise Listening Port: 29401

Fortecgi Location: http://henna.forte.com/cgi-forte/fortecgi.exe

Plug-in URL: http://henna.forte.com/web.forte

HTML Document Root: %{FORTE_ROOT}/html/docs

Default Browser: c:\PROGRA~1\Netscape\Communicator\Program\netscape.exe

2. Type HTMLtutApp in the application name field and press the Tab key.

The Title and WebEnterprise Service Name fields are filled automatically with default values based on the HTML application name.

NOTE You must *not* use spaces in the HTML application name, because this string is the application name in the Repository Workshop, and the basis for several file names.

What are these other fields? The WebEnterprise Listening Port specifies the port number of the socket upon which the WebEnterprise application's AccessServer service object "listens" for web requests. WebEnterprise Designer automatically generates a random number for this property when you first create the HTML application model with the HTML Application Wizard. When you eventually deploy your application, this port must not be in use on the AccessServer service object node. In such a case, you can modify this application property. This number should be set to an integer value between 1025 and 65535.

The values of the other four fields are defined by WebEnterprise-specific environment variables, usually set at install time. See the *WebEnterprise Installation Guide* for more information. For a description of all fields on this dialog, press your Help key to display context-sensitive Help.

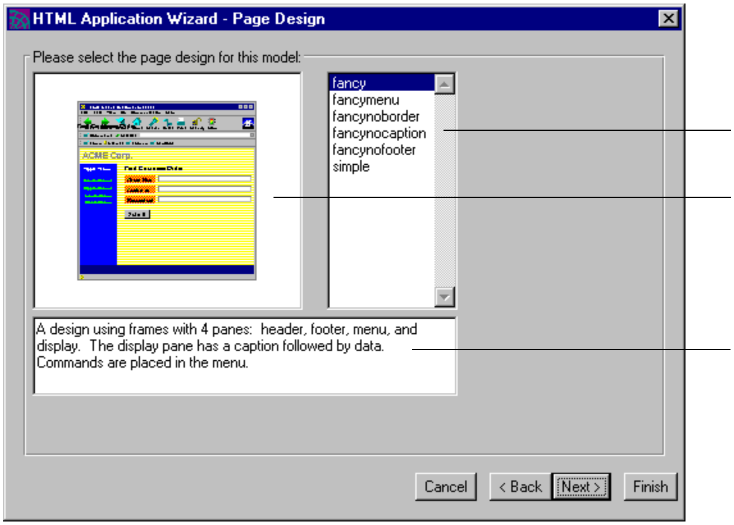
3. Change the title by typing `HTML Tutorial` in the Title field.

   The title appears at the top of all the application's pages in the browser (see Figure 2-3 on page 50).

   NOTE You *can* put spaces in the application title, because it is just text that appears at the top of the application's pages.

4. Click Next.

   The HTML Application Wizard displays a page for specifying the page design to use:



HTML Application Wizard - Page Design

Please select the page design for this model:

A design using frames with 4 panes: header, footer, menu, and display. The display pane has a caption followed by data. Commands are placed in the menu.

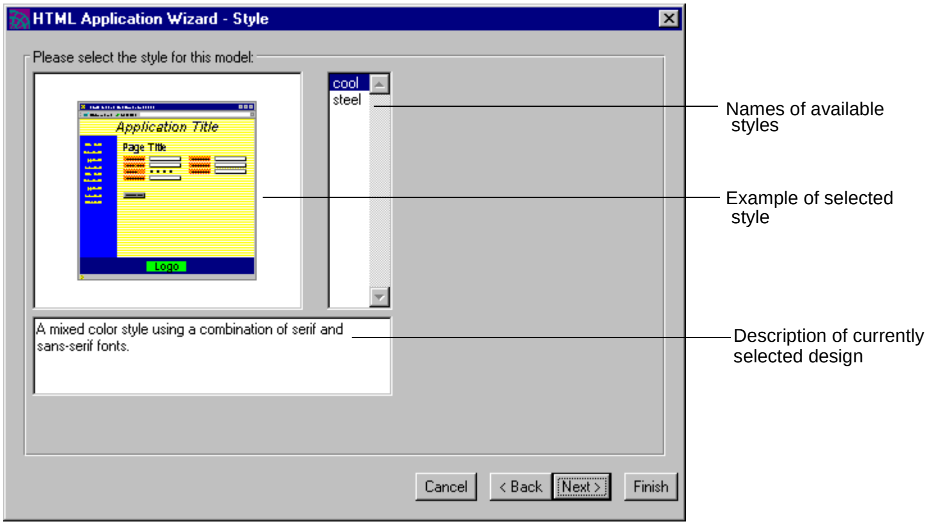
Names of available designs

Example of selected design

Description of currently selected design

5. Select the “fancy” design and click Next.

   The HTML Application Wizard displays a page for selecting a style for your page style:

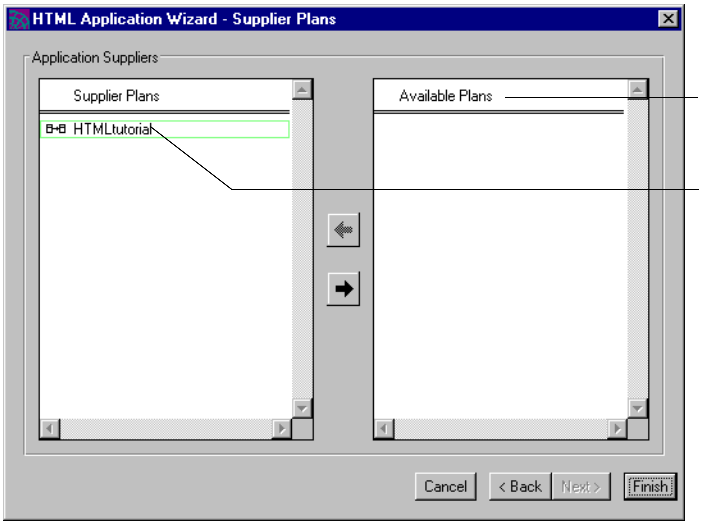

HTML Application Wizard - Style
Please select the style for this model:
Application Title
Page Title
Logo
cool
steel
Cancel
< Back
Next >
Finish
Names of available styles
Example of selected style
Description of currently selected design

A mixed color style using a combination of serif and sans-serif fonts.

   For brief information on page designs and styles, see “About Page Designs and Styles” on page 49. For information on how to customize them, see *Customizing WebEnterprise Designer Applications*.

6. Select the “cool” style and click Next.

   The HTML Application Wizard now displays a page for specifying supplier plans to the application, as shown below.



List of all other business models (if any) in your workspace

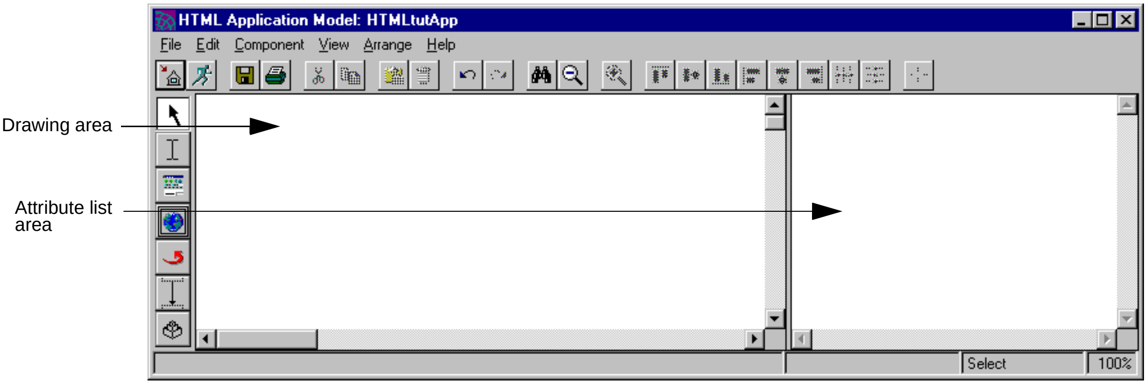
The business model from which you started the HTML Wizard

   You would add another business model as a supplier if you wanted to use a class from that model as a page in your HTML model. In this tutorial, you do not want to do that, so you will ignore this page.

NOTE When you start the HTML Application Wizard from the Repository Workshop instead of the Business Workshop, there are no default suppliers. In that case, you would have to select at least one plan from the Available Plans pane and move it to the Supplier Plans pane.

7. Click Finish.

   The HTML Application Model Workshop appears.


HTML Application Model: HTMLtutApp
File Edit Component View Arrange Help
Drawing area
Attribute list area
Select
100%

You have finished defining the basic properties of your HTML application. Your next task is to create an HTML page.

To pause here, choose File > Close to exit the HTML Application Model Workshop. You can resume later, starting with the next section.

Creating a Business Class Page

Now you will create an HTML page based on the Customer business class. This class stores the customer data: name, address, phone number, and customer number. You will create a simple form window to display this information, with appropriate commands for searching a particular customer and navigating through the customer records.

Starting the Page Wizard

You create pages with the help of the Page Wizard. For complete information on all pages of this wizard, see iPlanet UDS online Help. For a description of all fields on a particular page of the Page Wizard, simply press the Help key when the page is the active window.

➤ To start the Page Wizard

1. If the HTML Application Model Workshop is not already open, open it from the Repository Workshop by double-clicking on the HTMLtutApp application.

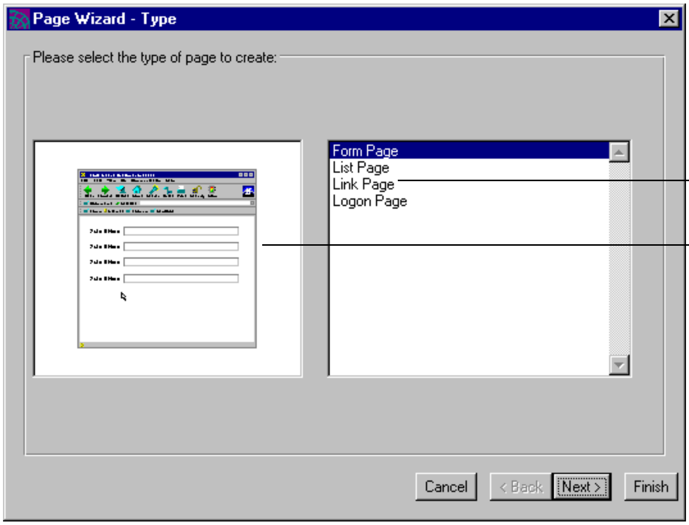
2. Click the New HTML Page button.

   Alternatively, you can choose Component > New > HTML Page.

   The cursor turns into an upward arrow.

3. Click anywhere in the workshop's drawing area.

   The Page Wizard appears:



Please select the type of page to create:

Names of available page types

Example of selected page type

Continue on to the next section to define your page.

Defining a Form Page

There are several types of pages available: form, list, link, and logon. Here you will create a form page, which will display record one at a time.

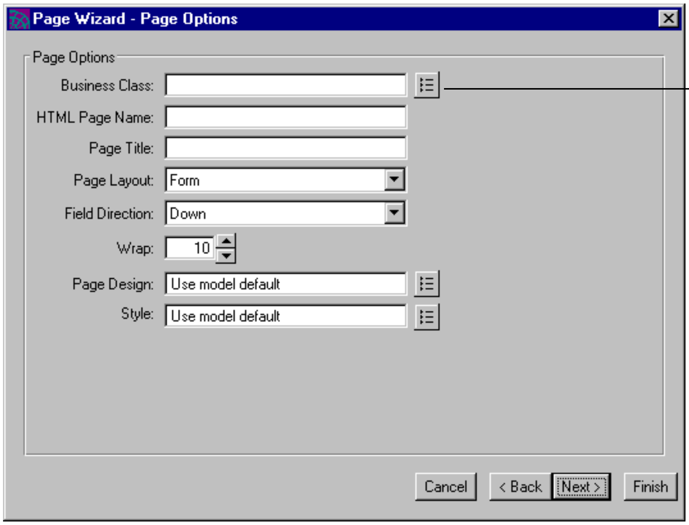
➤ To define a form page based on the Customer business class

1. If not already selected, select Form Page from the list in the Page Wizard.

   An example Form Page is displayed in the left window.

2. Click Next.

The Page Wizard displays a page for specifying basic page properties:



Business Class:

HTML Page Name:

Page Title:

Page Layout: Form

Field Direction: Down

Wrap: 10

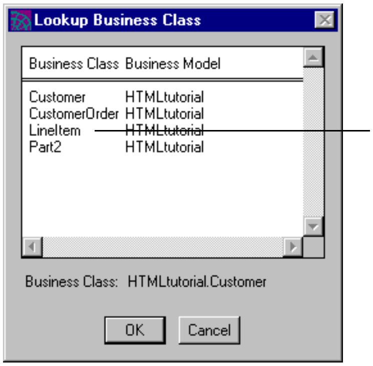
Page Design: Use model default

Style: Use model default

Browser button

3. Click the browser button to look up possible business classes on which to base the page.

A lookup window appears, showing classes available from the business models you specified as supplier plans in Step 6 on page 56:



Business Class	Business Model
Customer	HTMLtutorial
CustomerOrder	HTMLtutorial
LineItem	HTMLtutorial
Part2	HTMLtutorial

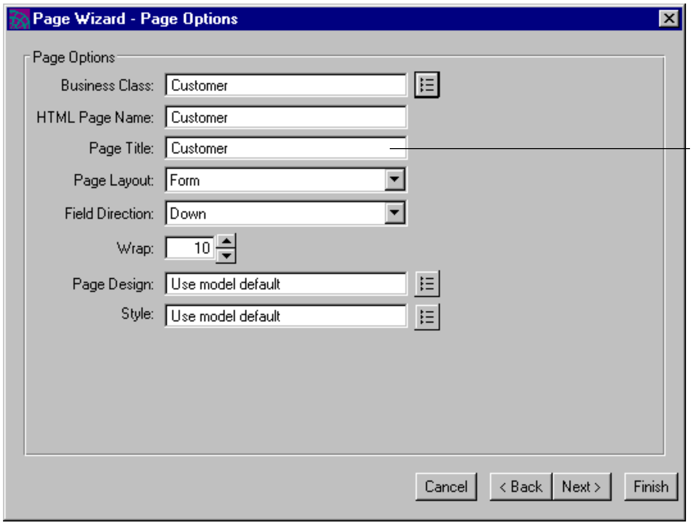
Business Class: HTMLtutorial.Customer

Classes in the current business model

NOTE Your list may look different from this one. Refer to "About Editing Class Names" on page 41.

4. Select Customer and click OK (or double-click Customer).

The Page Wizard fills the property values on the Page Options page:



Business Class: Customer

HTML Page Name: Customer

Page Title: Customer

Page Layout: Form

Field Direction: Down

Wrap: 10

Page Design: Use model default

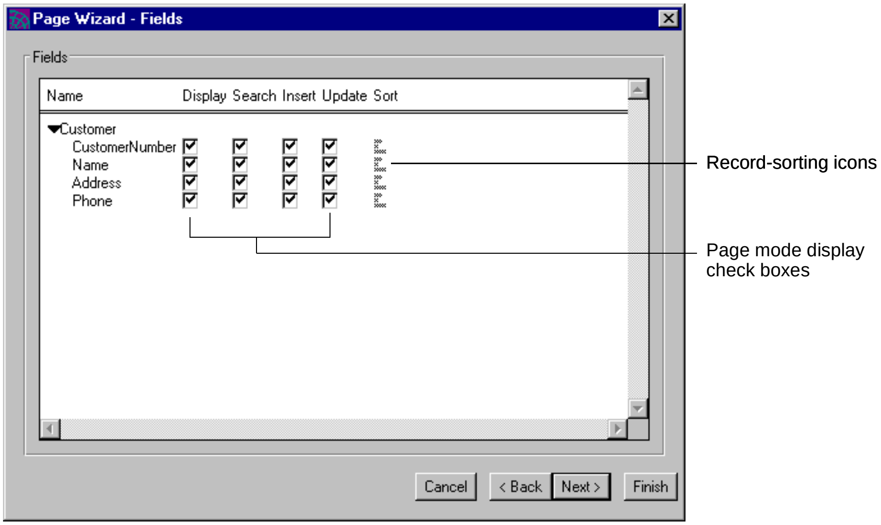
Style: Use model default

Change title to “Tip-a-Canoe Customers”

5. Replace the default page title (Customer) with Tip-a-Canoe Customers and click Next.

NOTE The page title *can* have spaces in it, because it is just text that appears at the top of a portion of the application’s pages. The HTML page name must *not* contain any spaces, because this string is the basis for several file names.

The Page Wizard displays a page for adjusting properties for fields (attributes), as shown below:



Page Wizard - Fields

Fields

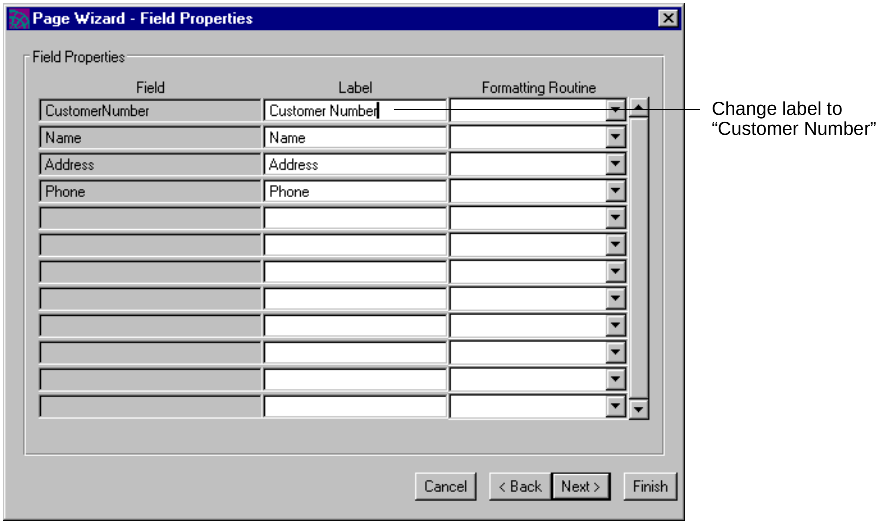
Name	Display	Search	Insert	Update	Sort
Customer					
CustomerNumber	☑	☑	☑	☑	
Name	☑	☑	☑	☑	
Address	☑	☑	☑	☑	
Phone	☑	☑	☑	☑	

Record-sorting icons

Page mode display check boxes

Ignore these field properties for now. They will be discussed later. For information on the properties on this page, press your Help key when this page is active.

6. Click Next to go to the Field Properties page of the Page Wizard.



Page Wizard - Field Properties

Field Properties

Field	Label	Formatting Routine
CustomerNumber	Customer Number	
Name	Name	
Address	Address	
Phone	Phone	

Change label to “Customer Number”

7. Use this page to replace the default labels (which are based on the attribute names) with more readable names.

   Even if you edited the class attributes in the Business Model Workshop (see "About Editing Class Names" on page 41), you still must add a space in the label for CustomerNumber (to make it Customer Number).

   Leave the "Formatting Routine" fields blank so that WebEnterprise Designer will use the default formatting values determined by your browser.

8. Click Next twice to skip over the Validation page to get to the Command Options page.

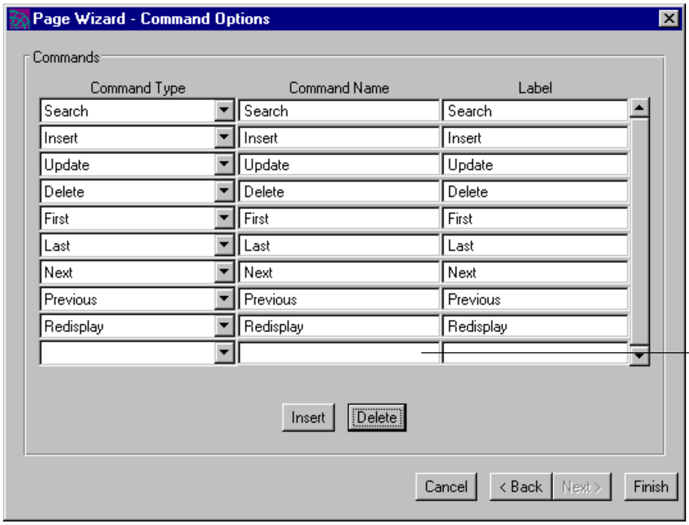
   Validation will be illustrated in "Setting a Field Validation Option" on page 117.

Specifying Command Options

Your task now is to specify which commands you want on the page, their order, and the text of their labels.

➤ To modify command options for the Customer page

1. On the Command Options page, delete the Go To command by placing the cursor anywhere in the Go To command row and clicking Delete.



Page Wizard - Command Options

Commands

Command Type	Command Name	Label
Search	Search	Search
Insert	Insert	Insert
Update	Update	Update
Delete	Delete	Delete
First	First	First
Last	Last	Last
Next	Next	Next
Previous	Previous	Previous
Redisplay	Redisplay	Redisplay

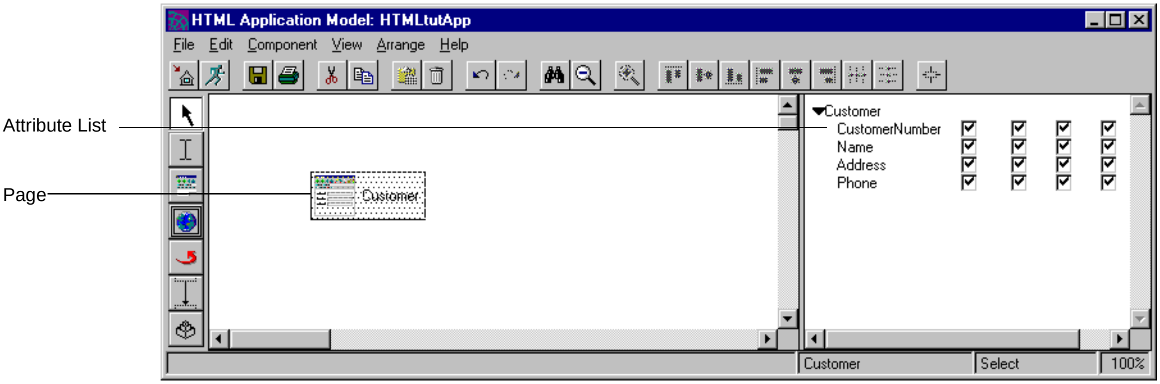
Go To row removed

Go To is useful when the number of records expected to be returned from a database will not fit on a page. The user can type in a record number and click the GoTo button to view the record. However, this application has very few records, so the Go To button is not necessary.

2. On the same page, add the word “Customer” to the labels of the First, Last, Next, and Previous commands (for example, “First Customer,” “Last Customer”).

3. Click Finish to exit the Page Wizard.

   The HTML Application Model Workshop displays the Customer page:



You have just finished creating your first business class page in your HTML application model.

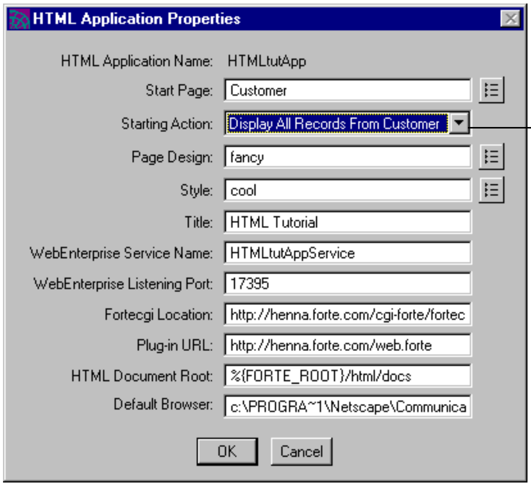
Setting the Starting Action of the Application

You can set whether the application starts with all records displayed on the page, with no records displayed, or in the Search mode of the page. By default, the page starts in Display mode containing no records. You will set the Customer page to start with all records displayed.

➤ To set the starting action of an HTML application

1. In the HTML Application Model Workshop, choose File > Properties.

   The HTML Application Properties dialog appears.



HTML Application Properties

HTML Application Name: HTMLtutApp

Start Page: Customer

Starting Action: Display All Records From Customer

Page Design: fancy

Style: cool

Title: HTML Tutorial

WebEnterprise Service Name: HTMLtutAppService

WebEnterprise Listening Port: 17395

Fortecgi Location: http://henna.forte.com/cgi-forte/fortec

Plug-in URL: http://henna.forte.com/web.forte

HTML Document Root: %{FORTE_ROOT}/html/docs

Default Browser: c:\PROGRA~1\Netscape\Communica

Starting action drop list

2. Select Display All Records From Customer in the Starting Action drop list.

3. Click OK to apply the changes and dismiss the HTML Application Properties dialog.

Generating HTML and TOOL Code

At this point, the default settings for the Customer page are sufficient to run the application. In the next step, you generate the HTML template files and Web application server code.

Deferring code generation You can defer generating code, because iPlanet UDS automatically generates the necessary code before it runs the application, which you will do shortly.

➤ **To generate Web application services and HTML page files**

1. In the HTML Application Model Workshop, click the Generate Web Application Server Code button, or choose the File > Generate Web Application Server Code command.

   A status box appears, reporting the progress of the generation. When code generation is completed, the message “Code generated.” appears in the status bar.

Viewing the Generated Code

WebEnterprise Designer generates both HTML template files and TOOL services from the HTML application model. The HTML template files support the pages that are displayed in a Web browser, which is the user interface to the application. The TOOL services handle requests from the Web server and generate dynamic HTML content by contacting the business services that interact with the database.

In particular, generated code includes:

- HTML template files that define the pages in the application

  These HTML template files contain HTML statements and iPlanet UDS tags that are replaced by dynamic data (including database data) at runtime.

- a TOOL project that includes:

  - an AccessService service object to handle HTTP requests from the pages by way of the Web server, pass them to the Scanner Service, and later cache the result set passed back from the Scanner Service

  - a ScannerService service object to transform the HTML template files into HTML pages, which are passed back and rendered by the Web browser

HTML Template Files

The HTML files are generated into the directory defined by the FORTE_WW_DOCUMENT_ROOT environment variable, or, if that is not set, by default in `%{FORTE_ROOT}/html/docs/`*html_model*. (*html_model* is the name of the HTML application model.) HTML files are most easily readable with an HTML editor, although you can also read them in a text editor. Two text files—`manifest.txt` and `report.txt`—are also generated, which report on the process of generating HTML template files.

For a complete description of generated HTML template files, see *Customizing WebEnterprise Designer Applications*.

Two identical sets of HTML files are created: One set is created under `%{FORTE_WW_DOCUMENT_ROOT}/`*html_model* and one in `%{FORTE_WW_DOCUMENT_ROOT}/`*html_model*`/.base`.

CAUTION The .base directory files are used internally by WebEnterprise Designer and should not be modified by the user.

The files created for the one-page application you just built are:

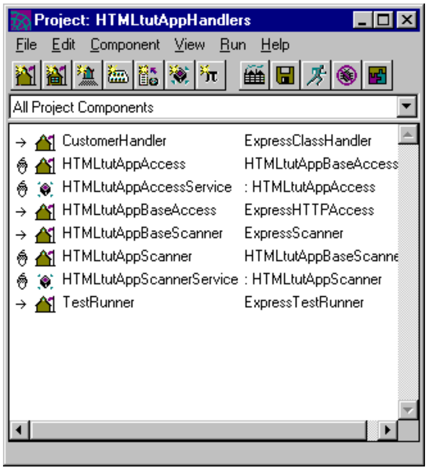
File Name	**Description**
Main_Customer.html	Basic HTML layout of Customer page.
Data_Customer.html	Provides the tags used to put the data (but not nested data) in the page to be displayed.
Display_Customer.html	Layout of the Data Display portion of the Customer page, including nested pages.
Insert_Customer.html	Insert mode of Customer page.
Menu_Customer.html	Menu area for Customer page.
Search_Customer.html	Search mode of Customer page.
Update_Customer.html	Update mode of Customer page.
Footer_Customer.html	Defines the footers of all the model's pages.
Header_Customer.html	Defines the headers of all the model's pages.
Start.html	The starting page of the application.
AccessError.html	Displays an error message for an unauthorized user. Used only if security is implemented (usually in conjunction with a Logon page).
manifest.txt	Lists all the HTML files generated and so serves as a catalog of the directory.
report.txt	Reports on the process of generating the HTML templates, and what has been customized.

Web Application Server Code

To view the generated Web application server code, open the HTMLtutAppHandlers project from the Repository Workshop.

The HTMLtutAppHandlers project includes the following classes:

Figure 2-5 Generated Server Code for HTMLtutApp


Project: HTMLtutAppHandlers
File Edit Component View Run Help
All Project Components

CustomerHandler	ExpressClassHandler
HTMLtutAppAccess	HTMLtutAppBaseAccess
HTMLtutAppAccessService	: HTMLtutAppAccess
HTMLtutAppBaseAccess	ExpressHTTPAccess
HTMLtutAppBaseScanner	ExpressScanner
HTMLtutAppScanner	HTMLtutAppBaseScanne
HTMLtutAppScannerService	: HTMLtutAppScanner
TestRunner	ExpressTestRunner

Generated server classes are subclasses of the ExpressHandlers project. The generated classes are described in the following table:

Generated Class	Superclass	Description
*bus_class_page*Handler	ExpressClassHandler	Receives requests from Scanner to get data. One *bus_class_page*Handler is generated per business class page.
TestRunner	ExpressTestRunner	Sets up whatever is required to test run the application, including starting the business server, and launching the Web browser. One TestRunner class is generated per HTML application.
*html_model*BaseAccess	ExpressHTTPAccess	Handles HTML requests from the HTTP Web server and responds with the appropriate Web page.

Generated Class	Superclass	Description
*html_model*Access	*html_model*BaseAccess	Customizable subclass of the **BaseAccess** class.
*html_model*BaseScanner	ExpressScanner	Responds to HTTP requests from the Access server for pages by opening the specified template file and processing the iPlanet UDS tags to generate a Web page. It also passes requests to the iPlanet UDS business model server partition.
*html_model*Scanner	*html_model*BaseScanner	Customizable subclass of the *html_model*`BaseScanner` class.

In addition, two service objects are generated:

Generated Service Object	Type	Description
*html_model*AccessService	*html_model*Access	Receives HTTP requests from the Web server and passes them to the ScannerService, then receives the response from the ScannerService and passes the page information back to the Web server.
*html_model*ScannerService	*html_model*Scanner	Converts requests from the AccessService service object into pages.

Running an HTML Application

At this point, you have defined a business class, created its associated business class page, and generated an HTML application. Now you can run this application and examine the functionality WebEnterprise Designer provides by default.

When you test run the application in WebEnterprise Designer, iPlanet UDS partitions the application so that as much of it as possible will run on your local machine. Any service objects that the local node cannot support are assigned to a suitable node in your development environment, allowing you to test the application independently of its deployment environments.

Starting the Application in a Web Browser Manually

An HTML application's pages are displayed in a Web browser. When you start your HTML application, you can enter the application's starting URL into the URL field of a running Web browser. This causes the application's starting page to be displayed in the browser.

➤ **To start the application in the browser manually**

1. In the HTML Application Model Workshop, click the Run Application button.

   Alternatively, you can select HTMLtutAppHandlers in the Repository Workshop and click the Run Application button from there.

   If you run the application from the workshop, WebEnterprise Designer always checks that the generated classes are up-to-date in both workshops before attempting to run an application. Therefore, if you haven't generated the classes yet (or if you made changes to either model and forgot to regenerate), WebEnterprise Designer does this for you before it runs the application.

   If you run the application from the Repository Workshop, you must explicitly generate the classes beforehand, if they are not up-to-date.

2. Open the Launch Server window and find the line containing the string:

   ```
   ***URL of starting page is =URL_application_string
   ```

3. Start your browser, if it is not already running.

4. Copy the above line from the iPlanet UDS Launcher window into the URL line of your browser.

Exiting the application After you finish running an application started in this manner, you must stop it explicitly by using the Cancel Run command. You do not need to close the browser.

Starting the Application in a Web Browser Automatically

For testing convenience, WebEnterprise Designer will automatically launch a browser window and issue the starting URL to your application. This feature is activated by defining a default browser.

Setting the Default Browser

You create a default browser setting in two ways:

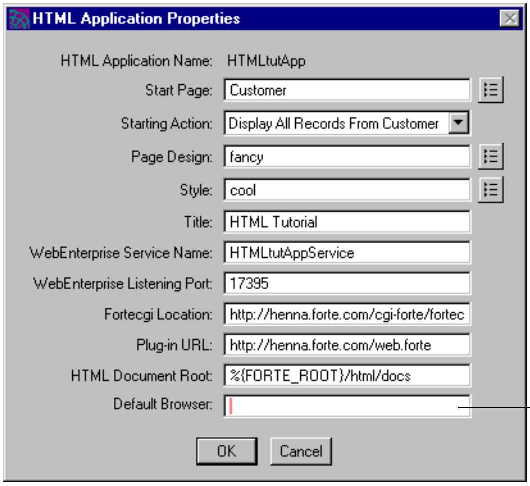
- type in the setting in the HTML application's properties dialog
- set the FORTE_DEFAULT_BROWSER environment variable

In either case, you use the path specification of your browser application as the default browser setting.

➤ To set the HTML application's default browser property

1. From the HTML Application Workshop, choose File > Properties.

   The HTML Application Properties dialog appears.



HTML Application Properties

HTML Application Name: HTMLtutApp

Start Page: Customer

Starting Action: Display All Records From Customer

Page Design: fancy

Style: cool

Title: HTML Tutorial

WebEnterprise Service Name: HTMLtutAppService

WebEnterprise Listening Port: 17395

Fortecgi Location: http://henna.forte.com/cgi-forte/fortec

Plug-in URL: http://henna.forte.com/web.forte

HTML Document Root: %{FORTE_ROOT}/html/docs

Default Browser:

Default browser field

2. Type in your browser application's path specification in the Default Browser setting.

NOTE On NT, it is likely the path to your browser contains an embedded space. You must use a "shortname" instead. Typically, the shortname version of your browser will be:

- For Netscape:

  `C:\PROGRA~1\Netscape\Communicator\Program\netscape.exe`

- For Microsoft Internet Explorer:

  `C:\PROGRA~1\Plus!\MICROS~1\IEXPLORE.EXE`

For information on shortnames, see the iPlanet UDS online Help topic "Use a shortname for the Default Browser field."

3. Click OK to apply your change and dismiss the HTML Application Properties dialog.

➤ **To set the FORTE_DEFAULT_BROWSER environment variable**

1. See *A Guide to the iPlanet UDS Workshops* for information on setting iPlanet UDS environment variables in general.

2. Set this variable using this syntax:

   FORTE_DEFAULT_BROWSER *file_specification*

   file_specification is your browser application's path specification, as above.

   For further information on setting the FORTE_DEFAULT_BROWSER variable, see the online Help topic for this variable.

NOTE On NT, the WebEnterprise Designer installer provides the opportunity to set the FORTE_DEFAULT_BROWSER environment variable at install time. For information, see the *WebEnterprise Installation Guide*.

Starting Your Application

When there is a value set for the default browser, you can start your application as follows.

➤ **To start an HTML application when there is a default browser value**

1. In the HTML Application Model Workshop, click the Run Application button.

   Alternatively, you can select HTMLtutAppHandlers in the Repository Workshop and click the Run Application button from there.

   WebEnterprise Designer always checks that the generated classes are up-to-date in both workshops before attempting to run an application. If you run an application before generating the classes (or if you make changes to either model and forget to regenerate), WebEnterprise Designer automatically generates the classes before running the application. Note that if you run the application from the Repository Workshop, you must explicitly generate the classes beforehand, if they are not up-to-date.

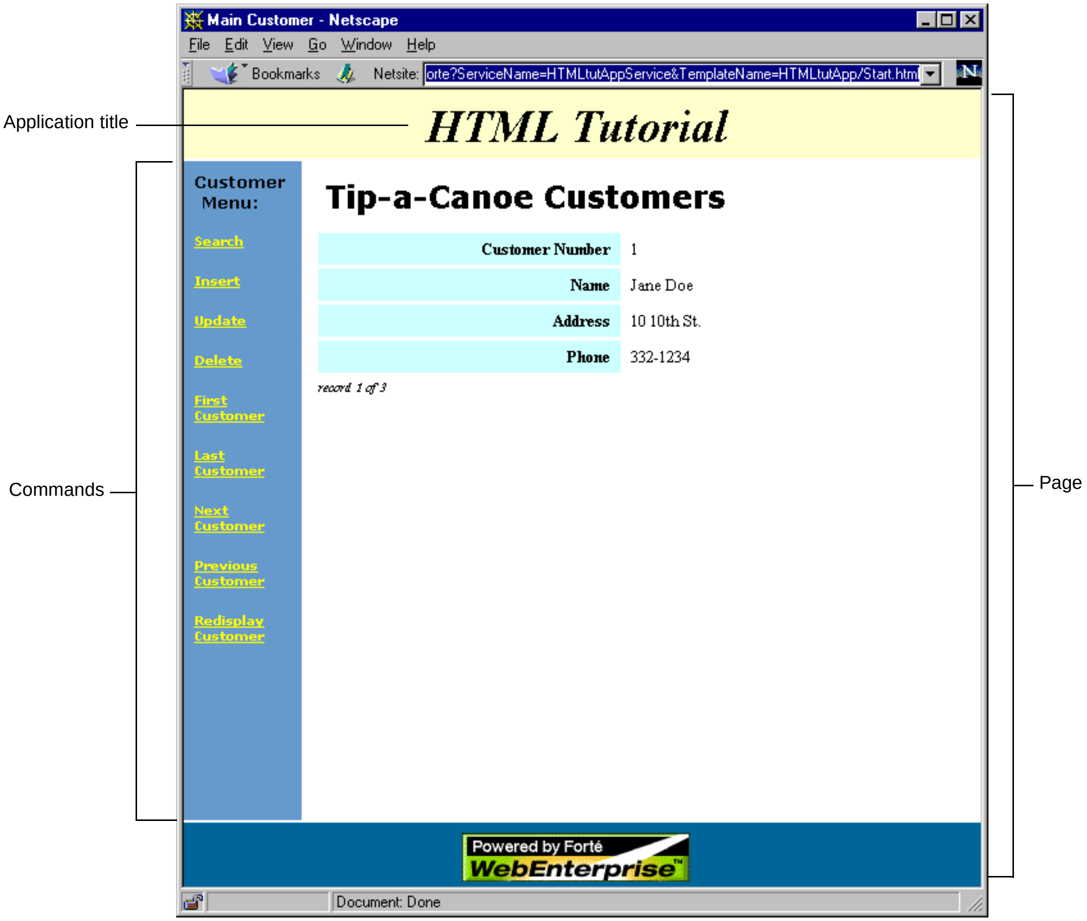
Exiting the application If you are running on Windows NT, the browser runs synchronously. When you exit the browser, the application test run quits as well.

If you are running on UNIX, the browser runs asynchronously. When you exit the browser, the application test run continues. You must stop it explicitly with the Plan > Run >Cancel Run command.

Testing an HTML Application

After you start the HTML application, the default Web browser is started and displays the Customer page, as shown in Figure 2-6.

Figure 2-6 Customer Page (Fancy Design)


Application title
Commands
Page
Main Customer - Netscape
File Edit View Go Window Help
Bookmarks
Netsite: orte?ServiceName=HTMLtutAppService&TemplateName=HTMLtutApp/Start.html
Document: Done

HTML Tutorial

Customer Menu:

- Search
- Insert
- Update
- Delete
- First Customer
- Last Customer
- Next Customer
- Previous Customer
- Redisplay Customer

Tip-a-Canoe Customers

Customer Number	1
Name	Jane Doe
Address	10 10th St.
Phone	332-1234

record 1 of 3

Powered by Forté WebEnterprise™

Scrolling through records To scroll through the records one by one, click `Next Customer` in the Commands column. Use the other commands to jump to first, last, and previous record.

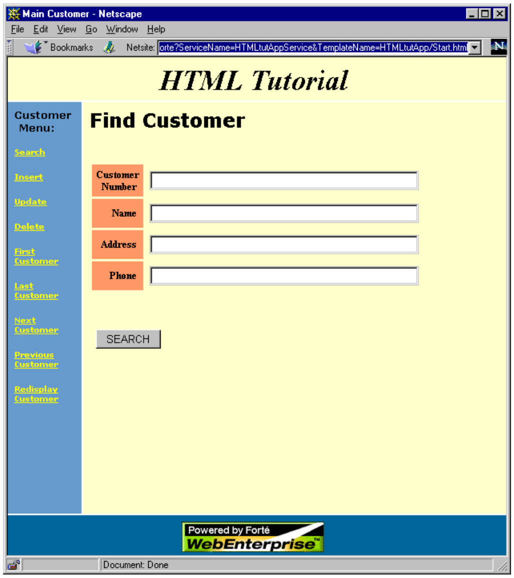
Searching Records

Wildcard character WebEnterprise Designer supports the use of the wildcard character (`%`), which allows end users to use partial search criteria. For example, if you type `J%` in the customer name field, WebEnterprise Designer returns all customers with names that begin with J. Conversely, using `%J` returns all customers whose names end with J. Any other wildcard mechanisms are database-dependent.

➤ To search for a record

1. In the HTML page displayed in your browser, click on the Search command.

   The Search mode of the Customer page appears:



2. Type 1 in the Customer Number field and click the Search button.

   The application retrieves all records with a Customer Number of 1 (there should be only one).

   If there is more than one, the first record appears on the form; the Current Record of Total Records indicator displays how many records were retrieved (in this case, the indicator reads, "record 1 of 1").

3. Using your browser's navigational tools, return to the Customer main page.

4. To conduct another search, click the `Search` command.

5. In the Search page, type `J%` in the Name field and click the Search button.

   The application retrieves all records with a Customer Name starting with J (there should be two).

6. Click Search again, and in the Search page that displays, leave the Number field blank and just click the Search button.

   The application retrieves all records in the Customer table (there are three), and displays the first record.

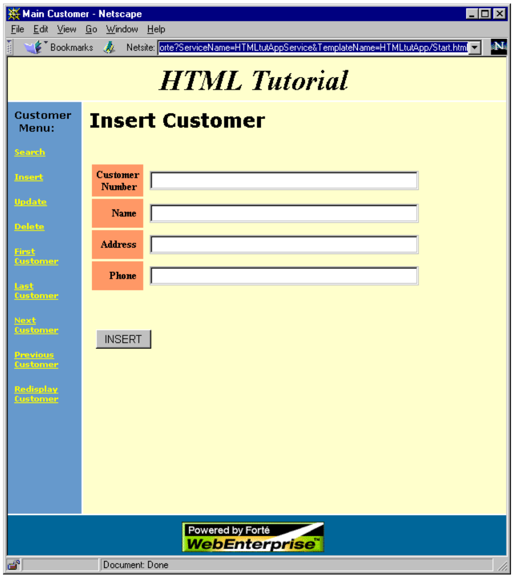
Inserting and Deleting Records

You can insert a new record into the Customer table using the Customer page.

➤ To insert a record

1. In the CUSTOMER Home page, click `Insert Customer` in the commands column.

   The Insert page appears:



2. Type an integer in the Customer Number field.

   If you enter a Customer Number already taken (1, 2, or 3), you will get an error.

3. Fill in a name, address, and phone number in the Name, Address, and Phone fields.

4. Click the Insert button.

   The new record has been saved to the database.

➤ **To retrieve the record you just entered**

1. Click the `Search` command.

2. Type the integer you entered above in the Customer Number field.

3. Click the Search button.

   The record you just entered appears in the window.

➤ **To delete the record you just entered**

1. In a page showing the record you just entered, click the `Delete` command.

   A confirmation message appears (“Delete this record?”).

2. Click OK.

   The record is deleted.

Exiting the Application

As with starting your application, the way you exit the application depends on how you have set up the default browser.

If the default browser property is set Exit your HTML application by exiting the browser.

If the default browser property is not set On Windows NT, the browser runs synchronously. When you exit the browser, the application test run quits, as well.

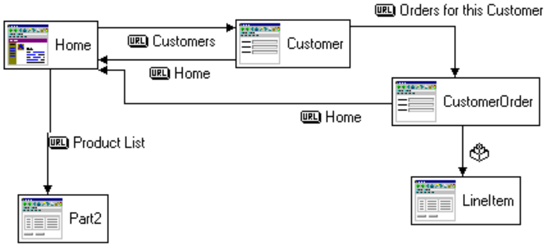
On UNIX, the browser runs asynchronously. When you exit the browser, the application test run continues. You must stop it explicitly with the Plan > Run > Cancel Run command. You can leave your browser running at the same URL, while you go back to iPlanet UDS and make changes to your model and then rerun the application. Then, just refresh the browser to see your changes (assuming you did not change the model’s starting page and action).

Finishing the Tutorial HTML Application

Now that you have learned the basics for creating, generating, and running a page, you will create a more complex application.

The application you create in this section includes a home page with links to pages for viewing, modifying, or adding to the customer list, the customer order list, and the product list. The final model will look like this:

Figure 2-7 The HTMLtutApp Model



When you start the application, the Home page displays, showing links to the Customer page and the Product List page. Activating the Product List link displays a list of products in a separate browser window. The user can then use this as a reference for product data while using the main browser window to add items to customers' orders.

To browse through customer data, the user activates the Customers link on the Home page. To view a particular customer's orders, the user clicks the Orders for this Customer link on the Customer page. This action displays a master-detail page of orders and line items for the selected customer.

To add an item to an order, the user activates the Insert command in the Line Item section of the master-detail page, enters the required data in the Insert page that appears (using the Product List page for reference), and then clicks the Add button. This redisplays the master-detail page with the added order.

This section shows you how to perform the following tasks:

- "Creating a Link Page" on page 77
- "Setting the Application's Start Page" on page 78
- "Creating a Command Link" on page 79
- "Adding a Reference Page" on page 84

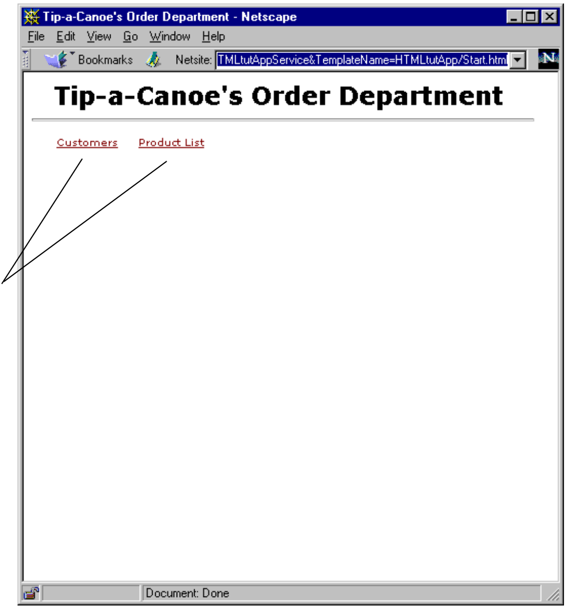
- “Creating a Page that Displays Graphics” on page 85
- “Creating a Separate Browser Link” on page 90
- “Creating a Master-Detail Page” on page 93
- “Running the HTMLtutApp Application” on page 101

Creating a Link Page

The home page will be a special page called a “link page,” and is not based on a business class. This page will be an extremely simple page, displaying hypertext links to several other pages.

The link page you will create will look like this at runtime:

Figure 2-8 Home (Link) Page



Netsite: TMLtutAppService&TemplateName=HTMLtutApp/Start.html

Tip-a-Canoe's Order Department

Customers Product List

Document: Done

Links to other pages

➤ To create a Link page

1. If the HTML Application Model Workshop is not already open, open it from the Repository Workshop by double-clicking on the HTMLtutApp application.
2. Choose the New > HTML Page command.
3. Click anywhere in the workshop's drawing area.

   The Page Wizard appears. For complete information on the Page Wizard, see iPlanet UDS online Help.
4. Select Link Page from the list in the Page Wizard and click Next.

   The Page Wizard displays the only options you can set on a link page: HTML Page Name and Link Page Title. For a description of these fields, press your Help key.
5. Type Home for the HTML page name and Tip-a-Canoe's Order Department for the Link page title (or whatever you want).
6. Click Finish.

You have created a link page. Since it is to be the starting page of your application, you must specify that in the HTML Application Properties dialog.

Setting the Application's Start Page

When your application consisted of one class page, you had no choice but to start with that page. You could, however, set the starting action of that page (refer to "Setting the Starting Action of the Application" on page 63). The HTMLtutApp application will start on a link page (the Home page). This page is not based on a business class and has no records associated with it. There is no starting action for such a page. You just have to tell iPlanet UDS that this is the page to start on.

NOTE If you do not set a start page, the default is the first page you created. If you delete the start page, no other page is set by default. When you try to generate, you get a message to specify the start page.

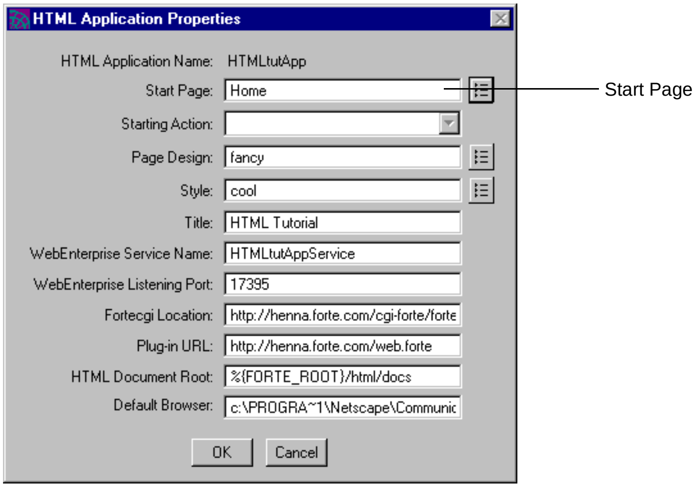
➤ To set an HTML application’s start page

1. In the HTML Application Model Workshop, choose File > Properties.

   The HTML Application Properties dialog appears.

2. Use the browse button to locate the Home page and double-click it.

   The HTML Application Properties dialog reappears, showing the new start page.



HTML Application Properties

HTML Application Name:	HTMLtutApp
Start Page:	Home
Starting Action:	
Page Design:	fancy
Style:	cool
Title:	HTML Tutorial
WebEnterprise Service Name:	HTMLtutAppService
WebEnterprise Listening Port:	17395
Fortecgi Location:	http://henna.forte.com/cgi-forte/forte
Plug-in URL:	http://henna.forte.com/web.forte
HTML Document Root:	%{FORTE_ROOT}/html/docs
Default Browser:	c:\PROGRA~1\Netscape\Communic

Start Page

   Notice that the starting action drop list is now deactivated.

3. Click OK.

You are finished with the Home page. Now you will create a link between the Home page and the Customer page.

Creating a Command Link

A *link* is the way one page calls another page. Creating a link is similar to creating an association in the business model: you connect two pages and define the connection’s properties. You specify link properties by means of the Link Properties dialog, which you open by double-clicking on the link (you can also select the link and choose the Component > Properties... command). This tutorial briefly covers link properties. Refer to iPlanet UDS online Help for more information on links and link dialogs.

WebEnterprise Designer offers two types of links: command links and nested links. In this section you will create two command links: one from Home to Customer and one from Customer to Home.

Linking the Home Page to the Customer Page

You will link the Home page to the Customer page.

➤ **To create a command link from the Home page to a target page**

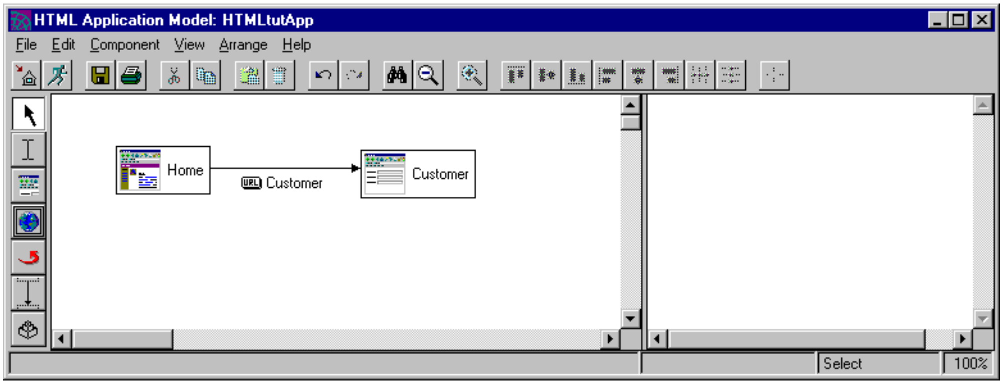
1. Click on the New Link button.

   Alternatively, you can choose Component > New > Link, or right-click in the drawing area and choose New Link from the popup menu.

2. Drag the cursor from the Home page to the Customer page.

   The direction of the link is important; start at the From page and drag to the To page.

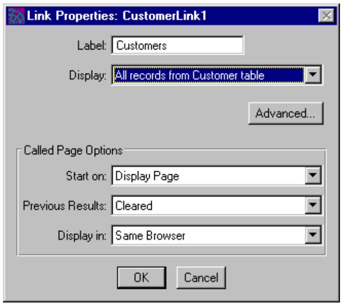
   A line appears with a command link icon, as shown below.



3. Double-click on the command link to open the Link Properties dialog.
4. Click the Advanced button to open the advanced part of the dialog.
5. Type Customers in the Label field.

6. Select All records from Customer table in the Display drop list.

   The Link Properties dialog should look like this:



Link Properties: CustomerLink1

Label: Customers

Display: All records from Customer table

Called Page Options

Start on: Display Page

Previous Results: Cleared

Display in: Same Browser

   For information on the properties of this type of command link, press your Help key.

7. Click OK.

You are finished with the Customer link. Now you will create a link from the Customer page to the Home page.

Linking the Customer Page to the Home Page

From the Customer page, a user can return to the Home page using the Back button. However, it is more elegant to have a specific button to return to the Home page. When the application is finished and navigation is more complicated, a well-designed application has a direct route to the Home page from all major pages.

➤ **To create a Home page link**

1. Click on the New Link button.

2. Drag the cursor from the Customer page to the Home page.

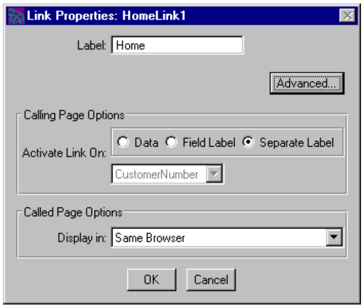
3. Double-click on the new Home link to display the Link Properties dialog.

4. Click the Advanced button to open the advanced part of the dialog.

5. Leave the text Home in the Label field.

6. If not already enabled, click Separate Label for the Activate Link On property.

   The Link Properties dialog should look like this:



Link Properties: HomeLink1

Label: Home

Calling Page Options

Activate Link On: ○ Data ○ Field Label ● Separate Label

CustomerNumber

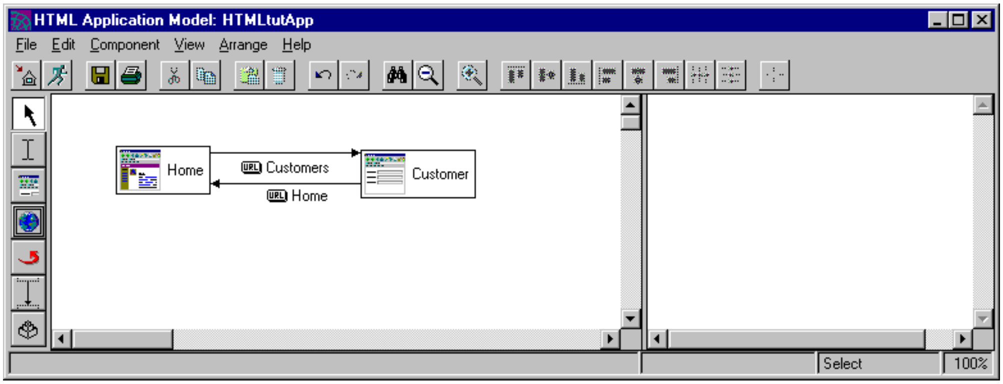
Called Page Options

Display in: Same Browser

   For information on the properties of this dialog, press your Help key.

7. Click OK.

   Your model should look like this:



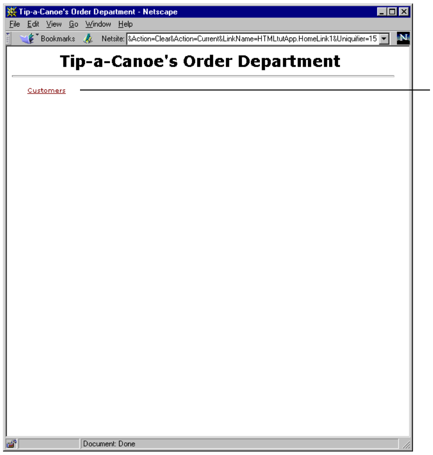
Testing Your Application

To test your work so far, run your application as you did the single-page application (see “Running an HTML Application” on page 68.

The Web browser displays the Home page, as shown in Figure 2-8 on page 77, except only the Customers link is on the page.

➤ To test the HTMLtutApp application

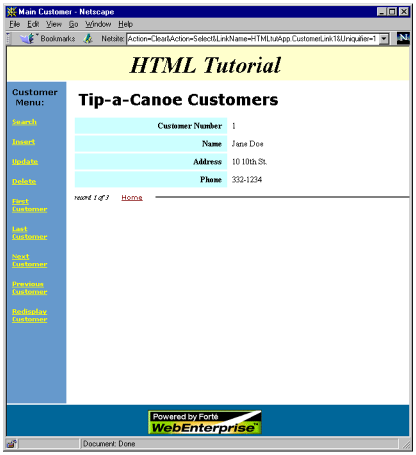
1. Start the application. The Home page is displayed:


Tip-a-Canoe's Order Department - Netscape
File Edit View Go Window Help
Bookmarks
Netsite:
Tip-a-Canoe's Order Department
Customers
Document: Done

&Action=Clear&Action=Current&LinkName=HTMLtutApp.HomeLink1&Uniquifier=15

`Customer` link

2. Click on the `Customers` link. The default Customer page is displayed.



Action=Clear&Action=Select&LinkName=HTMLtutApp.CustomerLink1&Uniquifier=1

Customer Number	1
Name	Jane Doe
Address	10 10th St.
Phone	332-1234

record 1 of 3

`Home` link

3. On the Customer page, click on the `Home` link. The Home page is redisplayed.

4. Exit the application as described in "Exiting the Application" on page 75.

Now you will create some pages to link to.

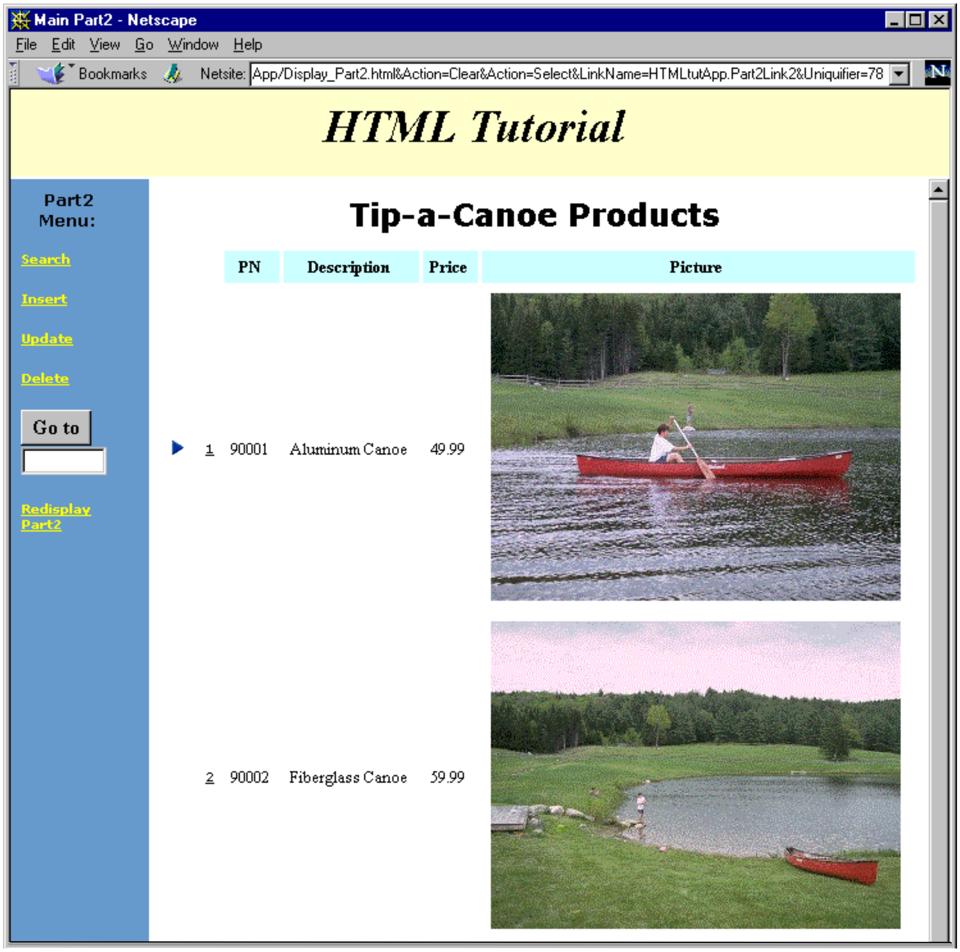
Adding a Reference Page

Now you will create another target page, only this time, you will have it display in a separate browser window. This will be a reference page showing all the available products. When checking orders or adding new ones, the Order Entry people can refer to information on this page while navigating around the rest of the application. In this section, you will create this page.

In addition, you will learn how to create a page that displays graphics.

The page you create in this section will look like Figure 2-9 at runtime.

Figure 2-9 Page Displaying Graphics



App/Display_Part2.html&Action=Clear&Action=Select&LinkName=HTMLtutApp.Part2Link2&Uniquifier=78

HTML Tutorial

Tip-a-Canoe Products

	PN	Description	Price	Picture
1	90001	Aluminum Canoe	49.99	
2	90002	Fiberglass Canoe	59.99	

Creating a List Page

You will create the product list page as a list page, so that you can display multiple records on it.

➤ **To create a list page based on the Part2 business class**

1. In the HTML Application Workshop, click the New HTML Page button.

2. Click anywhere in the drawing area.

3. Choose List Page and click Next.

4. On the Page Options page, base the page on the `Part2` business class.

5. Also on the Page Options page, type Tip-a-Canoe Products for the page title.

6. Adjust Visible Rows to 2.

   The graphics are large, so this will display only two rows at a time.

7. On the Field Properties page, type in the Label fields:

   ```
   PN
   Description
   Price
   Picture
   ```

8. On the Command Options page, leave all commands enabled.

9. On the same page, add the word “`Product`” to the labels of the First, Last, Next, and Previous commands (for example, “First Product,” “Last Product”).

10. Click Finish.

You have finished creating the target page. Now you will set the page to display graphics.

Creating a Page that Displays Graphics

One of the attributes of the `Part2` business class is gifAddress. Data for this field are file system addresses for graphic files that can be displayed on an HTML page. Using mechanisms in the business model and the HTML application model, you can display the file indicated by the data as a graphic on your HTML page.

To display graphics on a page, you need:

- graphic files in a format displayable in a browser (such as .gif or .jpg format)
- a directory for storing graphic files that is accessible to your Web server and application environment
- a column in a table in the business model's database schema for specifying the directory and file name of a graphic file
- the Is URL property set on the corresponding graphic-directory attribute in the business model
- the Display As Image property set on the corresponding graphic-directory attribute in the HTML application model

Graphic Files

The *Tip-a-Canoe and Paddles, Too* company sells four products: two types of canoes and two types of paddles. Graphics for these four products are stored in the following files:

Product	**Graphic File Name**
Aluminum Canoe	canoea.gif
Fiberglass Canoe	canoef.gif
5-Foot Oar	paddle1.gif
6-Foot Oar	paddle2.gif

The graphic files are stored in the `examples/wedtut/images` subdirectory of the `FORTE_ROOT` subdirectory under the Web server root directory. Refer to "Important Files on the Web Server Node" on page 136.

The Underlying Table

The Part2 table consists of four attributes, one of which specifies graphic file locations:

- PartNumber—the item's part number, and the primary key for the table
- Description—the name of the product
- Price—the product's price

- gifAddress—the file system address of the product’s graphic

  The file specification for this attribute must be given relative to the Web server root, which is where your Web server is installed. For example, if your Web server is installed in `d:\Netscape\Suitespot`, then the Web server portion of WebEnterprise Designer is installed in the `d:\Netscape\Suitespot\forte` directory. Note in the script below how data for the gifAddress attribute is specified in the insert statements.

Part2 was created with this script:

```
create table Part2 (
  PartNumber    int,
  Description   varchar(25) null,
  Price         money null,
  gifAddress    varchar(80) null,
  PRIMARY KEY (
    PartNumber
  )
)
go
insert into Part2 values (90001, ‘Aluminum Canoe’, 49.99,
  ‘/forte/examples/wedtut/images/canoea.gif’);
insert into Part2 values (90002, ‘Fiberglass Canoe’, 59.99,
  ‘/forte/examples/wedtut/images/canoef.gif’);
insert into Part2 values (90003, ‘5 Foot Oar’, 9.99,
  ‘/forte/examples/wedtut/images/paddle1.gif’);
insert into Part2 values (90004, ‘6 Foot Oar’, 12.99,
  ‘/forte/examples/wedtut/images/paddle2.gif’);
go
```

The Is URL Attribute Property—Business Model

To enable the gifAddress field of the Part2 page to display a graphic, the Is URL property of the corresponding business class attribute must be enabled. You enable this property in the business model.

➤ **To enable the Is URL property on a business class attribute**

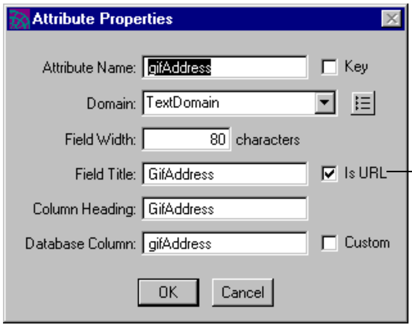
1. In the Repository Workshop, double-click on the HTMLtutorial business model.

   The Business Model Workshop appears, displaying the HTMLtutorial business model.

2. On the `Part2` class, double-click on the gifAddress attribute.

   The Attribute Properties dialog appears.

3. Enable the Is URL property, as shown:



Attribute Name: gifAddress ☐ Key

Domain: TextDomain

Field Width: 80 characters

Field Title: GifAddress ☑ Is URL

Column Heading: GifAddress

Database Column: gifAddress ☐ Custom

Enable this field for graphic display

For information on the properties of this dialog, press your Help key.

4. Click OK to apply your change and close the Attribute Properties dialog.
5. (Optional) Click the Generate Server Code button.

You can close the Business Model Workshop, if you wish.

The Display As Image Attribute Property—HTML Model

Fields based on attributes that have the Is URL property set can be displayed on a page in your HTML application in two ways:

- as an image
- as text—use this if you want to be able to update the image file's location at runtime

You might want the address text displayed if you want to be able to edit it at runtime, but in this tutorial, you want the image displayed, not the text of the attribute. To specify that the image be displayed, you will adjust a property of the gifAddress field, as follows:

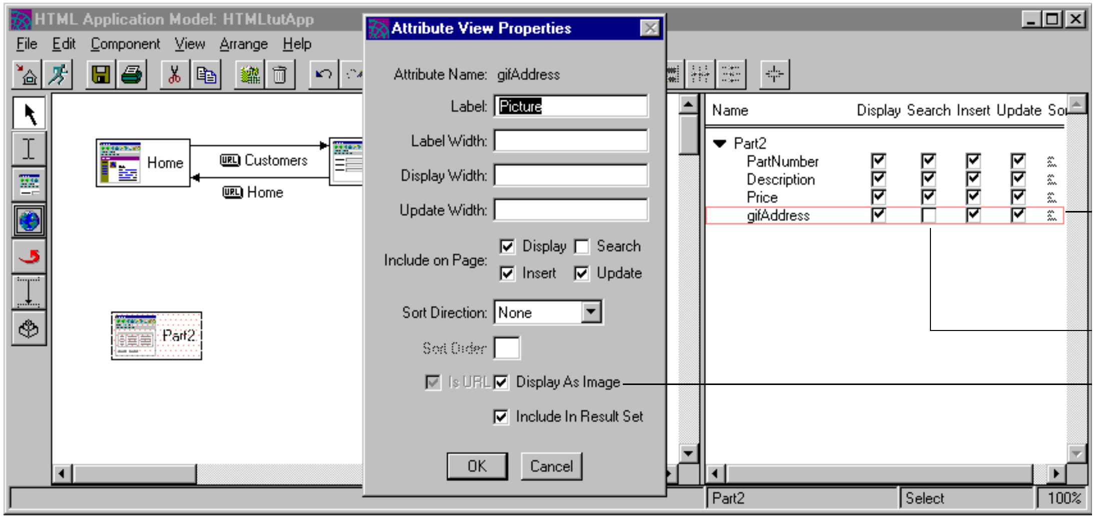
➤ To set a field to display a graphic

1. In the HTML Application Model Workshop for HTMLtutApp, select the Part2 page.

   Part2 is displayed in the attribute list (the right pane of the model).

2. Double-click on the gifAddress field.

   The Attribute View Properties dialog appears, as shown below:



Attribute View Properties

Attribute Name: gifAddress

Label: Picture

Label Width:

Display Width:

Update Width:

Include on Page: ☑ Display ☐ Search ☑ Insert ☑ Update

Sort Direction: None

Sort Order:

☑ Is URL ☑ Display As Image

☑ Include In Result Set

OK Cancel

Name	Display	Search	Insert	Update	So
Part2					
PartNumber	☑	☑	☑	☑	
Description	☑	☑	☑	☑	
Price	☑	☑	☑	☑	
gifAddress	☑	☐	☑	☑	

HTML Application Model: HTMLtutApp
File Edit Component View Arrange Help
Home
Customers
Home
Part2
Part2
Select
100%

3. Enable the Display As Image field.

4. Click OK to apply your changes and close the Attribute View Properties dialog.

5. In the attribute list, disable the Search page mode option for the gifAddress field.

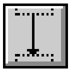
   This specifies that the gifAddress field will not display on a Search page. This is sensible, because there is no way to search an image.

Creating a Separate Browser Link

You will now create a link to this page so that it displays in a separate browser. This is appropriate for a reference page.

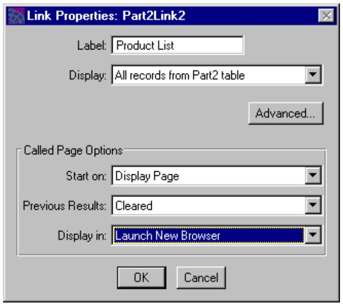
➤ To create a command link to a separate browser

1. Click on the New Link button.



2. Drag the cursor from the Home page to the new Part2 page.
3. Double-click on the new Part2 link to display the Link Properties dialog.
4. Type Product List in the Label field.
5. Select All records from Part2 table from the Display drop list.
6. Click the Advanced button to open the advanced part of the dialog.
7. Select Launch New Browser in the “Display in” field.

   The Link Properties dialog should look like this:



Link Properties: Part2Link2

Label: Product List

Display: All records from Part2 table

Called Page Options

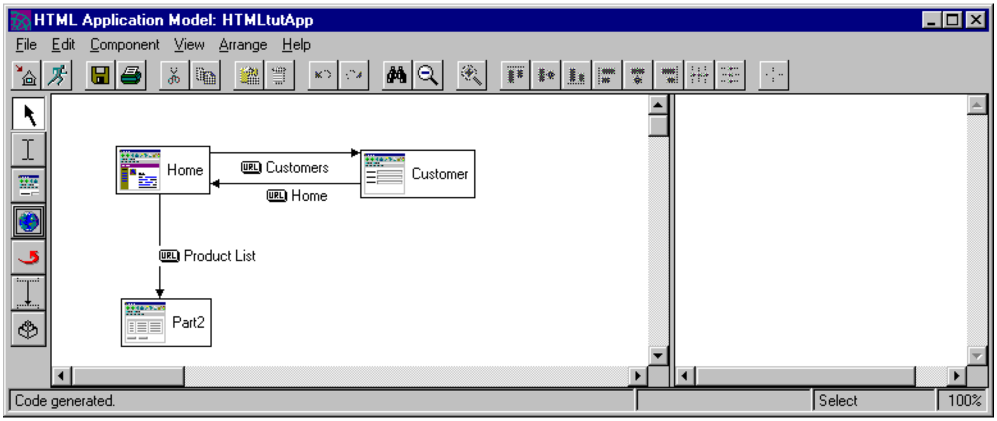
Start on: Display Page

Previous Results: Cleared

Display in: Launch New Browser

8. Click OK.

   Your model should look like this:



HTML Application Model: HTMLtutApp
File Edit Component View Arrange Help
Home
Customers
Customer
Home
Product List
Part2
Code generated.
Select
100%

You will not create a Home link for this page, because this page in a separate browser is for reference purposes. The user need only navigate to Home in the main browser window.

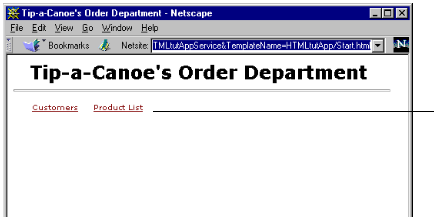
Testing Your Application

To test your work, do as you did in "Testing Your Application" on page 82.

➤ To test the HTMLtutApp application

1. Start the application.

   The Home page is displayed:


Tip-a-Canoe's Order Department - Netscape
File Edit View Go Window Help
Bookmarks
Netsite:

TMLtutAppService&TemplateName=HTMLtutApp/Start.html

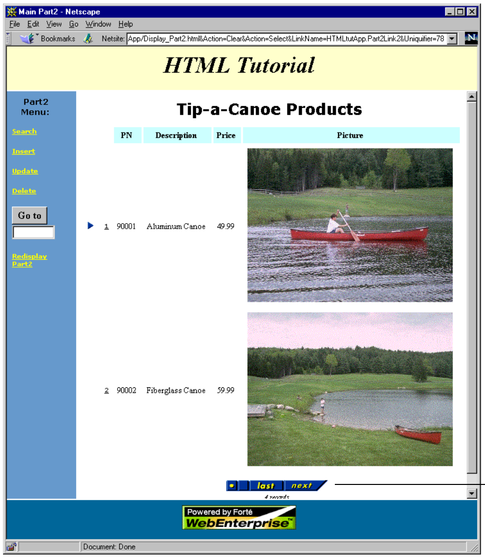
Tip-a-Canoe's Order Department

Customers Product List

`Product List` link

2. Click on the `Product List` link.

   A separate browser window opens, displaying the Part2 page:



App/Display_Part2.html&Action=Clear&Action=Select&LinkName=HTMLtutApp.Part2Link2&Uniquifier=78

HTML Tutorial

Tip-a-Canoe Products

		PN	Description	Price	Picture
▶	1	90001	Aluminum Canoe	49.99	
	2	90002	Fiberglass Canoe	59.99	

4 records

Document: Done

Navigation buttons for a list page

3. Use the navigation buttons at the bottom of the page to view the other two products.

Exit the application as described in "Exiting the Application" on page 75.

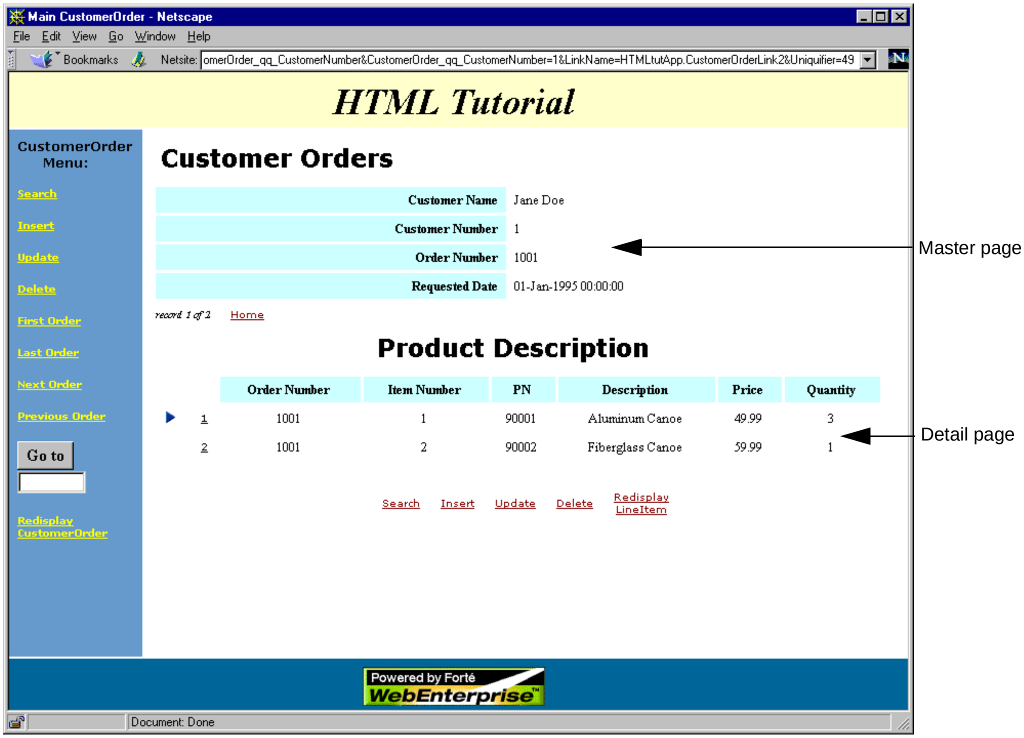
Now you will create a master-detail page to view and edit the customer orders.

Creating a Master-Detail Page

For the next part of the tutorial, you will create a master-detail page of customer orders and their corresponding line items. To do this, you will make a page based on the LineItem business class and link it with the Customer Order page with a nested link.

The master-detail page you will create will look like this:

Figure 2-10 Master-Detail Page



omerOrder_qq_CustomerNumber&CustomerOrder_qq_CustomerNumber=1&LinkName=HTMLtutApp.CustomerOrderLink2&Uniquifier=49

HTML Tutorial

CustomerOrder Menu:

Customer Orders

Customer Name	Jane Doe
Customer Number	1
Order Number	1001
Requested Date	01-Jan-1995 00:00:00

record 1 of 2 Home

Master page

Product Description

		Order Number	Item Number	PN	Description	Price	Quantity
▶	1	1001	1	90001	Aluminum Canoe	49.99	3
	2	1001	2	90002	Fiberglass Canoe	59.99	1

Detail page

Document: Done

In the next few sections, you will create a form page for the master page and a list page for the detail page.

Creating the Master Page

In this section, you will create the master page, which is based on the `CustomerOrder` business class.

➤ **To create a form page based on the CustomerOrder business class**

1. Create a form page as before.

   For information about creating a form page, see “Defining a Form Page” on page 58.

2. On the Page Options page, base the page on the `CustomerOrder` class.
3. Change the default page title to Customer Orders.
4. Click Next to bring up the Fields page.

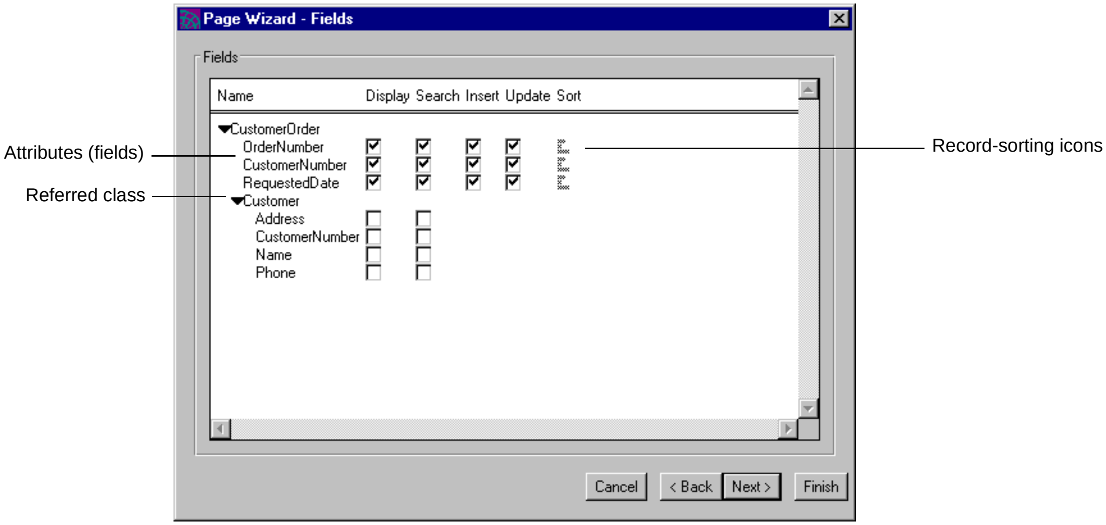
You will now modify some of the field properties.

Changing the Order and Accessibility of Fields

Notice that not only the CustomerOrder fields are displayed in the list, but also the Customer class fields. This is because Customer is referenced by `CustomerOrder`. Click on the triangle to the left of Customer to display its fields. Notice that the fields of the referred class are not displayed by default. If you want to display them, check the appropriate boxes for the page modes you want them displayed on. They cannot display on Insert or Update modes.

For information about the Fields page, press your Help key.

Figure 2-11 Page Wizard Displaying Fields Page



Name	Display	Search	Insert	Update	Sort
▾CustomerOrder					
OrderNumber	☑	☑	☑	☑	⇅
CustomerNumber	☑	☑	☑	☑	⇅
RequestedDate	☑	☑	☑	☑	⇅
▾Customer					
Address	□	□			
CustomerNumber	□	□			
Name	□	□			
Phone	□	□			

For this tutorial, you will move RequestedDate to follow OrderNumber and also make the Customer.Name attribute visible and read-only. You then rename all the visible fields.

➤ To modify field properties

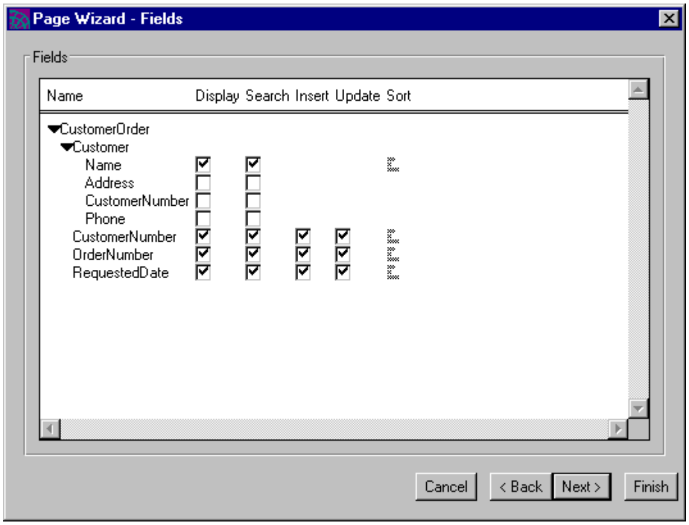
1. Using drag and drop, arrange the CustomerOrder class fields in this order:

   CustomerNumber
   OrderNumber
   RequestedDate

2. Click on the Customer arrow to display the fields of this referred class page.
3. To the right of the Customer.Name field, enable both the Display and Search toggles.

4. Click on the Customer page label and drag it above OrderNumber.

   The Fields page should look like this:



Name	Display	Search	Insert	Update	Sort
▼CustomerOrder					
▼Customer					
Name	☑	☑			
Address	☐	☐			
CustomerNumber	☐	☐			
Phone	☐	☐			
CustomerNumber	☑	☑	☑	☑	
OrderNumber	☑	☑	☑	☑	
RequestedDate	☑	☑	☑	☑	

   Next, you will refine the labels of the fields that will appear on the page.

5. Click Next to go to the Field Properties page of the Page Wizard.

6. Replace the default labels (the attribute names) with more readable names, such as:

   Customer Name
   Customer Number
   Order Number
   Requested Date

7. Click Next twice to get to the Command Options page.

8. Add “`Order`” to the labels of the First, Last, Next, and Previous commands (for example, “First Order,” “Last Order”).

9. Click Finish.

You are finished with the master page and will now create the detail page.

Creating the Detail Page

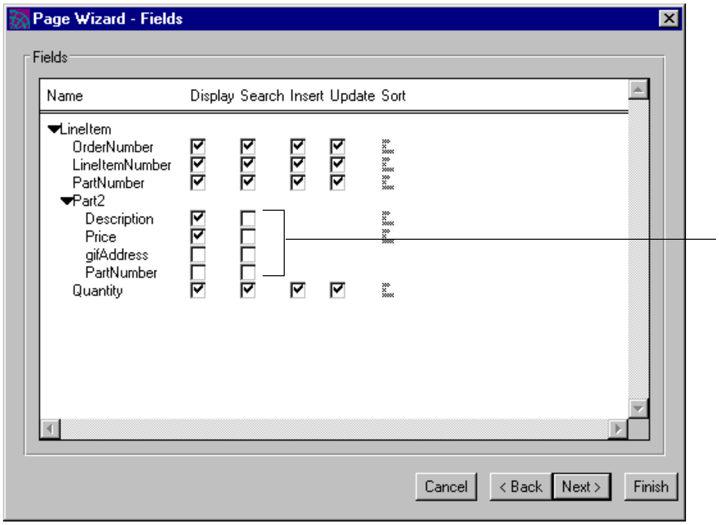
In this section, you will create the detail page, which is based on the LineItem business class. You expect a customer order to contain more than one record, so you will create this as a list page, to display multiple records.

➤ To create the List page for the Master-Detail page

1. Create a list page as described in "Creating a List Page" on page 85.
2. Base this page on the `LineItem` business class.
3. On the Page Options page, make the page title Product Description.
4. Click Next to bring up the Fields page.

   Now you will display the Description and gifAddress fields from the Part2 business class on the Line Item page.
5. On the Fields page, open the Part2 page to display its fields.
6. Click in the boxes in the Display column to enable them for the Description and Price fields.
7. Drag Part2 under the PartNumber field.

   The Fields page should look like this:



Name	Display	Search	Insert	Update	Sort
▼LineItem					
OrderNumber	☑	☑	☑	☑	
LineItemNumber	☑	☑	☑	☑	
PartNumber	☑	☑	☑	☑	
▼Part2					
Description	☑	□			
Price	☑	□			
gifAddress	□	□			
PartNumber	□	□			
Quantity	☑	☑	☑	☑	

Referenced attributes

8. Click Next.

9. On the Field Properties page, edit the field labels as follows:

   ```
   Order Number
   Item Number
   PN
   Description
   Price
   Quantity
   ```

10. On the Command Options page, delete all commands except Search, Insert, Update, Delete, and Redisplay.

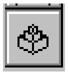
11. Click Finish.

Creating a Nested Link

You use a nested link to link a Master page to the Detail page.

➤ **To create a nested link**

1. Click the New Nested Connection button.

   Alternatively, you can choose Component > New > Nested Connection.

2. Drag the cursor from the Customer Order page to the LineItem page.

   As with command links, the direction of a nested link is important: start at the Master page and drag to the Detail page.

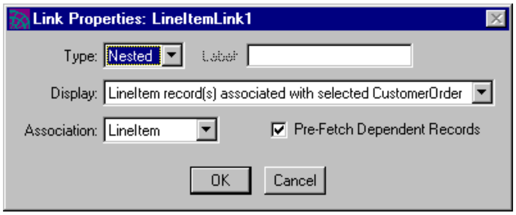
3. Double-click on the new nested link to display the Link Properties dialog.

4. If not already selected, select Nested from the Type field’s drop list.

5. Select LineItem record(s) associated with selected CustomerOrder in the Display field.

6. If not already enabled, enable the Pre-Fetch Dependent Records field.

   The Link Properties dialog should look like this:



Link Properties: LineItemLink1

Type: Nested

Label:

Display: LineItem record(s) associated with selected CustomerOrder

Association: LineItem

☑ Pre-Fetch Dependent Records

7. Click OK.

You have finished creating your master-detail page for customer orders and line items. Now you will link this page to the Customer page and the Home page.

Linking the Master-Detail Pages to the Application

In this section, you will create a command link from the Customer page to the CustomerOrder page, and from the CustomerOrder page to the Home page.

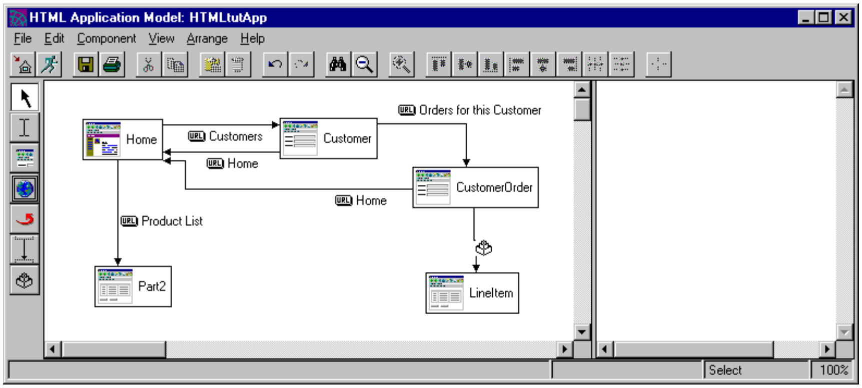
➤ To link the Customer page to the application

1. Create a link from Customer to CustomerOrder.
2. Double-click on the new CustomerOrder link to display the Link Properties dialog.
3. Change the label to Orders for this Customer.
4. Select CustomerOrder record(s) associated with selected Customer from the for Display drop list.

   When you created the Orders link in the business model (see Step 5 on page 43), you set up conditions to accomplish this step. Without the Orders link, the only choices for the Display field would be either No Records or All Records.
5. Click the Advanced button to open the advanced part of the dialog.
6. If not selected, click Separate Label for Activate Link On, and click OK.

7. Create a link from CustomerOrder to the Home page, as you did in "Linking the Customer Page to the Home Page" on page 81.

   Your model should look like the following figure.



Congratulations! You have finished your first multi-page HTML application model.

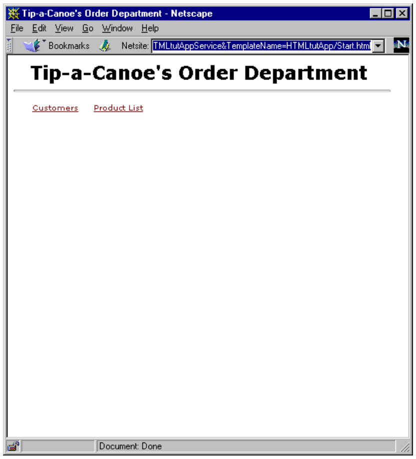
Running the HTMLtutApp Application

You are now ready to run your completed HTML application.

➤ To test run the HTMLtutApp application

1. Run the HTMLtutApp application as you did in "Testing Your Application" on page 82.

   The Web browser displays the Home page, as shown below.


Tip-a-Canoe's Order Department - Netscape
File Edit View Go Window Help
Bookmarks
Netsite:
Document: Done

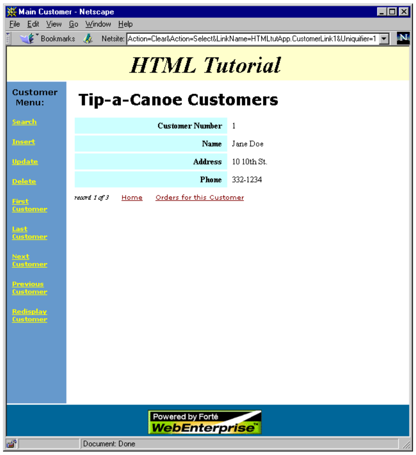
Netsite: TMLtutAppService&TemplateName=HTMLtutApp/Start.html

Tip-a-Canoe's Order Department

Customers   Product List

2. Click on the `Customers` link.

   The default Customer page is displayed.



Netsite: Action=Clear&Action=Select&LinkName=HTMLtutApp.CustomerLink1&Uniquifier=1

HTML Tutorial

Customer Menu:

Search
Insert
Update
Delete
First Customer
Last Customer
Next Customer
Previous Customer
Redisplay Customer

Tip-a-Canoe Customers

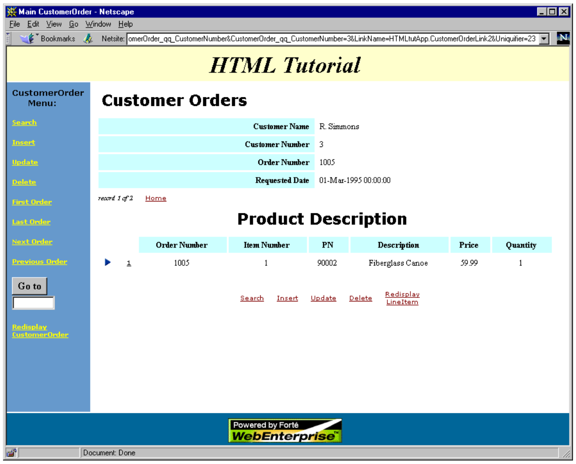
Customer Number	1
Name	Jane Doe
Address	10 10th St.
Phone	332-1234

record 1 of 3   Home   Orders for this Customer

3. In the command column, click `Next Customer` until you reach the record for the customer R. Simmons.

4. Under the customer information section, click the `Orders for this Customer` link.

   The records for R. Simmons are displayed, showing the Order 1005.


Main CustomerOrder - Netscape
File Edit View Go Window Help
Bookmarks
Netsite: omerOrder_qq_CustomerNumber&CustomerOrder_qq_CustomerNumber=3&LinkName=HTMLtutApp.CustomerOrderLink2&Uniquifier=23
HTML Tutorial
CustomerOrder Menu:
Search
Insert
Update
Delete
First Order
Last Order
Next Order
Previous Order
Go to
Redisplay CustomerOrder
Powered by Forté WebEnterprise™
Document: Done

Customer Orders

Customer Name	R. Simmons
Customer Number	3
Order Number	1005
Requested Date	01-Mar-1995 00:00:00

record 1 of 2 Home

Product Description

		Order Number	Item Number	PN	Description	Price	Quantity
▶	1	1005	1	90002	Fiberglass Canoe	59.99	1

Search Insert Update Delete Redisplay LineItem

5. Click `Next Order` to view Order 1006.

   There are two line items for order 1006. View them by scrolling down.

   To summarize, R. Simmons has ordered the following items:

Order No.	Item No.	PN	Description	Price	Quantity
1005	1	90002	Fiberglass Canoe	59.99	1
1006	1	90004	6 Foot Oar	59.99	2
	2	90001	Aluminum Canoe	59.99	1

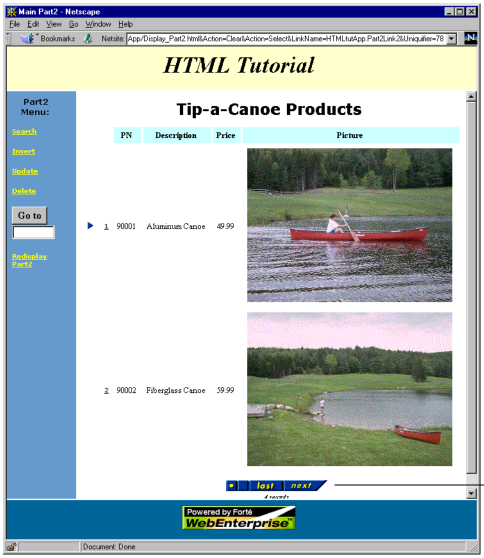
Now you will give R. Simmons an oar to go with his fiberglass canoe. First, you will display the product list in a separate browser for reference.

6. Click `Home` to get to the Home link page.

7. On the Home page, click `Product List`.

   The Product List page appears in a separate browser.

8. Scroll to the bottom of the page until you can see a bar to access the next set of records.



Main Part2 - Netscape
File Edit View Go Window Help
Bookmarks
Netsite: App/Display_Part2.html&Action=Clear&Action=Select&LinkName=HTMLtutApp.Part2Link2&Uniquifier=78
HTML Tutorial
Part2 Menu:
Search
Insert
Update
Delete
Go to
Redisplay Part2
last
next
4 records
Powered by Forté WebEnterprise™
Document: Done

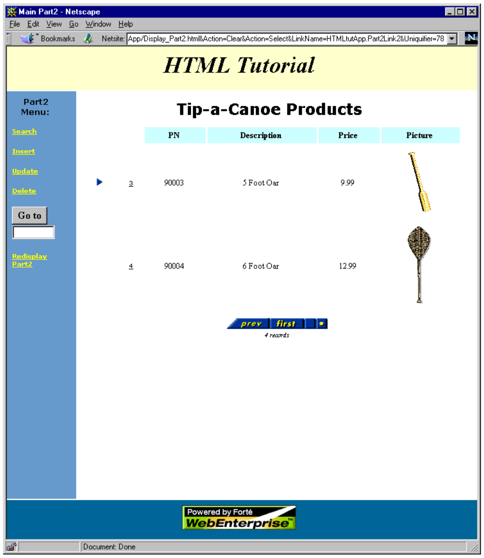
Tip-a-Canoe Products

		PN	Description	Price	Picture
▶	1	90001	Aluminum Canoe	49.99	
	2	90002	Fiberglass Canoe	59.99	

Access to next set of records

9. Click Next on the access bar.

A page with oars is displayed:



Main Part2 - Netscape
File Edit View Go Window Help
Bookmarks
Netsite: App/Display_Part2.html&Action=Clear&Action=Select&LinkName=HTMLtutApp.Part2Link2&Uniquifier=78
HTML Tutorial
Part2 Menu:
Search
Insert
Update
Delete
Go to
Redisplay Part2
prev
first
4 records
Powered by Forté WebEnterprise™
Document: Done

Tip-a-Canoe Products

		PN	Description	Price	Picture
▶	3	90003	5 Foot Oar	9.99	
	4	90004	6 Foot Oar	12.99	

You will add the 5-foot oar to order 1005.

10. In the application browser window, repeat Steps 2-4 to get to R. Simmons' order page.

11. On the Customer Orders page for R. Simmons (displaying Order Number 1005), click the `Insert` command under the Product Description detail.

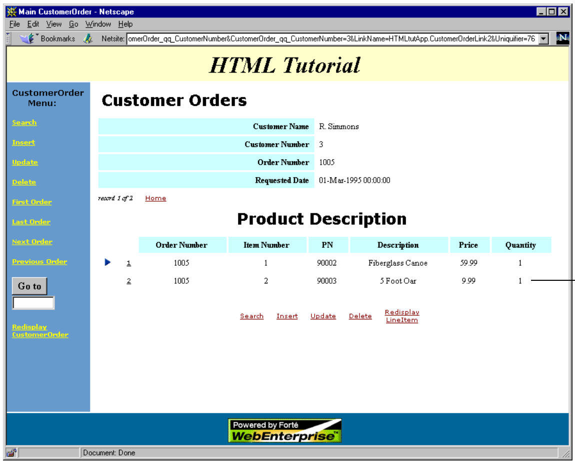
12. Type in the following:

    Order Number: `1005`
    Item Number: `2`
    PN: `90003`
    Quantity: `1`

13. Click the Insert button.

    This returns you to the Customer Orders page, where you can see your new order added.

    Note that the referenced fields do not display. Reload the page to see that there are now two items (including the 5-foot oar) on order 1005.


Main CustomerOrder - Netscape
File Edit View Go Window Help
Bookmarks
Netsite: omerOrder_qq_CustomerNumber&CustomerOrder_qq_CustomerNumber=3&LinkName=HTMLtutApp.CustomerOrderLink2&Uniquifier=76
HTML Tutorial
CustomerOrder Menu:
Search
Insert
Update
Delete
First Order
Last Order
Next Order
Previous Order
Go to
Redisplay CustomerOrder
Powered by Forté WebEnterprise
Document: Done

Customer Orders

Customer Name	R. Simmons
Customer Number	3
Order Number	1005
Requested Date	01-Mar-1995 00:00:00

record 1 of 2 Home

Product Description

		Order Number	Item Number	PN	Description	Price	Quantity
▶	1	1005	1	90002	Fiberglass Canoe	59.99	1
	2	1005	2	90003	5 Foot Oar	9.99	1

Search
Insert
Update
Delete
Redisplay LineItem

The added order

Exiting the Application

Exit the application as described in "Exiting the Application" on page 75.

About WebEnterprise Designer Session Management

In WebEnterprise, the *session management* features allow you to manage state, identity, and security information. Session management allows you to build continuity into an application as users navigate, however randomly, through a Web site. Session management also gives users the feeling that the application is tailored to them individually. Setting up session management is beyond the scope of this tutorial, but the following discussion outlines what you need to know and where to find the information.

Session Management Requirements

Different types of Web applications have different requirements regarding session management. Consider the following Web application types:

- applications that have no need to track the actions of their Web users

  Users of such applications can visit any Web page without restriction or control, and the Web application maintains no user-specific information ("state"). An example is an on-line catalog that Web users can browse through freely. If this application does not allow users to order items, then session management is not important.

  In this type of application, the default WebEnterprise Designer session management is sufficient. See "Default WebEnterprise Designer Session Management."

- "shopping cart" applications that must keep track of data users enter into the system

  Such applications do not restrict access to the individual HTML pages, but must track a sequence of HTML requests together as a transaction.When the transaction is completed, the request must include identifying data (such as name, address, credit card information) to ensure that the submission is valid.

  In WebEnterprise Designer applications, you can use Web "cookies" to create the transaction. These are described fully in *A Guide to WebEnterprise*.

- applications that permit access to specified pages only to validated users

  Typically, these applications present a Logon page to the user, who supplies an "identity" (usually a username and password) that must be validated before access to the application's HTML pages is granted. The identity becomes a property of each Web request and may be used by the application to restrict access to certain pages or activities.

Procedures for implementing this type of security are provided in *Customizing WebEnterprise Designer Applications*.

Default WebEnterprise Designer Session Management

When a WebEnterprise Designer application does not have a Logon page defined for it, all generated HTML templates are defined with the SESSION_AUTOCREATE session property. A WebEnterprise session is then automatically created whenever a Web client requests an HTML template. This "Web session" is persistent, binding subsequent Web requests from that browser client together, until it eventually expires (based on the value of the SessionTimeOutInterval attribute). In this scenario, all application pages are accessible without any logon or authentication procedure.

Partitioning and Deploying the Tutorial Application

Up to now, you have used the WebEnterprise Designer workshops to develop a logical application without worrying about details of the environment in which the application will be deployed. In this last step of the development process, you will partition this logical application, make a distribution, and then run it in distributed mode.

In this section, you will:

- learn a bit about the partitioning and deployment process
- learn about the projects and services generated for an HTML application
- partition the application using the default configuration
- test the application in a distributed environment
- deploy the application

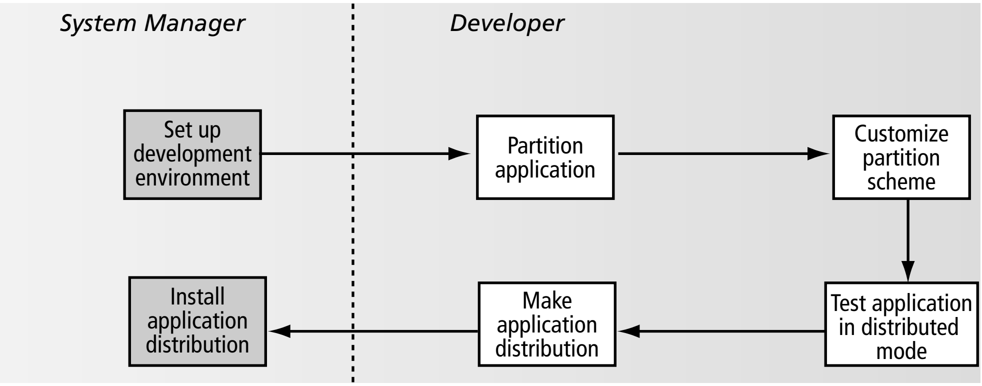
For further information, see:

- *A Guide to the iPlanet UDS Workshops* for information on the Partition Workshop
- *iPlanet UDS Programming Guide* for advanced deployment concepts
- *Customizing Express Applications* for business model projects and services
- *Customizing WebEnterprise Designer Applications* for advanced WebEnterprise partitioning and deployment information

The Partitioning and Deployment Process

Figure 2-12 shows the steps required to complete the partitioning and deployment process.

Figure 2-12 Making a Distribution and Installing the Application


System Manager
Developer
Set up development environment
Partition application
Customize partition scheme
Test application in distributed mode
Make application distribution
Install application distribution

Setting up the development environment In general, before you can partition the application, the system manager must set up the iPlanet UDS environment in which the application is partitioned. If you are running iPlanet UDS in distributed mode, you are already connected to an environment.

In this section, you will not customize the partition scheme; you will use only the default configuration. For information on customizing the default configuration, see *Customizing WebEnterprise Designer Applications* and *A Guide to WebEnterprise*.

About HTML Application Projects and Service Objects

When you generate the code for your HTML application, the following iPlanet UDS projects are created:

- *html_model*Handlers—the main project for the application

  For a description and example, see "Web Application Server Code" on page 67.

- *business_model*Services—a supplier project for *html_model*Handlers

  For a description and example, see "Viewing Generated Server Code" on page 44.

If your application model is based on more than one business model, there will be one *business_model*Services project for each business model, and these are all suppliers to the *html_model*Handlers project.

When you partition your HTML application using the Partition Workshop, iPlanet UDS creates a default configuration for the application based on the application's service objects. Both the "Handlers" and "Services" projects contain service objects, which are described in the following table.

Service Object	**Partition**	**Visibility**	**Dialog Duration**	**Description**
*html_model*AccessService	iPlanet UDS Web Access Server	Environment	Session	The application contains one AccessService service object, which provides the primary link between a Web browser and the iPlanet UDS application. It responds to HTTP requests from the Web client by returning the appropriate Web page.
*html_model*ScannerService	iPlanet UDS Web Access Server	Environment	Session	The application contains one ScannerService service object, which responds to HTTP requests from AccessService by opening the HTML template file in the request and processing the iPlanet UDS tags to generate a Web page.
*service_name*Service	Application Server	Environment	Message	The application contains one Service service object for each Express service defined in the Business Model Workshop. You can change the replication of this service object to provide load balancing and/or failover.
*service_name*DBService	Application Server	User	Session	The application contains one DBService service object for each Service service object. The DBService service object will always be in the same partition as the Service service object with which it is associated.

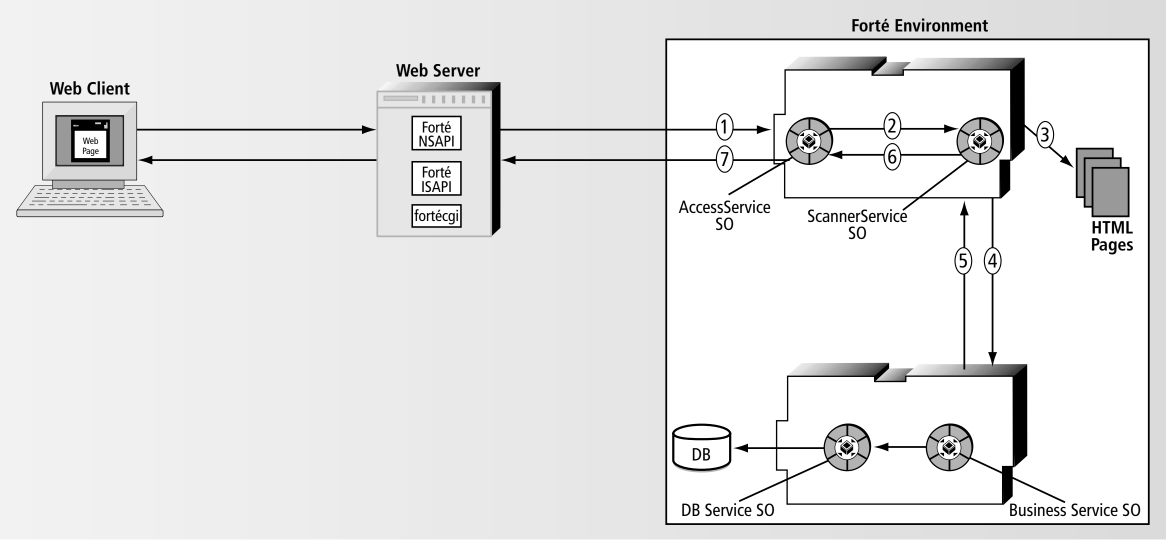
Relationship Between the Service Objects

The interactions between the four service objects in the HTML application are:

1. The AccessService service object receives HTTP requests for pages from fortecgi or an iPlanet UDS Web server plug-in on the Web server.
2. The AccessService service object invokes a method on the ScannerService service object, requesting the appropriate page and passing any parameter values relevant to the page.
3. The ScannerService opens the requested HTML page template and processes the iPlanet UDS tags to create the page, and executes methods, which may request data from the database.
4. Database requests are sent to the application server where DBService provides the database session used by the Service service object to access the database.
5. Result sets are returned to the ScannerService, which processes them and populates the page template with the result set. The entire result set is cached in the service in case the user needs to browse through the result set.
6. The ScannerService sends the requested page information back to the AccessService service object.
7. The AccessService service object delivers the HTML information to the Web server, which passes it to the client Web browser, which displays the page.

Figure 2-13 illustrates:

Figure 2-13 Relationship between AccessService, ScannerService, Service, and DBService



Partitioning the Tutorial Application

When you use the Partition Workshop to partition your application, iPlanet UDS creates logical partitions and assigns them to various nodes in the environment based on the properties of each node. Each *partition* is an independent process that can run on its own machine. The result of assigning partitions to nodes in an environment is called a *configuration*.

When you partition a WebEnterprise application, iPlanet UDS assigns the Express business model service objects to one partition and the WebEnterprise service objects to another.

Exceptions to default configuration The default partitioning will be altered if any of the following conditions are true:

- any of the service objects have replication turned on

  Replication is discussed in *A Guide to WebEnterprise*.

- external resource managers for the service objects differ in type or location

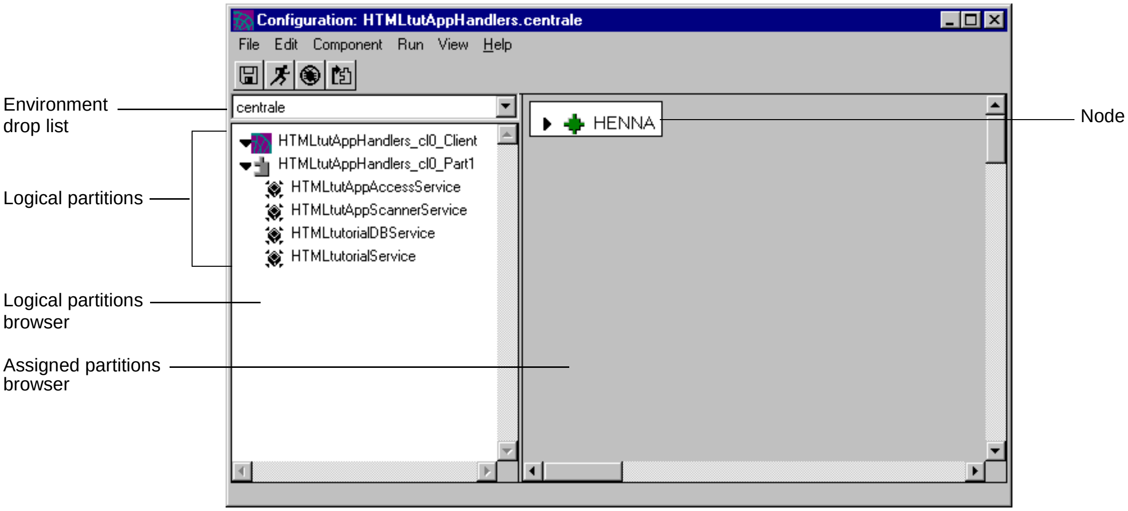
  This could occur if pages are built from multiple business models that have different resource managers. Selecting a resource manager for a business model was discussed in “Specifying a Resource Manager” on page 33. In this case, the Partition Workshop assigns each server partition to the node where the required external resource is installed.

➤ To partition the HTMLtutApp application

1. In the Repository Workshop, select the HTMLtutAppHandlers project and click the Partition button.

   Alternatively, you can choose the Plan > Run > Partition command. Or, you can open the HTMLtutAppHandlers project and click the Partition button or choose Run > Partition from there.

The Partition Workshop is displayed with the default configuration for the HTMLtutApp application, as shown in the following figure.

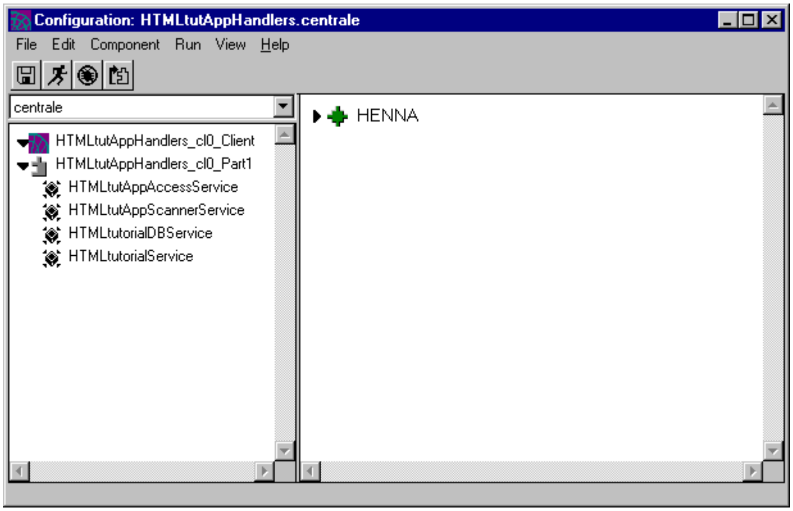

Configuration: HTMLtutAppHandlers.centrale
File Edit Component Run View Help
centrale
HTMLtutAppHandlers_cl0_Client
HTMLtutAppHandlers_cl0_Part1
HTMLtutAppAccessService
HTMLtutAppScannerService
HTMLtutorialDBService
HTMLtutorialService
HENNA
Environment drop list
Logical partitions
Logical partitions browser
Assigned partitions browser
Node

The left pane, called the Logical Partition browser, displays the application's logical partitions. The right pane, called the Node browser, displays the names of the nodes available in your environment. Just above the Logical Partition browser, there is a drop list for choosing the environment in which to partition the application.

The node browser of your Partition Workshop window will look different. You might have only one node in your environment or you might have several, but their names and types will differ from the one shown in the figure. You should be able to see the name of your own machine displayed in the node browser.

If the Logical Partition browser is not large enough to show the names of the logical partitions in their entirety, click the cursor on the vertical line dividing the browsers. The cursor changes into a cross-hair cursor, which you then can drag to expand the browser to the right.

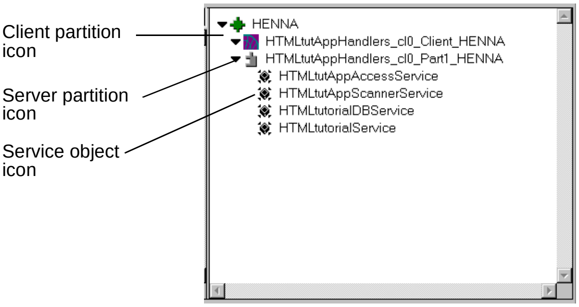
2. Choose View > Node Outline to change the view of the assigned nodes in the Node browser, as shown:


Configuration: HTMLtutAppHandlers.centrale
File Edit Component Run View Help
centrale

HTMLtutAppHandlers_cl0_Client
HTMLtutAppHandlers_cl0_Part1
HTMLtutAppAccessService
HTMLtutAppScannerService
HTMLtutorialDBService
HTMLtutorialService

HENNA

3. To see the contents of your logical partitions, click on the arrows to the left of the partition icon. Note the use of the icons to identify logical partition types.



HENNA
HTMLtutAppHandlers_cl0_Client_HENNA
HTMLtutAppHandlers_cl0_Part1_HENNA
HTMLtutAppAccessService
HTMLtutAppScannerService
HTMLtutorialDBService
HTMLtutorialService

The HTML Client Partition

The client partition (HTMLtutAppHandlers_Cl0_Client) is used to keep the server partitions running in the development environment and to launch, if selected, the browser window. In a deployed environment, it has no function. Because of this, the process of deploying and running a WebEnterprise Designer application is different from other iPlanet UDS applications. The WebEnterprise Designer process is described in "Deploying the Tutorial Application" on page 114.

Testing the Tutorial Application

You have put the HTMLtutApp application through a test run from the HTML Application Model Workshop, and now you should test it in a distributed environment from the Partition Workshop.

➤ **To test run the tutorial application in distributed mode from the Partition Workshop**

1. In the Partition Workshop (displaying the HTMLtutAppHandlers configuration), click the Run button.

   Alternatively, you can choose the Run > Run command.

2. Exit the application and close the Partition Workshop.

   If you are unable to exit the application, you can use the Run > Cancel Run command at any time to cancel execution.

   This command cancels the client partition for the application. The remote partitions will continue to run; use the Run > Stop Remote Partitions command to stop remote partitions.

➤ **To test run your application in distributed mode from other workshops**

NOTE You can only do this after you have partitioned your application once.

1. In the Repository Workshop, select the HTMLtutAppHandlers project and choose Plan > Run > Run Distributed.

   Alternatively, open the HTMLtutAppHandlers project and choose Run > Run Distributed.

2. Exit the application and close the Partition Workshop.

   If you are unable to exit the application, you can use Plan > Run > Cancel Run (Repository Workshop) or Run > Cancel Run (Project Workshop) at any time to cancel execution.

See *A Guide to the iPlanet UDS Workshops* for further information about testing applications in a distributed environment.

Deploying the Tutorial Application

After testing, you can make an *application distribution*. During this stage, iPlanet UDS creates all the files needed to install the application into a deployment environment. During installation, which is typically done by the system manager, code for each partition is installed on its assigned node.

The application distribution is a representation of the application outside the repository—you use the application distribution to install the application in an environment.

➤ To deploy your HTML application

1. In the Repository Workshop, double-click the main project for your application (the HTMLtutAppHandlers project, in this case).

   This opens the Project Workshop for the HTMLtutAppHandlers project.

2. In the Project Workshop, choose the File > Configure as > Server command.

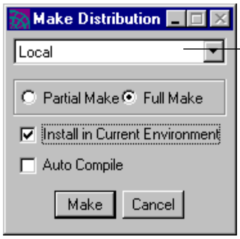
   This opens the Partition Workshop for the HTMLtutAppHandlers project.

   NOTE If, after configuring for deployment, you want to modify this application, you must reconfigure the HTMLtutAppHandlers project as a client.

3. Choose File > Make Distribution.

   iPlanet UDS displays a dialog for specifying the node where you want the files created, whether you want a full or partial distribution, and whether you want it to compile specified partitions.

   The first time you make a distribution, you should choose the Full Make option. Subsequently, you can choose Partial Make.



Local

○ Partial Make ● Full Make

☑ Install in Current Environment

□ Auto Compile

Location where you want Forte to place application distribution files. The choice is Local or Remote.

4. Specify where you want the application distribution files placed by selecting an item from the drop list.

   For the purpose of this tutorial, make the distribution on your local machine.

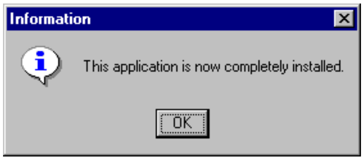
5. Select the Full Make radio button.

6. Enable the Install in Current Environment option.

7. Leave the Auto Compile toggle off, because you have not marked any partitions to be compiled.

8. Click the Make button.

   iPlanet UDS creates the distribution files and installs them on your machine. When it is finished, iPlanet UDS displays the following message to inform you when the distribution is complete.



Information

This application is now completely installed.

9. Click the OK button to close the dialog.

10. Close the Partition Workshop.

Running the Tutorial Application

Now you can run your deployed application.

➤ **To run the deployed tutorial application**

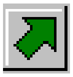
1. Start the Environment Console.

2. Double-click the node on which you made the distribution to open it.

   Alternatively, select the node and choose Component > Open.

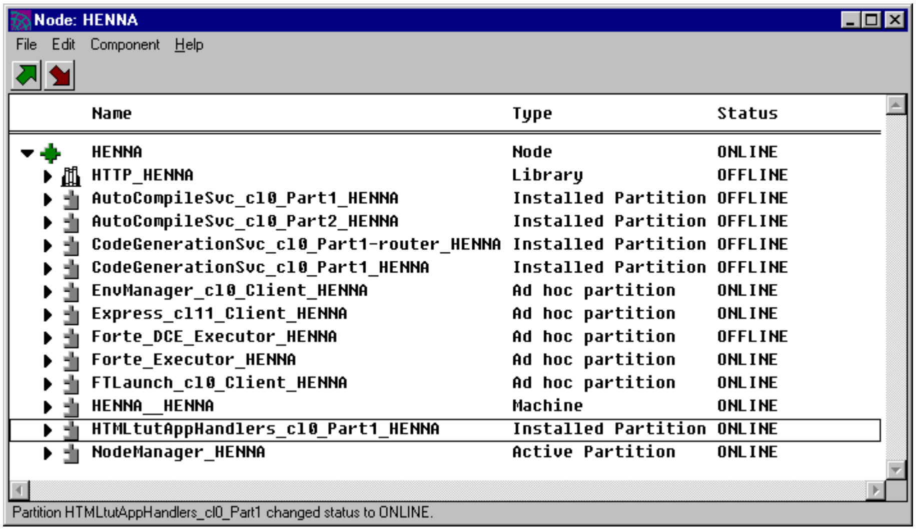
3. From the node window, click the triangular control to display the node’s contents.

4. Select the application and click the Start Up button.



Alternatively, select a server partition and choose Component > Start Up.

When the partitions are marked “on-line,” your application is running.



Name	Type	Status
HENNA	Node	ONLINE
HTTP_HENNA	Library	OFFLINE
AutoCompileSvc_cl0_Part1_HENNA	Installed Partition	OFFLINE
AutoCompileSvc_cl0_Part2_HENNA	Installed Partition	OFFLINE
CodeGenerationSvc_cl0_Part1-router_HENNA	Installed Partition	OFFLINE
CodeGenerationSvc_cl0_Part1_HENNA	Installed Partition	OFFLINE
EnvManager_cl0_Client_HENNA	Ad hoc partition	ONLINE
Express_cl11_Client_HENNA	Ad hoc partition	ONLINE
Forte_DCE_Executor_HENNA	Ad hoc partition	OFFLINE
Forte_Executor_HENNA	Ad hoc partition	ONLINE
FTLaunch_cl0_Client_HENNA	Ad hoc partition	ONLINE
HENNA__HENNA	Machine	ONLINE
HTMLtutAppHandlers_cl0_Part1_HENNA	Installed Partition	ONLINE
NodeManager_HENNA	Active Partition	ONLINE

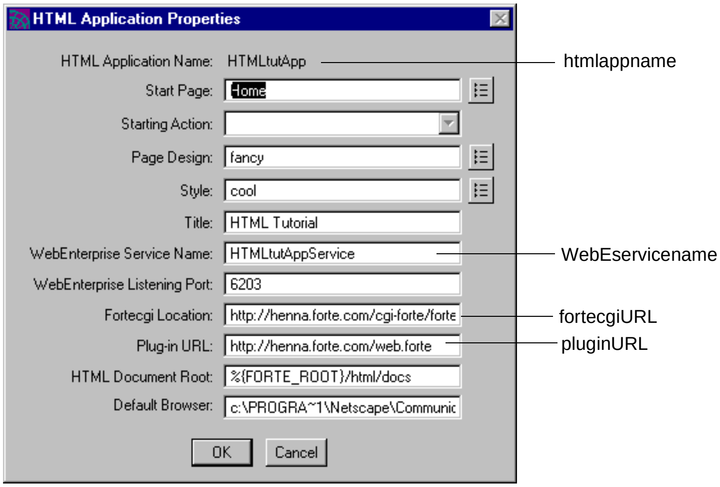
Partition HTMLtutAppHandlers_cl0_Part1 changed status to ONLINE.

5. Start your browser, if it is not already running.

6. Type in an URL with the following syntax:

```
]
[pluginURL|fortecgiURL]?servicename=WebEservicename=templatename
&templatename=htmlappname/Start.html
```

Note that this is all one string with no breaks. You obtain the values for the variables in this string from the HTML Application Properties dialog.



HTML Application Properties

HTML Application Name:	HTMLtutApp
Start Page:	Home
Starting Action:	
Page Design:	fancy
Style:	cool
Title:	HTML Tutorial
WebEnterprise Service Name:	HTMLtutAppService
WebEnterprise Listening Port:	6203
Fortecgi Location:	http://henna.forte.com/cgi-forte/forte
Plug-in URL:	http://henna.forte.com/web.forte
HTML Document Root:	%{FORTE_ROOT}/html/docs
Default Browser:	c:\PROGRA~1\Netscape\Communic

htmlappname

WebEservicename

fortecgiURL

pluginURL

Congratulations! You have just created your first WebEnterprise Designer application.

Enhancing the Tutorial Application

You can leave your tutorial application as it is, or you can enhance it with any or all of the enhancements provided in this section. These include:

- “Setting a Field Validation Option,” which follows
- “Auto-Formatting a Field” on page 119
- “Creating a URL Reference” on page 121

Setting a Field Validation Option

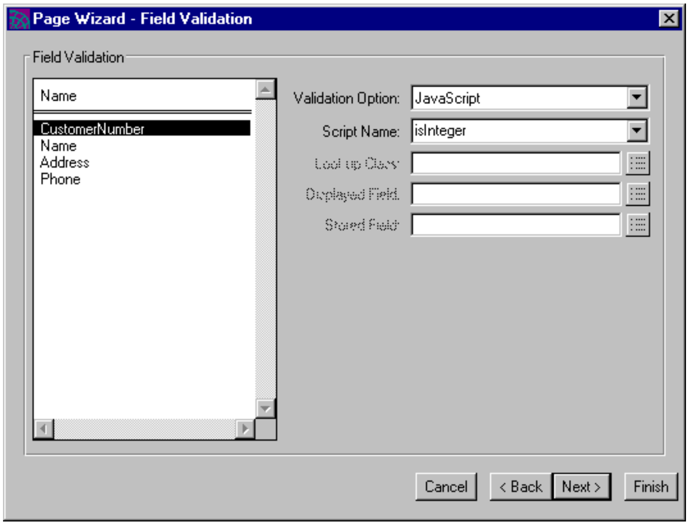
WebEnterprise Designer provides several JavaScripts that enforce constraints on user entry data types (for example, to disallow numeric entry in an alphabetic field). These validations pop up an explanatory message if a user enters a disallowed type of data. In this section, you will set the isInteger validation on the CustomerNumber field of the Customer page.

➤ To set a validation option

1. In the HTML Application Model Workshop, double-click the Customer page to display the Page Wizard.
1. Click Next until you get to the Field Validation page.
2. Select the CustomerNumber field in the left-hand list.
3. Select JavaScript from the drop list of the Validation Option field.

   The Script Name field is now activated so that you can select the appropriate script.
4. Select isInteger from the drop list for the script name.

   Your Field Validation page should look like this:



Page Wizard - Field Validation

Field Validation

Name

CustomerNumber
Name
Address
Phone

Validation Option: JavaScript

Script Name: isInteger

Look up Class:

Displayed Field:

Stored Field:

   Press your Help key for more information on this and other validation scripts available from this page.
5. Click Finish.

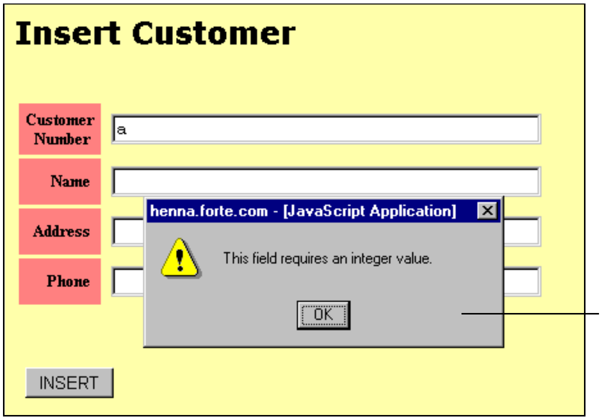
Now you should test your work.

➤ To test the validation script

1. Run the HTMLtutApp application.
2. Click `Customers` on the Home page to display the Customer page.
3. Click `Insert` to display the Insert mode of the Customer page.

4. In the Customer Number field, type a.
5. Leave the field by clicking in another field.

   The isInteger warning is displayed, as shown below.



Insert Customer

Customer Number: a

Name:

Address:

Phone:

henna.forte.com - [JavaScript Application]

This field requires an integer value.

The isInteger script warning window

Auto-Formatting a Field

WebEnterprise Designer provides several JavaScripts for automatically formatting user input in a field. In this section, you will add a record into the Part Description page. One of the fields is Price, and you will use a script that applies a US currency format to data entered into that field.

Because one of the fields is a picture of the new product, you will also learn how to add, at runtime, a record that includes a graphic. (The graphic is provided with the tutorial example.)

➤ To auto-format a field

1. In the HTML Application Model Workshop, double-click the Part2 page to display the Page Wizard.
2. Click Next till you get to the Field Properties page.
3. For the Price field, select asDollars from the drop list in the Formatting Routine column.
4. Click Finish.

Now you will insert a record to test your work.

➤ To test the asDollars formatting routine

1. Run the HTMLtutApp application.
2. Click `Product List` on the Home page to display the Tip-a-Canoe Products page.
3. Click `Insert` to display the Insert mode of the Products page.
4. Enter the following values on the Insert page:

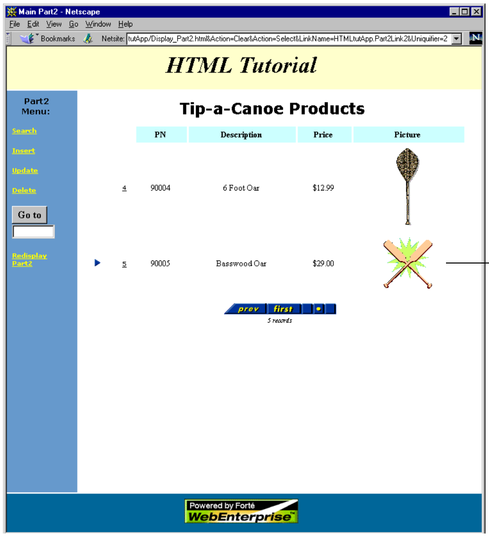
   PN: `90005`
   Description: `Basswood Oar`
   Price: `29`
   Picture: `/forte/examples/wedtut/images/2paddles.gif`

   NOTE The file specification is assumed to be under the Web server root. See "Tutorial Image Files" on page 34 for the location of tutorial image files on your system.

5. Click the Insert button to redisplay the Display page.

6. In the Go to field, enter 5 and click the Go to button.

The new record is displayed:



Main Part2 - Netscape
File Edit View Go Window Help
Bookmarks
Netsite: tutApp/Display_Part2.html&Action=Clear&Action=Select&LinkName=HTMLtutApp.Part2Link2&Uniquifier=2
Part2 Menu:
Search
Insert
Update
Delete
Go to
Redisplay Part2
prev
first
Powered by Forté WebEnterprise

HTML Tutorial

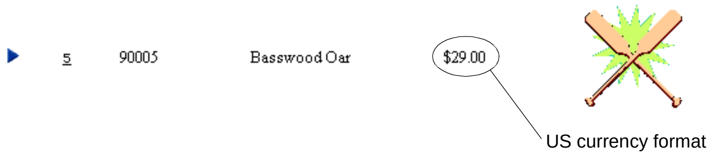
Tip-a-Canoe Products

		PN	Description	Price	Picture
	4	90004	6 Foot Oar	$12.99	
▶	5	90005	Basswood Oar	$29.00	

5 records

New record with picture

Observe that the Price field is properly formatted:



▶	5	90005	Basswood Oar	$29.00

US currency format

Creating a URL Reference

The last thing to add to the HTMLtutApp application is a reference to a Web site that is external to our application. Customers calling in may want advice on clothes and accessories to use with their new canoes and paddles. In this section, you will create a reference to the Web site of the recreational clothiers of your choice.

➤ To create a URL reference

1. Click the New URL Reference button.

   Alternatively, you can choose Component > New > URL Reference, or right-click in the drawing area and choose New URL Reference from the popup menu.

2. Click anywhere in the drawing area of the workshop.

   The Custom URL Properties dialog appears. For a description of this dialog, press your Help key.

3. Type ClothestoGo in the Name field.
4. Set the Custom toggle on.
5. Type the URL of your favorite outdoor clothiers' Web site in the URL field.
6. Click OK.

You have created a URL Reference page. Now you will create a command link to it.

➤ To create a link to the URL reference page

1. Create a link from the Home page to the ClothestoGo URL Reference page.

   The Link Properties dialog for this kind of link appears. For a description of this dialog, press your Help key.

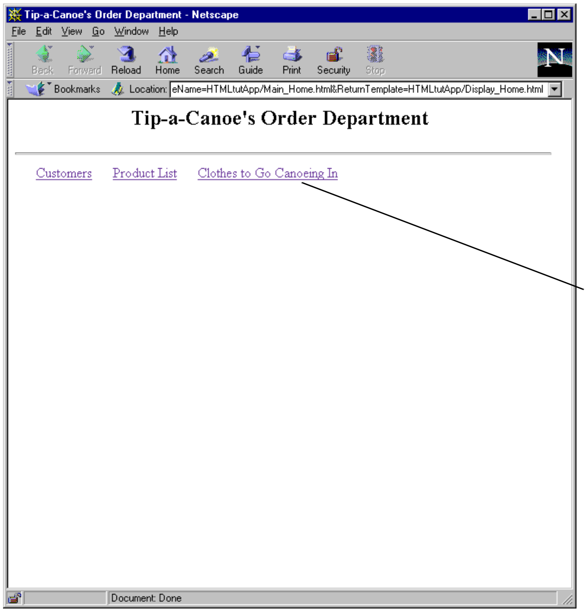
2. In the Label field, type in Clothes to Go Canoeing In.
3. Click the Advanced button.
4. Set this page to display in the same browser.
5. Click OK.

Now you should test your work.

➤ To test the URL reference

1. Rerun the HTMLtutApp application.

   The Home page now looks like this:



Tip-a-Canoe's Order Department - Netscape

Location: eName=HTMLtutApp/Main_Home.html&ReturnTemplate=HTMLtutApp/Display_Home.html

Tip-a-Canoe's Order Department

Customers Product List Clothes to Go Canoeing In

Document: Done

New link on the Home page

2. Click on the new `Clothes to Go Canoeing In` link.

   Your favorite outdoors clothiers’ home page is displayed.

3. To return to the application, click Back.

Appendix A

WebEnterprise Designer Example Applications

iPlanet UDS provides two example applications that illustrate how to use WebEnterprise Designer. This appendix provides instructions on how to install the examples, and a section describing each example in detail. Typically, you run an example application, then examine it in the various iPlanet UDS Workshops to see how it is implemented. You can modify the examples if you wish.

About the Examples

WebEnterprise Designer contains two examples to demonstrate various aspects of the product.

HTML Tutorial Example

The HTML Tutorial example is an online copy of the application described in Chapter 2, “WebEnterprise Designer Tutorial.” It does not include the enhancements described in the last section of the chapter. We recommend you go through the Tutorial to create your own application. Then, if you need to, you can load the HTML tutorial example and run and examine it. You can then add the enhancements manually, if you wish.

Movie Example

The Movie example creates a web site that displays information on movies, including their ratings, a brief description, information on their directors and producers, as well as where the movies are being shown and show times.

Installing WebEnterprise Designer Example Applications

You can run the WebEnterprise Designer example applications only after you have imported them into your repository, run a database script to create and populate the required database schema, modified the service properties in the business models, and regenerated the applications.

Importing the Examples into your Repository

Each WebEnterprise Designer example consists of two projects: the business model project and the HTML application model project. These projects are stored as import (.pex) files, which you import by choosing the Plan > Import command from the Repository Workshop.

The example import files are in the `FORTE_ROOT/install/examples/web/`*example_name* directory, as described in the following table.

example_name	File name	Description
movie	moviebm.pex	MovieBM project—Movie example business model
	moviehtml.pex	MooVie project—Movie example HTML application model
wedtut/pex	htut_b.pex	HTMLtutorial project—WebEnterprise Designer tutorial business model
	htut_a.pex	HTMLtutApp project—WebEnterprise Designer tutorial HTML application model

Creating Database Schema and Inserting Data

The WebEnterprise Designer examples rely on database tables. Each example has its own script to create and populate the tables it needs. These scripts are in the `FORTE_ROOT/install/examples/web/`*example_name* directory, as described in the following table.

example_name	File name	Description
movie	ormovie.sql	Creates tables used by the Movie example in Oracle SQL format.
	ordata.sql	Adds data to the tables created by ormovie.sql.
	ordrop.sql	Drops the tables added by ormovie.sql.
	msmovie.sql	Creates tables used by the Movie example in SQLServer SQL format.
	msdata.sql	Adds data to the tables created by msmovie.sql.
	msdrop.sql	Drops the tables added by msmovie.sql.
wedtut/install	makehtut.ora	Creates and populates tables used by the HTML Tutorial example in Oracle SQL format.
	drophtut.ora	Drops the tables created by makehtut.ora.
	makehtut.syb	Creates and populates tables used by the HTML Tutorial example in Sybase or SQLServer SQL format.
	drophtut.syb	Drops the tables created by makehtut.syb.

Creating Movie Example Tables

The Movie example uses a script to create its tables, and another script to populate them.

For Oracle: Use SQL*Plus to import the ormovie.sql script first, then the ordata.sql script.

For Microsoft SQLServer or Sybase: Use the appropriate SQL command line interpreter to import the msmovie.sql and msdata.sql scripts. You can edit these scripts for use with other databases.

Creating HTML Tutorial Example Tables

The HTML Tutorial example uses a single script to create and populate a database schema.

For Oracle: Use SQL*Plus to load the makehtut.ora script.

For Microsoft SQLServer or Sybase: Edit the first line of makehtut.syb to name an existing database. For example, create a database called "testapps," then edit maketut.syb to start with:

```
use testapps
```

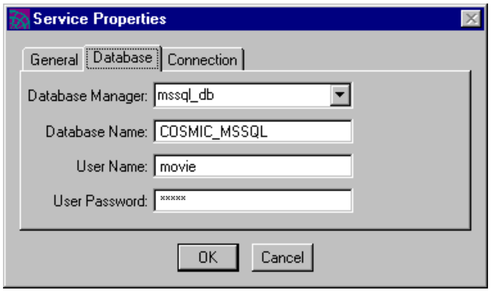
Then use the standard mechanism for redirecting the makehtut file to load the data into your database.

Modifying Service Properties in the Business Model

Before you can run an example, you must modify the service properties in the business model so that they are correct for your database.

For each example you plan to run, open the Business Model Workshop and double-click on the business service in the Business Class Service list. The Service Properties dialog opens:.

Figure A-1 Service Properties Dialog



Database Manager: mssql_db

Database Name: COSMIC_MSSQL

User Name: movie

User Password: *****

You must update the database fields (Database Manager, Database Name, User Name, User Password) with values appropriate for your environment. Once you have updated the fields, click OK.

NOTE The database manager is the name given for the resource manager for your node, which you specify with the iPlanet UDS Environment Console. You can review this name by following the procedures given in "Specifying a Resource Manager" on page 33.

Regenerating Services

Once you have imported the example and made the appropriate modification, you must generate the business server code. In the Business Model Workshop for the example business model, choose the File > Generate Server Code command. Regenerating the code propagates the new service properties you have set into the DBSession service object throughout the *business_model*Services project.

Detailed descriptions of how to run each example are provided in the "Application Descriptions" on page 131.

Example Image Files

Both WebEnterprise Designer examples display graphic files on their Web pages. The graphic files are installed on the Web server node by the WebEnterprise installer, and must be located in the following directory:

> *web_server_webenterprise_dir*/examples/wedtut/images

The `web_server_webenterprise_dir` directory is where your the WebEnterprise Designer files for the Web server are installed. For example:

- c:\Netscape\Suitespot\forte—for Netscape Enterprise Server on NT
- /usr/netscape/suitespot/forte—for Netscape Enterprise Server on UNIX
- c:\InetPub\forte—for Microsoft IIS

See your System Administrator (or the person who installed WebEnterprise) for the exact location of your WebEnterprise Web server files, if you don't know where they are.

Using the Browser

Review the options in "Running an HTML Application" on page 68 before running any of the examples. You must decide whether or not you want WebEnterprise Designer to start the browser automatically when you run the application. In either case, you must examine the example's HTML application model preferences and modify the Default Browser property accordingly.

Removing the Database Tables

When you no longer need the database tables, you can drop them using scripts described in the table provided in "Creating Database Schema and Inserting Data" on page 127.

Dropping Movie Example Tables

For Oracle: Use SQL*Plus to load the ordrop.sql script to remove the tables created by ormovie.sql.

For Microsoft SQLServer or Sybase: Use the appropriate SQL command line interpreter to import the msdrop.sql script to remove the tables created by msmovie.sql. Edit this script for use with other databases.

Dropping HTML Tutorial Example Tables

For Oracle: Use SQL*Plus to load the drophtut.ora script to remove the tables created by ormovie.sql.

For Microsoft SQLServer or Sybase: Use drophtut.syb; edit the first line to use the database you chose above. For example, if you created a database called testapps, then edit `droptut.syb` to start with:

```
use testapps
```

Then use the standard mechanism for redirecting the drophtut.syb file to remove the tables from your database.

Application Descriptions

The example has five sections describing it.

The `Description` section defines the purpose of the example, what problem it solves, and what WebEnterprise Designer features it illustrates.

The `Pex Files` section gives you the subdirectory and file names of the exported projects. You will need to import all the files listed to run the application. Be sure to import the files in the order given so that dependencies will be satisfied.

The `Graphic Files` section identifies the graphics the example uses.

The `Special Requirements` section identifies any special setup procedures you may need to follow.

Finally, the `To Use` section tells you how to step through the application's functions.

CAUTION Every example requires standard setup steps. Even if you only want to import one of the example applications, you will still need to follow the instructions given in "Installing WebEnterprise Designer Example Applications" on page 126 for building database schema, updating service properties, regenerating the application, and setting the default browser property.

Movie Example

Description The Movie example creates a web site for searching information on current movies, including their ratings, a brief description, information on their directors and producers, as well as where the movies are being shown and show times. Movie ratings use a label and a graphic of one to four stars that indicate the movie's relative merit.

Pex Files movie/moviebm.pex, movie/moviehtml.pex.

Graphic Files star1.gif, star2.gif, star3.gif, and star4.gif.

Special Requirements Remember to set your browser configuration. For information, see "Running an HTML Application" on page 68.

➤ To use the Movie application

1. Open the MooVie HTML Application Model.
2. Click the Run button.

   iPlanet UDS automatically generated Web server code before running the application.

   Experiment with searching for a variety of information about each movie.

HTML Tutorial Example

Description HTML Tutorial is an on-line copy of the application described in Chapter 2, "WebEnterprise Designer Tutorial." We recommend you go through the Tutorial chapter to create your own application. Then, if you need to, you can load the Tutorial example we provide and run and examine it.

Pex Files wedtut/htut_b.pex, wedtut/htut_a.pex.

Graphic Files canoea.gif, canoef.gif, paddle1.gif, paddle2.gif, and 2paddles.gif.

Special Requirements Remember to set your browser configuration. For information, see "Running an HTML Application" on page 68.

➤ To use the Tutorial application

1. Open the HTMLtutApp HTML Application Model.
2. Click the Run button.

   iPlanet UDS automatically generated Web server code before running the application.

   Procedures for querying, updating, inserting, and deleting records are described in Chapter 2, "WebEnterprise Designer Tutorial."
3. (Optional) Implement the enhancements as described in "Enhancing the Tutorial Application" on page 117.

Appendix B

Starting WebEnterprise Designer

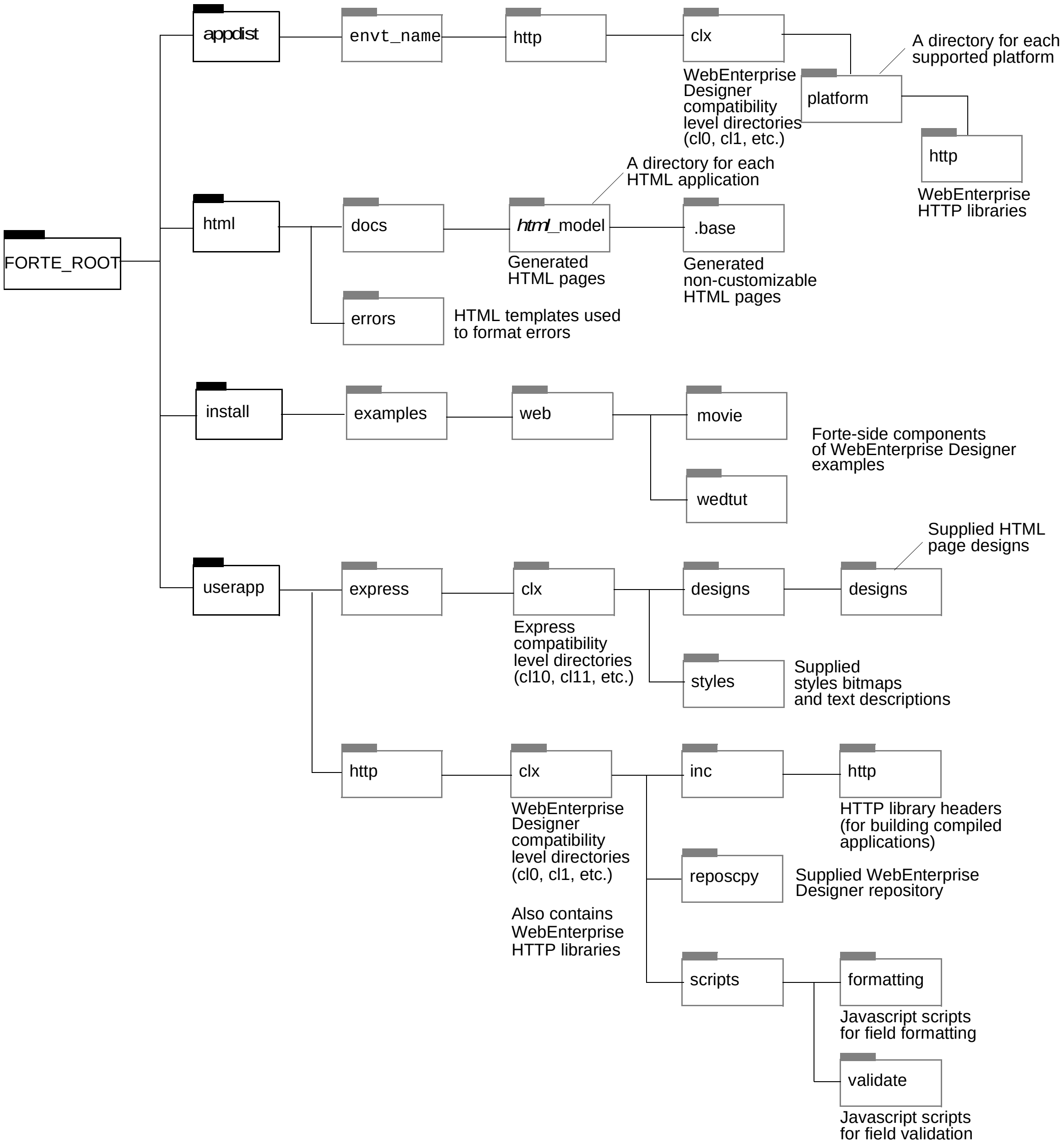
This appendix describes the elements of the WebEnterprise Designer software, and how to start WebEnterprise Designer. Topics covered in this appendix include:

- getting around the WebEnterprise Designer directory structure
- getting around the iPlanet UDS Web server directory structure
- WebEnterprise Designer environment variables
- installing the WebEnterprise Designer repository
- starting WebEnterprise Designer

Understanding the WebEnterprise Designer Directory Structure

Figure B-1 shows a typical iPlanet UDS System Software directory structure after installation of WebEnterprise Designer. Items of note specific to iPlanet UDS and Express are described in the *iPlanet UDS System Installation Guide* and *Express Installation Guide*. Figure B-1 details only those items of specific interest to the WebEnterprise Designer developer.

Figure B-1 WebEnterprise Designer Directories within the iPlanet UDS Directory Structure


FORTE_ROOT
appdist
envt_name
http
clx
WebEnterprise Designer compatibility level directories (cl0, cl1, etc.)
platform
A directory for each supported platform
http
WebEnterprise HTTP libraries
html
docs
html_model
A directory for each HTML application
Generated HTML pages
.base
Generated non-customizable HTML pages
errors
HTML templates used to format errors
install
examples
web
movie
wedtut
Forte-side components of WebEnterprise Designer examples
userapp
express
clx
Express compatibility level directories (cl10, cl11, etc.)
designs
designs
Supplied HTML page designs
styles
Supplied styles bitmaps and text descriptions
http
clx
WebEnterprise Designer compatibility level directories (cl0, cl1, etc.)
Also contains WebEnterprise HTTP libraries
inc
http
HTTP library headers (for building compiled applications)
reposcpy
Supplied WebEnterprise Designer repository
scripts
formatting
Javascript scripts for field formatting
validate
Javascript scripts for field validation

HTTP libraries During development, the HTTP header files under the `FORTE_ROOT/userapp/http/cl`*x*`/inc` directory are used when compiling WebEnterprise Designer applications (or any other applications that use HTTP as a supplier).

Compiled and interpreted HTTP libraries to be used with deployed applications are in the `FORTE_ROOT/appdist/`*envt_name*`/http/cl`*x*`/`*platform*`/http` directory. (Compiled libraries are named `http.dll`, interpreted libraries are named `http.lib`.) There is a separate directory for each supported platform.

Generated HTML pages HTML pages that are automatically generated from your HTML application model are written to the directory specified as the "document root." By default, this directory is `%{FORTE_ROOT}/html/docs` (note the use of portable syntax, which is required for this specification). For more information about setting the value of "document root," please see iPlanet UDS online Help. Another set of HTML files is generated in the .base subdirectory of the application and used by WebEnterprise Designer to preserve customizations to the generated files. Note that you should not modify the files in the .base directory.

Example files Subdirectories of the `FORTE_ROOT/install/examples/web` directory hold the iPlanet UDS-side components of the WebEnterprise Designer examples. These include import (.pex) files of the business and HTML application models, and scripts for importing database schema required by the examples. Web server-side files used by these examples are described in "Important Files on the Web Server Node," below. For information on installing and using the examples, see Appendix A, "WebEnterprise Designer Example Applications."

Formatting scripts Scripts used for field formatting are kept in the `FORTE_ROOT/userapp/http/cl`*x*`/scripts/formatting` directory. Currently, iPlanet UDS supplies four scripts, described in iPlanet UDS online Help (see the Field Properties page topic). For information about using a formatting script, see "Auto-Formatting a Field" on page 119. WebEnterprise developers can supply additional scripts. Any scripts placed in the formatting directory will be available for use within the WebEnterprise Designer HTML Application Model Workshop.

Validation scripts Scripts used for field validations are kept in the `FORTE_ROOT/userapp/http/cl`*x*`/scripts/validate` directory. Currently, iPlanet UDS supplies three scripts, described in iPlanet UDS online Help (see the Field Validation page topic). For information about using a validation script, see "Setting a Field Validation Option" on page 117. WebEnterprise developers can supply additional scripts. Any scripts placed in the validate directory will be available for use within the WebEnterprise Designer HTML Application Model Workshop.

Page Designs Subdirectories of the `FORTE_ROOT/userapp/express/cl`*x*`/designs` directory contain files that define designs for the HTML applications generated by WebEnterprise Designer. In addition, each subdirectory contains the definition text (a .txt file) and example graphic (a .bmp file) that display in the Page Designs dialog when you select the design. iPlanet UDS currently provides six designs: Fancy, FancyMenu, FancyNoBorder, FancyNoCaption, FancyNoFooter, and Simple. WebEnterprise developers can supply additional designs. Any directories placed in the designs directory will be available as page design choices in the WebEnterprise Designer HTML Application Model Workshop.

Styles Files in the `FORTE_ROOT/userapp/express/cl`*x*`/styles` directory include the definition text (a .txt file) and example graphic (a .bmp file) that display in the Styles dialog when you select a style. The files that define the styles are on the Web server node (refer to "Important Files on the Web Server Node," which follows this section). iPlanet UDS currently provides two styles: Cool and Steel. WebEnterprise developers can supply additional styles. Any .txt and .bmp files placed in the styles directory will display as style choices in the Styles dialog.

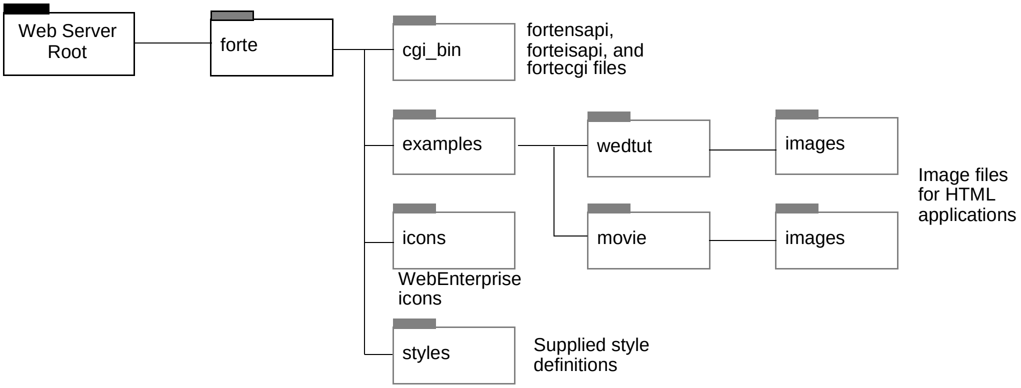
WebEnterprise Designer repository iPlanet UDS supplies a special WebEnterprise repository, wwseed, that includes all the required projects and libraries for the product. Using this repository is convenient, especially when you are using WebEnterprise Designer for the first time. For other repository configuration options, see "Installing the WebEnterprise Designer Repository" on page 140.

Errors The errors directory contains HTML templates used to format errors encountered while processing user requests.

Important Files on the Web Server Node

In addition to the files located in the FORTE_ROOT directory, the WebEnterprise install program places files under the Web server directory. Figure B-2 illustrates the location of these files.

Figure B-2 WebEnterprise Designer Directories on the Web Server


Web Server Root
forte
cgi_bin
fortensapi, forteisapi, and fortecgi files
examples
wedtut
images
movie
images
Image files for HTML applications
icons
WebEnterprise icons
styles
Supplied style definitions

cgi_bin files This directory contains WebEnterprise executables and libraries used by your HTML application to interact with the Web server. Specifically, the AccessServer service object of your HTML application interacts with these executables and libraries. For more information on these executables and libraries, see *A Guide to WebEnterprise*.

Example files Subdirectories of the examples directory hold graphic files used by HTML applications. These files are in a format displayable in a browser (such as .gif, .jpg, or .html). The HTML Tutorial files are located in the wedtut/images subdirectory, and are discussed in "Creating a Page that Displays Graphics" on page 85. The Movie example files are located in the movie/images subdirectory.

Icons This directory contains image files of the WebEnterprise icons.

Styles This directory contains the Cascading Style Sheet files used by applications on this Web server. These files should end in .css and have the same name as those in the `FORTE_ROOT/userapp/express/clx/styles` directory. See the definition of Styles in the previous section.

WebEnterprise Designer Environment Variables

This section describes the environment variables used by WebEnterprise Designer. On Windows NT platforms, the WebEnterprise installer prompts the user to specify values for these variables during installation. On UNIX platforms, these environment variables must be set manually. The procedures for setting iPlanet UDS environment variables are discussed in *A Guide to the iPlanet UDS Workshops*.

iPlanet UDS Server Environment Variables

The following table lists the optional environment variables that may be set on the WebEnterprise development or deployment node.

On Windows NT, if you set these variables with the WebEnterprise installer, they are set in the `HKEY_LOCAL_MACHINE\Software\ForteSoftwareInc\Forte\`*version* folder of the NT registry.

Variable Name	Description*	Data-type	Default Value	Example
FORTE_WW_DOCUMENT_ROOT	Root directory of all generated HTML files. Used by WebEnterprise in development, and by the ScannerService SO after deployment.	String	`%{FORTE_ROOT}/html/docs` This variable must be set in portable syntax. Defaults to this value if environment variable is not set.	See the entry for this environment variable in iPlanet UDS online Help.
FORTE_DEFAULT_BROWSER	Directory of the executable for the default Web browser. Used in development for testing WebEnterprise Designer applications.	String	None	See the entry on default browser in iPlanet UDS online help.
FORTE_WW_HTMLEDITOR	Specifies the HTML editor the Page Handler Customization Wizard should use when opening .html files.	String	None	`D:\frontpage\bin\fpeditor.exe`
FORTE_WW_EDITOR	Specifies the text editor the Page Handler Customization Wizard should use when opening files other than .html files.	String	None	c:\Winnt\System32\notepad.exe

Variable Name	Description*	Data-type	Default Value	Example
WWW_CGI_URL_BASE	The fortecgi URL used to register the AccessService SO with your Web server.	String	None	`http://`*node*`.forte.com/cgi-forte/fortecgi.exe` (Windows NT) `http://`*node*.`forte.com/cgi-forte/fortecgi` (UNIX)
WWW_PLUGIN_URL_BASE	Used only when the NSAPI or ISAPI plug-in is used.	String	None	`http://`*node*.`forte.com/web.forte`

A WebEnterprise environment variable for the iPlanet UDS server is FORTE_CGI_REG_PORT. This is used mainly in autoregistration. For a description of this variable, see iPlanet UDS online Help.

Web Server Environment Variables

The following table lists the environment variable for the Web server.

On Windows NT, if you set this variable with the WebEnterprise installer, it is set in the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment` folder of the NT registry, which governs the “System” environment variables set in the NT Control Panel.

Variable Name	Description	Data-type	Default Value	Example
FORTE_CGI_REG_FILE	Path name for the fortecgi.dat file. If not set, the fortecgi.dat file is written to the same directory as the fortecgi program.	String	*web_server_root*\ `forte\cgi_bin\fortecgi.dat`	c:\netscape\suitespot\forte\cgi_bin\fortecgi.dat

Other WebEnterprise environment variable for the Web server are FORTE_CGI_REG_PORT and FORTE_WW_HANDLER_CONFIG_FILE. For a description of these variables, see iPlanet UDS online Help.

Installing the WebEnterprise Designer Repository

WebEnterprise Designer provides you with a repository that includes all the projects and libraries required for using WebEnterprise Designer. This is the "wwseed" repository.

Using the wwseed repository is convenient, unless you already have iPlanet UDS or Express repositories that you would rather use.

If you don't install wwseed, and use instead either an existing or new iPlanet UDS or Express repository, you will be prompted to upgrade your repository when you start WebEnterprise or Express. Follow the prompt directions and your repository is automatically updated.

➤ **To install the wwseed repository**

1. Copy the two wwseed files (`wwseed.btx` and `wwseed.btd`) in the `FORTE_ROOT/userapp/http/cl0/reposcpy` directory.

   Refer to Figure B-1 on page 134.

2. Paste the two files into the `FORTE_ROOT/repos` directory.
3. Rename the wwseed.* files to a name of your choice (for example, `myweb.btd`, `myweb.btx`)
4. Start the iPlanet UDS Control Panel.
5. Type the following in the Repository Name field:

Platform	Command Syntax
Windows NT	bt:%FORTE_ROOT%\repos\myweb
UNIX	bt:$FORTE_ROOT/repos/myweb

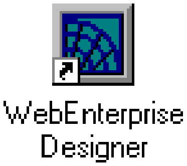
Alternatively, you can set the FORTE_REPOSNAME environment variable.

6. Click OK to close the iPlanet UDS Control Panel.
7. Start WebEnterprise Designer (see "Starting WebEnterprise Designer" on page 141).

Starting WebEnterprise Designer

Depending on your installation, you can start WebEnterprise Designer either by selecting and opening a window system icon or by using a command line, as described below. Figure B-3 illustrates a window system icon for starting WebEnterprise Designer:

Figure B-3 WebEnterprise Designer Windows Icon



If you want to start WebEnterprise Designer from the command line, you use the Express command with the following syntax:

Portable

```
ftcmd run express [-fs] [-docroot path][-fr repository] [-fw workspace] [-fnd
node_name]
                    [-fmn model_node_name] [-fm memory_flags] [-fl logger_flags]
[-fcons]
```

The following table explains the command line flag:

This Flag	Specifies
-fs	Run the development session in standalone mode. In standalone mode, the node is not connected to the distributed development environment, and you cannot partition projects in the Partition Workshop or use a central repository. The default mode is distributed, which connects you to the name server and environment manager (see the *iPlanet UDS System Management Guide* for information about the distributed development environment).

This Flag	Specifies
-docroot *path*	Run the development session using *path* as the root directory of all generated HTML files. Overrides the value set by FORTE_WW_DOCUMENT_ROOT or the Documentroot value of the ScannerService service object. See "iPlanet UDS Server Environment Variables" on page 138.
-fr *repository*	The repository to be used for the development session. See below for information on specifying a repository.
-fw *workspace*	The workspace for the development session. See below for information on specifying a workspace.
-fnd *node_name*	Specifies the node name to use for this session. If you do not specify the node name in the express command, the default node name depends on the operating system. On Windows and Macintosh, the default node name is set by the FORTE_NODENAME environment variable. On all other platforms, the actual node name is used.
-fmn *model _node_ name*	Specifies the model node name to use for this session. If you do not specify a model node name in the express command, iPlanet UDS uses the value of the FORTE_MODELNODE environment variable. If the environment variable is not set, the node is not treated as a model node.
-fm *memory _flags*	Specifies the space to use for the memory manager. See Appendix B in *A Guide to the iPlanet UDS Workshops* for details.
-fl *logger_flags*	Specifies the logger flags to use for the session. See Appendix B in *A Guide to the iPlanet UDS Workshops* for details. If you do not set the logger flags in the express command, iPlanet UDS uses the value of the FORTE_LOGGER_SETUP environment variable. Note that you can change the logger settings from the Repository Workshop.
-fcons	Displays the trace window. Use this flag to display the trace window on startup.

The following are examples of the express command:

```
express -fs
express -fr bt:$FORTE_ROOT/repos/examples
express -fl "%stdout(trc:os:1:1 trc:err)" -fm "(n:16000,x:64000)"
```

Selecting a repository To specify the repository in the express command:

- For a central repository, specify a repository service name. See your system manager for information about your central repository.
- For a private B-tree repository, specify the repository name using the following format: **bt:** *private_repository_name*.
- For a shadow repository, specify the repository name using the following format:
  bt: *shadow_repository_name*.

See *A Guide to the iPlanet UDS Workshops* for information about the different kinds of repositories.

Default repository If you do not specify a repository in the express command, the default repository depends on the operating system. For all platforms, iPlanet UDS uses the setting of the FORTE_REPOSNAME environment variable, and if the environment variable is not set, iPlanet UDS uses the distributed repository called "CentralRepository."

Selecting a workspace If you do not specify a workspace in the express command, the default workspace depends on the operating system. For all platforms, iPlanet UDS uses the setting of the FORTE_WORKSPACE environment variable, and if the environment variable is not set, iPlanet UDS opens the Repository Workshop a default workspace named FirstWorkspace.

Index