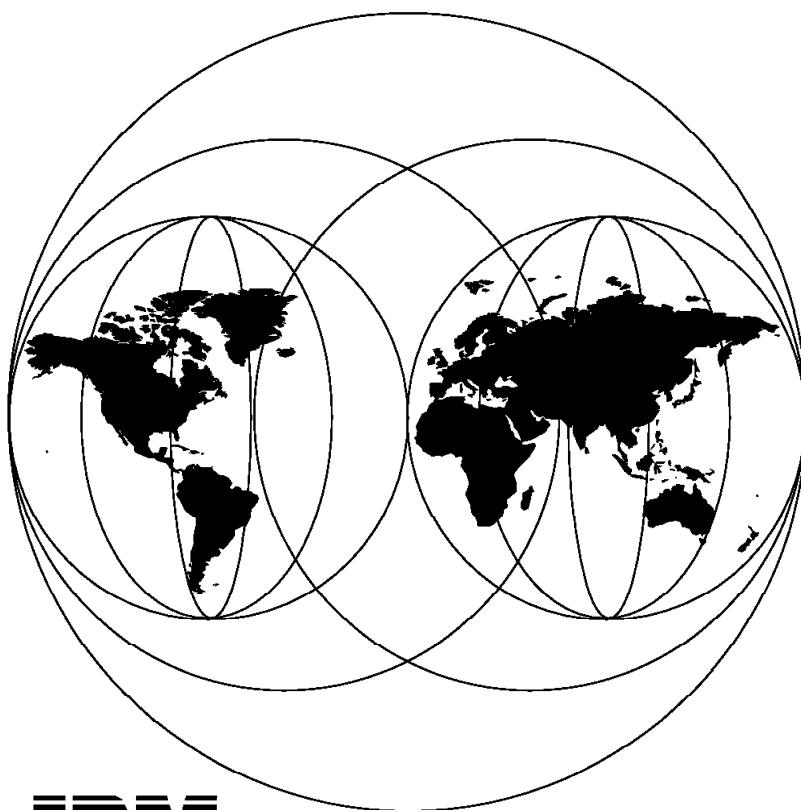


# Integrating Net.Commerce with Legacy Applications

June 1997



**International Technical Support Organization  
Raleigh Center**





International Technical Support Organization

SG24-4933-00

## **Integrating Net.Commerce with Legacy Applications**

June 1997

**Take Note!**

Before using this information and the product it supports, be sure to read the general information in Appendix B, "Special Notices" on page 273.

**First Edition (June 1997)**

This edition applies to Version 1.1 of IBM Net.Commerce, Program Number 5697-A16, for use with the AIX and Windows NT operating systems.

Comments may be addressed to:  
IBM Corporation, International Technical Support Organization  
Dept. HZ8 Building 678  
P.O. Box 12195  
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1997. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Figures</b> .....	vii
<b>Tables</b> .....	xi
<b>Preface</b> .....	xiii
The Team That Wrote This Redbook .....	xiii
Comments Welcome .....	xiv
<b>Chapter 1. Electronic Commerce - The Fundamentals</b> .....	1
1.1 General Concepts .....	1
1.2 Net.Commerce Overview .....	2
1.2.1 Net.Commerce Site Manager .....	4
1.2.2 Net.Commerce Store Manager .....	5
1.2.3 Net.Commerce Template Designer .....	6
1.2.4 Related System Components .....	7
1.3 Electronic Payment Overview .....	8
1.3.1 Net.Commerce Payment for Merchants .....	10
1.4 Customization .....	11
1.4.1 Net.Commerce Commands .....	11
1.4.2 Net.Data Macros .....	13
1.4.3 Net.Commerce Application Program Interface (API) .....	14
<b>Chapter 2. Designing Your Individual Shopping Site</b> .....	17
2.1 Products and Their Classification .....	17
2.1.1 Choosing the Right Products .....	17
2.1.2 Product Categories .....	18
2.2 Graphical Page Layout .....	20
2.2.1 Using Recognizable Styles .....	21
2.2.2 Using Graphics and Colors .....	21
2.2.3 Using Templates .....	21
2.2.4 Using Headers and Footers .....	21
2.2.5 Using a Common Set of HTML Features .....	21
2.2.6 Using Images and Image Formats .....	22
2.2.7 Using Different Layouts for Different Customers .....	23
2.3 Preliminary Aspects of Operation .....	23
2.3.1 Offering a Shopping Cart Facility .....	23
2.3.2 Updating Your Contents .....	23
2.3.3 Keeping Your Links Accurate .....	23
2.3.4 Defining an Adequate Order Process .....	24
2.3.5 Defining an Adequate Fulfillment Process .....	24
2.3.6 Some Words on Publicity .....	24
2.4 The Anatomy of a Net.Commerce Shopping Trip .....	24
2.4.1 Home Page .....	29
2.4.2 Catalog Browsing .....	30
2.4.3 Shopping Cart .....	31
2.4.4 Registration .....	31
2.4.5 Address Book .....	31
2.4.6 Shipping Information .....	32
2.4.7 Ordering and Payment .....	32
2.4.8 Default Shopping Trip .....	32
2.5 Post-Order Processes .....	33

2.6 Possible Store Scenarios	34
2.6.1 Small Single Store	34
2.6.2 Large Single Store	35
2.6.3 Small Mall	35
2.6.4 Large Mall	36
2.7 An Example of a Small Shop	36
2.7.1 The Ma Baker Environment	36
2.7.2 Shopping at Ma Baker's	37
2.7.3 Customizing the Order Process	38
<b>Chapter 3. Advanced Planning</b>	<b>43</b>
3.1 Planning a Net.Commerce Project	43
3.1.1 Skills Profile for Advanced Scenarios	43
3.1.2 Time Schedule	44
3.1.3 Who Will Be Your Service Provider?	44
3.1.4 Marketing and Design	45
3.1.5 Hardware and Software Requirements	45
3.2 System Selection	47
3.2.1 Using a Single System Configuration	47
3.2.2 Using a Multiple System Configuration	48
3.3 Database Considerations	49
3.3.1 Database Space Requirements	49
3.3.2 Accessing Data and the Database	50
3.3.3 Understanding the Database Structure	50
3.3.4 Database Tables	52
3.3.5 Net.Commerce Database Recovery	59
3.3.6 Database Recovery Configuration Example	61
3.4 Web Server Considerations	63
3.5 Network and Security Considerations	66
3.5.1 The Threat	66
3.5.2 The Value	67
3.5.3 The Objective	67
3.5.4 The Weapons	68
3.5.5 Firewall Considerations	74
3.5.6 Additional Security Approaches	76
3.6 Examples of Aspects of Local Legislature	77
3.6.1 Regulations for Electronic Commerce on the Internet	77
3.6.2 General Regulations for Commerce	79
<b>Chapter 4. Installation and Configuration</b>	<b>81</b>
4.1 Installation and Basic Configuration on AIX	81
4.1.1 Preparation	81
4.1.2 Installation of the Files	83
4.1.3 Basic Configuration	86
4.1.4 Verification of the Installation	97
4.1.5 Installing the Demo Mall	101
4.1.6 Uninstalling the System	103
4.2 Installation and Basic Configuration on Windows NT	104
4.2.1 Preparation	104
4.2.2 Installation of the Files on the Net.Commerce Server	106
4.2.3 Installation of the Files on the Database Server	108
4.2.4 Basic Configuration of the Database Server	109
4.2.5 Basic Configuration of the Net.Commerce Server	112
4.2.6 Verification of the Installation	125
4.2.7 Installing the Demo Mall	128

4.2.8	Configure ODBC Support for the Net.Commerce Database	131
4.2.9	Basic Performance Tuning	133
4.2.10	Uninstalling the System	134
4.3	Using Lotus Approach to Access the Net.Commerce Database	135
<b>Chapter 5. Integrating Net.Commerce with CICS</b>		<b>137</b>
5.1	General Concept	137
5.1.1	External Presentation Interface (EPI)	137
5.1.2	External Call Interface (ECI)	137
5.2	Platform Implementations	137
5.2.1	AIX Platform	137
5.2.2	Windows NT Platform	139
5.3	System Environment	142
5.3.1	Additional System Requirements	144
5.4	The Application Scenario	144
5.5	Mainframe Definitions	144
5.5.1	VTAM/NCP Definitions	145
5.5.2	CICS/ESA Definitions	152
5.6	Configuring the SNA Connection with IBM Personal Communications	155
5.6.1	Configure the SNA Node	155
5.6.2	Configure 3270 Terminal Emulation	161
5.7	Installing and Configuring the Transaction Systems	163
5.7.1	Installing and Configuring IBM Transaction Server for Windows NT	164
5.7.2	IBM Transaction Server Basic Configuration	167
5.7.3	Configuring IBM Transaction Server As a Transaction Router	170
5.7.4	Testing the Transaction Server Configuration	177
5.7.5	Creating Users	178
5.7.6	Creating a Listener Definition for TCP/IP Client	180
5.7.7	Installing IBM CICS Client for NT	181
5.7.8	Configuring the IBM CICS Client	182
5.7.9	Testing the CICS Client Configuration from a CICS Terminal	183
5.7.10	Configuring IBM CICS Internet Gateway on NT03	183
5.7.11	Testing the CICS Client Configuration from a Web Browser	184
5.8	Configuring Net.Commerce to Interface with CICS	186
5.8.1	Using Your Own APIs with the System	186
5.8.2	Using Your own Macros with the System	188
5.8.3	Running the Application without Net.Commerce	189
5.8.4	Running the Application With Net.Commerce	189
5.8.5	What Happens in Case of a CICS Error?	191
5.9	The CICS API Program and Customized Net.Commerce Macros	194
5.9.1	Program Source Code	194
5.9.2	Macro Definitions	208
5.9.3	Include Files	216
<b>Chapter 6. Integrating Net.Commerce with MQSeries</b>		<b>227</b>
6.1	General Concept	227
6.2	Implementation Scenarios	227
6.2.1	Possible Scenario 1	227
6.2.2	Possible Scenario 2	228
6.3	System Environment	229
6.3.1	Additional System Requirements	230
6.4	The Application Scenario	231
6.5	Installing and Configuring the System	231
6.5.1	Installing MQSeries on NT03	231
6.5.2	Installing MQSeries on NT04	232

6.5.3	Initial Configuration of MQSeries Server on NT03	232
6.5.4	Initial Configuration of MQSeries Server on NT04	233
6.5.5	Description of the MQSeries Configuration Files on NT03	233
6.5.6	Configuring MQSeries on NT03	234
6.5.7	Description of the MQSeries Configuration Files on NT04	234
6.5.8	Configuring MQSeries on NT04	235
6.6	Configuring Net.Commerce to Interface with MQSeries	235
6.6.1	Using Your Own APIs with the System	235
6.6.2	Using Your Macros with the System	237
6.6.3	Running the Application without Net.Commerce	238
6.6.4	What Happens in the Case of an MQSeries Error	241
6.7	The MQSeries API Program and Customized Net.Commerce Macros	244
6.7.1	Program Source Code	244
6.7.2	Macro Definitions	252
6.7.3	Include Files	261
<b>Appendix A. ITSO System Environment</b>		<b>271</b>
<b>Appendix B. Special Notices</b>		<b>273</b>
<b>Appendix C. Related Publications</b>		<b>277</b>
C.1	International Technical Support Organization Publications	277
C.2	Product Documentation	277
C.3	Redbooks on CD-ROMs	277
C.4	Other Publications	278
C.5	World Wide Web Resources for Information	278
<b>How to Get ITSO Redbooks</b>		<b>279</b>
How IBM Employees Can Get ITSO Redbooks		279
How Customers Can Get ITSO Redbooks		280
IBM Redbook Order Form		281
<b>List of Abbreviations</b>		<b>283</b>
<b>Index</b>		<b>285</b>
<b>ITSO Redbook Evaluation</b>		<b>287</b>



---

## Figures

1.	The Net.Commerce System	3
2.	Net.Commerce Site Manager	5
3.	Net.Commerce Store Manager	6
4.	Template Designer	7
5.	Product Relationships	8
6.	Secure Electronic Transactions (SET)	10
7.	Net.Commerce Payment Interface	11
8.	Net.Commerce Command Structure	12
9.	Net.Commerce Macro Structure	13
10.	Net.Commerce System Structure	14
11.	Configuring for Multiple Parents	19
12.	Database Tables for Categories, Products and Items	20
13.	Database Tables for the Ordering Process	28
14.	A Simple Set of Category and Product Pages	30
15.	Net.Commerce Default Shopping Trip	33
16.	The Ma Baker Home Page	37
17.	A Ma Baker Product Page	38
18.	DB2WWW.INI on Windows NT with Flat File Environment Registered	41
19.	Net.Commerce Implementation Team	44
20.	Net.Commerce Single System Configuration	47
21.	Net.Commerce Multiple System Configuration	49
22.	Entity Relationships Model for Net.Commerce	51
23.	Referential Integrity Model for Net.Commerce	52
24.	Database Tables for Shoppers	54
25.	Database Tables for Merchants	55
26.	Database Tables for Categories and Products	56
27.	Database Tables for Shipping	57
28.	Database Tables for Orders and Payment	58
29.	Other Database Tables	59
30.	Plotting Web Server Peak Utilization	65
31.	Secure Sockets Layer (SSL)	70
32.	Creating Certificates	72
33.	SET Roles	73
34.	Firewall Models	74
35.	Simple Firewall Setup	75
36.	Complex Firewall Setup	76
37.	Creating an AIX Group for DB2	82
38.	Creating an AIX User for DB2	82
39.	Installing Net.Commerce Components on AIX	84
40.	Installation Options on AIX	85
41.	Certification Process Using a Certification Authority	87
42.	Certification Process Using Self-Signed Certificates	88
43.	Creating Encryption Keys, AIX	89
44.	Key System Information, AIX	90
45.	Current Key Ring File, AIX	91
46.	Receive Certificate, AIX	92
47.	Designate Trusted Root Key, AIX	93
48.	Select Default Key, AIX	94
49.	Net.Commerce Server Manager, AIX	95
50.	Net.Commerce Configuration Files	98
51.	AIX mserver.conf File for Single System	98

52.	AIX httpd.conf File	99
53.	AIX db2www.ini File	100
54.	Net.Commerce Administrator, AIX	101
55.	Adding User Defined Functions to the Net.Commerce Database on AIX	102
56.	Net.Commerce Demo Mall Home Page, AIX	103
57.	Creating a Windows NT Account for DB2	105
58.	Windows NT Shut Down/Log Off Dialog	105
59.	DB2 Client Setup Options, NT	107
60.	Net.Commerce Fix Pak #1 Component Selection, NT	108
61.	DB2 Server Setup Options, NT	109
62.	Setting DB2 Services Startup Configuration, NT	110
63.	Setting DB2 Communication Environment Variable, NT	111
64.	Setting ICSS Service Startup Configuration, NT	113
65.	Creating Encryption Keys, NT	114
66.	Key Ring Password	115
67.	Key System Information, NT	116
68.	Certificate Request Information	117
69.	Current Key Ring File, NT	118
70.	Receive Certificate, NT	119
71.	Designate Trusted Root Key, NT	120
72.	Select Default Key, NT	121
73.	Net.Commerce Administration Web Server	122
74.	Net.Commerce Server Manager, NT	123
75.	NT mserver.ini File for Multiple Systems	126
76.	Windows NT httpd.cnf File	127
77.	Windows NT db2www.ini File	127
78.	Net.Commerce Administrator	128
79.	MSADFUNC Command File	130
80.	Net.Commerce Demo Mall Home Page, NT	131
81.	DB2 Client ODBC Administrator	132
82.	Listing Windows NT ODBC Drivers	132
83.	Listing Windows NT ODBC User Data Sources	133
84.	Opening Net.Commerce Databases with Lotus Approach	136
85.	CICS Scenario 1 on AIX	138
86.	CICS Scenario 2 on AIX	139
87.	CICS Scenario 1 on Windows NT	140
88.	CICS Scenario 2 on Windows NT	141
89.	CICS Scenario 3 on Windows NT	142
90.	System Environment for Net.Commerce and CICS Integration	143
91.	VTAM Options for SA18	146
92.	Models VTAMLST Entry	148
93.	Display of PU W05253 from VTAM	149
94.	Display of WTR05253 from VTAM	150
95.	ISTADJCP Display	150
96.	ISTCDRDY Display	150
97.	VTAM APPN Directory Entry for WTR05253	151
98.	CICS Application Major Node	151
99.	Display of CICS Application RAIAC from VTAM	151
100.	Mode Table Entry for LU62APPC	152
101.	CICS Configuration Using the CEDA Transaction	152
102.	CICS Connection Profile Configuration	153
103.	CICS Session Profile Configuration	154
104.	SNA Node Operations	155
105.	SNA Node Configuration	156
106.	Node Definition	156

107. Define a LAN Device	157
108. Define a LAN Connection - Basic	158
109. Define a LAN Connection - Advanced	158
110. Define a LAN Connection - Security	159
111. Define a Partner LU 6.2 - Basic	159
112. Define a Partner LU 6.2 - Advanced	160
113. Define a Mode - Basic	160
114. Define a Mode - Advanced	161
115. Customize Communication Window	162
116. Customize 3270 Terminal Session	162
117. Configure Local System	163
118. Select Product To Install Panel	164
119. Select Production Environment To Install	165
120. CICS Server Setup Type	165
121. Select Server Destination Directory Panel	166
122. CICS Server Setup Information Panel	166
123. IBM Transaction Server New CICS Region (CICSNT01)	167
124. Local Listener Definition CICSNT01	168
125. Configuring Local CICS Client in Definition File (Server)	168
126. Start CICS NT Region	169
127. IBM Transaction Server Administration Utility	169
128. IBM Transaction Server Local Terminal	170
129. Properties of the Region CICSNT01	171
130. Listener Definition LOCALSNA	172
131. Communication Definition CESA - General	173
132. Communication Definition CESA - SNA	174
133. Transaction Definition EPI1 - General	175
134. Transaction Definition EPI1 - Transaction	176
135. Transaction Definition EPI1 - Context	177
136. Testing The Transaction Router from a Local Terminal	178
137. User Definition EPIUSER - General	179
138. User Definition EPIUSER - Security/DCE	179
139. Listener Definition for TCP/IP Clients	180
140. CICS Client Install Panel	181
141. CICS Client Components	181
142. CICS Client Installation Options	182
143. Configuring CICS Server in Definition File (Client)	182
144. Routed Transaction EPI1 from Client CICS Terminal	183
145. HTTPD.CNF Entries for CICS Internet Gateway	184
146. Configuration Page of the CICS Internet Gateway	185
147. Routed Transaction EPI1 From Web Browser	186
148. Registration Form for CICS Scenario	190
149. Registration Successful with CICS	191
150. Registration Form for CICS Error Scenario	192
151. Exception Macro for CICS	193
152. CICS User Error Page	194
153. MQSeries Scenario 1	228
154. MQSeries Scenario 2	229
155. System Environment for Net.Commerce and MQSeries Integration	230
156. MQSeries Installation Options Panel	232
157. MQSeries Remote Queue on NT03	233
158. MQSeries Local Transmission Queue on NT03	234
159. MQSeries Channel Initiator on NT03	234
160. MQSeries Sender Channel on NT03	234
161. MQSeries Local Queue on NT04	235

162.	MQSeries Application Initiation Process on NT04	235
163.	MQSeries Receiver Channel on NT04	235
164.	Registration Form for MQSeries Scenario	240
165.	Successful Registration with MQSeries	241
166.	Registration Form for MQSeries Error Scenario	242
167.	Exception Macro for MQSeries	243
168.	MQSeries User Error Page	243

---

## Tables

1.	Database Table Groups	52
2.	Component Installation (Net.Commerce Server, AIX)	83
3.	Net.Commerce Server - System Configuration Parameters, AIX	96
4.	Net.Commerce Server - Access Control Parameters, AIX	96
5.	Component Installation (Net.Commerce Server, NT)	106
6.	Component Installation (Database Server, NT)	108
7.	Net.Commerce Server - System Configuration Parameters	123
8.	Net.Commerce Server - Access Control Parameters	124
9.	Net.Commerce Server - Server Machine Parameters	124
10.	ITSO System Environment for Net.Commerce	271



---

## Preface

This redbook covers the planning and legacy system integration considerations for the IBM Net.Commerce product, Version 1.1, on the AIX and Windows NT platforms, and it provides examples of Net.Commerce implementations. To assist novice users, it offers a brief introduction to electronic commerce and the product itself as well as a section on the basic steps of product installation, configuration and usage.

This redbook helps you understand advanced aspects of planning for and designing an electronic shopping mall using the IBM Net.Commerce product. Since Net.Commerce is bundled with IBM Internet Connection Server, Version 4.1, and IBM Database Server, Version 2.1.2, planning and design issues related to those components are also discussed. Selected case study scenarios illustrate the addressed design questions.

An advanced customization section shows how to integrate Net.Commerce with legacy backend applications, such as CICS and MQSeries, using built-in Net.Commerce functions as well as customized macros and APIs. Last but not least, Net.Commerce commands, macros and database structures are also explained in this redbook.

---

## The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the Systems Management and Networking ITSO Center, Raleigh. The leaders of this project were Eamon Murphy and Martin W. Murhammer.

**Eamon Murphy** is Networking Systems Curriculum Manager in IBM Education and Training in the UK. Prior to this he was a Senior International Technical Support Specialist at the Systems Management and Networking ITSO Center, Raleigh. He graduated as a Bachelor of Science in Mathematics and Economics from the University of Sussex.

**Martin W. Murhammer** is an International Technical Support Specialist at the ITSO Raleigh Center. Before joining ITSO in 1997, he was a Systems Engineer in the Systems Service Center at IBM Austria. He has 12 years of experience in the personal computing environment, including areas such as heterogeneous connectivity, server design, system recovery, and Internet solutions. He is a Certified OS/2 Engineer and a Certified LAN Server Engineer and has previously co-authored six redbooks during residencies at the ITSO Raleigh and Austin Centers.

**Kurt Bergler** is a Solution Engineer in the Network Computing Solutions Unit in Vienna, Austria. He graduated as a Master of Science in Computer Engineering from the Technical University of Vienna. He has eight years experience in UNIX and TCP/IP-based networking, middleware architecture, Internet security and Web site administration. He has worked with IBM for four years.

**David Liu** is an Information Technology Specialist with IBM in Taiwan. In his home country, he is working in the commercial online services department assisting customers to develop electronic business solutions.

**Martha Carmina Muñoz Gomez** is a Network Computing Specialist in IBM Mexico. She has a Bachelor's degree in Management Information Systems at University of La Salle in Mexico City. She has two years of experience in Internet environment solutions, and is currently working with electronic business products and strategies.

**Maurice van der Sterren** is a Systems Engineer in the Application Development and Systems Integration business unit of IBM Netherlands. He has a degree in Industrial Design Engineering from the University of Delft. He is currently working for the Multimedia Business Solutions team developing electronic commerce solutions with customers in his home country.

**Pasquale Maurizio Coda** is working in the NC Research Center of IBM SEMEA Sud in Bari, Italy. He has been working for one year in a European project, developing an object-oriented persistence framework in C++ vs ODBC data sources. Since the beginning of 1997, he has developed a stronger interest in NC technologies, especially Java and the integration of new Internet applications and existing legacy applications using MQSeries and CICS.

Thanks to the following people for their invaluable contributions to this project:

Ray Hawkes  
IBM Education & Training, Cary

Dieter Wackerow, Shawn Walsh, Carla Sadtler, Rob McGregor,  
Linda Robinson, Dave Shogren  
International Technical Support Organization, Raleigh Center

Yumman Chan, Tapas Banerjee, Samantha Shurety  
IBM Net.Commerce Development Team, Toronto

Juha Nevalainen  
IBM Internet Center of Competence, Hursley

Nell Palistrant, Bruce Larson  
IBM Research Triangle Park

John Sweeney  
IBM UK

Paul St. Germain  
IBM Canada

Catherine Ezvan, Manuel Sanchez, Gerard Ducloux  
IBM France

Guy Denton  
IBM South Africa

---

## Comments Welcome

**Your comments are important to us!**

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:



- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 287 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Home Pages at the following URLs:

For Internet users                      <http://www.redbooks.ibm.com>

For IBM Intranet users                <http://w3.itso.ibm.com>

- Send us a note at the following address:

[redbook@vnet.ibm.com](mailto:redbook@vnet.ibm.com)



---

## Chapter 1. Electronic Commerce - The Fundamentals

Increasingly, the Internet is becoming an essential tool for our time. Students, business men and women, children, and housekeepers all use it in their daily lives. The technology, now more than ever, has given to the world the ability to access an enormous quantity of information from different organizations from any part of the world. The Internet is *the* new communication channel and arguably the easiest way to reach people. With the emergence of this phenomenon, all societies will have to re-think how to make the best use of this new medium in order to get from the Internet an added value for institutions, organizations, and companies.

Electronic commerce is one of the answers. One benefit that we can get from the Internet is to make it a primary profit source by creating a marketplace where companies will want to invest. The time has come to turn the Internet into a major business center for your company.

In this book we cover Net.Commerce, a product that finally makes it possible to engage millions of users, anytime, anywhere in the age old process of buying and selling goods, thus making the Internet truly the world's biggest marketplace.

---

### 1.1 General Concepts

After becoming aware of, and familiarizing themselves with the World Wide Web, companies generally pass through the following three phases: Presence, integration and commerce.

*Presence* means the way business information is made available to the Internet community, no matter what kind of business. Although a "wait-and-see" attitude usually surrounds this stage of a company's Web strategy, they quickly realize that they cannot ignore the almost 40 million (and growing daily) Internet users, and must let them know who they are and what they can do for them.

*Integration* is the next step, whereby companies can set up their internal systems such that they are compatible with the Internet. That is, their internal systems and Internet presence will have the same interface utilizing the facilities of a Web browser, and thus making useful information available to anyone inside or outside the company. Now, with Net.Commerce, integration is becoming even easier, allowing companies to not only establish their corporate image over the Internet, but also to sell merchandise directly throughout this new medium.

*Commerce* is the final, and perhaps most anticipated phase of the Internet phenomenon, and is the main goal of any company. The potential business value of the Internet has led to major changes in the way the Internet community is trying to reach their audience, which includes both clients and partners. Arguably, users of the WWW are the largest mass market in history, where millions of users all over the world, can be reached any time, any where, in a very easy way. For relatively small capital outlay, any company, through its Web presence, can engage in a dialog with the entire world, reaching out to potential customers, distributors, and business partners alike.

IBM has developed a set of products that will help you be part of this new marketplace in the WWW. Commerce.Point is the name of the strategy that has all different kinds of products for different business needs, depending on the infrastructure and the sector of the Internet a business wants to reach, be it selling to other businesses or directly to consumers, and all the while selling securely.

Companies that sell products or services to wholesalers and manufacturers need strong customer relationships, which often means secure communications as well as secure transactions. No need to let the competition know what you and your partners are up to.

Companies selling directly to consumers, on the other hand, may be more concerned with a Web presence that is more open and inviting, thus making customers feel as though they are receiving more personal attention. Some companies will achieve this by duplicating the shopping experience that consumers are already familiar with, for example, by creating an online virtual mall. Yet, although the virtual mall must be openly accessible to all, the customer must ultimately have enough confidence in the retailer to engage in electronic commerce. This paradox of accessibility and security must be overcome if the Internet is to realize its business potential.

There are different products and services that IBM has for each aspect/sector of electronic commerce, but there is one that can fit all of them, namely, Net.Commerce.

Net.Commerce allows you to be part of the electronic business world, from store setup or mall home page, to secure transactions with the newest methods and techniques of security on the Internet. You can access gateways that work with internal systems of the companies or gateways working directly with financial institutions for payment methods.

Net.Commerce is easy-to-use software that allows you to have online stores with dynamic pages. It works with the highest standards of security including SSL and SET, and works with DB2 in order to integrate existing systems. It is both flexible and scalable. It can be used for business-to-business transactions, as well as for direct consumer sales, in both cases providing state of the art transaction security.

In the following chapters we explain how Net.Commerce works, and what it can do for you, by working through some typical scenarios in which Net.Commerce can be used.

---

## 1.2 Net.Commerce Overview

Net.Commerce, the protagonist of this book, and arguably the best product for achieving this new marketplace goal, is a complete product that allows you to set up a store or multiple stores in a very easy way. It works together with a relational database and secure Web server in order to give users and companies a simple and secure environment to work with.

The following figure shows the complete Net.Commerce system.

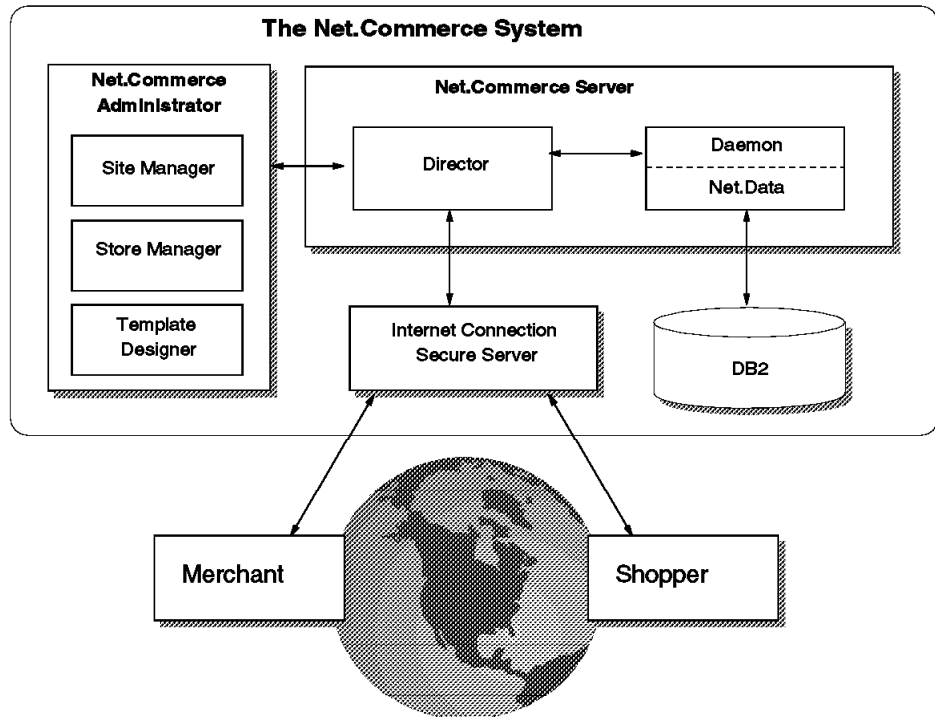


Figure 1. The Net.Commerce System

Net.Commerce as a server has two main components:

- Net.Commerce Server
- Net.Commerce Administrator

These two components together are the heart of the system. The Net.Commerce Server is composed of the *Net.Commerce Server Daemon* and the *Net.Commerce Server Director*. The first one is the daemon that enables communication with the database controlling the shopping process and executing macros that need information from the database. It supports the SSL protocol for security.

The second component, Net.Commerce Server Director, which works via Transmission Control Protocol/Internet Protocol (TCP/IP) connections, handles the communication between the Internet Connection Secure Server (ICSS) and the Net.Commerce Server daemon for quickly accessing the database. This is a Common Gateway Interface (CGI) program.

With the *Net.Commerce Administrator* you can easily create and maintain a shop or mall. With the *Net.Commerce Site Manager*, *Net.Commerce Store Manager* and *Net.Commerce Template Designer* you have the tools to manage your mall or shop. We briefly describe these parts later on.

Also, Net.Commerce needs to work together with ICSS and IBM Database 2 (DB2). They come with the product if you have the CD-ROM version, but you can also get them separately or maybe you are already working with them.

## 1.2.1 Net.Commerce Site Manager

As noted above, Net.Commerce Administrator has three main tools: Site Manager, Store Manager and Template Designer.

Net.Commerce Site Manager is the tool that allows merchants to create and manage an online store or multiple stores in a mall. With this tool you specify the location of the files for the mall's home page, the access control and store authorities, and the shopper information. You can also create a list of the shipping companies that will work with the system. These are the options you will find on the Site Manager home page configuration with a brief description of each:

<b><i>Mall Front:</i></b>	Sets up the mall home page and directory of stores, and specifies the various directories for the appropriate files.
<b><i>Store Records:</i></b>	Displays and keeps the store records form.
<b><i>Shopper Information:</i></b>	Displays and keeps the shopper information form.
<b><i>Template Designer:</i></b>	Creates and modifies templates for your mall home page.
<b><i>Access Control:</i></b>	Displays the access control form and the store authorities form.
<b><i>Task Management:</i></b>	Customizes mall, store or shopping tasks.
<b><i>Shipping Providers:</i></b>	Creates and maintains a list of shipping companies.

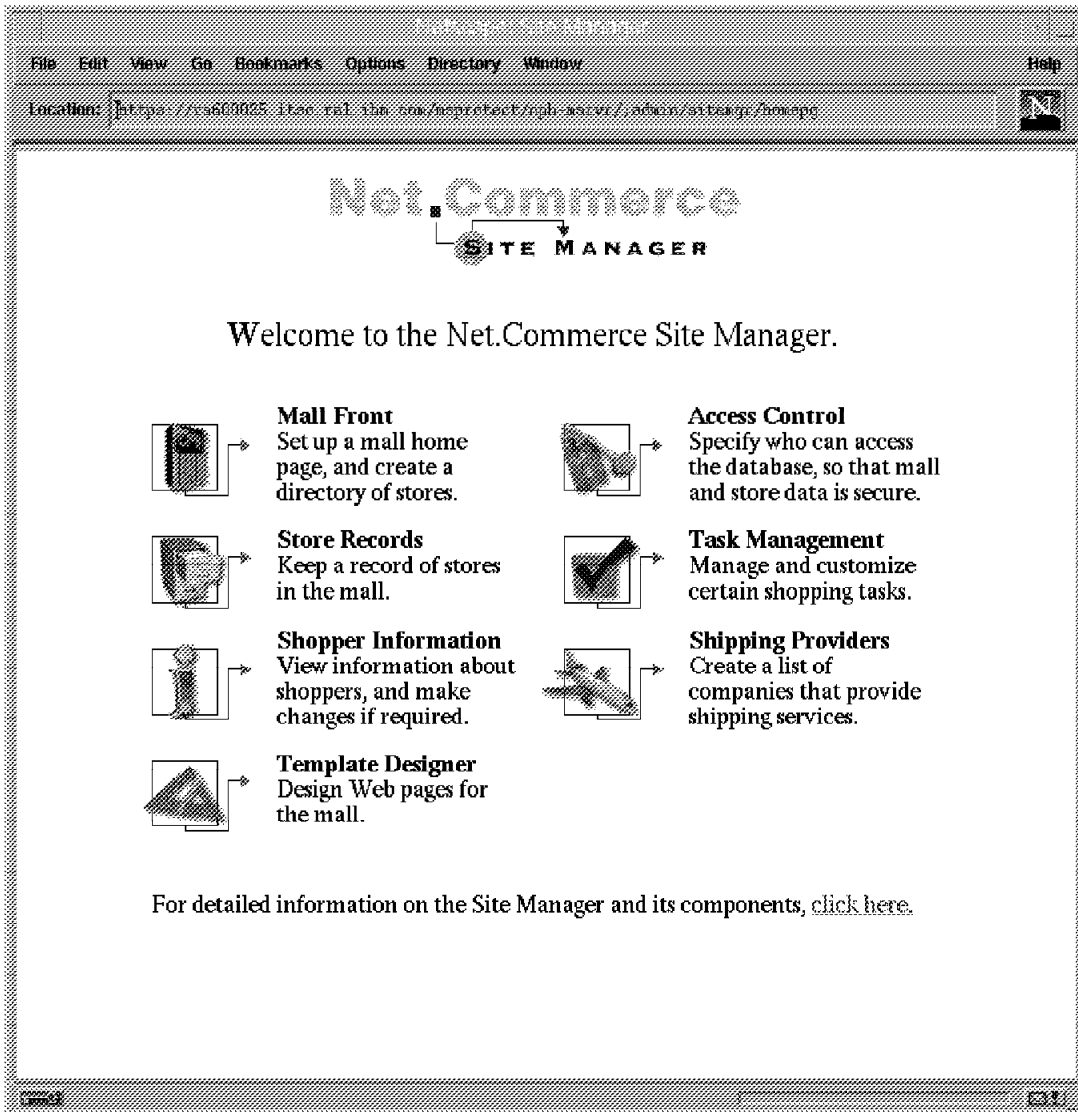


Figure 2. Net.Commerce Site Manager

## 1.2.2 Net.Commerce Store Manager

The Store Manager allows you to create and maintain the catalogs for the online stores. You can enter and update the store information and catalogs, create product categories, assign shopper groups, and view shopper information among other things. Here is a brief description of the options you will find on the Store Manager home page:

**Store Information:**

Displays the store information and allows you to enter or change data about your store.

**Product Information:**

Allows you to enter and maintain information about the products, services and items in the store.

**Shopper Groups:**

Allows you to create and modify groups of shoppers for your store.

**Shipping Services:**

Displays shipping codes and defines the way your products will be shipped.

- Product Categories:** Creates paths to navigate through your store.
- Template Designer:** Allows you to design Web pages for the store.
- Customer Information:** Displays all customer data.

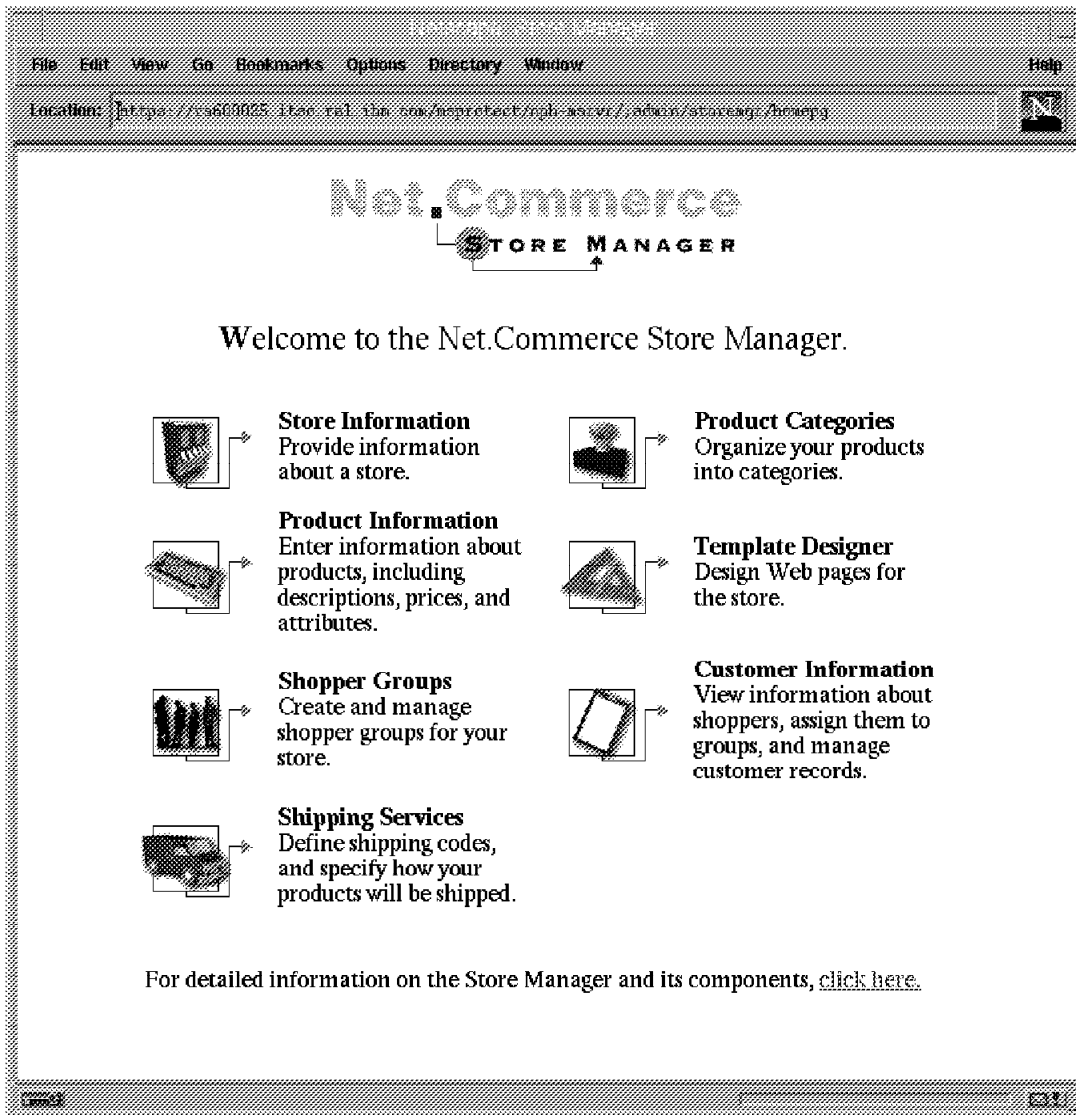


Figure 3. Net.Commerce Store Manager

### 1.2.3 Net.Commerce Template Designer

Finally, the Template Designer is an important tool that helps you to design and modify templates with a graphical HTML editor. Based on a Java applet, this tool has the utilities of a graphic and text editor, but is more than this. With the Template Designer, you can also create a store, mall, product, items and categories home pages. It works with a WYSIWYG (What You See Is What You Get) interface that translates your design into HTML. Also, you can customize the macros that work with SQL and HTML for accessing information from the database. In this way your Net.Commerce site will work with dynamic pages, so you can quickly and easily set up and maintain the mall or store on the World Wide Web.



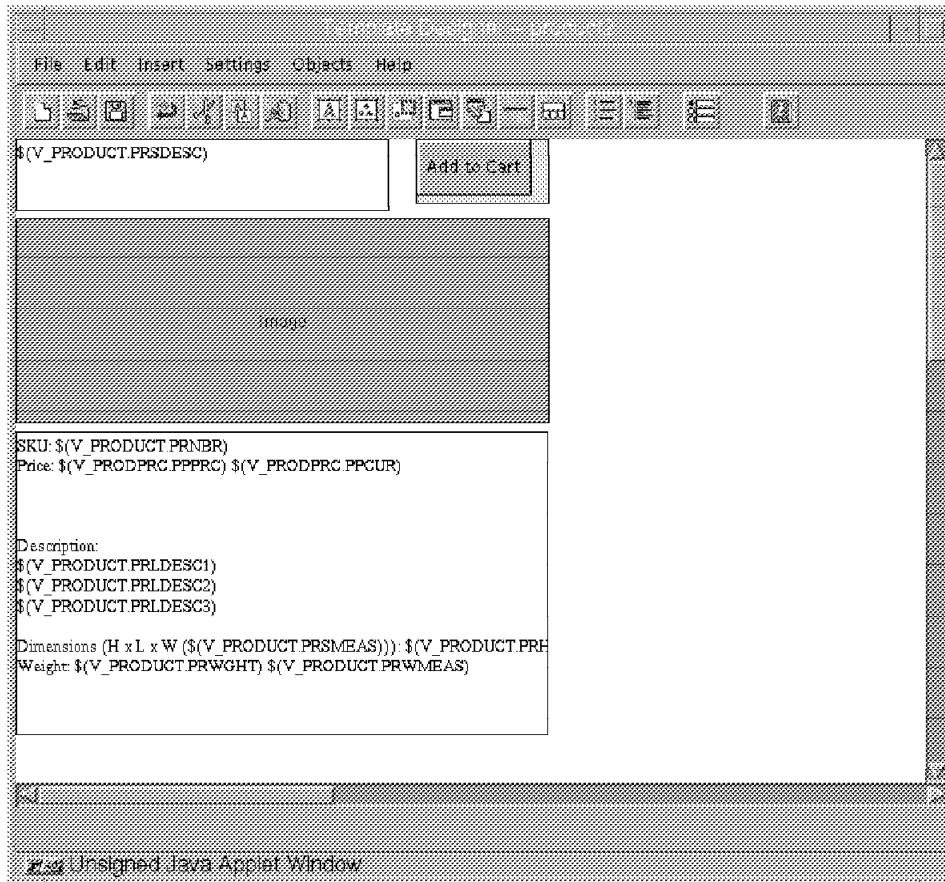


Figure 4. Template Designer

## 1.2.4 Related System Components

As mentioned before, DB2 and ICSS are intimately related to the performance of the Net.Commerce system which is shown in Figure 5 on page 8.

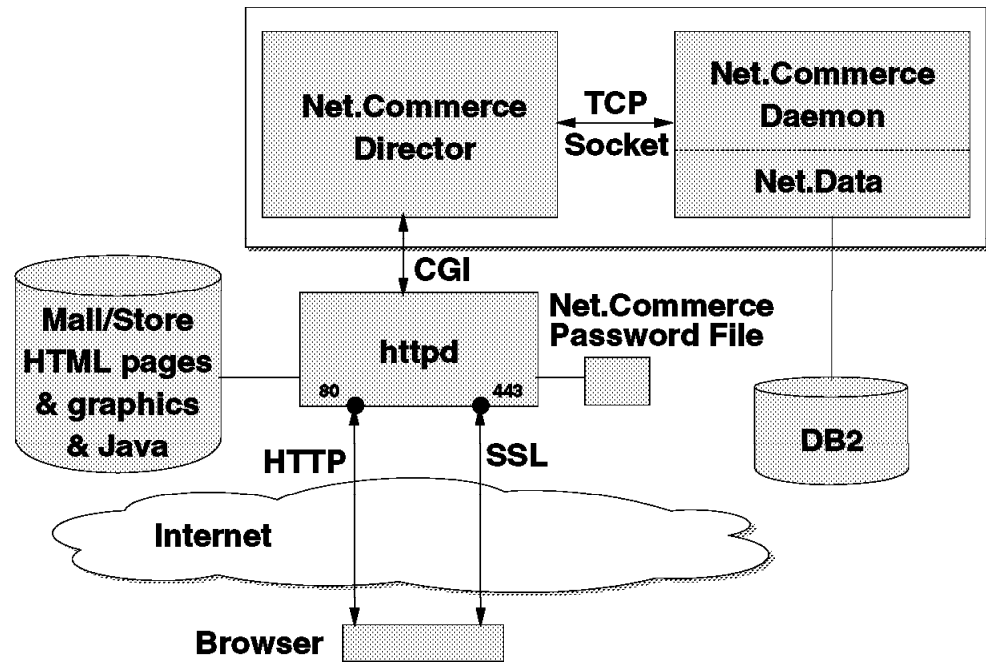


Figure 5. Product Relationships

#### 1.2.4.1 The Database

DB2 allows you to manage and keep the data you enter about the store and its merchandise. The DB2 data management system consists of a collection of tables that correspond to the forms in the Net.Commerce Administrator. Every time you fill in a form and save it, the information is stored and/or updated in the corresponding table in the database.

#### 1.2.4.2 The Web Server

The Internet Connection Secure Server (ICSS) is a very important part of your Net.Commerce system. It controls the flow of secure transactions using the SSL protocol in order to ensure that transactions are only performed by authorized users, which gives shoppers and merchants confidence in the transaction. Net.Commerce can also use other security systems for payments, for example, the SET protocol, which is available for even more secure transactions.

SSL is described in more detail in 3.5.4.2, "Secure Sockets Layer (SSL) and Secure HTTP (S-HTTP)" on page 69.

---

### 1.3 Electronic Payment Overview

Because security is so important, let's talk for a moment about payment systems and the SET protocol.

The entire process of an electronic payment transaction includes the browsing process, purchase process, payment authorization process and payment capture process which is defined as follows:

- The cardholder accesses an electronic store, browses the products and services and prepares an order.
- The cardholder will receive information about the order, as well as a description of the products or services, and the total price including shipping and handling.
- The purchaser must confirm this order and supply the payment information.
- The merchant will use the payment information to request an authorization and a capture order with the payment gateway.
- The payment gateway will send a request with the payment information to the cardholder's financial institution.
- The institution will send a response to the payment gateway and if the authorization is approved, the merchant will ship the goods or perform the services indicated in the order.
- The merchant will use the capture order to request payment from the payment gateway.

The capture process always occurs after the payment authorization. These processes will only occur if they are all totally secure and all members of the transaction have valid certificates. The appropriate responses are provided by the SET protocol.

SET has been designed to allow businesses to receive payment for goods and services in a safe, reliable, and consistent manner. The protocol works with the requirements that any business has for secure transactions: Confidentiality, authentication, integrity, interoperability. These requirements are addressed in SET using combinations of cryptographic techniques to address the business specifications of a secure transaction. A functional diagram of SET is shown in Figure 6 on page 10.

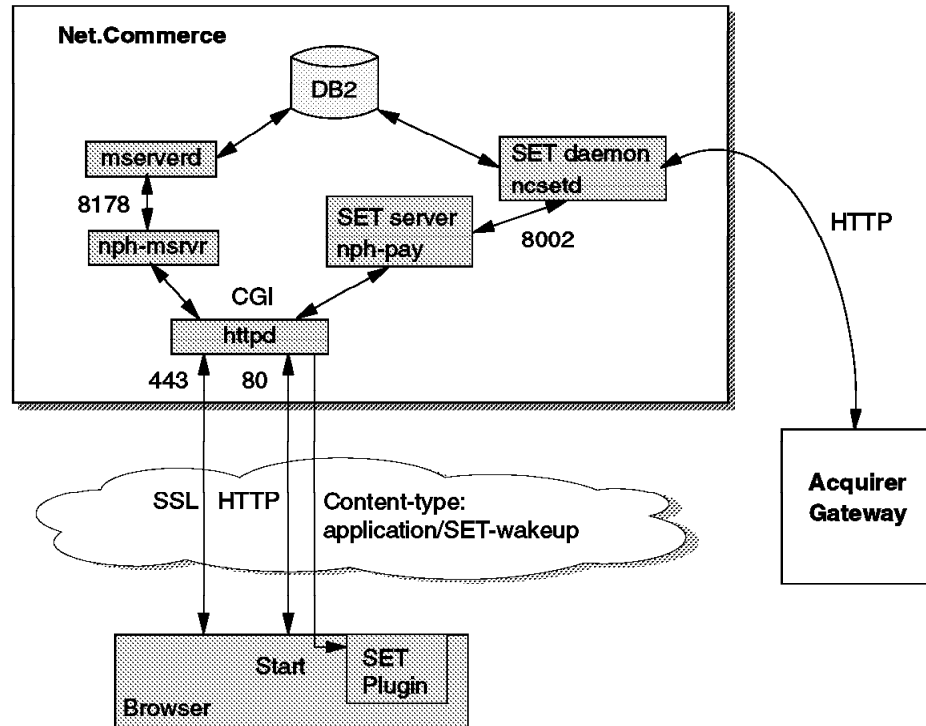


Figure 6. Secure Electronic Transactions (SET)

### 1.3.1 Net.Commerce Payment for Merchants

Payment for Merchants in the Net.Commerce system comes as an extension to the product and is available on the AIX and Windows NT platforms. Net.Commerce Payment for Merchants provides secure Internet transactions for the merchant, performing the critical role of handling SET transactions from the Internet. It serves in the combined roles of message management, encryption, certification and record keeping.

Net.Commerce Payment for Merchants:

- Acquires customer orders over the Internet.
- Forwards customer card information and certification to the bank.
- Supports all SET messages and processes.
- Manages SET messages at the merchant level.
- Provides cryptographic functions.
- Offers record keeping functions.

The integration of the Net.Commerce SET extension and its interaction with Net.Commerce is shown in the following diagram.

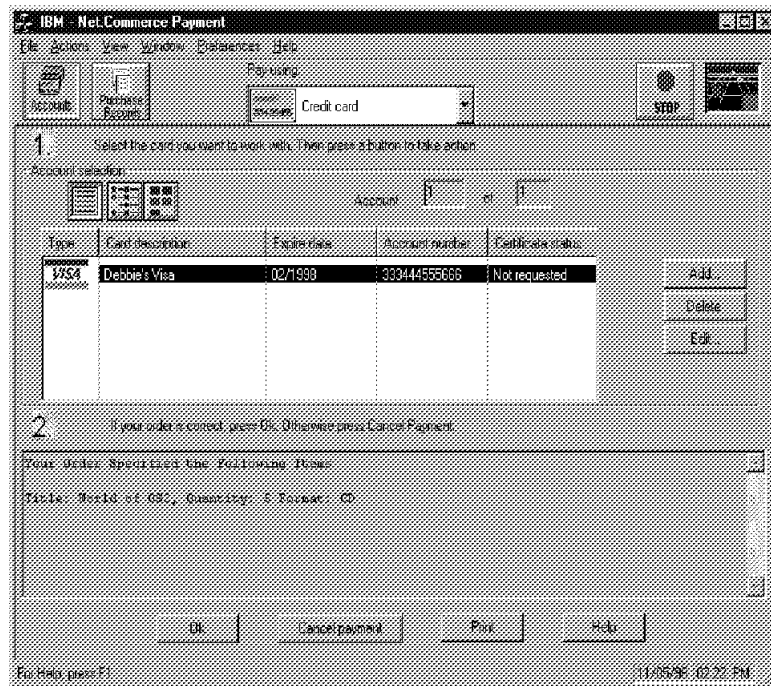


Figure 7. Net.Commerce Payment Interface

**Note:** Payment for Merchants will be a standard part of future versions of Net.Commerce.

For more information on the SET protocol and payment systems, see *Secure Electronic Transactions: Credit Card Payment on the Web in Theory and Practice*, SG24-4978.

## 1.4 Customization

Another feature of the Net.Commerce product is its ability to give your shop or mall a unique image. You can do this easily by customizing the macros and commands that come standard as part of the Net.Commerce product. Commands and macros are statements in each home page that allow the user's browser to interact with the Net.Commerce system. They are comprised of both HTML and SQL statements, which work with the Net.Data statements found in Net.Commerce itself.

### 1.4.1 Net.Commerce Commands

Net.Commerce commands are used to invoke the various mall and store functions for the shopping process. Commands are sent as part of a URL from the shopper's browser to the Net.Commerce system. They are interpreted as arguments to the `nph-msrvr` CGI program (the Net.Commerce server director).

There are two types of commands:

- *UI commands* (user interaction) cause a macro to be invoked which will retrieve information from the Net.Commerce database and send a page to the shopper's browser.

- *Non-UI commands* (no user interaction) process and write information to the Net.Commerce database. (They will then branch to a URL which might be a UI command.)

There are seven groups of commands for different mall functions:

- *display* - UI command used to display category or product templates (macros) based on a category or product number.
- *execmacro* - UI command to run a specific Net.Data macro. Some mall functions (for instance address book pages, search, mall directory) are invoked using this mechanism.
- *adrbk* - Non-UI commands to maintain address book data.
- *register* - Used to display and maintain shopper registration data.
- *shopcart* - Used to display and maintain a shopper's shopping cart.
- *shipto* - Used to display and update shipping addresses for an order.
- *order* - Used to display orders and obtain payment information.

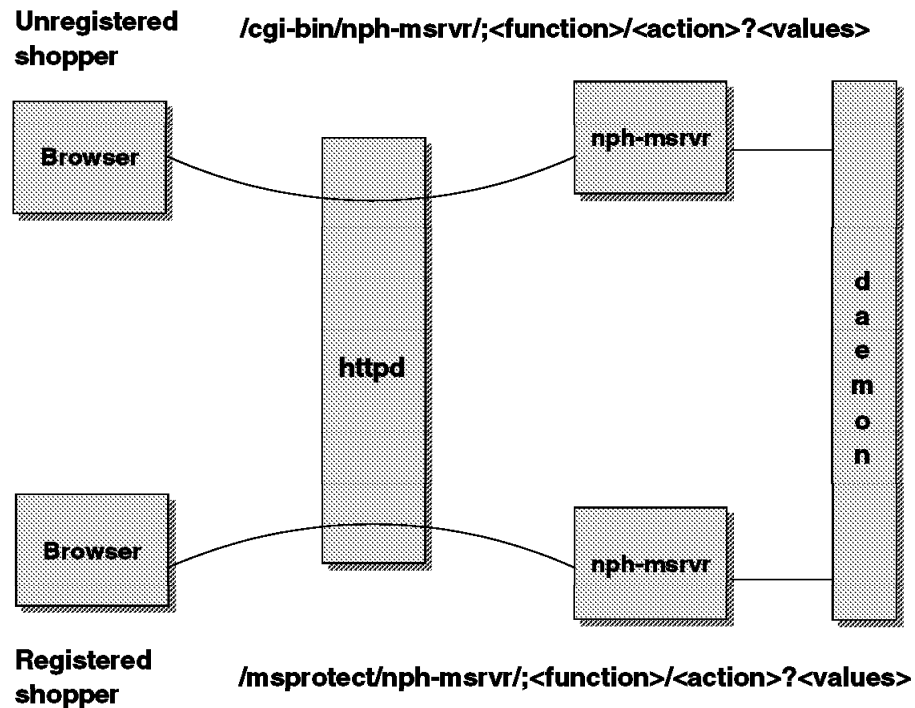


Figure 8. Net.Commerce Command Structure

The nph-msrvr CGI program can be invoked in two ways:

- From the `cgi-bin` directory as an unregistered shopper. A temporary `SESSION_ID` cookie is then assigned by the server and a row is created in the `SHOPPER` database table for that ID to keep track of the shopper's activities.
- From the `msprotect` directory requiring a shopper to log in with a valid user ID and password. The server sets the `SESSION_ID` cookie to the value of the user ID to track that (registered) user's activities.

For more information on the commands and macros for Net.Commerce customization, please see *IBM Net.Commerce: Customization Guide and Reference*, SC09-2370.

## 1.4.2 Net.Data Macros

Net.Commerce includes the Net.Data run-time libraries. The mserverd daemon uses these to execute macros for a range of mall and store tasks. Macro execution is performed as follows:

1. The macro file is loaded.
2. Input parameters are passed from HTML forms, command parameters and other global variables (for instance SESSION\_ID). These can be used in the macro to control processing.
3. Net.Data syntax controls macro flow, and one or more SQL statements are executed to retrieve data from the database.
4. The data is formatted as an HTML page and returned to the browser.

Simple Net.Data macros can be created with the Template Designer, such as those used for store home pages, category pages and product pages. More complex pages will often be created by customizing the samples supplied with Net.Commerce.

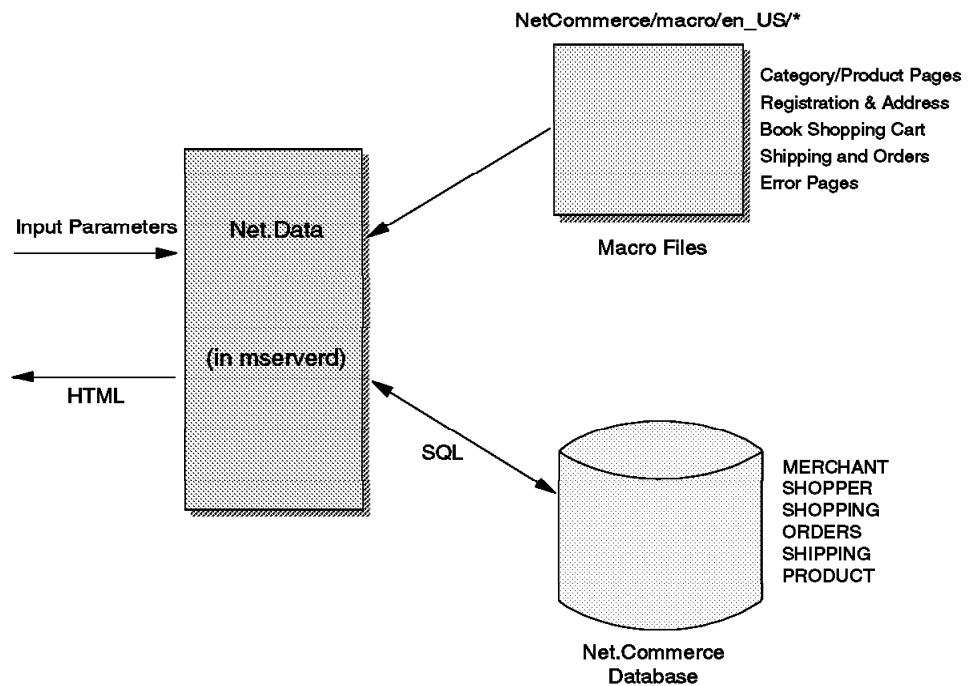


Figure 9. Net.Commerce Macro Structure

Basically, Net.Data macros can have four sections:

1. The %DEFINE section assigns variables used in the macro.
2. The %SQL section defines an SQL statement and the HTML markup used to display the results of the SQL query. Multiple %SQL sections can appear in the same macro.

3. The %HTML\_INPUT section represents a full HTML page containing Net.Data statements to assign and reference variables, control the flow and execute %SQL sections.
4. The %HTML\_REPORT section is similar to the %HTML\_INPUT section; those sections can be used as independent entry points into the macro.

For more information on the macros and commands for Net.Commerce customization, please see *IBM Net.Commerce: Customization Guide and Reference*, SC09-2370.

### 1.4.3 Net.Commerce Application Program Interface (API)

Net.Commerce APIs, similar to Net.Data macros, are called and executed as the resulting action of a Net.Commerce command. An API is an external dynamic link library (DLL) which has been created by compiling and linking a C or C++ program.

**Note:** If you have to develop your own APIs, you need the Microsoft Visual C++ compiler, Version 4.2 in order to produce DLLs that can be used with Net.Commerce on the Windows NT platform.

The APIS and MACROS database tables contain information about which API function or macro has to be used by a specific command. By customizing these tables, you can specify different APIs and macros to be executed for most commands. This is necessary if you want to use your customized macros or APIs to perform certain functions, such as order processing, inventory checks, or shipping.

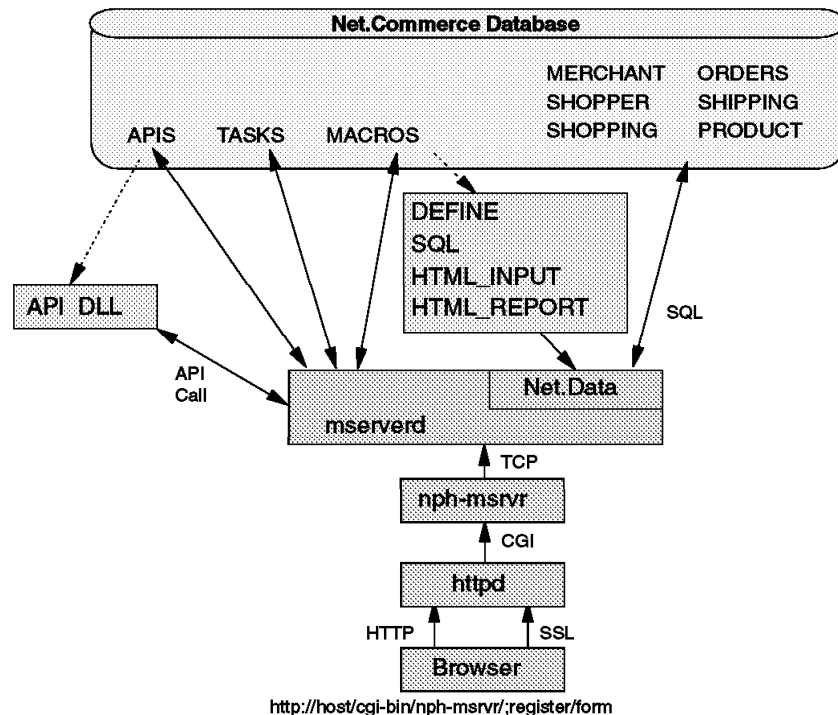


Figure 10. Net.Commerce System Structure



This soft copy for use by IBM employees only.

For more information on the APIs for Net.Commerce customization, please see *IBM Net.Commerce: Customization Guide and Reference*, SC09-2370.



---

## Chapter 2. Designing Your Individual Shopping Site

This chapter discusses some general topics that should be decided upon before establishing an electronic store.

---

### 2.1 Products and Their Classification

In order to set up an electronic store, the first thing you need to decide is which products you want to sell and how to classify them in categories.

There are some points to consider:

- Which products are suitable?
- Which group of shoppers do you address?
- How can you keep your product table up-to-date to gain an advantage over a paper catalog?
- Do you intend to offer specials, sales, etc.?
- Do you intend to provide cross references to related products, accessories, detailed descriptions, alternatives if sold out?
- How can you leverage shopping information, analyze data, react on typical behavior?
- How can you react on shoppers' demographic information (income, age, hobbies, etc.)?
- How can you make the ordering process easy, for first-time users as well as for experienced users?
- How can you give shoppers a reason to come back?
- How can you set your fulfillment system up?
- How can you integrate a virtual storefront to your legacy systems?

#### 2.1.1 Choosing the Right Products

Some products are more suitable for selling over the Internet than others. For most products, the shopping process on the Internet is different from the traditional shopping process of going to a shop, taking a product from the shelf, feeling it, reading the labels and possibly asking a sales person for advice. Most products that a consumer doesn't know about, he or she will first want to feel, try, or hear advice about from people he or she trusts, before buying. When buying a steak, you want to see it before you buy it: not much fat, right size, etc. When buying a drill, you want to feel whether it feels solid and fits your hands.

The consumer wants to be sure he or she has chosen the right product. A product that he or she has bought before, and likes, will probably be purchased again, possibly through your Internet shop. Generally, mass-produced pre-packaged products are more suitable than others, especially the well-known brands. A can of Coca Cola is a can of Coca Cola.

Why would a potential shopper choose your product over another product? This is an essential question to the success of your online store, especially if you plan on selling rather expensive products. So you might want to include product

reviews, forums or newsgroups where you respond to questions or problems from customers.

If you are selling sports-related goods, such as sneakers, shirts, etc., you could, for instance:

- Add a page with links to sites where news about sports is presented.
- Run the highlights of the latest sports results (by yourself or in cooperation with the source of the information).
- Add a newsgroup where people can ask questions about the products.
- Add a chat-environment, where people interested in sports could meet.

This way you attract people interested in sports to your shop that you can seduce with offers. You also create an environment that is more than just a company trying to sell things.

Information-intensive products are also suited for this medium. These products can be transmitted over the same channel as they have been ordered. The problem of the customer being able to copy or even resell the information is being tackled at the moment by technology such as Cryptolope. Cryptolope uses containers that secure and guarantee payment for intellectual property.

Market research is another critical factor to the success of your online store. Find out what the customers want, what your competition is offering (probably over the Internet as well) and what your added values could be over your competitors. No one is going to buy from you if they can get the same product elsewhere (not necessarily cheaper but in a more convenient way), unless you have something extra to offer. People want information on what to choose in order to be able to make a better choice. Therefore you might consider to offer customization options for individual product packages, information on how to better use your products, or offer entertainment as well.

### **2.1.2 Product Categories**

To create a shop where people can easily feel at home, make sure they can find the products they need in the places they expect them. But, contrary to a real life shop, you can put products in more than one place without losing expensive shop space.

Use multiple parents if you want to include the product in more categories. For example, you could include milk in the cold drinks category as well as in dairy products. That way, you can make sure that a product, in this case milk, will be found by customers who take different ways while browsing through your online store.

The figure below illustrates how the Net.Commerce Administrator can be used to set up multiple parent relationships.

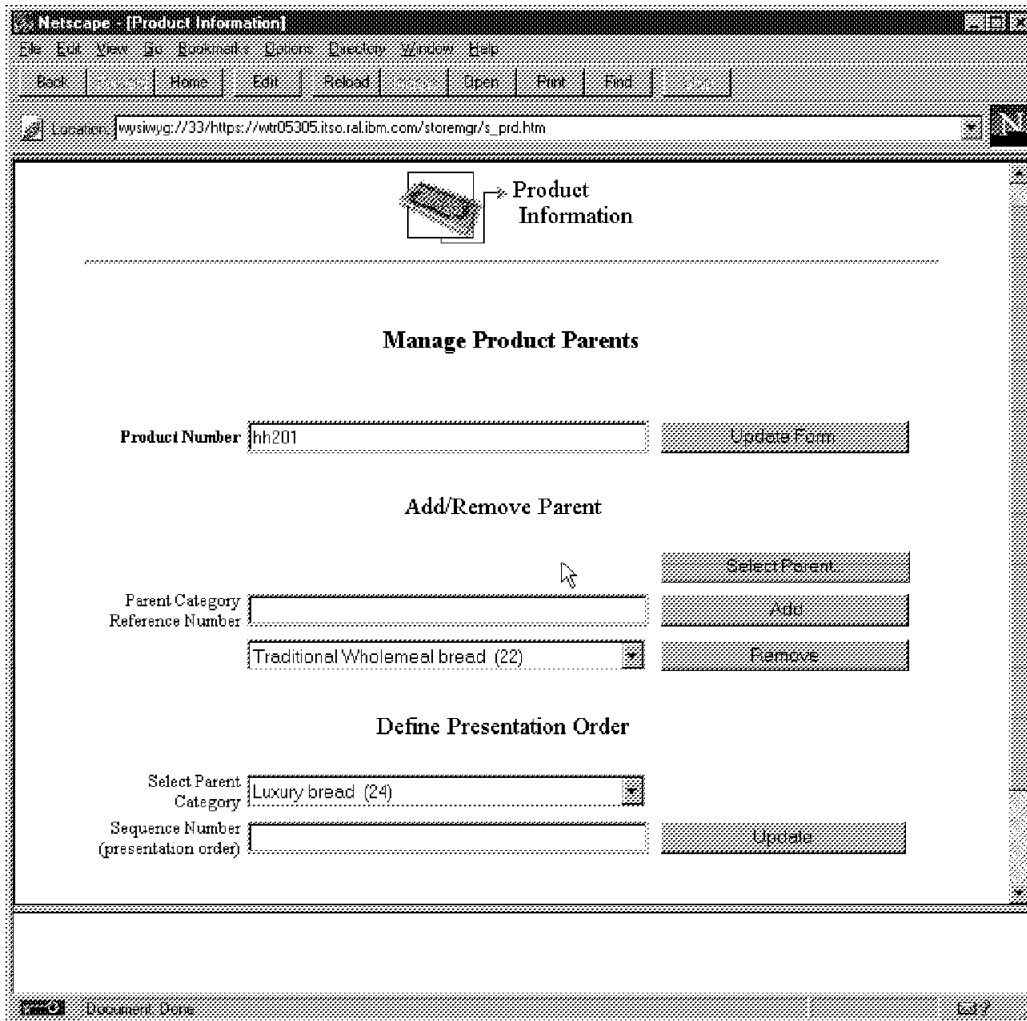


Figure 11. Configuring for Multiple Parents

Most experienced users know what they want, and they want to get there fast. Include a way for them to get to the products they have bought before, and related products. Especially on slow connections, having to browse multiple categories is not a good idea. Depending on the products you are selling customers accept more or less complicated shopping and order processes.

When shopping for groceries you don't want to browse two categories and one product page for every product you want to buy. For groceries you may just want a list in which you can check items and specify the amount you need.

But for the more expensive products, people are willing to invest time and effort in the process of finding the right product. Offer all information the shopper needs to make a decision, and build an easy but logical navigation structure around it.

Net.Commerce provides you with great flexibility to find and implement the appropriate design for products and categories and also for multiple variations of the same product (also referred to as attributes or items). Figure 12 on page 20 illustrates how such structures and relations are reflected in the Net.Commerce database:

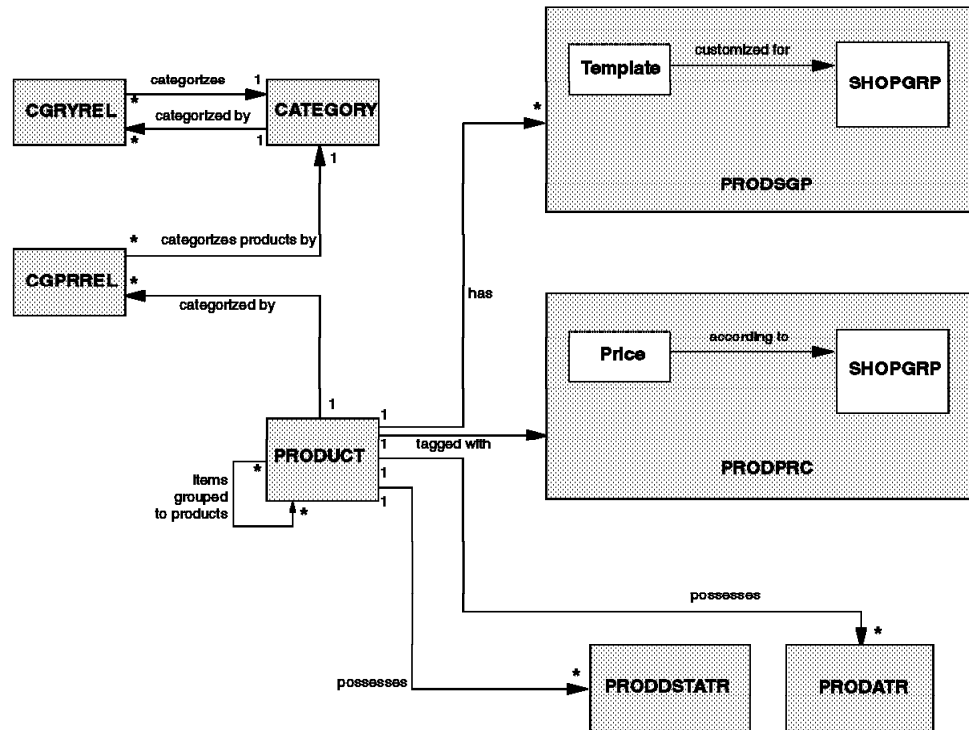


Figure 12. Database Tables for Categories, Products and Items

## 2.2 Graphical Page Layout

It is important to make a professional impression with your shop. To achieve that you need professional looking pages that fit the image you want to create for your shop. Just like in the real world, where you judge shops by the way the building, their staff, catalog and advertising look, your Web pages which substitute almost all those items should look good as well.

For selling products over the Internet, you need to find the right balance between "hip" pages that look nice but are not very functional and boring, business-like pages. Remember, you are selling products, so make it easy to order those products.

Let a professional graphic designer who is used to designing interactive media and who is aware of the image you want to create do the page layout for you. Shoppers browsing the net are used to nice pictures and well designed pages, and a nice design makes them come back more often. *Do not underestimate the graphical design of your Web pages!*

In general, consider the following things before designing a Web page:

## 2.2.1 Using Recognizable Styles

Try to find a basic layout that appears on all of your pages, and is unique for your company. This includes headers and footers, colors, fonts, location of images, etc.

If you decide to design your site yourself, you can design most macros (dynamic HTML pages that interact with the Net.Commerce database) using the Template Designer, provided by Net.Commerce. However, for designing professional looking HTML pages, the Template Designer contained in Net.Commerce 1.1 is not very well suited. You might be better off using dedicated HTML editors for this task. These programs offer more control over the layout of the page and also offer the latest HTML additions. Although some editors claim WYSIWYG control over the pages they create, it is still the browser that does the final formatting, so check the appearance of your pages regularly using popular browsers to avoid surprises.

**Note:** If you decide to edit the macros using a text editor, don't expect the Net.Commerce Template Designer to further edit the file. It is not built to interpret these changes.

## 2.2.2 Using Graphics and Colors

Do not design your pages like lists on a 3270 screen. But do not overload your pages with too many fancy pictures and animations that take a long time to load. Keep in mind that a lot of users still have slow connections to the Internet.

## 2.2.3 Using Templates

Use different layouts for different product categories. You certainly want to use another background picture or text font for winter coats than for summer swimsuits.

## 2.2.4 Using Headers and Footers

Create headers that include your logo, colors and slogans that you also use in other media and let shoppers know which store they are in at all times. You can use the footer for offering links to the shopping cart, the order pages, and back to the product and category pages. You can also use the footer line to add links to sales offerings or to general product categories that might be interesting for all shoppers (for example, candies or fancy mousepads).

You could also design a more complicated setup with frames, so you can show your shopcart all the time, or show category and product pages together. Shoppers have to recognize you when browsing through the net.

**Note:** Not all browsers support frames, so don't forget to include a notification of some sort on your pages when using advanced HTML features. You may also include pages without frames for the benefit of those users whose browsers cannot support them.

## 2.2.5 Using a Common Set of HTML Features

Try to use a subset of HTML that most browsers support. There are a lot of HTML extensions out there that can be used to produce fancy graphics, animations, etc., but which can only be viewed with the latest browsers or special browser plug-ins. You might deter shoppers from visiting your site if they are forced to change to another browser or obtain and install additional code for viewing. Use it only if that's the way you need to make the impression,

and only for "nice-to-have" features, not for essential information. Be aware of your target group. Are you targeting young people that like keeping up with the latest technology or are your customers people that look at browsers as a necessary evil.

## 2.2.6 Using Images and Image Formats

Perhaps the most important factor in the look of your catalog are the pictures. It is important to do this right.

Produce images (photos) of all the products you want to sell via the Internet. Do not underestimate this step. These pictures must be of good quality and contain just the product. Do not use photos of the whole family of products on all product pages. Likewise, don't use badly scanned images from brochures with some text across the picture.

Making pictures is expensive and time consuming. Try the manufacturer of the products you are selling for good images of all products. You will need these images in JPEG or GIF format, in order to include them in your shopping site. These two formats are basically different.

JPEG images can contain up to 16 million colors but they use "lossy" compression. The image degrades during compression, but the file size is just a fraction of the original. Some image-editing programs, such as PaintShop Pro (available as shareware on the Internet) or Adobe Photoshop allow you to specify the compression ratio. The higher the compression ratio, the lower the file size, but also the lower the quality of the image. Also, during the editing process, in some programs you lose quality every time you open a picture and save it as JPEG. Depending on the number of pictures, the bandwidth and the type of pictures you can optimize this. In color photos, even larger compression is hardly visible. You need to experiment to find the optimum.

For graphics using plain colors, JPEG images are less suitable because of the distortion that comes with the data compression. GIF images are better for these types of graphics. GIF images can contain only 256 colors and use only non-lossy data compression, so they can be fully reconstructed. Most programs allow reducing the number of colors to 128 or even less, in order to further decrease the file size. Here also you have to try to find the best solution.

GIF images allow for a "transparent color". Some editors allow you to designate one color to be transparent, so that the background (color or image) will shine through. This gives the picture a nice, integrated look. When designing your own graphics, use the anti-aliasing option that most editors have, to get the edges more fluent. When using a transparent background, anti-alias to the color that best describes your background.

You might omit images for self-explaining items, such as for example a memory extension for a PC, but in general your customers will prefer buying things they have seen in advance.



## 2.2.7 Using Different Layouts for Different Customers

If your shoppers have registered and have entered personal information such as age, gender, income, address, job, etc., you can use this data to show special sales offerings to them, or to use different page layouts.

---

## 2.3 Preliminary Aspects of Operation

Before you get serious about implementing an electronic store, there are other issues to consider apart from Web page design and product classification. Imagine yourself as a user who wants to shop at your site and try to find answers to the following questions:

- How will users learn about the existence of your store?
- What problems might a user encounter while shopping online?
- How will users like the shopping process you offer them?
- What are the users' expectations towards order fulfillment?

The following sections provide guidance on how to approach the aforementioned issues.

### 2.3.1 Offering a Shopping Cart Facility

The shopping cart contains products that the shopper bought before, or at least found interesting. Otherwise the shopper could use the search macro. In order to really make use of the search capabilities of Net.Commerce you may want to include additional information in the custom fields of the database or in the product attributes. A gift shop could include the target group so you can just specify buying something for your grandmother and the system will come up with ideas, or you could add some extra keywords on the use of a product, or the problems it solves.

### 2.3.2 Updating Your Contents

Consider adding headlines, including the latest news of your company or your products. It is very annoying to hear about a product and then not find anything about it on the Web site, or to find announcements for events that happened a month ago. Show on your pages the date of the last updates so that shoppers know that they have retrieved the latest information. They want to know what has changed since their last visit. Most shoppers like dynamic shops, as long as they still can find their way.

### 2.3.3 Keeping Your Links Accurate

If you remove information, don't forget to remove the dependent links to this information. Would you like it, if when you clicked on a link you got the following message?

```
Error 404  
Not found - file doesn't exist or is read protected  
[even tried multi]
```

You probably wouldn't. Avoid expired links! Within Net.Commerce the program only shows up-to-date links, but hardcoded pages such as the home page and additional HTML pages are static and prone to expiration. Especially be aware that pages with external links that are not under your control (links to manufacturers, user organizations, etc.) can change without notice.

### 2.3.4 Defining an Adequate Order Process

Organize the order process so it is easy. Guide the shopper unambiguously through the steps of order preparation and ordering by using easy-to-understand symbols and by avoiding unnecessary links to other pages. The user interface should accommodate first time users as well as experienced shoppers. For first time users it is important to be able to find the products they like in an easy way. The Net.Commerce catalog with its categories and products is quite suitable for this approach. If the speed of their Internet connection is appropriate, most new shoppers like to browse by images. Offering a text-only page speeds up shopping for shoppers who don't need images to make their selections.

### 2.3.5 Defining an Adequate Fulfillment Process

Choose the most appropriate system to get your products to the customer. People expect pizzas within 30 minutes, but a \$3000 computer can be delivered some days after ordering. In some cases you can use an existing parcel service, sometimes you have to set up your own. Being faster than the competition can be a selling point as well. In some conditions you might even want customers to pick up the goods themselves after ordering through your system.

### 2.3.6 Some Words on Publicity

Let potential customers know you are there. Perhaps the easiest forgotten step is letting the people know you exist. Use part of your budget for promoting your shop.

- Advertise in your existing documentation and packaging.
- Advertise on radio or TV.
- Have clickable ads on popular (among potential customers) Web sites.
- See your suppliers for means of advertising.
- Look for free sources of advertising. If you have something special to offer, some radio, TV, newspapers or magazines may be interested in writing about you. Advertising in newsgroups and unsolicited e-mail is *not* a good idea unless you have something *really* good to offer.

---

## 2.4 The Anatomy of a Net.Commerce Shopping Trip

You have to decide on the design of the shopping trip that your shoppers will be confronted with. This can be compared to the arrangement of aisles inside a big supermarket, where you will try to guide your customers through most of the departments.

Make the shopping trip exciting, uncomplicated and straightforward for the shoppers. Assume that shoppers are not used to the Internet, nor to buying goods via the World Wide Web. They need to be guided carefully, and must not be confronted with unnecessary complex tasks or technical terms. KISS - Keep it short and simple.

Take into account the dynamic nature of the Internet and of Net.Commerce. Think about putting specials and sales offerings directly on your home page or from your footers. If you have collected demographic data of your shopper, you can use this to design pages individually, according to age, gender or income. You may also include cross-references to detailed product descriptions or to

product evaluations. You might also offer optional add-on products or accessories, and add links to alternative products when you are out of stock for a specific item.

For small shops especially, the shopping process needs to be easy, short and clear. This is true not only for the categories, but also for the ordering process. If it is common for customers to just buy one product for themselves and then leave the store, they don't need other options such as an address book. Simply offer them a button to jump from the product page to the page where they can enter their payment information. (They must be known to the system before they can place an order.)

If most customers don't need the information you are displaying on the product page, you may even offer a shortcut button on the category page. As an example, you can do this by entering the default information into the SHIPTO/UPDATE macro and display the ORDER/DISPLAY page using the ;order/display?status=P command, as shown below:

```
%{---
Net.Commerce Product Template for products with no items

Offers 2 shopping links...
- Prepare Order - Adds item to shopping cart and goes to ;shipto/list
- Quick Order - Adds item to SHIPTO table and goes to ;order/display?status=P
  (Only available for registered shoppers)
---%}

%SQL(HeaderSQL){
  select methead,metfoot from merchant where merfnbr=$(prmenbr)
  %SQL_REPORT{
    @DTW_assign( MerchantHeader, V_methead )
    @DTW_assign( MerchantFooter, V_metfoot )
  }
  %SQL_MESSAGE{100:{ %} :continue %}
%}

%SQL(ProductFullSQL){
  select PRODUCT.PRFULL
  from PRODUCT
  where prrfnbr=$(prrfnbr) and prmenbr=$(prmenbr)
  %SQL_REPORT{
    
  }
  %SQL_MESSAGE{100:{ %} :continue %}
%}

%define GroupNumber="-1"
%SQL(GroupNumberSQL){
  select mcsgnbr from mcstinfo , shopper
  where mcshnbr=shrfnbr and shlogid='$(SESSION_ID)' and mcmenbr=$(prmenbr)
  %SQL_REPORT{
    @DTW_assign( GroupNumber, V_mcsgnbr )
  }
  %SQL_MESSAGE{100:{ %} :continue %}
%}

%define ShopperAddress="-1" 1
%SQL(ShopperAddressSQL){
  select sarfnbr from shaddr , shopper
  where sashnbr=shrfnbr and saadrflg='P'
  and shlogid='$(SESSION_ID)' and sanick=shlogid
  %SQL_REPORT{
    @DTW_assign( ShopperAddress, V_sarfnbr )
  }
  %SQL_MESSAGE{100:{ %} :continue %}
%}
```

```

%define MShipMode="-1" 2
%SQL(MShipModeSQL){
  select distinct spmmnbr from shipping, product
  where prpsnbr=spsmnr and prrfnbr=$(prrfnbr)
  %SQL_REPORT{
    @DTW_assign( MShipMode, V_spmmnbr )
  }
  %SQL_MESSAGE{100:{ %} :continue %}
}

%SQL(ProductText1SQL){
  select PRODUCT.PRSDESC, PRODUCT.PRNBR, PRODPRC.PPCUR,
  formatcur( PPPRC, PPCUR, char('N') ) as PPPRC
  from PRODUCT, PRODPRC
  where PPPRNR=PRRFNBR and prrfnbr=$(prrfnbr) and prmenbr=$(prmenbr)
  AND ppmenbr=$(prmenbr) AND ppprc is not NULL
  AND (ppsgnbr is NULL OR ppsgnbr=$(GroupNumber))
  AND (ppprnbr=$(prrfnbr) OR ppprnbr=
    (SELECT prprfnbr
     FROM product
     WHERE prmenbr=$(prmenbr) AND prrfnbr=$(prrfnbr)
    )
  )
  AND (ppdeffs is NULL OR ppdeffs<CURRENT DATE OR
    (ppdeffs=CURRENT DATE AND (ppteffs is NULL OR ppteffs<=CURRENT TIME))
  )
  AND (ppdefff is NULL OR ppdefff>CURRENT DATE OR
    (ppdefff=CURRENT DATE AND (ppteffs is NULL OR ppteffs>=CURRENT TIME))
  )
  ORDER BY pppre DESC, ppsgnbr, ppprnrbr DESC

  %SQL_REPORT{
    %ROW{
      %if( ROW_NUM == "1" )
      <H3>$(V_PRSDESC)</H3>
      <P>Product Number: $(V_PRNBR)
      <P>Price: $(V_PPCUR)$(V_PPPRC)
      %endif
    }
    %SQL_MESSAGE{100:{ %} :continue %}
  }

  %HTML_REPORT{
  <HTML><head>
  <TITLE>Home Computing Product</TITLE>
  </HEAD>
  <BODY BGCOLOR=#ffffff TEXT=#000000 LINK=#0000ff VLINK=#ff00ff ALINK=#ff0000>

  %EXEC_SQL(HeaderSQL)
  %if ( MerchantHeader != "" )
  %include "$(MerchantHeader)"
  %endif

  <CENTER>
  <table width=100%>
  <tr><td rowspan=2 colspan=2 width=66%>
  %EXEC_SQL(ProductFullSQL)
  </td><td width=34%>
  %EXEC_SQL(GroupNumberSQL)
  %EXEC_SQL(ProductText1SQL)
  </td></tr>
  <tr><td width=34%>
  <FORM ACTION="/cgi-bin/nph-msrvr/shopcart/add" METHOD="POST">
  <INPUT TYPE=HIDDEN NAME="merchant_rn" VALUE="$(prmenbr)">
  <INPUT TYPE=HIDDEN NAME="product_rn" VALUE="$(prrfnbr)">
  <INPUT TYPE=HIDDEN NAME="ur1" VALUE="/cgi-bin/nph-msrvr/
  shipto/list?merchant_rn=$(prmenbr)">
  <INPUT TYPE="submit" VALUE="Prepare Order">
  </FORM>
  (Specify shipping address and quantity)
  </td></tr>
  <tr><td colspan=3 width=100%><hr>
  %EXEC_SQL(ShopperAddressSQL) 3
  %EXEC_SQL(MShipModeSQL)

```

```
%if ( ShopperAddress != "-1" ) 4
  <FORM ACTION="/cgi-bin/nph-msrvr/shipto/update" METHOD="POST">
  <INPUT TYPE="HIDDEN" NAME="addr_rn" VALUE="$(ShopperAddress)">
  <INPUT TYPE="HIDDEN" NAME="product_rn" VALUE="$(prrfnbr)">
  <INPUT TYPE="HIDDEN" NAME="shipmode_rn" VALUE="$(MShipMode)">
  Quick order. Just specify the number of products you need and we'll ship
  it to your door as fast as possible.<br>
  Number: <INPUT TYPE="text" NAME="quantity" VALUE=1 SIZE=3 MAXLENGTH=3>
  <INPUT TYPE="HIDDEN" NAME="url" VALUE="/cgi-bin/nph-msrvr/
  order/display?status=P&merchant_rn=$(prmenbr)">
  <INPUT TYPE="submit" VALUE="Quick Order">
  </FORM>
%endif
</td></tr>
</table>
</CENTER>

%if ( MerchantFooter != "" )
  %include "$(MerchantFooter)"
%endif
</BODY>
</HTML>
%}
```

The following is an explanation to the points highlighted in the macro example above:

- 1** Selecting shopper address
- 2** Selecting shipmode
- 3** Triggering the selection queries
- 4** Building the form with the Quick order button

People having different wants and needs should also be able to order, so offer a complete solution (if you want to use it) as well.

Even for big shops or malls the order process should be kept simple. Because the shop is more complex and there are more options, this may be harder to achieve. As a general rule, try to adapt the order process to the shopper's behavior and not to the shop's organizational structure.

If typical shoppers of your store buy several products during one shopping trip, or if they usually want their orders shipped to more than one address, then make this easy for them. If they buy more than one product from the same category then link the "add to shopcart/shipto" button straight back to the category page. To do so, pass the category number into the product macro, and use it in the URL section of the HTML form around the "add" button. If a product uses accessories then route the shopper to a page where these are offered.

To shorten the order process, have a good look at the database structure and table relationships for the ordering process which is shown in Figure 13 on page 28.

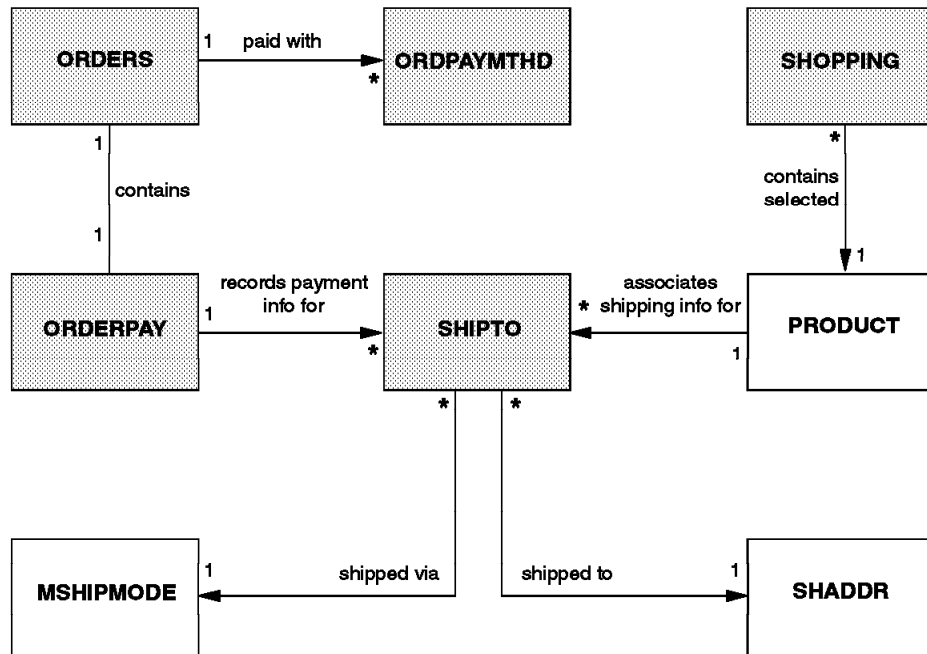


Figure 13. Database Tables for the Ordering Process

When a shopper puts a product in his or her shopping cart, a record for this will be created in the SHOPPING table. When the shopper issues a SHIPTO command, an entry in the SHIPTO table will be made. At the moment of ordering, a shipto address (SHADDR table), a shipmode (MSHIPMODE table) and an amount must be added.

When a customer is placing an order, a row in the ORDPAYMTHD table will be made with the credit card details; a row in the ORDERPAY table will contain the tax data; and a row in the ORDERS table will be made containing the totals. The ORDERS table also contains the ORLOCK column to lock orders.

In general, a Net.Commerce shopping process consists of seven tasks:

- Looking at mall and store home pages
- Catalog browsing
- Adding products to the shopping cart
- Registration
- Filling out an address book entry
- Providing shipping information
- Ordering and payment

## 2.4.1 Home Page

This is the first thing a shopper sees when surfing to your shopping site. Besides using graphical elements to lure the shopper into your store, you may consider putting special sales offerings on this page.

Don't offer too many links to the shoppers. Simply try to get them into the store.

Let the shoppers into your store without registering in advance. A registration process might be a barrier for shoppers and might cause them to leave without even having seen your products. Place the registration process toward the end of the shopping trip. Of course, as long as you don't know who your shoppers are, you won't be able to present special offers to them, so give registered shoppers the opportunity to log on.

In the default shopping process this home page is a flat HTML file. For easier maintenance, a dynamic page would be preferable for stores with frequently changing products that still want to make offers from the home page.

The URL of the dynamic home page is quite complicated. You can get around this by adding a pass statement in the Web server configuration file to point from the short name of your page to the dynamic name.

```
Pass          /mall.html
http://wtr05267.itso.ral.ibm.com/cgi-bin/nph-msrvr/;execmacro/mall_dir.d2w/report
```

Here's another way to create a static HTML introduction page from where you transfer shoppers to the dynamic page.

```
<html>
<META HTTP-EQUIV="Refresh" CONTENT="5; 1
  URL=http://wtr05305.itso.ral.ibm.com/cgi-bin/nph-msrvr/;
  execmacro/mall_dir.d2w/report">
<head>
<title>Welcome</title>
</head>
<body bgcolor="#FFFFFF" LINK="#009C00" VLINK="#FF0000" ALINK="#FFFF00">
<p>
<center>
<a href="http://wtr05305.itso.ral.ibm.com/cgi-bin/nph-msrvr/;
  execmacro/mall_dir.d2w/report">
</a> 2
</center>
</body>
</html>
```

The following is an explanation to the points highlighted in the HTML example above:

**1** This tells your browser to refresh this page after 5 seconds with the mall page.

**2** The clickable image gives the customer the opportunity to enter the mall earlier.

Make sure you include a nice picture but at the same time make it quick enough so it's not annoying.

## 2.4.2 Catalog Browsing

The catalog offers access to product categories, subcategories and products. It consists of two sets of pages:

- The category pages link shoppers to the groups of products or services available in the store. You can compare it to a table of contents in a paper catalog or signs in a real store. They have to include category title and images, descriptions, and of course hyperlinks to subcategories or product pages.
- The product pages include descriptions, attributes, images and the price of the products. They might also contain a link to related products or to detailed product evaluations. Additionally, the product pages should include a link for adding the product to the shopping cart.

The following figure shows a simple example of the hierarchical structure of category and product pages. This is one branch of the demo mall that is included with Net.Commerce V1.1.

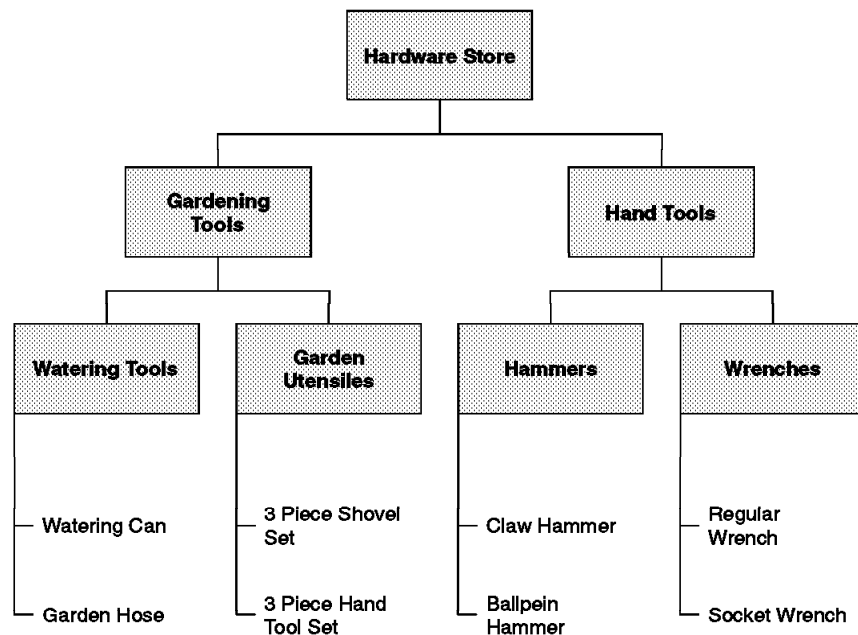


Figure 14. A Simple Set of Category and Product Pages

Catalog browsing works much the same way as browsing a paper catalog, but the electronic version has the big advantage of placing hyperlinks to totally different pages.



### 2.4.3 Shopping Cart

The shopping cart or basket displays the items that the shopper potentially wants to order. Look at it as a tool for the shoppers to gather interesting products, not as a necessary step in the shopping process. It should be there for the shoppers that need it, and optional for the people that don't need it. It should not be in the typical order process.

Items remain in the cart until they are removed by the shopper. For registered users, the shopping basket is not only valid for the current session, but also for all subsequent sessions. Anonymous users lose their shopping cart information after leaving the store. You can select products and register later during your session without losing your products, though.

You also have to decide on the scope of the shopping cart. The shopping cart may be valid for the whole mall or only for one store in the mall.

### 2.4.4 Registration

When shoppers register, they get a unique user ID and password, which can be used for subsequent visits to the store.

The merchant decides when the customer should register. As stated before, you may want to wait to ask the customer to register until he or she is close to really buying goods. Registration at this time is more natural than asking for identification when entering the shop. The customer has to specify a shipping address anyway.

However, some functions will not be possible until a shopper is registered:

- Maintaining an address book
- Keeping items in the shopping cart
- Being a member of a shopper group (and thus getting discounts)

For merchants it can be an advantage to know who they are dealing with. Take this into account when deciding on the shopping trip anatomy. Try to make it attractive for customers to register, but leave the option for shopping as an anonymous user.

The default registration page contains many fields. You may want to leave some data out, the demographic data for example, and concentrate on the necessary data for delivering the product to the customer. This way the registration form looks less overwhelming.

### 2.4.5 Address Book

The address book is quite a useful feature if shoppers want to order items not only for themselves but also for others, such as friends or family members. It offers the possibility to enter those friends and relatives into the database, refer to them via a short nickname and buy products for them and have the products shipped straight to them.

It is up to you to decide (or to discuss with whoever you are planning an electronic shopping site for), whether it should be possible to ship products to other addresses than that of the person who places orders, and whether shoppers can use the address book or have to enter shipping addresses manually.

Having the ability to maintain an address book makes the shopping process flexible but might confuse some people.

## 2.4.6 Shipping Information

Shipping information consists of three parts:

- Entering the shipping address (from the address book or manually)
- Entering the items from the shopping cart to be shipped to this address
- Selecting the shipping provider

Once again, you have to decide which options would be appropriate for the type of store that you want to set up on the Internet. A small store might easily suffice without an address book and with just one shipping provider for all orders. A big department store selling products from a variety of manufacturers, or a large mall hosting a number of individual stores might certainly need the ability to choose from a range of shipping providers.

## 2.4.7 Ordering and Payment

The last steps after assigning shipping information to the selected products, is to confirm the order and to initiate the payment process.

The shopper is presented with an order form listing the selected products, prices and recipients. In addition, the payment method must be selected now. The standard procedure is to use credit card payment and to invoke the SET-based payment process. The SET protocol provides you with an almost certain way of identifying the shopper. The credit card payment method does not. Anyone who has access to credit card details (a waiter in a restaurant, someone at the store where you bought something, or even between you and an old fashioned non-secure Internet shop) could enter these details and order merchandise. So you might want to check these before you ship the products.

Alternatively, you can choose any other payment method which can be invoked by calling user-defined functions or programs from within the Net.Commerce macros. Think of CyberCash, DigiCash, First Virtual and others.

You will find the complete order information in the Net.Commerce database for further processing.

## 2.4.8 Default Shopping Trip

The shopping trip that is provided by Net.Commerce in the demonstration shopping mall is outlined in Figure 15 on page 33. Depending on the store you're building you may have to modify it.

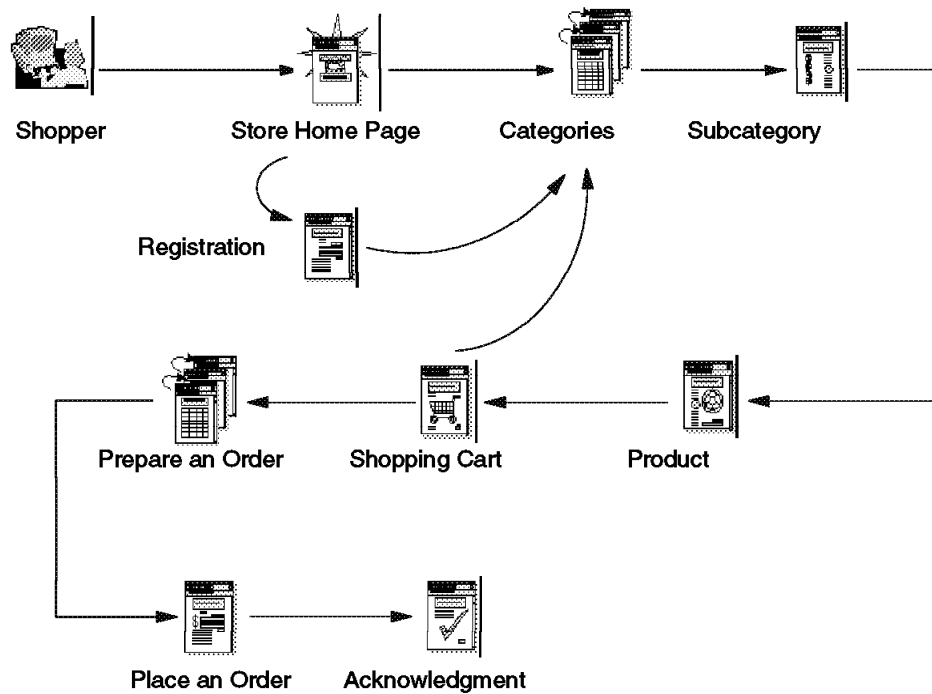


Figure 15. Net.Commerce Default Shopping Trip

Once you have designed your shopping site, including page layout and product pages, it's a good idea to perform usability tests on the site prior to putting it out to the Web.

**Note:** Version 1.1 of Net.Commerce does not provide the function of a *staging server*, a server acting as a test site for the stores and malls you are creating, to check out your shopping sites before going online. Therefore you have to be cautious and take your system offline while updating your site. Net.Commerce Version 2 will have staging server capabilities.

---

## 2.5 Post-Order Processes

Order fulfillment is an individual process that varies from company to company. Typically, the following tasks have to be dealt with:

- Accounting
- Tax calculation
- Shipping
- Inventory

What Net.Commerce does for you is put all necessary information into the database. It is your responsibility to use this information. At the moment there is no tool provided by Net.Commerce for viewing that data, so you have to choose another way for further processing.

Whichever way you decide to go from there is largely dependent on whether or not you already have backend systems in place to take care of the fulfillment

processes, and what interfaces those systems provide to exchange information with Net.Commerce. The list below provides some examples:

- Use tie-in applications such as Taxware or Prologistics.Merchant.
- Save output in a flat file to be picked up by other processes or batch programs.
- Send extracted information as an e-mail.
- Send extracted information as a fax.
- Convert data to Electronic Data Interchange format (EDI) which is sometimes used for communications between existing merchant applications.
- Export information to other database systems (for instance Lotus Notes) or to DB2 systems on other platforms.
- Develop your own APIs to send information online to existing legacy applications (for instance CICS or SAP R/3).
- Develop your own APIs to send information to legacy systems using MQSeries.

Most of these options rely on the translation of the information from one format to another, even to flat files. This operation can be done online or as a batch process, depending on the fulfillment system you choose.

---

## 2.6 Possible Store Scenarios

To give you an idea in what kind of situations Net.Commerce could be used, and how it could be used, let's take a look at the following four scenarios:

1. Small single store
2. Large single store
3. Small mall
4. Large mall

### 2.6.1 Small Single Store

A scenario like that might consist of the following:

- 1 shop
- 2 main categories
- 20 subcategories
- 100 products

Small stores are unlikely to have specifications of all of their products in a database.

A small store can be maintained using Net.Commerce's built-in editing tools. Depending on the products that the merchant sells, editing will be needed more or less frequently.

A single system containing both Web server and database will be sufficient in most cases. The server could be run locally or be placed in a server farm with other servers hosted by third parties. This machine does not need to be a dedicated machine. The administrator of the system will have other tasks to

perform so a fax or e-mail system to the merchant or even the manufacturer or shipping provider may be used for communicating orders.

## 2.6.2 Large Single Store

- 1 shop
- 10 main categories
- 100 subcategories
- 10000 products
- Connection to backend systems

A big store will typically be an existing department store or mail order company. They already have a database with specifications of the products to sell. The system will be maintained by a dedicated staff. The company will host the system itself.

Whether the store will look like a single store or like a mall is a marketing decision. Some big stores might prefer looking like many smaller, specialized companies.

The question whether the company wants to use the existing database or set up a new one depends on the security of the system. Especially for mission-critical systems speed and reliability should not be much affected by adding an online Net.Commerce application.

For maintaining the information in the database, the merchant probably won't rely on the HTML-based customization forms provided by Net.Commerce. This system is complicated (for updating a product you may need to change up to 5 pages) and not very fast. For maintaining 10000 products, an administrator will need direct access to the database(s). He or she will want to directly manipulate columns and tables for changing prices, availability etc. Net.Commerce currently does not provide a tool for this. Third-party software such as Lotus Approach or Microsoft Access could provide solutions here.

The Web server(s) and the database will be on separate machines. This way you can scale your setup as needed. The database can be on any platform, as long as it is DRDA-compatible. Scaling the frontend can be done by using more or faster machines.

If you use more machines there must be a way to divide the incoming requests over the multiple machines, such as a network dispatcher.

## 2.6.3 Small Mall

- 3 shops
- 15 categories
- 60 subcategories
- 300 products

This mall will be maintained by a dedicated staff, probably a small content hosting company doing this as their main business. The shops could be considered as three small shops, but they could have common parts and share a common database, server, security setup, etc. Some shops will want their own shopping cart, navigation and ordering process, some may not. For customizing

this you can use the Task Manager to specify which macro or API to use for which task for which shop.

The merchants could use a common promotion site, where products of all companies are featured. This could be a macro that searches the database for all current specials.

A combination of shops that complement each other and have a common fulfillment system could resemble a department store. If they make an integrated look, they could compete as a high-quality shop/mall where you can buy all you need in one stop.

## 2.6.4 Large Mall

- 50 shops in 5 categories
- 500 categories in 50 main categories
- 25000 products
- Some conversions of existing databases
- Connection to backend systems

A big mall could consist of 50 shops or more. Big shops will host their own site. The hosting company will run a server farm, with the following probable main tasks:

- Keeping the hardware running and up-to-date
- Administering the mall (site manager)
- Writing applications to run in the mall or connecting to legacy systems
- Helping merchants design their shops
- Running a help desk for merchants and/or customers

In a big mall the possibility of merchants offering the same kind of products is very great. Use of search tools across the mall is interesting to customers, possibly more than for most merchants themselves, because it encourages competition, especially on price.

---

## 2.7 An Example of a Small Shop

In this section we introduce a small scenario that we have set up to illustrate certain aspects of planning and customization, such as a simplified order process and shipping facility.

### 2.7.1 The Ma Baker Environment

For this example we take a closer look at Ma Baker. Ma Baker is a shop that sells real Dutch bread through the Internet. Ma Baker has only a limited number of products it sells worldwide.

Because Ma Baker does not produce all bread itself, it sends the incoming orders to the bakery. Every morning the bakery receives a fax with the orders, in the afternoon they prepare the orders and the shipping provider collects the bread at the manufacturer to directly deliver them to the customer, except when the customer orders bread from more than one manufacturer. In this case Ma Baker repacks the orders and all products are shipped from the Ma Baker office.

Ma Baker set up a single Net.Commerce system on Windows NT. Because of the small amount of products, entering product information was not a problem. All photos were made by Ma Baker.

## 2.7.2 Shopping at Ma Baker's

Let's take a quick look at Ma Baker's home page:

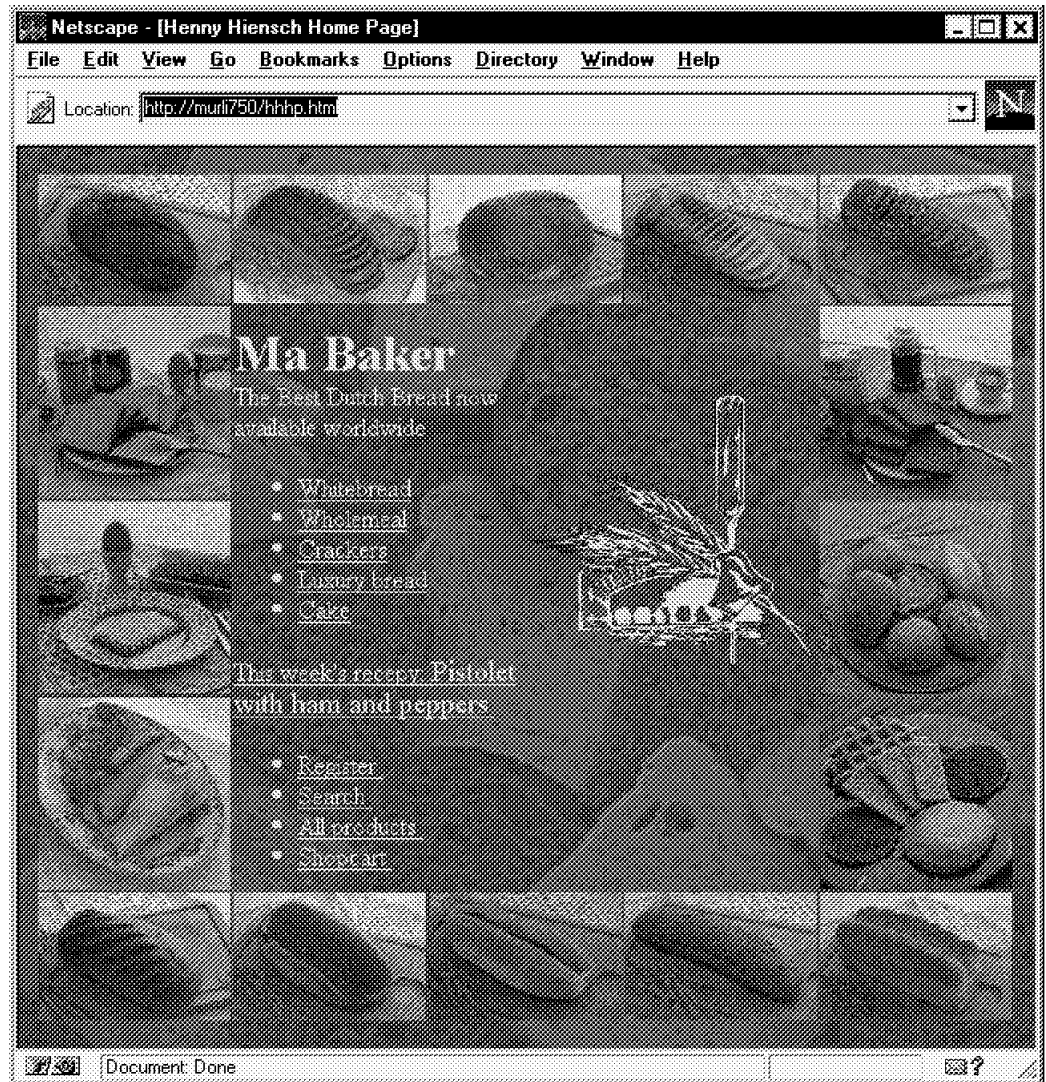


Figure 16. The Ma Baker Home Page

**Note:** Ma Baker is an online store that is currently under development for the Henny Hiensch Bakkerij in the Netherlands. The pages shown in this section serve as examples only and are not fully translated or complete in their functionality.

Select the bread you want to buy by clicking on the appropriate image. Or, you can select your favorite bread from a list that is displayed when you click on **All products**. Figure 17 on page 38 shows one of Ma Baker's product pages:



Figure 17. A Ma Baker Product Page

If you want to add the bread to your shopping cart, select the **Ja, Lekker!** (meaning "yes, that's great!") button.

The design of Ma Baker allows shoppers to go to a product page directly from the home page thus speeding up the shopping process significantly. The shop also offers five categories, listed on the home page, from where shoppers can eventually get to the products they want to buy.

### 2.7.3 Customizing the Order Process

To prepare for the fax, the macro orderacct.d2w was edited to include some lines to produce a flat file.

```
%{=====
Edited orderacct.d2w macro to export to flat file "c:/d2wtest.txt"

Change db2www.ini as well to include:
ENVIRONMENT (DTW_FILE) DTWFFI ( )
=====}%

%define {
%}

%DEFINE myFile = "c:/d2wtest.txt" 1
%DEFINE myTable = %TABLE
%DEFINE DTW_SAVE_TABLE_IN = "myTable"

%SQL(BuildShiptoData) {

select strfnbr, sanick, stquant, prsdsc, prnbr, shaddr.salname, shaddr.safname,
shaddr.saaddr1, shaddr.sacity, shaddr.sastate, shaddr.sacntry, shaddr.sazipc,
formatcur(stprice,stpcur,char('N')) as stprice, formatcur((stprice*stquant),
stpcur,char('N')) as total, stornbr, stshnbr from ORDER_COMP, SHIPTO,
product, shaddr
```



```

        where ORRFNBR=$(order_rn)
        AND shipto.stsanbr=shaddr.sarfibr
        AND STORNBR=ORRFNBR
        AND shipto.stprnbr=product.prrfnbr
        order by sanick

%SQL_REPORT{ %}

        %SQL_MESSAGE{ default: { %} %}
%}

%SQL(InitSQL){

        select mhthead, mhtfoot, mhtbase from mall

%SQL_REPORT{
%ROW{
        @DTW_assign( MallHeader, V_mhthead)
        @DTW_assign( MallFooter, V_mhtfoot)
        @DTW_assign( MallBase, V_mhtbase)
%}
%}

%SQL_MESSAGE{
        default: { %} :continue %}

%}

%SQL(merchant) {
        select mestname from order_comp, merchant where
        orrfnbr = $(order_rn) and
        ormenbr = merfnbr

%SQL_REPORT {
        %ROW {
                <p align=center>
                <font size=6>
                <strong>$(V_mestname)
                </strong>
                <br>
                <br>
                </font>
                <font size=4>
                Thank you very much for giving us your business.
                </font>
                <font size=5><br>
                </font>
                <font size=4>
                Your order has been accepted <br>
                and your <b>Order Reference Number is
                <strong>$(order_rn)
                </strong>.
                </b>
                </font></p>

        %}

%}

%SQL_MESSAGE {
        100: {
                </table>
                <p><b>No Merchant Name could be found for this order.</b>
                %}
        default: {
                </table>
                <font size=+3><b>Database Error:</b><p>
                A database error occurred when attempting to
                add your information to the database.
                Please contact the merchant server administrator.
                </font><BR>
                Returned an error code of $(SQL_CODE)

        %}

%}

%}

```

```

%SQL(nextorder) {
    select orrfnbr, ormenbr
    from ORDER_PEND
    where orshnbr=(select shrfnbr from shopper
                   where shlogid='$(SESSION_ID)')
        AND orstat='P'
    order by orrfnbr

    %SQL_REPORT {
        %ROW {
            %if ("$(ROW_NUM)" == "1")

                @DTW_assign(nextorrf, V_orrfnbr)
            %endif

        %}

        <center>
        <table cellspacing=0 cellpadding=0>
        <tr>
        <td align=center valign=bottom>
            <form action="/cgi-bin/nph-msrvr;/order/display">
            <input type=hidden name="status" value="P">
            <input type=hidden name="order_rn" value="$(nextorrf)">
            <input type=submit value="Process Next Order">
            </form>
        </td>
        <td align=center valign=bottom>
            <form action="/cgi-bin/nph-msrvr;/order/list">
            <input type=hidden name="status" value="P">
            <input type=submit value="View All Pending Orders"></p>
            </form>
        </td>
        </tr>
        </table>
        </center>
        %}

        %SQL_MESSAGE{
            -100: {
                </table>
                %}

            100: {
                </table>
                %} :continue
            default: {
                </table>
                <font size=+3><b>Database Error:</b><p>
                A database error occurred when attempting to
                add your information to the database.
                Please contact the merchant server administrator.
                </font><BR>
                Returned an error code of $(SQL_CODE)

            %}

        %}

    %}

    %HTML_REPORT {

    <html>

    <head>
    <META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
    <title>Order OK</title>
    </head>

    <body background="/ncsample/back1.gif" bgcolor="#FFFFFF">

    %EXEC_SQL(InitSQL)
    %include "$(MallHeader)"

```

```
<P>Faxing billing information..... 2
%EXEC_SQL(BuildShiptoData)
@DTWF_WRITE(myFile,"DELIMITED",";",myTable)

%EXEC_SQL(merchant)
%EXEC_SQL(nextorder)

%include "$(MallFooter)"

</body>
</html>

%}
```

The following is an explanation to the points highlighted in the macro example above:

- 1** Defining table and file to export
- 2** Exporting table myTable to file myFile

Net.Commerce does not register the flat file environment during installation of Net.Data. To do this yourself edit db2www.ini (in x:/www/html on Windows NT, where x is the drive letter you installed Net.Commerce on) to look like the following:

```
ENVIRONMENT (DTW_SQL) DTWSQL
              ( IN_DATABASE, LOGIN, PASSWORD, TRANSACTION_SCOPE, SHOWSQL, ALIGN)
ENVIRONMENT (DTW_FILE) DTWFFI ( ) 1
ENVIRONMENT (DTW_DEFAULT) DTWFUNC ( )
ENVIRONMENT (DTW_APPLET) DTWJAVA ( )
ENVIRONMENT (DTW_ODBC) DTWODBC
              ( IN_DATABASE, LOGIN, PASSWORD, TRANSACTION_SCOPE, SHOWSQL, ALIGN)
MACRO_PATH C:\IBM\NetCommerce\macro\en_US\ncsample;
            C:\IBM\NetCommerce\macro\en_US\demomall;
            C:\IBM\NetCommerce\macro\en_US
INCLUDE_PATH C:\IBM\NetCommerce\macro\en_US\ncsample;
             C:\IBM\NetCommerce\macro\en_US;
             C:\IBM\NetCommerce\html\en_US\ncadmin\teditor;
             C:\
IBM\NetCommerce\html\en_US
EXEC_PATH C:\IBM\NetCommerce\macro\en_US\ncsample
BIND_FILE C:\IBM\NetCommerce\bin\d2wsq1.bnd
HTML_PATH C:\WWW\HTML
DB2INSTANCE MSTERREN
DB2PATH C:\DB2\SQLLIB
```

Figure 18. DB2WWW.INI on Windows NT with Flat File Environment Registered

The following is an explanation to the point highlighted in the file example above:

- 1** The additional line to register the flat file interface.

The output that has been put into such a flat file constitutes an order form to be picked up by a fax software and sent to a shipping company for further handling.



---

## Chapter 3. Advanced Planning

This chapter discusses planning aspects for Net.Commerce and its related products. It is our intention to illustrate that, though Net.Commerce comes as a shrink-wrap product on a single CD-ROM, there is a lot more to consider about this product than just installing it out of the box, connecting to the Internet, and being ready to receive orders.

---

### 3.1 Planning a Net.Commerce Project

To set up a Net.Commerce shop you need a variety of specific skills where the quantity of required resources depends on the level of customization needed for the shop. In the easiest case, the system has already been set up for you, and all you have to do is build a small, simple store that is similar to the default or the demo mall provided. You then may not need more time and skills than you would need to design your static and dynamic pages and customize the shopping process a little.

#### 3.1.1 Skills Profile for Advanced Scenarios

If you have more ambitious plans and would like to get started on your own, you have a great challenge ahead since you need all of the following skills:

- AIX or Windows NT platform skills
- Web server setup, customization and tuning skills
- HTML skills for the static pages
- DB2 setup, administration and problem determination skills
- SQL and Net.Data programming skills for the macros
- Graphical design skills including photography
- User interface/interaction design skills and information design
- Java, JavaScript and multimedia application development skills
- API programming skills using the C language and an appropriate compiler
- Payment system (SET) skills
- Network skills (TCP/IP, security, Internet access)
- Marketing and sales experience
- Project management experience

You will realize that, unless you happen to have most of the skills above or know someone who does, the design and development efforts for a complex Net.Commerce site should be given to a team of specialists. You may also add a business consultant to that team to assist in the transition process from your existing store to selling over the Internet.

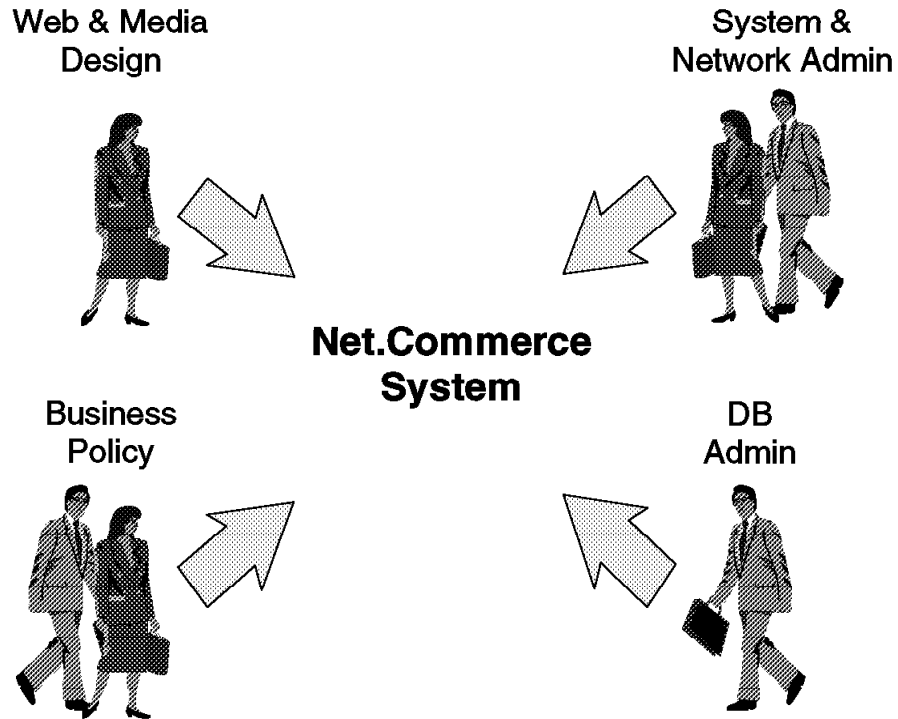


Figure 19. Net.Commerce Implementation Team

### 3.1.2 Time Schedule

As with all IT projects, the time schedule very much depends on the knowledge available about the topics to be covered. If you have to acquire all needed skills before or while creating the shop, it will be a long and hard process. Even if you have (access to) most of the skills, hardware and software, installing a shop could be a matter of weeks.

Other factors to consider are:

- Connections to backend systems
- Connections to payment systems
- Amount of customization required, especially programming work to be done
- Completeness of the information
- Availability of product images, logos, etc.

### 3.1.3 Who Will Be Your Service Provider?

For a Net.Commerce project and shopping site, you can certainly choose to develop and host everything at your own location. Or you may want to have a service provider, for instance IBM, do everything at their site. Or you may choose a combination of the two. We don't want you to be misled by the fact that Net.Commerce is sold as a product that enables merchants to build their own shop on the Internet without much help. As the previous explanations have shown that may not always be the case. In most instances the merchant does not have the required skills and resources.

One solution would be that the merchant has the content hosted by a dedicated service company. That way, setup and maintenance of the hardware, the network, and the customization and connection of most of the software and security components are done elsewhere. This is not as easy as hosting a normal (passive) Web site, though, because a connection to the database is needed, and testing and updating macros and APIs can be difficult.

Joining a mall that provides you with solutions on the payment gateway might be the best solution until SET becomes more widely available.

### **3.1.4 Marketing and Design**

Graphical design, Java, and multimedia programming are highly specialized jobs that are best left to professionals. User interface and information design for the simplest shops can be done by common sense, but for larger shops professional advice might substantially improve the appearance on the Web. User interface and information designers can help decide what information to include for products and categories as well as design a useful layout of the page. This makes the shop more transparent and makes it easier to find the products the customer wants.

Last but not least, you need to think about marketing. Many shops are set up from the technical point of view. Much can be improved by having a good marketing vision before setting up the shop.

So while it is possible to build your own shop as a small merchant, it is just not realistic to expect a single person to establish a large shopping site without help; the scope of skills is just too wide.

### **3.1.5 Hardware and Software Requirements**

Net.Commerce is currently available for Windows NT and for AIX. Versions for other platforms, such as ES/390, AS/400, and Sun Solaris are currently being developed. Hardware and software requirements for AIX and NT are shown in the following sections. Requirements regarding the Net.Commerce database are described in 3.3.1, "Database Space Requirements" on page 49.

#### **3.1.5.1 AIX System Requirements**

- Hardware
  - RS/6000 or Power Series (Model C20 or higher)
  - 120 MHz PowerPC 604 processor
  - 96 MB RAM minimum, 128 MB RAM recommended
  - 350 MB free disk space minimum for program files
  - 2 to 3 times the RAM size for paging space
  - 5 to 100 MB disk space for HTML pages, macros and graphics depending on multimedia usage
  - LAN adapter that is supported by the TCP/IP protocol
  - CD-ROM drive
  - Graphics-capable monitor with a recommended resolution of 1024x768 or higher
  - Mouse or other pointing device

- Software
  - AIX Version 4.1.5 or later
  - DB2/6000 Version 2.1.2
  - Internet Connection Secure Server Version 4.1.1
  - Note:** DB2 and ICSS are part of the Net.Commerce product package.
  - Web browser that is:
    - SSL-compliant
    - Java and JavaScript-enabled
    - Capable of supporting tables and frames
    - Capable of handling cookies
    - Capable of supporting self-signed certificates (which is very useful for testing and Intranet installations)

### 3.1.5.2 Windows NT System Requirements

- Hardware
  - Pentium 166 MHz processor
  - 64 MB RAM minimum, 128 MB RAM recommended
  - 100 MB free disk space for program files
  - 100 MB to 150 MB paging space
  - 5 to 100 MB disk space for HTML pages, macros and graphics depending on multimedia usage
  - LAN adapter that is supported by the TCP/IP protocol
  - CD-ROM drive
  - Graphics-capable monitor with a recommended resolution of 1024x768
  - Mouse or other pointing device
- Software
  - Windows NT Server Version 3.51 or later
  - DB2/NT Version 2.1.2
  - Internet Connection Secure Server Version 4.1.1
  - Note:** DB2 and ICSS are part of the Net.Commerce product package.
  - Web browser that is:
    - SSL-compliant
    - Java and JavaScript-enabled
    - Capable of supporting tables and frames
    - Capable of handling cookies
    - Capable of supporting self-signed certificates (which is very useful for testing and Intranet installations)



---

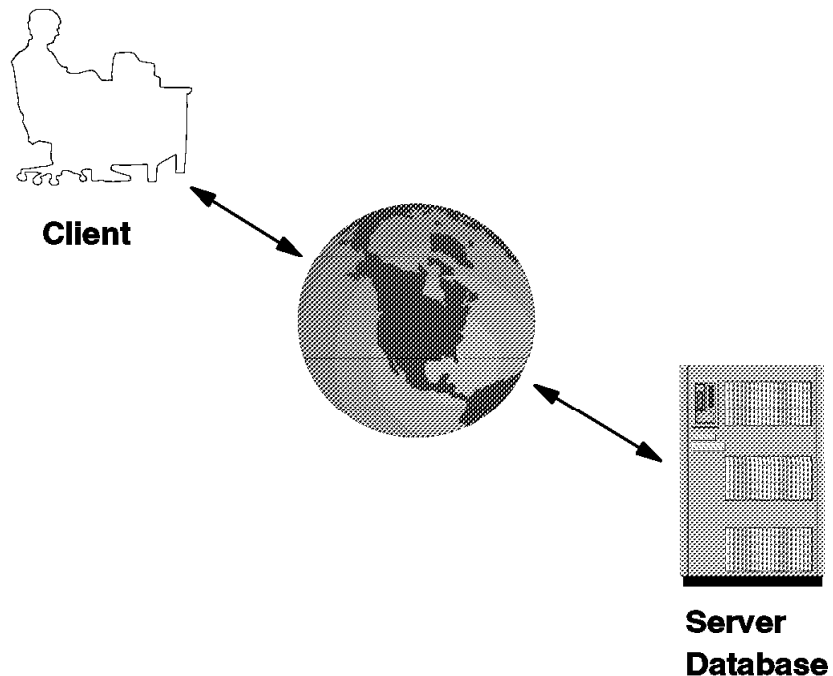
## 3.2 System Selection

With Net.Commerce you have the option of installing a shopping site on a single system or splitting the site over a cluster of servers to provide for load balancing, scalability, etc.

### 3.2.1 Using a Single System Configuration

A single system configuration is a good way to start your electronic business. If you are not sure about the number of visitors and the amount of business you can expect on your Web site, you can start small and scale up later, if necessary.

The following figure shows a single system configuration



*Figure 20. Net.Commerce Single System Configuration*

All parts of the Net.Commerce system are located on one physical machine:

- The Internet Connection Secure Server
- The DB2 Database Server
- The Net.Commerce Server

Of course, you need a connection to the Internet from this system.

Depending on your existing IT environment, on the existing skills and on your personal preferences, you can choose between an AIX or a Windows NT-based installation. Net.Commerce is available on both operating system platforms with equivalent functionality.

### 3.2.2 Using a Multiple System Configuration

The following sections discuss the benefits of distributing the Net.Commerce components to multiple systems.

**Increased performance:** By using more than one CPU, you will have more computing power to use. One possibility in this area is to distribute the database engine from the Net.Commerce server machine. In this case, the database engine will be able to process more transactions in a smaller period of time. Another way is to use several systems as Net.Commerce server systems. In this case, even the Net.Commerce server systems are able to share the workload.

**Availability:** By using more than one system for the Net.Commerce server, or more than one system for the database server, you can achieve a fault-tolerant configuration. In conjunction with High Availability Cluster Multiprocessing (HACMP/6000) you can configure one or more

systems as standby, if the main system encounters a problem and cannot continue its work. On the Windows NT platform, similar solutions for fault tolerant systems are available, for instance StandbyServer for NT from Vinca Corporation. In addition you can use intelligent routers that redirect the network traffic from the Internet to another system, if one system is no longer reachable. An example of this would be the IBM Interactive Network Dispatcher which is available on the AIX and Windows NT platforms.

**Higher security for the database:** By placing the Net.Commerce server in front of your company's firewall and the database server behind the firewall, you can ensure that nobody, except your Net.Commerce server, accesses the database.

The following picture shows a Net.Commerce multiple system configuration, where you have one Net.Commerce machine, handling incoming requests and a database machine, where all the information is stored.

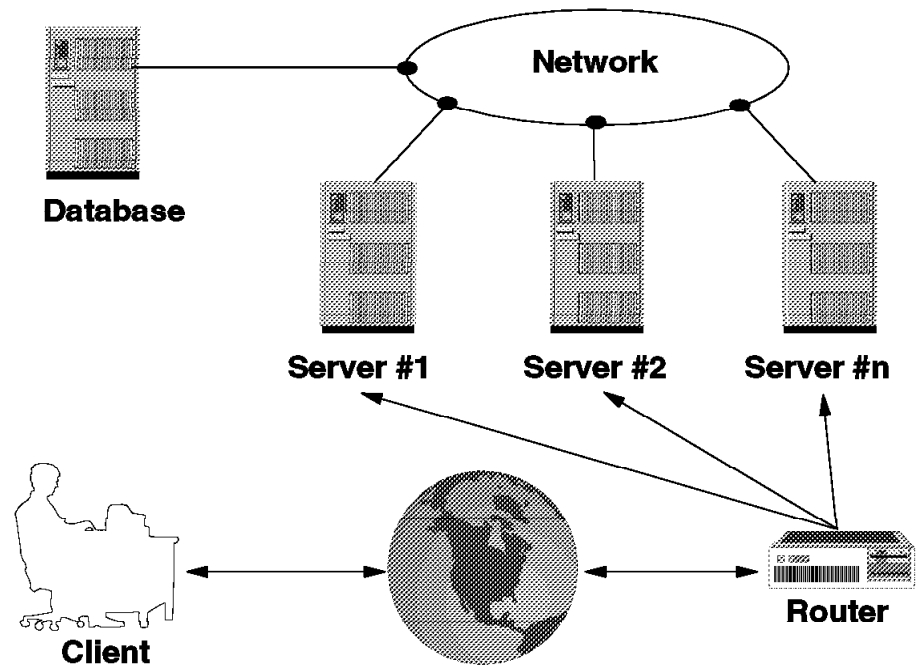


Figure 21. Net.Commerce Multiple System Configuration

### 3.3 Database Considerations

Net.Commerce uses a complex database scheme to store information about all aspects of your electronic store. This includes data about the store, the products, the shoppers, the product prices, taxes, shipping, and so on.

This section gives a description of what kind of information is needed for the Net.Commerce database. It does not include all tables and fields of the database, but only those that have information you must provide. For a detailed description of the Net.Commerce database scheme refer to *IBM Net.Commerce: Customization Guide and Reference*, SC09-2370.

#### 3.3.1 Database Space Requirements

Depending on the number of products, shoppers (and stores if you run a mall), your database may become very large. Basic figures to allow an estimate are shown below:

- Per unregistered shopper: 1 KB
- Per registered shopper: 2.5 KB
- Per product: 2 KB
- Per order: 2 KB
- Log files. Database log files will grow depending on the number of transactions and the frequency of cleaning. By default, DB2 is configured to implement circular logging and creates three primary and two secondary log files, each with a size of 1 MB. If you change the configuration to retained

logging, all transactions since the time of the last full backup will be kept in the log files which can then become much bigger. Please refer to 3.3.5, "Net.Commerce Database Recovery" on page 59 for more details on logging options.

### 3.3.2 Accessing Data and the Database

There are several possibilities to access the data in the database:

**Use standard Net.Commerce tools:** This means using HTML forms and entering information manually. This is sufficient for adding certain information, like administrators, shoppers, or shopper groups. For bulk data, such as product information and prices, which need frequent updates, you will likely not want to enter this data manually.

**Access the database via ODBC:** Net.Commerce uses an ordinary DB2 relational database, which can be accessed via ODBC from any ODBC-enabled client application, such as Lotus Approach. This is a good way to browse the database and to make updates.

**Note:** There is no simple or default way to control access to database tables when the Net.Commerce database is accessed via ODBC. The standard Net.Commerce tools have built-in access protection mechanisms to prevent, for instance, one merchant in a mall from accessing another merchant's data. When using Lotus Approach, you have to provide such mechanisms yourself. If you only have a single store then this may not be such a big issue.

**Access the database using DB2 tools:** Another way to access the database is via standard DB2 mechanisms, such as DRDA. This enables you to connect to databases that are located on different platforms such as Windows NT, OS/2, UNIX, AS/400 or MVS. If, for example, you have an existing product database on MVS, you can connect to this database and don't need to keep the data twice.

**Net.Commerce mass import/update utility:** This function enables you to populate a Net.Commerce database with information from a flat file where you have previously stored data from your existing database. The mass import utility will be available with Version 2 of Net.Commerce.

**DataJoiner:** Using applications such as DataJoiner enable you to connect not only to other DB2 databases, but also to other relational database systems such as Oracle, Sybase and Informix.

**Note:** Net.Commerce Version 2 will support Oracle in addition to DB2.

### 3.3.3 Understanding the Database Structure

This section gives a brief overview of the database structure of the Net.Commerce database. For more detailed information see *IBM Net.Commerce: Customization Guide and Reference*, SC09-2370. The Net.Commerce database is named MSERVER by default. It contains all the information that the Net.Commerce system needs.

There are two main ways to customize the Net.Commerce database:

1. Use any column that is reserved for merchant customization.
2. Add a new table to the database, and link it to one or more existing tables through keys.

To understand how to customize the database you have to be familiar with the concepts of entity relationships model and referential integrity.

**Entity Relationships(ER) Model:** This is a method of software analysis and design. One of the outputs of this method is the ER model that describes with a graphic (diagram) how entities are related to each other. ER analysis and design is going to be substituted by the object-oriented analysis and design.

Figure 22 shows the ER model for the Net.Commerce database.

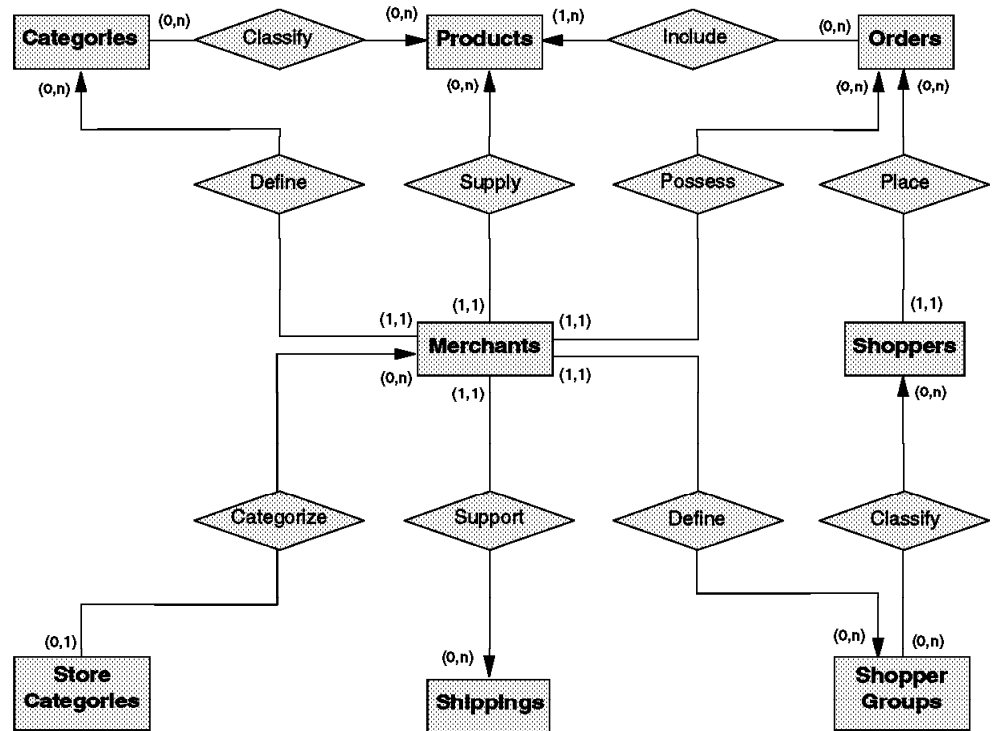


Figure 22. Entity Relationships Model for Net.Commerce

**Referential Integrity (RI):** This is the mechanism that the DB2 offers to control the relationships between data in more than one table. This includes rules for restricting deletes, cascading deletes, or setting null values on deletes of rows from a parent table. You can also restrict updates when the parent table does not contain a corresponding key value. RI constraints are used to enforce your rules on the data across one or more tables. For more information on referential integrity see *IBM DB2: Application Programming Guide*, S20H-4643 and *IBM DB2: SQL Reference*, S20H-4665.

Figure 23 on page 52 shows the RI model for the Net.Commerce database.

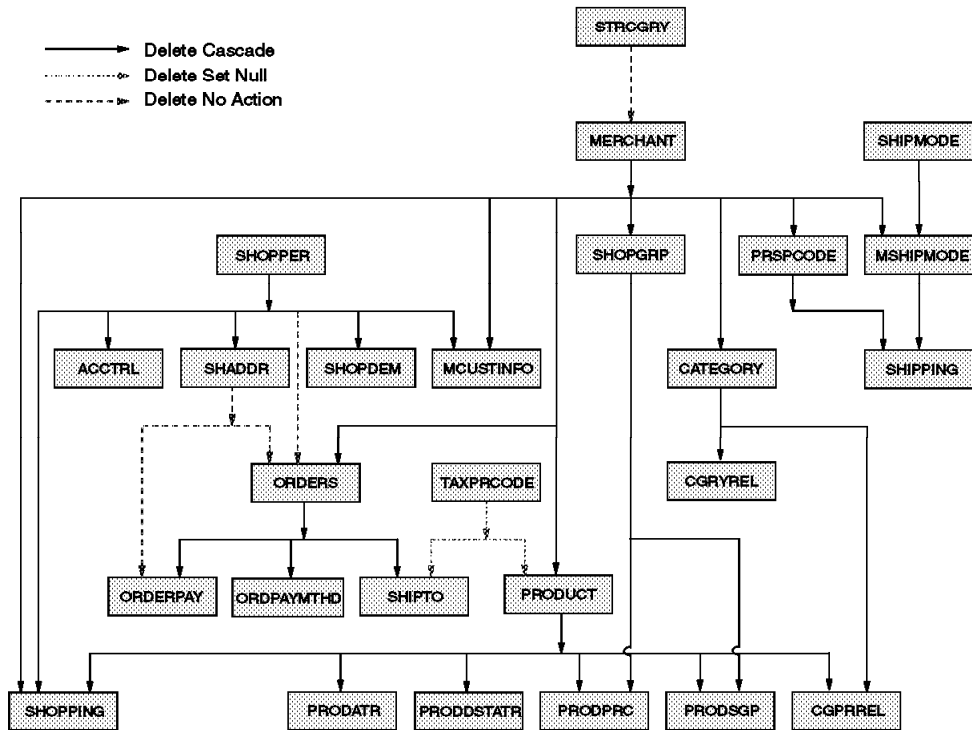


Figure 23. Referential Integrity Model for Net.Commerce

### 3.3.4 Database Tables

The database tables are divided into several categories, as shown in Table 1:

Table 1 (Page 1 of 2). Database Table Groups	
Group	Tables
Shoppers and Users	SHOPPER (Shopper Profile) SHOPDEM (Demographics) SHADDR (Address) ACCTRL (Access Control)
Merchants, Stores, Shopper Groups and Customers	STRCGRY (Store Category) MERCHANT (Merchant Profile) SHOPGRP (Shopper Group) MCUSTINFO (Customer Info)
Categories, Products and Items	CATEGORY (Categories) CGRYREL (Category Relationship) PRODUCT (Products and Items) CGPRREL (Cat./Product Relationship) PRODDPRC (Prices) PRODSGP (SG Product Template) PRODATR (Attribute) PRODDSTATR (Distinct Attribute)
Shipping	SHIPMODE (Shipping Mode) MSHIPMODE (Merchant Ship Mode) PRSPCODE (Product Ship Code) SHIPPING (Shipping)

<i>Table 1 (Page 2 of 2). Database Table Groups</i>	
<b>Group</b>	<b>Tables</b>
Orders	SHOPPING (Shopping Cart) ORDERS (Orders) ORDERPAY (Payment) ORDPAYMTHD (Payment Method) SHIPTO (Shipping Association)
System Tables	MALL (Mall Front) BROWSER (Browser) KEYS (Key Values) TASKS (Tasks) MACROS (Macros) APIS (APIs)

### 3.3.4.1 Shoppers and Users

Shoppers are people that visit your store on the Internet and buy goods. Users are merchant employees, who have been granted access to the Net.Commerce Administrator functions.

Usually you won't need to update this information frequently:

- The shopper information, such as name, address, age, gender, and address book is entered through the HTML forms by the shopper. If you have an existing customer database, you can think about matching the shopper information with this database. You can also start building a customer database with this information.
- The user information, which consists of the user's scope (mall or store) and level of access is entered through the administrator's HTML forms.

Figure 24 on page 54 shows the tables associated with shoppers and users and illustrates their relationships.

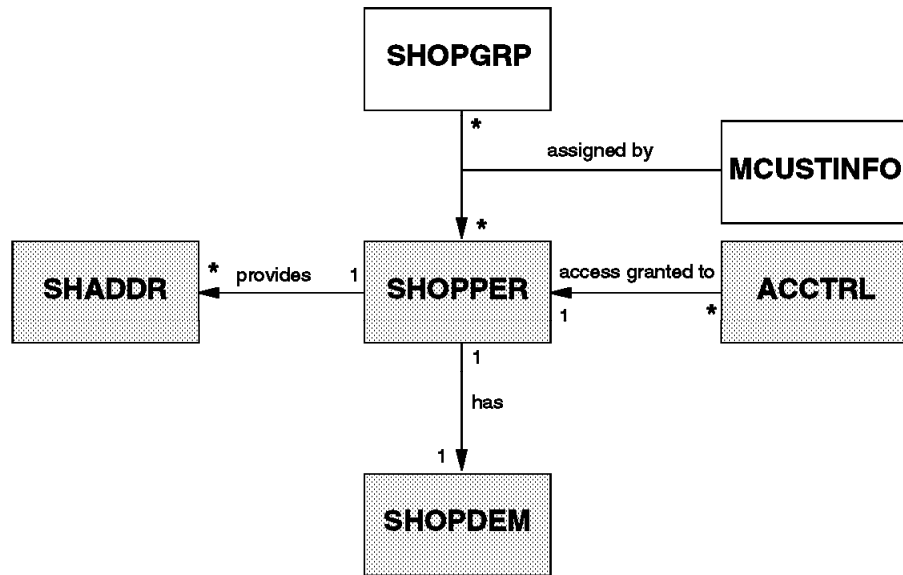


Figure 24. Database Tables for Shoppers

### 3.3.4.2 Merchants, Stores, Shopper Groups and Customers

This group of tables contains data about merchants, their stores, the shopper groups and the merchant's customers.

- Merchant information comes through the browser, so there is no need for automation.
- Shopper groups are maintained by the Store Administrator.
- Shopper group product templates are maintained through the Store Administrator, too.
- Merchant customer information associates shoppers with shopper groups and may hold additional information on shoppers, such as customer number.

Figure 25 on page 55 shows the tables associated with merchants and illustrates their relationships.



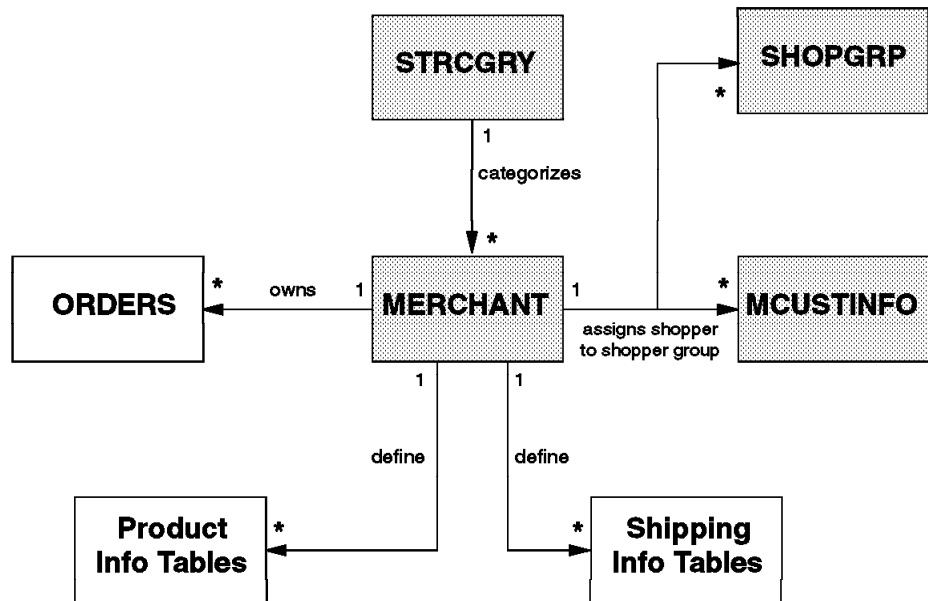


Figure 25. Database Tables for Merchants

### 3.3.4.3 Categories, Products, Prices and Items

This group of tables is likely to need frequent updates and connection to an existing system. Standard Net.Commerce HTML form editing will not be enough.

Figure 26 on page 56 shows the tables associated with categories and products and illustrates their relationships.

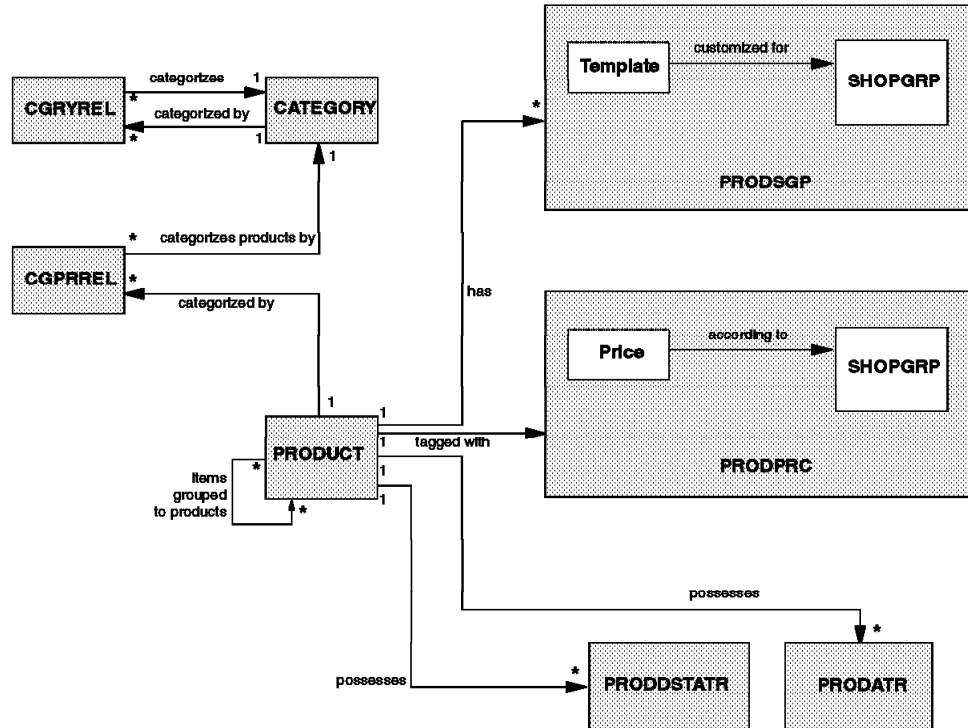


Figure 26. Database Tables for Categories and Products

### 3.3.4.4 Shipping

Figure 27 on page 57 shows the tables associated with shipping and illustrates their relationships.

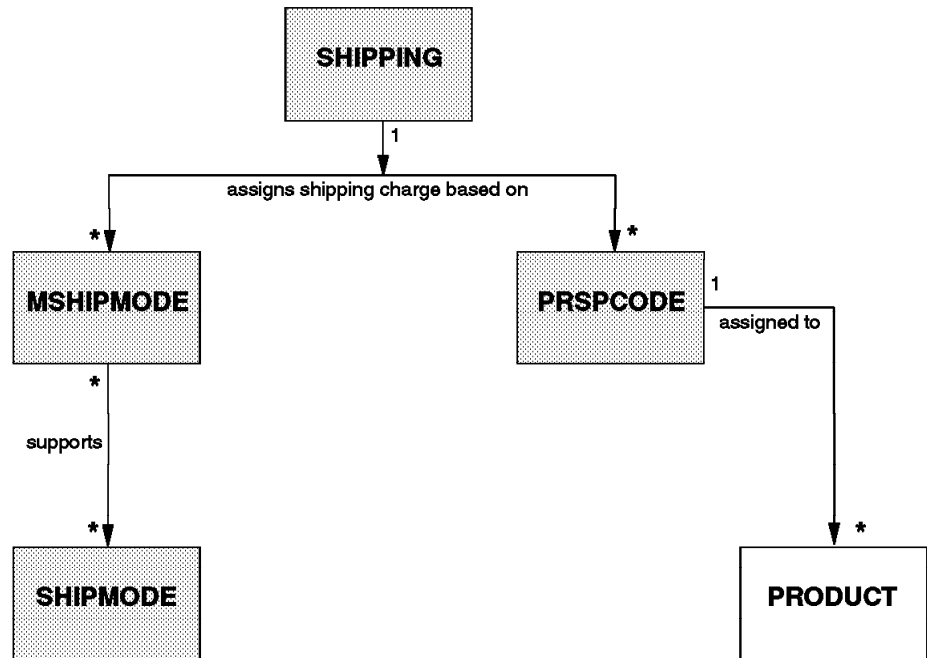


Figure 27. Database Tables for Shipping

### 3.3.4.5 Orders and Payment

Orders are stored in multiple tables:

- The SHOPPING table which contains the shopping cart contents
- The SHIPTO table which contains the details of the (possible) orders
- The ORDERPAY table which contains tax data
- The ORDPAYMTHD table which contains credit card information
- The ORDERS table which contains the totals for every order

Figure 28 on page 58 shows the tables associated with orders and payment and illustrates their relationships.

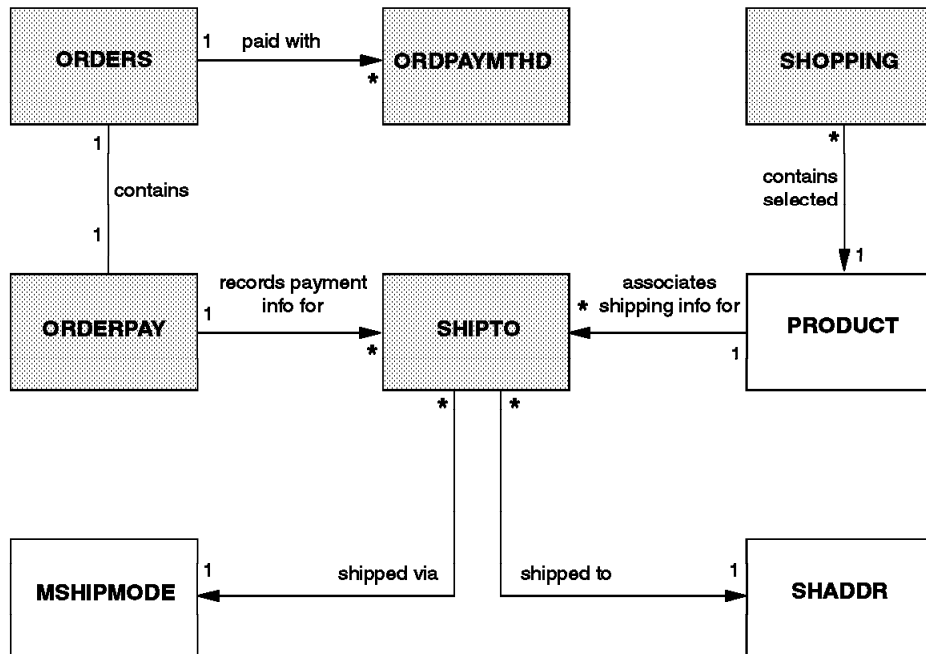


Figure 28. Database Tables for Orders and Payment

### 3.3.4.6 System and other Tables

Figure 29 on page 59 shows the other tables that remain in the Net.Commerce database.

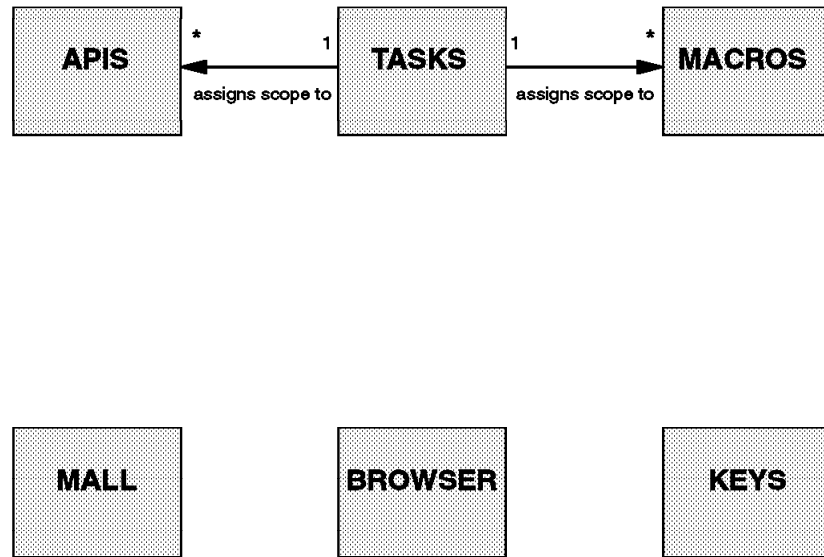


Figure 29. Other Database Tables

### 3.3.4.7 Tax Calculation

Tax calculation by default is performed by the Taxware system which is incorporated into Net.Commerce.

### 3.3.4.8 Accuracy of Data

In order to achieve a high degree of relevance with regard to the published data, you need to establish a way to keep data current. If you already have some kind of product and price database, you might not want to make your updates twice, both in your existing database and in the Net.Commerce database.

## 3.3.5 Net.Commerce Database Recovery

For a complete reference on database recovery see *IBM DB2: Administration Guide*, S20H-4580.

In this section we make some considerations about Net.Commerce database recovery and show an example of a possible configuration of the Net.Commerce database.

When you manage a database you have to be concerned about the problems on the database. You need to know the strategies available to you to help when there are problems with the database. Typically you will deal with media and storage problems, power interruptions, and application failures. You need to know that you can back up your database, or individual table spaces, and then rebuild them should they become damaged or corrupted in some way. The rebuilding of the database is called recovery. There are three ways to recover a damaged database: crash recovery, restore, and roll-forward.

**Crash recovery:** This method protects a database from being left in an inconsistent or unusable state. Transactions, or units of work, against the database are interrupted unexpectedly. For example, should a failure (power interruption, application failure) occur before all of the changes that are part of the unit of work are completed and committed, the database is left in an inconsistent and unusable state. The database then needs to be moved to a consistent and usable state. This is done by rolling back incomplete or in doubt transactions and completing committed transactions that were still in memory when the crash occurred. You can do this by entering a RESTART DATABASE command.

**Restore recovery:** This method allows for the restoration of a previous version or image of the database that was made using the BACKUP command. A database restore will rebuild the entire database using a backup image of the database made at some point earlier. A backup of the database allows you to restore a database to a state identical to the time when the backup was made. Every unit of work from the time of the backup to the time of the failure is lost.

**Roll-forward recovery:** This may be the next task to be done following a restore depending on the state of the database. A database roll-forward recovery method will follow the restore of the database with the application of database logs. The database logs record all changes made to the database. This method completes the recovery of the database to a state identical to the time just before the failure. You can apply the restore recovery and roll-forward recovery methods either to entire databases or to single table spaces.

To decide which database recovery method to use, you must consider the following key factors:

- Will the database be recoverable or non-recoverable?
- How near to the time of failure will you need to recover the database (the point of recovery)?
- How much time is associated with recovery? This would include:
  - Time between backups (will affect roll-forward recovery)
  - Time the database is usable or accessible (backing up online or offline based on data availability needs)
- How much storage space can be allocated for backup copies and archived logs?

We now discuss the following topics that are associated with the questions that we had made previously and we arrive at some conclusions about a better database configuration than the default configuration.

**Recoverable and non-recoverable databases:** If you can re-create data easily, then the database is a candidate to be a non-recoverable database. If you cannot re-create data easily, then the database is a candidate to be a recoverable database. The Net.Commerce database is definitively a recoverable database because it holds data from an Internet environment: You cannot ever re-create data coming from an Internet environment.

**Point of recovery, time between two backups and backup online or offline:** The restore and roll-forward methods of recovery provide different points of recovery. The restore method of recovery will restore the database from an offline backup image. That means you will lose all data from the point

at which the failure occurs back to the point at which you made the backup. The roll-forward method allows you first to restore a backup copy of the database and then apply a roll-forward on the retained logs so that you can establish the same situation that you had when the failure occurred.

For most Net.Commerce databases, roll-forward is a better recovery strategy. In an Internet environment, you cannot have your database offline very often to perform a backup that could take several hours. If the backup is performed offline, only the backup task needs to be connected to the database. Online backups are supported only if roll-forward recovery is enabled. If roll-forward recovery is enabled and you have a complete set of logs, you can rebuild the database.

**Storage and performance considerations:** To prevent a media failure from destroying a database and your ability to rebuild it, you should keep the database backup image, the database logs, and the database itself on different devices. For this reason, it is highly recommended that you use the *newlogpath* configuration parameter to put database logs on separate devices once the database is created. This will also improve performance because all the logs will be written on separate devices, thus freeing up database server resources.

For Net.Commerce, we recommend that you write logs to another disk. Then you can choose to:

- Dedicate enough disk space in the database log directory.
- Manually copy logs to a different storage other than the database log-path directory.
- Use a user-exit program to copy these logs to another storage device.

### 3.3.6 Database Recovery Configuration Example

In our examples we have a Net.Commerce system installed on two NT servers:

1. On the first server there is a DB2 database machine that we call SERVDB.
2. On the second server there is an ICSS and NET.Commerce system and Netscape browser that we call SERVNT (hostname=wtr05353.itso.ral.ibm.com).

We have taken the following steps:

1. To start Database Manager on SERVDB:
  - Click on **Settings --> Control Panel--> Services**. Choose **DB2 Security Server**. Click **Start**.
  - Click on **Settings --> Control Panel--> Services**. Choose **DB2**. Click **Start**.
2. Create a new database called MAUMALL, which will be the backup copy of the DEMOMALL database, and which resides on the C: drive of SERVDB.
  - Be sure that nobody is accessing the DEMOMALL database. Then, from a command line processor window, type: `db2 list application for database demomall`.
  - To back up the DEMOMALL database from a command line processor window, type: `db2 backup database demomall to c:.`
  - To restore the DEMOMALL database in MAUMALL, from a command line processor window, type: `db2 restore demomall from c: into maumall`.

3. To update the database configuration on SERVDB, from a command line processor window, type: update database configuration for maumall using newlogpath d:db2logmau logretain on. userexit on
4. The backup of the MAUMALL(OFFLINE) database should have been made on a different drive of the log database directory. We put it on drive D: of SERVDB by typing: backup database maumall to d:. (The time stamp is 19970408161532.)
5. Configure Net.Commerce System on SERVDB.
  - To start the Net.Commerce Administration web server, click on **Programs --> IBM Net.Commerce Server and Administrator --> Administration WebServer**.
  - To update the Net.Commerce System Configuration, start your Netscape browser, then type: <http://wtr05305.itso.ra1.ibm.com:4444>.
    - Click on **Reconfigure --> System Configuration**.
    - Enter the user ID and password.
    - Enter Database Name MAUMALL.
    - Click on **Submit Form**.
    - Enter **Userid** and **Password**.
  - To stop the Net.Commerce Administration Web Server, close the window.
6. To start the Net.Commerce Server on SERVNT, click on **Settings --> Control Panel --> Services**. Choose **Net.Commerce Server** and click **Start**.
7. To start the Internet Connection Secure Server on SERVNT, click on **Settings --> Control Panel --> Services**, then choose **IBM Internet Connection Secure Server** and click **Start**.
8. Next, we need to do something with Net.Commerce on SERVNT. From Netscape, type <http://wtr05305.itso.ra1.ibm.com/demomall/basemall> then update Net.Commerce. (That is, place some orders, or make some registrations, so the database will be different.)
9. To make a second backup of the MAUMALL(ONLINE) database on SERVDB, from a command line processor window type: BACKUP DATABASE MAUMALL ONLINE to D:. (The time stamp is 19970408171518.)
10. Then, simulate a disk crash on drive C:.
  - Copy all the database log files from d:db2logmau to another directory: copy d:logmau\*.\* d:ptemp.
  - Drop database MAUMALL. From a command line processor window type: db2 drop database maumall.
  - Stop the Net.Commerce Server on SERVNT. To do so, click on **Settings --> Control Panel--> Services**, then choose **Net.Commerce Server** and click **Stop**.
  - Stop the Internet Connection Secure Server on SERVNT. Click on **Settings --> Control Panel--> Services**, then choose **IBM Internet Connection Secure Server** and click **Stop**.
  - Stop Database Manager on SERVDB.
    - Click on **Settings --> Control Panel--> Services**, then choose **DB2 Security Server** and click **Stop**.



- Click on **Settings --> Control Panel--> Services**, choose **DB2** and click **Stop**.
  - Re-start Database Manager on SERVDB.
    - Click on **Settings --> Control Panel--> Services**, choose **DB2 Security Server** and click **Start**.
    - Click on **Settings --> Control Panel--> Services**, choose **DB2** and click **Start**.
  - Re-start Internet Connection Secure Server on SERVNT. Click on **Settings --> Control Panel--> Services**, choose **IBM Internet Connection Secure Server** and click **Start**.
  - Copy all the database log files back to d:db2logmau by typing: `copy d:ptemp*.* d:logmau`.
11. Restore the database DEMOMALL on SERVDB. From a command line processor window type: `restore database maumall from d: taken at 19970408171518 into c:`.
  12. Roll-forward the database DEMOMALL on SERVDB. From a command line processor window type: `rollforward database maumall stop into c:`.
  13. Start Net.Commerce Server on SERVNT. Click on **Settings --> Control Panel--> Services**, choose **Net.Commerce Server** and click **Start**.
  14. Check if everything is okay. From Netscape type URL: `http://wtr05305.itso.ral.ibm.com/demomall/basemall`. Begin to retrieve the information previously updated or inserted.

---

### 3.4 Web Server Considerations

When thinking about setting up your Net.Commerce Server, one concern that you will always face has to do with determining the right "size and strength" of the hardware so that it can provide the best response time to a client's request. There are several considerations you will have to think about it.

In this book we already have talked about the specifics and minimal considerations in hardware and software you need in order to set up and run Net.Commerce for NT and AIX platforms. Now we explain the more general factors that will have to be considered for a Web server in the Internet environment. These include:

**Hits per day:** A hit is anything that the Web browser catches whenever a home page, or element within the home page, is accessed. This includes everything from HTML code, graphics, and animations, to multimedia sources such as video or sound. The *hit* has become the main measure by which a Web server's accessibility and functionality are judged. Quite often, however, the number of hits on a Web server reflect the design of the pages, rather than the actual number of people accessing your page. While such numbers may be good for the advertising department, it is recommended to reduce the number of hits per page in order to have faster access.

**Sizes and types of files:** It is always important to know how many resources you need in order to store all the information in your server and what kind of information and applications are going to be placed on the server, that is, whether they are *static pages* or *dynamic pages*. Static pages are those that are going to be stored on the server without making changes to them

every time someone wants to access them. Such pages include most HTML text, graphics, video, sound, etc. Dynamic pages are those that must be created anew whenever someone hits them (CGI, forms, macros, etc.). Dynamic pages will always require many more server resources because the server, instead of returning an existing file, will have to first create the file to be returned.

On average, a WWW transaction takes from 1 to 5 KB if it is only text, 10 KB for text and graphics, 100 KB with multimedia and more than 100 KB for multimedia and applications.

It is important to note here that Net.Commerce most often uses dynamic pages.

With these two factors in mind, namely, hits per day and file type (static or dynamic) we can actually define four sizes of Web servers.

1. **Small Servers:** Normally run around one hundred pages, receive from five to ten thousand hits per day, and are primarily composed of statics pages.
2. **Medium Servers:** These sites typically house several hundred pages, receive from ten to perhaps even a hundred thousand hits per day, and again, are primarily composed of static pages.
3. **Large Servers:** These servers will generally house several hundred or even several thousand pages, receive millions of hits per day, and are composed of multimedia and other types of dynamic pages.
4. **Global Servers:** Global servers generally house many thousands of pages and have links to mirror sites as well. They receive millions and sometimes even tens of millions of hits per day that require multimedia and Web application processing; thus, they must be able to handle a tremendous volume dynamic page requests.

Regarding your Net.Commerce Web site, there are even more factors to consider:

**Number of concurrent users:** It's always good to have at least a rough idea of how the hits on your server will be distributed over a 24-hour period. This will allow you to identify, for example, your peak usage periods for concurrent users, and as a result give you a better idea of your server hardware needs. For example, consider the difference between a server handling one million hits per day spread over 24 hours, and one that has to handle the same volume (or even 80%) over an 8-hour "peak" period. Obviously, the latter scenario would require a much more powerful machine. Of course, concurrent use is not the only parameter you'll have to consider. As noted earlier, file size and type (dynamic or static), as well as peak hit periods must all be considered, and what's more, they must be considered together as part of an interactive process.

For more information about capacity planning, you can visit the following URL:

<http://www.ics.raleigh.ibm.com/capacity/howfast.htm>

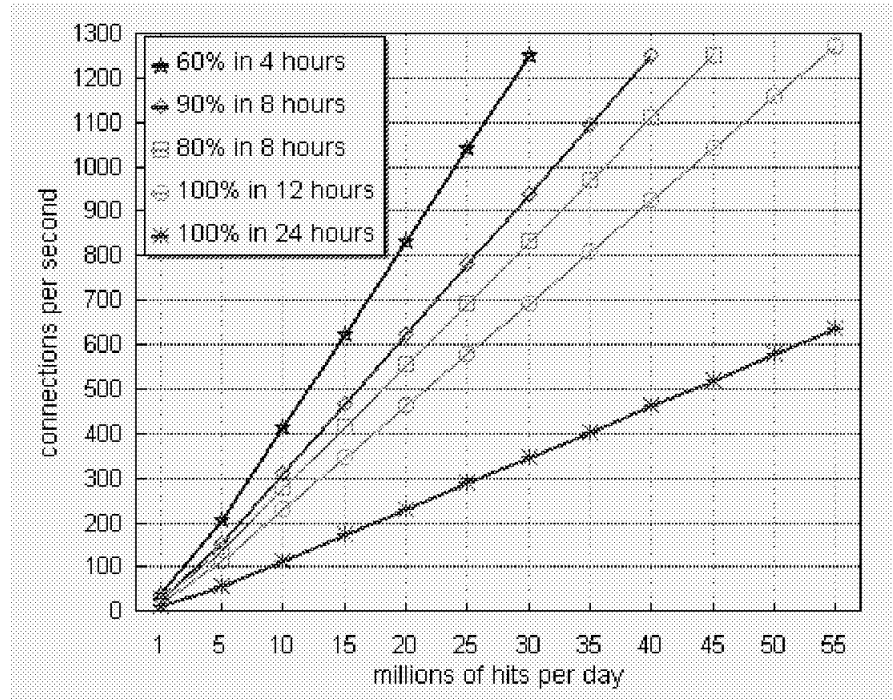


Figure 30. Plotting Web Server Peak Utilization

**Connection speed:** This factor is one of the most important in the performance of an electronic-commerce Web server system, and the one where quite often good results do not depend directly on the design of the solution. Here we have to consider bandwidth, internal network speed, and Internet Service Provider (ISP). The bandwidth recommended for any Web server with commerce applications on the Internet is at least T1. You also need to have good speed in your internal network if the Net.Commerce server will be accessing your legacy systems. Finally, the end user will receive information from your server according to the speed at which his or her ISP allocates it; thus, some factors will always be beyond your control.

The considerations that we have already mentioned are the most important factors that are going to give us good response times from our Web server. These include: hits per day, size and type of files, concurrent users, and connection speed. However, there are other factors that can help you define your needs in more detail.

One of these factors is to know your target audience. For example, if a Net.Commerce Web server is set up on your Intranet for the express purpose of selling internal products and providing special offers to your employees, then you can determine the number of potential users by simply calculating the total number of employees. Similarly, you should also be able to calculate with a fair degree of precision, the peak hours that your system will be accessed.

When designing a system that will require two machines, one for the Web server and another for file storage, the file server will typically require more memory and disk capacity than the Web server. Both the file server and the Web server require fast access to the network, and in some cases they will require extremely fast connections to each other.

Even though it is difficult to know how popular the site may become, you should still think about the potential growth and scalability of your Web server. On average, all WWW transaction sizes increase, so sooner or later you will have to think about increasing the capacity of your network connections to the Internet, the capacity of your server (CPU, disk space, bandwidth, and memory), and if your site becomes really popular, you may even have to consider some secondary machines or create mirror sites. Unfortunately, these really large systems also come with their own set of concerns:

**Homogeneity:** On mirror sites, homogeneity is great for fault diagnosis. It also makes some of the network planning and configuration more straightforward.

**Manageability:** When you have more than one Web server it is necessary to be able to control and monitor a number of systems from a single point by using a single interface.

**Single Footprint:** This is the ability to contain all the backend Web server systems in a single frame.

As you can see there are many factors that you have to consider when setting up a Web server for the Net.Commerce system. Yet, while it will always be difficult to determine the right size for your server, now that you have seen some of these considerations the task of sizing of your Web server should be considerably easier.

---

## 3.5 Network and Security Considerations

### The Internet is not a secure network!

This is why you have to take precautions in order to make your shopping site secure against potential intruders or malfunctioning systems or programs.

For a comprehensive discussion of security on the Internet, refer to *Safe Surfing: How to Build a Secure World Wide Web Connection*, SG24-4564. In this section we discuss the security issues that are relevant to Net.Commerce and to building a shopping site.

In general, you should try to aim for an appropriate level of security, based on the following factors:

1. The threat of an attack by hackers
2. The value of what you are protecting
3. The objective of your security measures

### 3.5.1 The Threat

As noted above, the threat in the Internet must be considered extremely high at all times. This is because there are always people out there trying to get unauthorized access to your systems and applications, not only for commercial gain, but also because they get their kicks that way.

The variety of attack types is extensive.

- Monitoring and capturing data traffic by:
  - Tracing Internet nodes

- Plugging into telecommunications equipment
- Decrypting encrypted data
- Gaining access to systems by:
  - Replacing system components (Trojan Horse attacks)
  - Dictionary attacks (guessing passwords)
  - Spoofing - masquerading as a trusted system
  - Exploitation of system security loopholes
- Denial of service attacks by:
  - Overloading your system (e-mail attack)
  - Redirecting data-traffic

In comparison to the open Internet (the *Wild West Web*), the threat to a company-wide Intranet, which is protected from the outside by firewalls, is more predictable. Because of the lack of anonymity, the probability of an inside attack is relatively small.

### 3.5.2 The Value

The level of security is highly dependent on the value of the data you are dealing with. For example, you will not be very interested in securing your company's profile or product catalog, because you want to make them accessible for everyone.

On the other hand, you will protect your customers' personnel and purchasing information, so that no unauthorized users can access them.

There are three different types of data that are transferred to and from a shopping site.

**Catalog Data:** This is the product catalog including pictures and product descriptions. This type of data is usually not sensitive and is available to everybody on the Internet.

**Pricing and Order Data:** Net.Commerce offers the possibility to define different shopper groups. The members of one group can get different discount rates on products than the members of another group. Therefore prices and information on orders and amounts might be treated as confidential data.

**Payment Data:** The most sensitive type of data is payment data, which includes total prices and credit card information.

### 3.5.3 The Objective

You may have different objectives concerning security. In general, there are five security categories, which all apply to electronic commerce needs.

**Access Control:** Assurance that the person or computer at the other end of the session is permitted to do what he, she, or it is requesting. You need tools to control and restrict access to certain data, like special prices or product pages.

**Identification and Authentication:** Assurance that the person or computer at the other end of the session really is what he, she, or it claims to be. This is closely related to access control. You cannot restrict information to a

certain group of shoppers if you haven't identified and authorized them definitely.

**Privacy:** Assurance that sensitive information is not visible to a snooper, usually achieved through encryption. Here you have to distinguish between the value of your information and the available encryption methods you can use.

**Integrity:** Assurance that the information that arrives is the same as when it was sent. Think about changing order information on its way from the shopper to the merchant.

**Accountability:** Assurance that any transaction that takes place can subsequently be proved to have taken place. This is also known as *non-repudiation*. The goal to achieve is that shoppers cannot deny that they have placed an order via the Internet, and that merchants cannot deny having accepted orders.

You will have to decide which of these objectives apply to you in which cases, depending on the business you are planning to run over the Internet. Usually, you will want to implement a solution that covers all of the security objectives.

## 3.5.4 The Weapons

This section gives you an overview of security mechanisms that exist today and that may be used to make your electronic commerce site safer.

### 3.5.4.1 HTTP Basic Security

This is a simple, but integrated authentication and access control mechanism. It lets you protect certain documents (for example, HTML pages and CGI programs) and permits you to assign access rights to certain users, groups, IP addresses or IP domains.

It is a standard mechanism on almost every HTTP server. For more information on setting up HTTP Basic Security refer to *IBM Internet Connection Secure Server for AIX: Up and Running!*, SC31-8203 and *IBM Internet Connection Secure Server for Windows NT: UP and Running!*, SC31-8278.

The user and password information for accessing the server's pages is kept in special password files, and the passwords are stored on the server in standard UNIX-encrypted format.

The problem, however, is that the user and password information is sent over the network without encryption. It uses a simple encoding technique, called Base64, which is described in the Multipurpose Internet Mail Extensions (MIME) standard (RFC 1521). This encoding mechanism is intended to convert binary data into a form that can be sent through mail gateways that can only handle 7-bit ASCII data, but *not* to protect data.

There is almost no effort necessary to decode this string back to its original format. It's no big deal for a hacker to listen to the string "authentication: basic" on any busy HTTP server or any line and capture and decode user ID and password information.

In addition, the Web browsers are caching the password once typed in and are resending it automatically every time you access a secured site. This increases the risk that the password is being captured.

The bottom line is that although the access control part of HTTP Basic Security is quite suitable for assigning access rights to HTML pages, the encoding of user ID and password is not sufficiently secure for electronic commerce purposes.

Net.Commerce uses HTTP Basic Authentication to protect documents for different groups of shopping site users, such as ordinary shoppers and administrators. For user IDs and passwords, however, a much more secure approach is used.

#### **3.5.4.2 Secure Sockets Layer (SSL) and Secure HTTP (S-HTTP)**

Most HTTP servers, such as IBM Internet Connection Secure Server, support these two security and encryption techniques.

Both protocols address the security issues of privacy, authentication and data integrity. SSL is a technique designed and implemented by Netscape, thus having gained high acceptance throughout the Internet world.

S-HTTP is a secure variant of HTTP developed by Enterprise Integration Technologies (EIT). Although more flexible in applying security options, it is not as widespread as SSL.

Net.Commerce uses SSL for achieving high security. This is why we describe the key features only of SSL in this section. For more information on S-HTTP refer to *Safe Surfing: How to Build a Secure World Wide Web Connection*, SG24-4564 or go to the following URL:

<http://www.eit.com/creations/s-http>

SSL is an alternative to the standard TCP/IP socket API which has security implemented in it. SSL creates a secure tunnel between the two communicating parties so that any TCP/IP-based protocol can be secured. In practice, however, SSL is used only for securing the HTTP protocol.

The SSL protocol consists of two parts:

1. The *SSL Handshake Protocol*, which is used for initial authentication and transfer of the encryption keys.
2. The *SSL Record Protocol*, which is used for transferring the actual user data, encrypted with previously agreed upon ciphers.

SSL uses both asymmetric and symmetric key methods to encrypt data. A functional diagram of SSL is shown in Figure 31 on page 70.

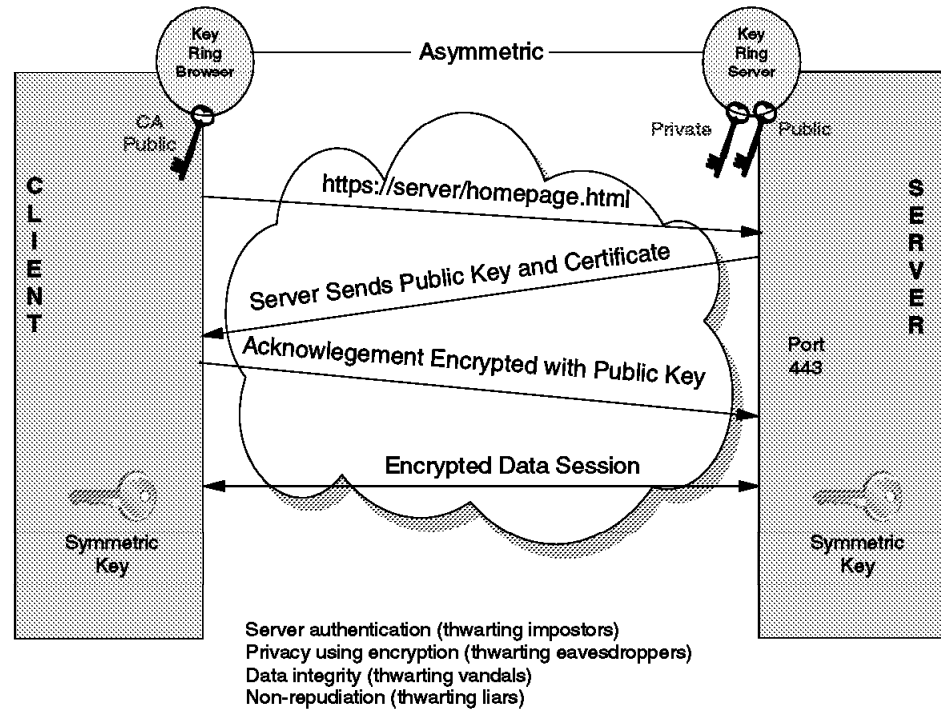


Figure 31. Secure Sockets Layer (SSL)

### 3.5.4.3 Encryption Techniques

The asymmetric public/private key method as defined by RSA is used during the handshake phase to agree upon the symmetric key. The advantage of the public/private key technique is that only the holder of the private key can decrypt what was encrypted by the corresponding public key and vice versa. This ensures that only the server can read what the client sends.

The drawback, however, is that asymmetric key encryption methods need more time to be performed, because they are based on a far more complicated mathematical model. This is why asymmetric methods are not used for encrypting the actual user data (bulk data), but only for exchanging the symmetric key.

SSL can use different symmetric key algorithms for encrypting bulk data, such as DES, RSA RC4 or IDEA.

In addition to using these encryption techniques, SSL also takes advantage of secure hash functions, such as MD2 and MD5 from RSA. These algorithms are designed to provide data integrity.

The principle is that a complex mathematical function, the hash function, generates a certain result, the hashed result, out of a data stream. This result has two characteristics:

- It is impossible (or at least extremely difficult) to reconstruct the original data from this result.
- The result is not predictable. That is, given one set of source data it is extremely difficult to find another set of data with the same hashed result.



If the sender of a data stream includes the hashed result of this data in the message, then it is easy for the receiver to verify the integrity of the data by applying the same hash function to the received data stream and comparing the result with the enclosed result generated by the sender. If the two results match, the data has not been altered.

By applying combinations of the previously described techniques, SSL can provide the following security features:

- Privacy, because the message is encrypted using a symmetric key cipher
- Integrity, because a secured hash function is used
- Server authentication, because the server can decrypt a message that was encrypted using its public key. Note that this relies on the integrity of the server's public key certificate.

For more information on SSL security, including client authentication, refer to *Safe Surfing: How to Build a Secure World Wide Web Connection*, SG24-4564 or have a look at the following URL:

<http://home.netscape.com/newsref/std/SSL.html>

In order to achieve a high level of security, Net.Commerce uses the SSL protocol. This is why you have to set up a secure Internet server (such as ICSS) before you can start setting up Net.Commerce.

#### **3.5.4.4 Regulations Regarding Cryptographic Products**

SSL provides an adequate level of encryption for all kinds of Net.Commerce transactions, when you use it within North America, where a 128-bit key is used. Due to the US export regulations for encryption technologies, SSL is only available using a 40-bit encryption key outside North America. Please refer to 3.6.1.1, "New US Export Regulations for Encryption Keys" on page 77 for more information on US export regulations and other issues regarding cryptographic techniques.

What does this mean?

According to Netscape, "*... a single RC4 40-bit encrypted message takes 64 MIPS-years of processing power to break ...*"

A 64 MIPS year is not too difficult to achieve with today's computer systems, although you need to have access to a cluster of powerful workstations and make them work together.

For a discussion on the strength of the 40-bit SSL protocol refer to:

<http://pauillac.inria.fr/~doligez/ssl/>  
[http://home.netscape.com/newsref/std/key\\_challenge.html](http://home.netscape.com/newsref/std/key_challenge.html)

However, 40-bit SSL might be adequate encryption in some cases. For some other situations 40-bit encryption might not be enough.

As an alternative to SSL, there are some solutions on the market that were not developed in the US and thus are not affected by the US export regulations, such as a new Australian 198-bit proxy-secure server which sits between a 40-bit Netscape SSL server and the Internet. These techniques solve the problem of higher security, but are not a "standard" product such as Netscape's SSL.

### 3.5.4.5 Creating Certificates

When you set up a Net.Commerce site using SSL security (which is used by default), you will need server certificates in order to authenticate yourself to the clients. Usually these certificates have to be ordered from an approved certification authority, such as VeriSign. A functional diagram of the certification process is shown in Figure 32.

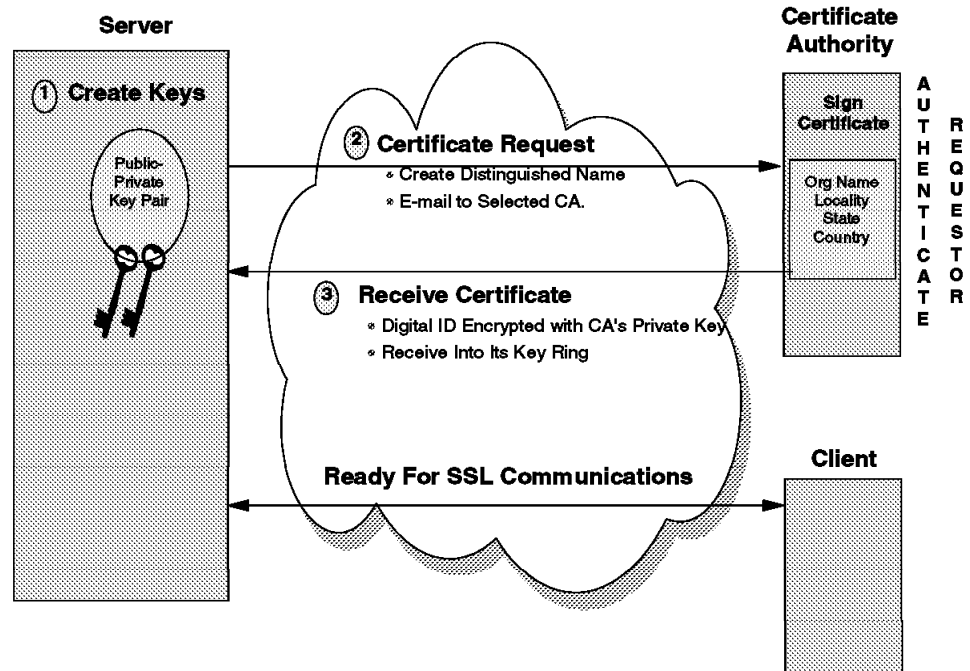


Figure 32. Creating Certificates

We strongly recommend you use a CA such as VeriSign if you are going to run a commercial, secure Web site. For instructions on how to get a certificate refer to *IBM Net.Commerce: Installation and Operation Guide, GC09-2373*.

### 3.5.4.6 Secure Electronic Transactions (SET)

When it comes to payment, the requirements for security are naturally very high. Usually a payment transaction is more than just a simple client to server request. It involves not only the shopper and the merchant, but other parties, such as the credit card provider and the shopper's bank.

This is why VISA and MasterCard commonly agreed upon a protocol, which defines the flow of transactions when making purchases over the Internet using a credit card. This protocol, *Secure Electronic Transactions (SET)*, defines different roles:

**The merchant:** The one who wants to sell something.

**The cardholder:** The one who wants to buy something using a credit card.

**The acquirer:** The organization that provides the credit card service, such as VISA and MasterCard.

**The issuer:** The organization that issued the card to the cardholder. Usually this is the cardholder's bank.

**The acquirer payment gateway:** The interface between the merchant and the existing financial network. Its function is to manage all of the SET functions on the merchant's side and to communicate with the existing legacy systems using their communication protocols and message formats according to the ISO 8583 standard. In fact it is part of the acquirer's system.

**The certification authority:** The organization that issues encryption certificates. SET processing uses public key cryptography, so each element of the system needs one or more public key certificates.

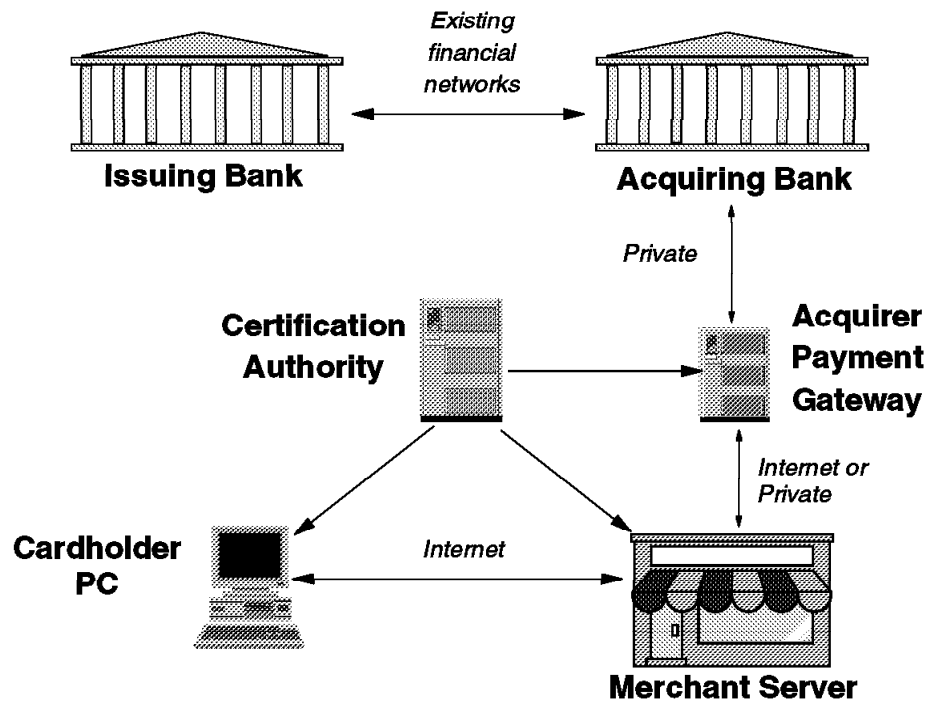


Figure 33. SET Roles

Due to the fact that SET is specifically designed for financial transactions, it uses strong encryption methods. Specifically it uses 1024-bit encryption for bulk data transfer and 2048-bit encryption for transferring the signed certificates.

These high security keys are approved not only for the USA, but also for outside North American use.

At the time of the writing of this redbook, SET products, such as IBM's Payment Gateway and IBM's Net.Registry for SET, were still in beta test.

For detailed information on SET refer to *Safe Surfing: How to Build a Secure World Wide Web Connection*, SG24-4564 and *Secure Electronic Transactions: Credit Card Payment on the Web in Theory and Practice*, SG24-4978, or go the following URLs:

<http://www.visa.com/cgi-bin/vee/sf/standard.html>  
<http://www.internet.ibm.com>

### 3.5.5 Firewall Considerations

Secure your Intranet and set up firewalls to deny access from outside.

There are three main methods of controlling access from the outside:

- Filtering routers: Normal IP routers with rules defined to control what packets will pass and what packets will be blocked.
- Bastion: A firewall in which sessions do not pass through, but instead are stopped on one side and restarted on the other. This can be called a dual-homed gateway.
- Combination: Normally, a firewall will be comprised of two or more systems and deployed in a combination of the two basic types. In this configuration, a LAN segment exists between the two firewalls, and is known as the De-Militarized Zone (DMZ). Only Web and FTP servers should be in the DMZ. Only Web and FTP servers should be in the DMZ.

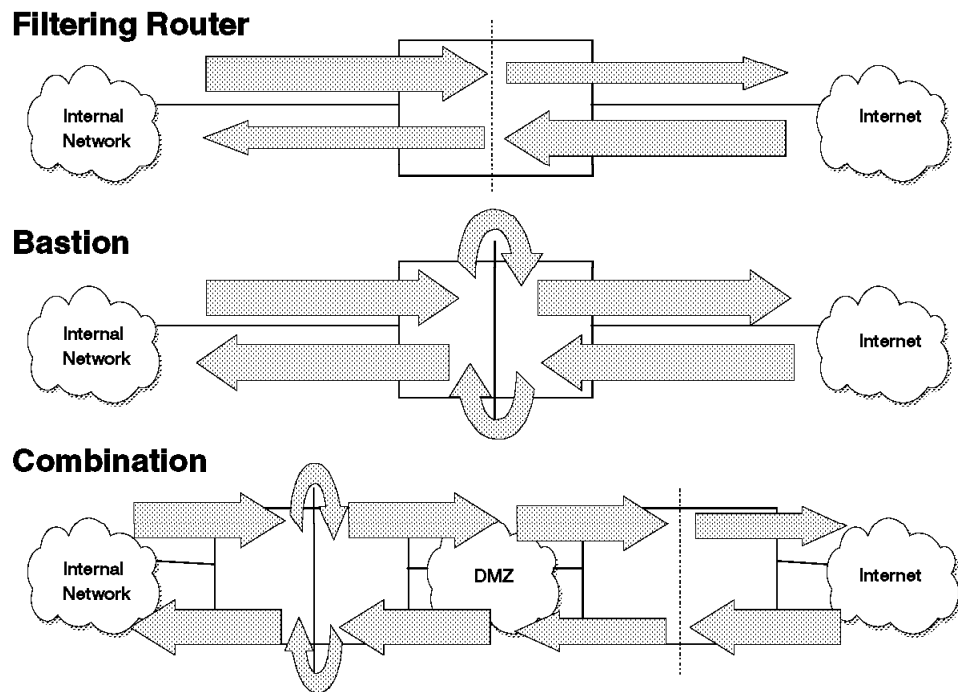


Figure 34. Firewall Models

To set up the firewall you can:

- Set up a single server with local administration.
  - Establish sessions for clients in the secure network to servers in the Internet. The objective is that these should not be direct (which would give an outsider information about the resources in the secure network), but should be broken by a proxy application at the inner firewall.
  - Sessions from outside clients to your servers are allowed to pass through the filters on the outer firewall, but blocked by the inner firewall. Only legitimate sessions will be allowed.

- Peer-to-peer sessions between servers such as SMTP and DNS are allowed through to the outer filters, but broken at the inner firewall. A relay application permits only the information that is necessary to reach the external nodes.
- Administrative sessions are passed through the inner firewall. These may be manual connections (maintaining the servers) or business data feeds (databases, for example). They may either be filtered through the inner firewall (but blocked at the outer one) or passed through using a proxy application.

## Single Server with Local Admin

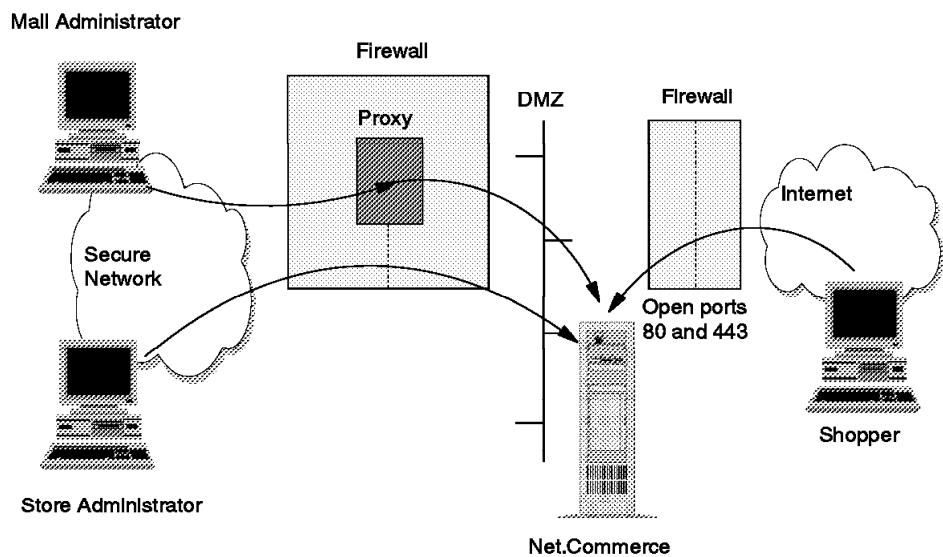


Figure 35. Simple Firewall Setup

- Separate database inside the firewall with outside administration
  - Sessions from the client in the secure network to servers in the Internet. The objective is that these should not be direct (which would give an outsider information about the resources in the secure network) but should be broken by proxy application at the inner firewall.
  - Sessions from outside to your server are allowed to pass through the filters on the outer firewall, but blocked by the inner firewall. Only legitimate sessions will be allowed.
  - Peer-to-peer sessions between servers such as SMTP and DNS are allowed through the outer firewall filters, but broken at the inner firewall. A relay application permits only the information that is necessary to reach the external nodes.
  - Administrative sessions are passed through the inner firewall. These may be manual connections (maintaining the servers) or business data feeds (databases, for example). They may either be filtered

through the inner firewall (but blocked at the outer one) or passed through using a proxy application.

## Separate Database Inside Firewall with Outside Admin

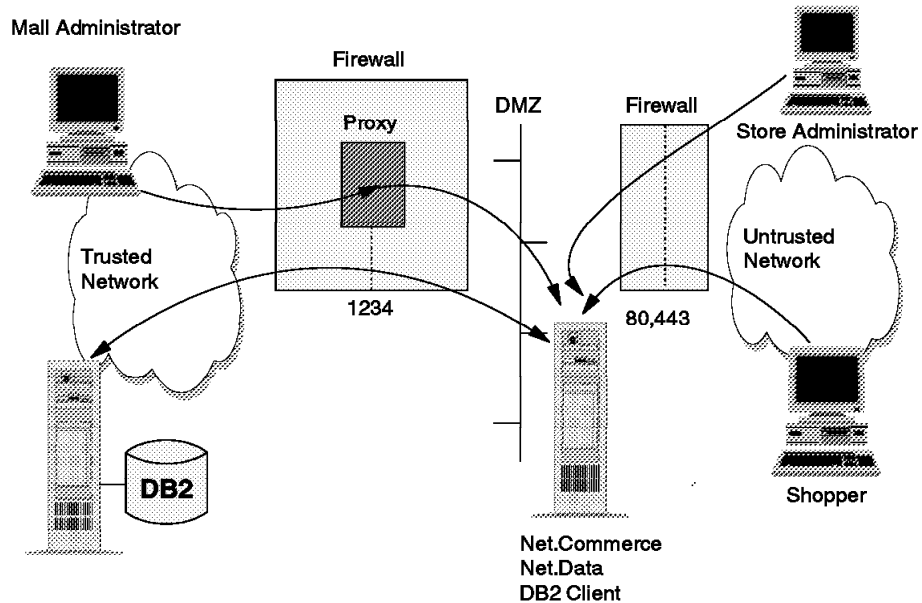


Figure 36. Complex Firewall Setup

### 3.5.6 Additional Security Approaches

In addition to the previously mentioned solutions, there are many more possibilities to make a shopping site secure. The following list gives some more suggestions on security, which should be taken into account, when setting up a secure site:

**Secure your endpoints:** Turn off all unnecessary services on your servers, such as FTP or Telnet. Remove direct user access from the Web server and from any database server.

**Change administrator passwords frequently:** Follow the discussion fora on the Internet and watch for security bugs.

**Separate your components:** Keep your systems as simple as possible. Do not try to put all functions (such as Web server and database server) on a single machine.

**Always use extensive logging:** Log everything that moves, even "normal" events. Remember that you want to ensure non-repudiation. Besides, you may want to track what happened on your system when something fails. Systems Management applications, such as NetView, can help you here.

**Hardware-based approaches:** If the above mentioned software solutions are still not enough, you can consider hardware-based cryptographic solutions, such as SmartCards, token devices, or the IBM 4755 Cryptographic Adapter.

---

## 3.6 Examples of Aspects of Local Legislature

In this section we briefly look at some legal implications that you may have to face when you intend to set up a Net.Commerce site.

Depending on the state or country where your store or mall is located, you are subject to local laws that regulate electronic commerce, if such laws are in place. It is your responsibility as the planner or owner of an electronic shopping site to obtain information on such regulations before you go online with your system. Places to turn to for more details on that matter would be, among others:

- City, County or District Chamber of Commerce
- Local association of lawyers
- State library of senate or congress, or equivalent, where all bills can be found that have either passed or otherwise been brought forward to those legislative bodies.

No matter what products you are going to sell over the Internet, it is more than likely that there are certain regulations in place and applicable to your business. The legal issue is especially important if you plan to install sites in different states or countries or if you are operating in a multi-cultural society where several official languages are spoken and/or required. If you are uncertain if or to what extent your online store would be affected by local legislature we strongly advise that you contact a lawyer.

We have done a little research on that matter and found some examples which we have included in the following sections. We have also tried to separate those issues that are primarily related to Internet commerce from such regulations that affect any kind of commercial interaction between merchants and customers.

### 3.6.1 Regulations for Electronic Commerce on the Internet

Listed below are examples of legal acts dealing specifically with electronic commerce or information exchange over the Internet

#### 3.6.1.1 New US Export Regulations for Encryption Keys

On October 1, 1996, the Clinton Administration announced a plan to make it easier for Americans to use stronger encryption products to protect their privacy, intellectual property and other valuable information. The plan envisions a worldwide key management infrastructure with the use of key escrow and key recovery encryption items to promote electronic commerce and secure communications while protecting national security and public safety. To provide for a transition period for the development of this key management infrastructure, this rule permits the export and re-export of 56-bit key length DES or equivalent strength encryption items under the authority of a License Exception if an exporter makes satisfactory commitments to build and/or market recoverable encryption items and to help build the supporting international infrastructure. This policy will apply to hardware and software.

Dates:

- Effective Date: This rule is effective December 30, 1996.
- Comment Date: February 13, 1997.

### **3.6.1.2 California Assembly Bill 3320 (1996)**

This bill regulates sales over the Internet, and other electronic media, in the state of California. It is the expansion of an existing bill regulating the sale, lease, or offering for sale or lease of goods or services by telephone, mail order, or catalog. The bill requires a vendor conducting business through the Internet or any other electronic means of communication to make specific disclosures to a buyer.

A merchant in California who wants to do business over the Internet would therefore have to do, among other required actions, one of the following within 30 days (or an otherwise clearly specified period of time) of payments received:

- Ship ordered goods or services.
- Mail a full refund or credit the account where payments have been transferred from if merchandise is not available or has not been shipped.
- Send a letter to the buyer advising of a delay in shipment, or offering a substitution of the goods or services ordered (which would have to be of equivalent or superior quality), or offering a full refund.

When a merchant outside the state conducts business electronically with a buyer in California, the merchant would then have to disclose to the buyer, in writing or electronically, the merchant's return and refund policy, the legal name under which the business is conducted, and under certain circumstances, the business address of the merchant.

It is likely that similar legislation will be introduced in other states and countries.

The full text of the bill can be obtained from the California Legislature Web site at the following URL:

<http://www.leginfo.ca.gov/cgi-bin/postquery?house=A&sess=PREV&number=3320>

### **3.6.1.3 Requirements for the Use of Cryptographic Software in France**

In France, cryptographic systems are regarded as weapons.

According to the law, any product capable of enciphering/deciphering user data should be granted a license from the French administration government service named Central Service for Information Systems Security (SCSSI) before being marketed. Then customers need to be authorized to use them on a case-by-case basis. It usually takes around six months before any such authorizations are granted.

In reality, two major and useful exceptions exist:

1. Routinely, licenses are granted that allow banks to use DES products on a global basis (no case-by-case authorization needed).
2. Routinely, global licenses are granted that allow anybody to use weak encryption (RC2/RC4 with 40-bit keys).

Although the ICSS 4.1 authorization request was submitted on August 1, 1996, it had not been approved when this book was published. As a result thereof, there is no general authorization for the use of Net.Commerce in France yet. Only one authorization has been granted for loaning Net.Commerce to a specific project. Any other loan is illegal.



In addition to that, as the new release of ICSS, Version 4.2, involves a more robust cryptographic algorithm, a new authorization request for that software is required. At the moment, you have to carefully review the issue of electronic commerce in France or with any business partners in France because the products are evolving faster than authorizations can be obtained.

When you plan or have a business need to exchange ciphered data with French partners, you have to check which options you are allowed to use. Normally 56-bit DES keys are allowed for banks and 40-bit keys are allowed for other users. Currently, it is also uncertain when DES keys will be allowed for SET protocols.

**Note:** If you plan to implement an electronic shopping site in France using Net.Commerce Version 2, you have the option of selecting a Web server other than IBM ICSS, for instance Netscape Enterprise Server which is authorized in France.

#### **3.6.1.4 Acceptance of Electronic Transactions As Court Evidence**

Other issues relate to repudiation of a transaction and electronic contracts in court. To date, no one has taken anyone to court as far as we are aware, but an electronic contract was still non-binding, for instance, in South Africa at the time this book was written. This issue has been raised by contacts from other countries as well.

### **3.6.2 General Regulations for Commerce**

Listed below are examples of legal acts dealing more generally with business requirements for commercial enterprises.

#### **3.6.2.1 National Language Requirements in Quebec, Canada**

Canada has two official languages, French and English. The majority of French speaking people in Canada reside in the province of Quebec, and the Quebec provincial government pays close attention to French language issues.

The laws affect merchants who wish to do business in their province. If a business has more than 35 employees in the province, they must provide the ability to do business in French. This affects merchants who have stores in the province. They may be requested by the Quebec government to provide shopping services (both the products in the store, which is the responsibility of the merchant, and the site itself, which is the responsibility of the site host, and includes the shopping basket and checkout capabilities) in French.

Although this does not prevent merchants from opening stores and going online, written requests from customers have been received in some cases to provide this service, with the expectation that this function would be available in the near future.

In such environments, the ability to select a language for checkout, driven by a table of terms such as "add to shopping basket", "name", "address", etc. would be a differentiating feature to a shopping site.

### **3.6.2.2 Bottle Return and Refund System in Sweden**

According to our information, a merchant who sells bottled goods, such as mineral water or juice, in Sweden is required by law to provide means to take back empty bottles and refund a certain amount of money to the customer who returns the bottles. Regulations like that are likely to be in place in other countries.

In this case, and in comparable situations, you have to develop your own macros and APIs for Net.Commerce because the system does not provide you with routines to handle such return and refund situations.

### **3.6.2.3 Inclusion of Applicable Taxes in Product Prices**

Some countries require, very much in contrast to the U.S., that the price for any item sold must include all applicable taxes and duties so that it constitutes the effective amount of money that a customer will have to pay for that item. Some countries also require merchants to explicitly specify any applied tax rates.

### **3.6.2.4 Monetary Control in South Africa**

Monetary control has the affect that any monies leaving the country, typically via the Internet, would first have to be cleared by the Central Reserve Bank of South Africa. This method is and has been found to be cumbersome, and as such is generally ignored, but it still is a legal requirement.

---

## Chapter 4. Installation and Configuration

This chapter describes the installation and basic configuration of the Net.Commerce systems that have been used throughout this project. That includes the following tasks:

1. For the AIX system:
  - Preparation
  - Installation of the system
  - Basic configuration
  - Verification of the installation
  - Installation the demo mall
  - Uninstallation of the system
2. For the Windows NT systems:
  - Preparation
  - Installation of the server and database systems
  - Basic configuration of the systems
  - Verification of the installation
  - Basic performance tuning
  - Installation the demo mall
  - Configuration of ODBC support
  - Uninstallation of the systems
3. Accessing the database with Lotus Approach

For a more detailed discussion of the Net.Commerce installation and configuration process, please refer to *IBM Net.Commerce: Installation and Operation Guide*, GC09-2373.

---

### 4.1 Installation and Basic Configuration on AIX

This section describes the installation and basic configuration of the Net.Commerce system on a single AIX system.

#### 4.1.1 Preparation

As described in 1.2, "Net.Commerce Overview" on page 2, all required software components (DB2, ICSS, Net.Commerce) are part of your Net.Commerce software package. For the specific system requirements, please refer to 3.1.5.1, "AIX System Requirements" on page 45 and 3.3.1, "Database Space Requirements" on page 49.

For our scenarios, we used Net.Commerce Version 1.1 which is a Fix Pak to Version 1.0. You can download the Fix Pak #1 for Net.Commerce from the following URL:

<http://www.internet.ibm.com/commercepoint/net.commerce/upgrades.html>

Before you start the installation of Net.Commerce, you have to create two groups and one user account for DB2:

**Note:** This user account ID must not exceed 8 characters or DB2 will fail to install.

1. From SMIT, go to **Security & Users** and select **Groups**, then select **Add a Group**. A panel will appear as shown in Figure 37 on page 82:

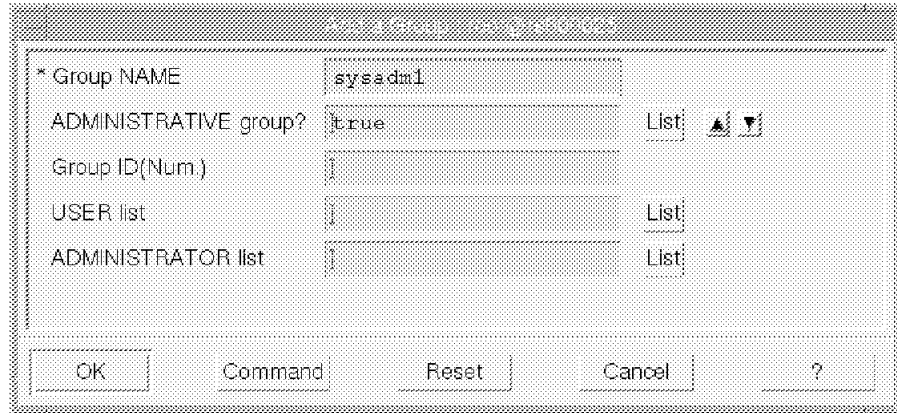


Figure 37. Creating an AIX Group for DB2

2. Fill in the group name as sysadm1.
3. Select **true** in the administrative group field, then click on **OK**.
4. Create another group and call it dbadmin1.
5. Go back to Security & Users and select **Users**, then select **Add a User**. A panel will appear as shown in Figure 38:

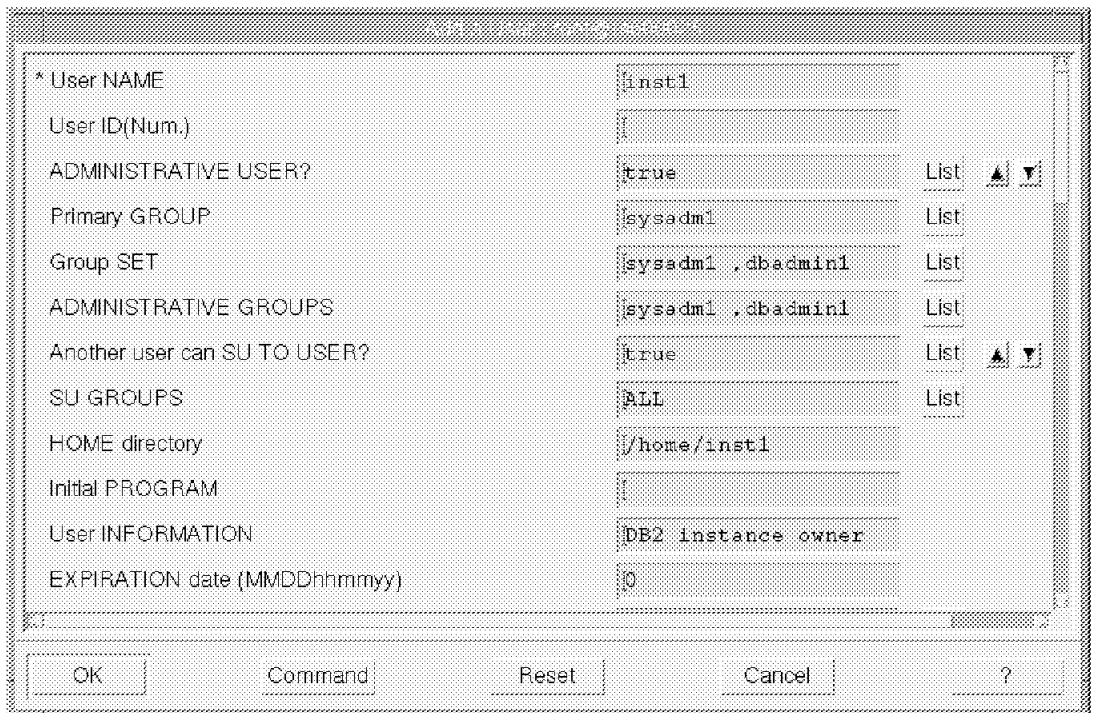


Figure 38. Creating an AIX User for DB2

6. Fill in the user name. In our case it is inst1 which is the default database instance name for DB2.
7. Select **yes** in the Administrative user field.
8. Select **sysadm1** as the administrative group for that user and make the user a member of the sysadm1 and dbadmin1 groups.

9. Select a home directory for that user and ensure that there is enough disk space available to hold the Net.Commerce database which will eventually be created there.
10. Click on **OK**.
11. Select **Change a User's Password** to set an initial password for that user.
12. Click on **OK**.

## 4.1.2 Installation of the Files

On this system, you need to install the following components:

1. DB2 server
2. ICSS to allow shoppers to access your system and to configure Net.Commerce
3. Net.Commerce server

Normally, you would install all components from the Net.Commerce CD-ROM using the integrated setup program. However, since our scenario required an installation that was different from the defaults, we installed each component separately. The table below illustrates the differences between the default installation and our setup:

ITSO Scenario	Default Installation
Net.Commerce 1.1 (with Fix Pak #1)	Net.Commerce 1.0

### 4.1.2.1 Installation of DB2 and Internet Connection Secure Server

Follow the instructions below to install the necessary DB2 and ICSS components from your Net.Commerce CD-ROM.

**Note:** We tried to use ICSS 4.2 because it offers, among a series of new features, a better performance than ICSS 4.1 which ships with the Net.Commerce product, but ICSS 4.2 is not compatible with Net.Commerce 1.1.

1. Mount the CD-ROM of your system over an available file system, in our case /cdrom.
2. From SMIT, go to **Software Installation and Maintenance** and select **Install and Update Software**, then select **Install and Update from ALL Available Software**.
3. As a source for the software to install, enter the /netcommerce directory of your CD-ROM, in our case /cdrom/netcommerce.
4. Click on **List** on the Software to install line.
5. From the list of available software, select the following:
  - db2\_02\_01 - ALL
  - internet\_server.Bnd
  - internet\_server.base
  - internet\_server.msg.en\_US
  - internet\_server.security.common
  - internet\_server.security.US\_secure
  - ipfx.rte

- ipfx.msg.en\_US
- ipfx.msg.En\_US

This is shown in the figure below:

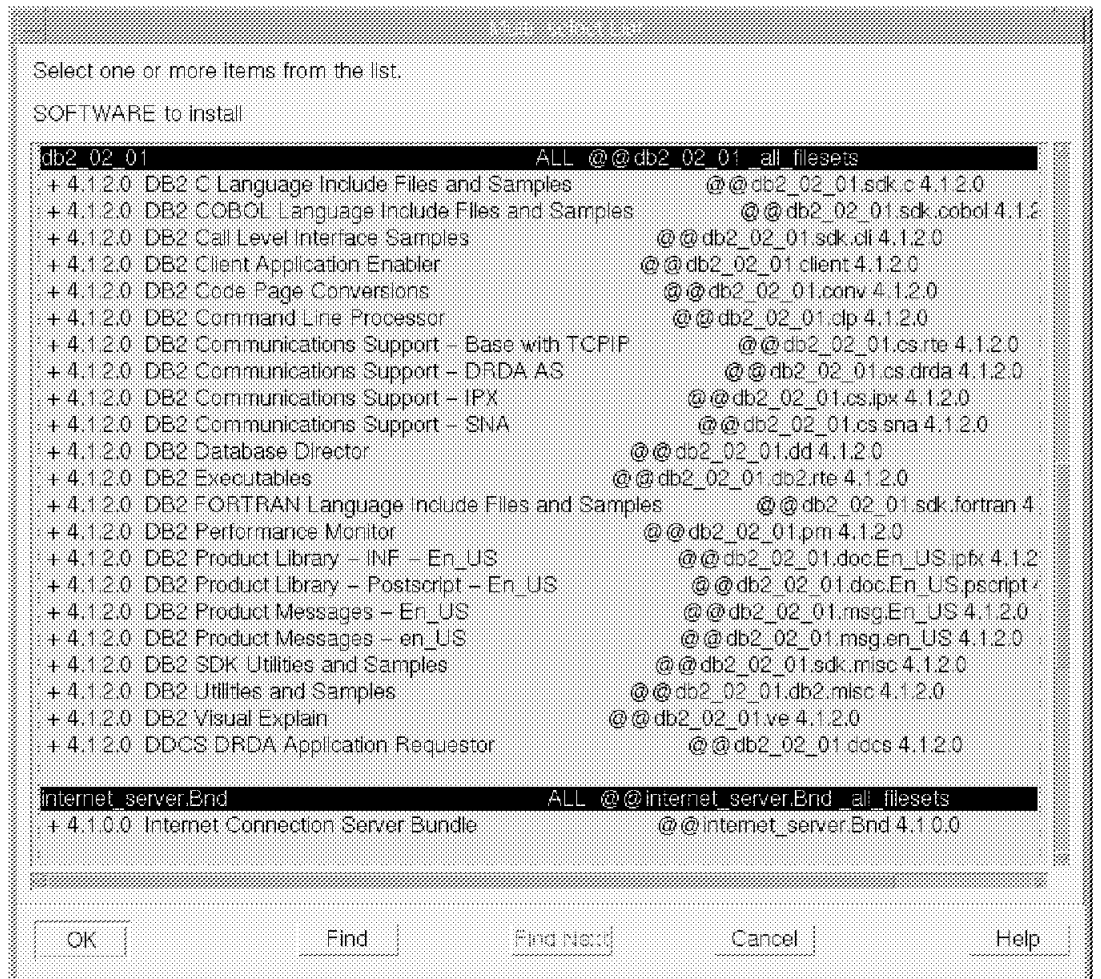


Figure 39. Installing Net.Commerce Components on AIX

**Note:** Do *not* select any Net.Commerce components as they will be installed from the Fix Pak #1 files.

6. Click on **OK** to confirm the selection. A panel will appear as shown in the figure below:

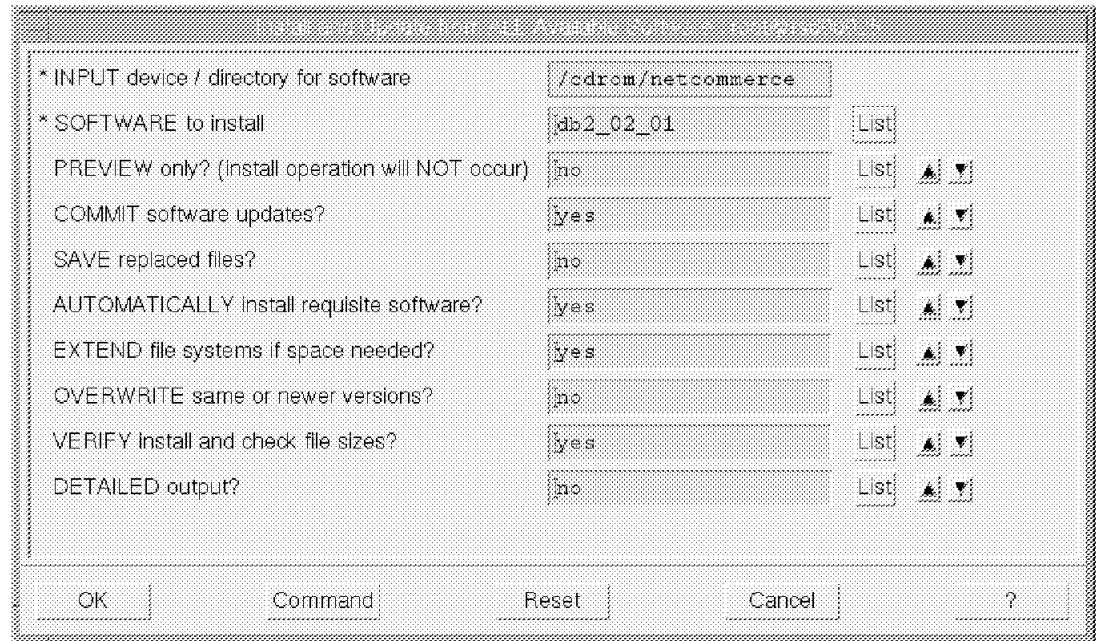


Figure 40. Installation Options on AIX

7. Select **yes** in the Verify field to allow the system to check for available disk space before any files are installed, and also select **yes** in the Commit field to have the software applied to the system permanently.
8. Click on **OK** to start the installation.
9. Click on **OK** to confirm.
10. When the installation has completed, unmount the CD-ROM drive.

#### 4.1.2.2 Creating the Database Instance Environment

You need to activate the DB2 environment for the instance owner before you can proceed.

1. Change to the /home/inst1/sqllib/bin directory, where inst1 is the name of your DB2 instance owner, and enter the following command from an AIX command prompt to check if the DB2 instance has been properly created:

```
./sqlist
```

2. If no instance is displayed, create one using the following command:

```
./db2icrt inst1
```

3. Change to the /home/inst1 directory and issue the following command:

```
echo ". sqllib/db2profile">>/home/inst1/.profile
```

That will initialize the DB2 environment whenever a user or a process logs in as inst1.

4. Log in as inst1 using the su - inst1 command.
5. Enter the following commands to check if DB2 works properly:

```
db2 start db manager  
db2 get db manager cfg  
db2 stop db manager
```

### 4.1.2.3 Installation of Net.Commerce

Follow the instructions below to install the Net.Commerce V1.1 files:

1. Receive the Net.Commerce Fix Pak #1 file into a temporary directory, in our case /usr/tempnc.
2. Uncompress and unpack the file.
3. From SMIT, go to **Software Installation and Maintenance** and select **Install and Update Software**, then select **Install and Update from ALL Available Software**.
4. As a source for the software to install, enter the temporary directory where the Fix Pak #1 files have been expanded.
5. Click on **List** on the Software to install line.
6. From the list of available software, select the following:
  - Net.Commerce.server
  - Net.Commerce.html
  - Net.Commerce.loc.en\_US
  - Net.Commerce.loc.En\_US
  - Net.Commerce.msg.en\_US
  - Net.Commerce.msg.En\_US
  - Net.Commerce.ps
  -
7. Select the **yes** option for the Verify and Commit fields again.
8. Click on **OK** to start the installation.
9. Click on **OK** to confirm.

## 4.1.3 Basic Configuration

The following configuration tasks are required on this system:

1. Create security keys and certificates for ICSS
2. Configure the Net.Commerce server
3. Create the Net.Commerce database

### 4.1.3.1 The Certification Process

You need to request a signed certificate from a certification authority and create a key ring file that contains the certificate that will be signed. This process is illustrated in Figure 41 on page 87.



## http://<machine\_name>/admin-bin/cfgin/initial

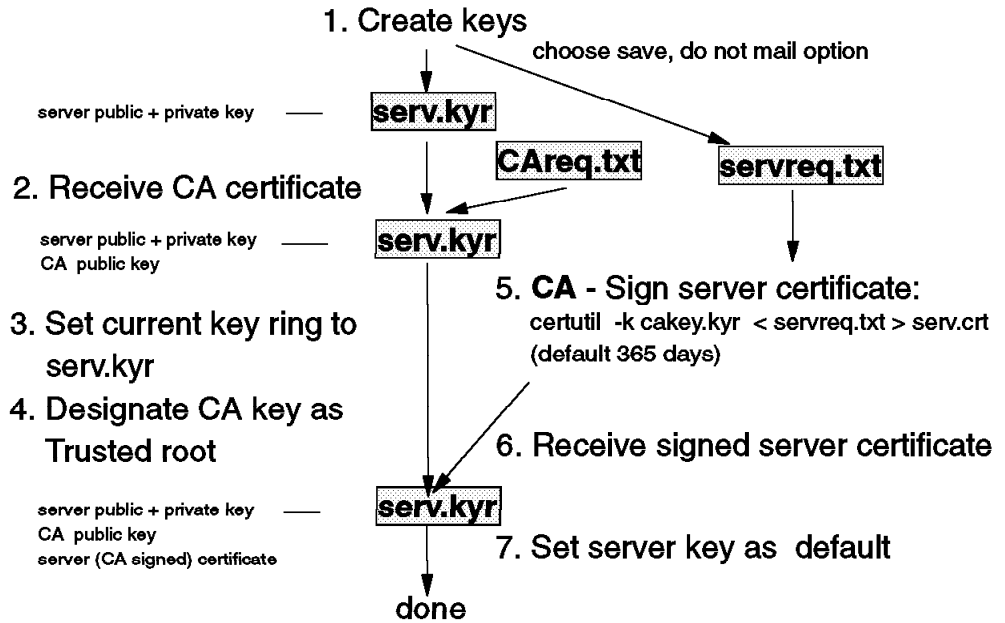


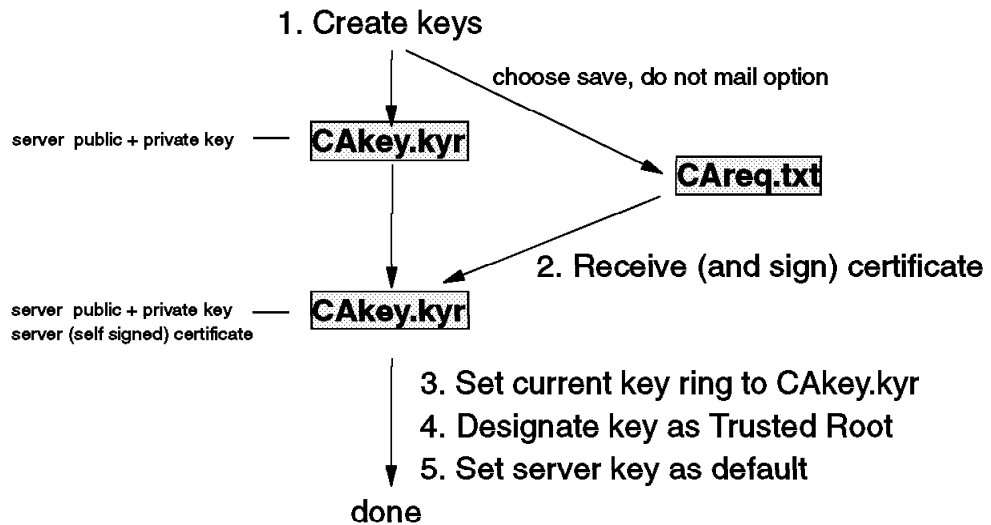
Figure 41. Certification Process Using a Certification Authority

Obtaining a security certificate signed by a certification authority is the process used to verify identity and to make sure that others are who they say they are. You, as the receiver, base your trust for the authentication of the sender and a third party (person or agency) that certified the sender. The third party is the certification authority (CA). Once you have requested and received the certificate you need a file in which the secure server will keep the certificate, public and private keys, passwords and trusted root keys.

To request this certificate and create the key ring file you can do it from the administration and configuration forms from the ICSS.

For testing purposes or Intranet-only installations of Net.Commerce, it is easier to use self-signed certificates. In that case you don't have to wait for a CA to send the signed certificate back to you, but you act as a CA yourself and sign your own certificate. ICSS allows you to do that, and the process is illustrated in Figure 42 on page 88.

**http://<machine\_name>/admin-bin/cfgin/initial**



**CA uses same steps (except 5) to create root certificate**

Figure 42. Certification Process Using Self-Signed Certificates

**Note:** Some browsers, such as Microsoft Internet Explorer V2.0, do not support self-signed certificates.

#### 4.1.3.2 Configure and Enable ICSS Security

For detailed information about how to create the key ring file and the certificate you can see *IBM Net.Commerce: Installation and Operation Guide*, GC09-2373.

For our scenario, it was sufficient to use self-signed certificates, so we describe that process below.

**Note:** This process is identical to AIX and Windows NT with the exception of the location of the key ring and certificate files.

1. Start your Web browser, if it is not running already.
2. Disable all caching (memory and file) and also disable the use of proxy and/or SOCKS servers.
3. Start ICSS, if it is not running already. To check if ICSS is running, enter the following command on an AIX command prompt:

```
ps -ef | grep httpd
```

To start ICSS, enter the following AIX command:

```
startsrc -s httpd
```

4. Go to URL `http://<your_hostname>/Frntpage.html` to access the ICSS configuration pages on your Net.Commerce server, where `<your_hostname>` is the TCP/IP domain name of your server.
5. Click on **Configuration and Administration Forms**.

6. Log on with a user ID of webadmin and a password of webibm which are the default values for the ICSS administrator.
7. Scroll down to Security and click on **Create Keys**.
8. Select **Other** and **Apply**.
9. Change the length of the encryption key to the largest available size. Also enter the name of the key and the key ring file.
10. Scroll down and enter the key ring password and also check **Automatic login**.  
All of this is shown in Figure 43:

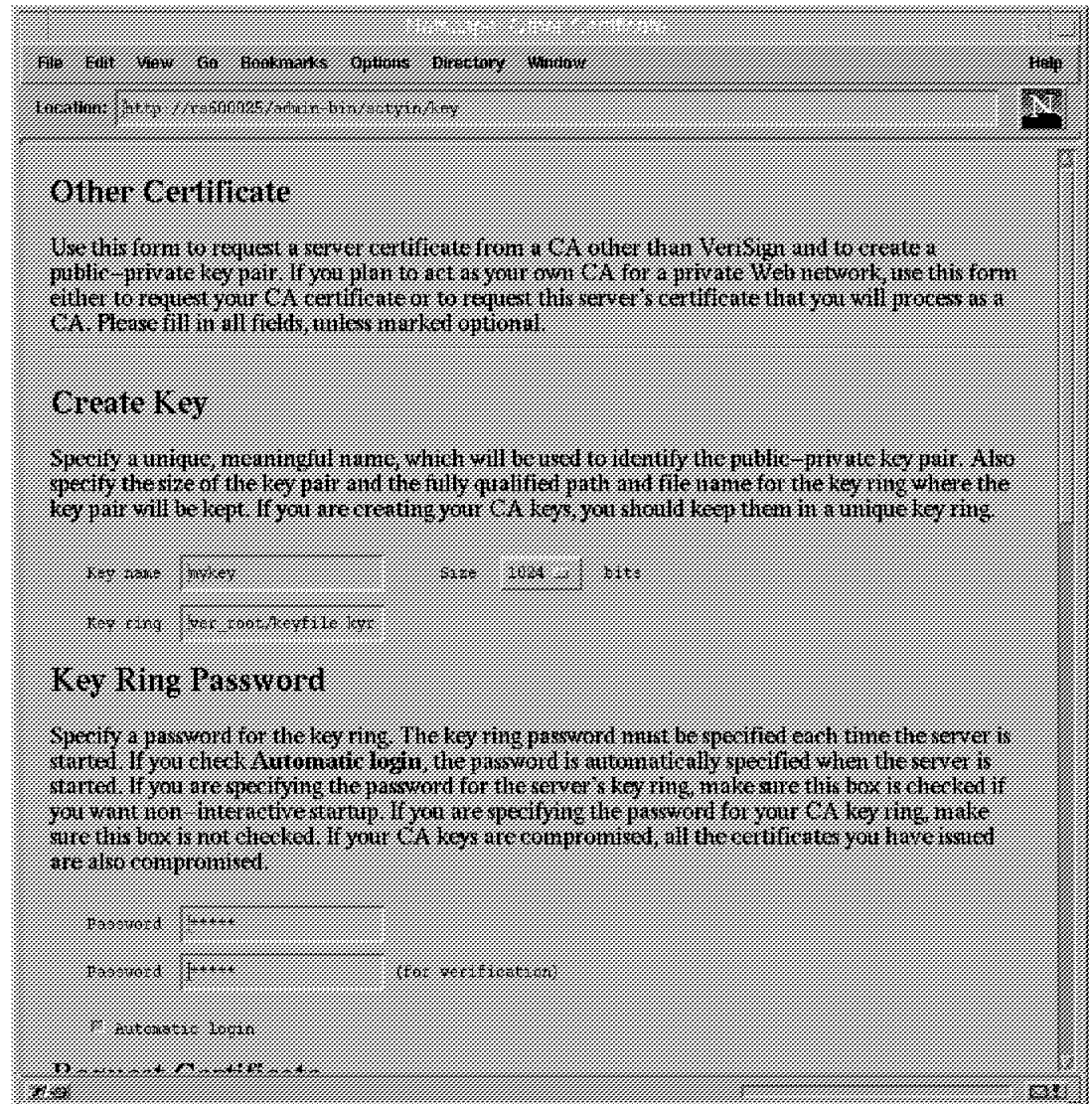


Figure 43. Creating Encryption Keys, AIX

11. Scroll down and enter the information about your server as shown in Figure 44 on page 90. This will be included in the certificate.
12. Scroll down and select **Don't mail** and enter the name of the file for the certificate request.

File Edit View Go Bookmarks Options Directory Window Help

Location: http://rs60025/admin-bin/actyin/key

### Distinguished Name

The Distinguished Name is a unique name that is associated with the certificate and public key. For this certificate, the Distinguished Name is the Server name and the location of the server. Server name is the X.500 common name. It is usually the fully qualified TCP/IP host name.

Server name	rs60025.itsa.ral.ibm.com	
Organisational unit	ITSa	(optional)
Organisation	IBM	
Locality/City	Gary	(optional)
State/Province	North Carolina	(minimum three characters)
Postal code	27513	
Country	US	

User's e-mail address should contain the address where you want the CA to mail the certificate.

User's e-mail address nobody@rs60025.itsa.ral.ibm.com

### Mail To

Choose the mailing option recommended by your CA. If you are acting as your own CA to request your CA certificate or to request this server's certificate that you plan to process as CA, choose **Don't mail**.

Mail to 
  
 Don't mail

**Save Copy**

Figure 44. Key System Information, AIX

13. Scroll down and click on **Apply**. A confirmation page will be displayed.
14. Return to the Configuration Page and select **Security Configuration**.
15. Make sure that both S-HTTP and SSL connection are enabled.
16. Scroll down and select a key ring file to be the current one, as shown in Figure 45 on page 91:

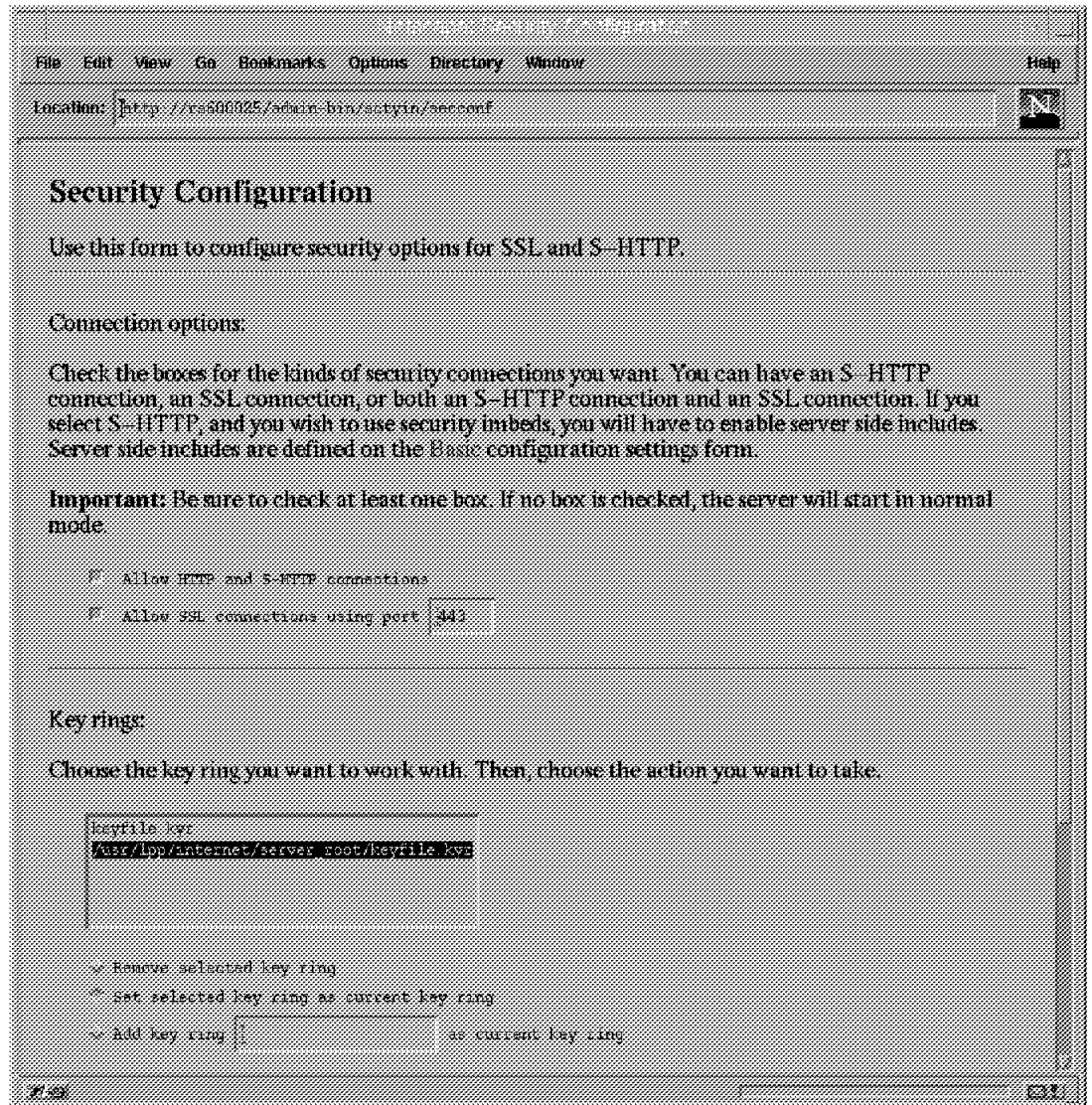


Figure 45. Current Key Ring File, AIX

17. Click on **Apply**. A confirmation page will be displayed.
18. Return to the Configuration page and select **Receive Certificate**.
19. Enter the name of the certificate request and key ring files as well as the key ring password, as shown in Figure 46 on page 92:



Figure 46. Receive Certificate, AIX

20. Click on **Apply**. A confirmation page will be displayed.
21. Return to the Configuration page and select **Key Management**.
22. Enter the key ring password and select **Designate Trusted Root Keys**.
23. Select a key to be designated as trusted root key as shown in the figure below:

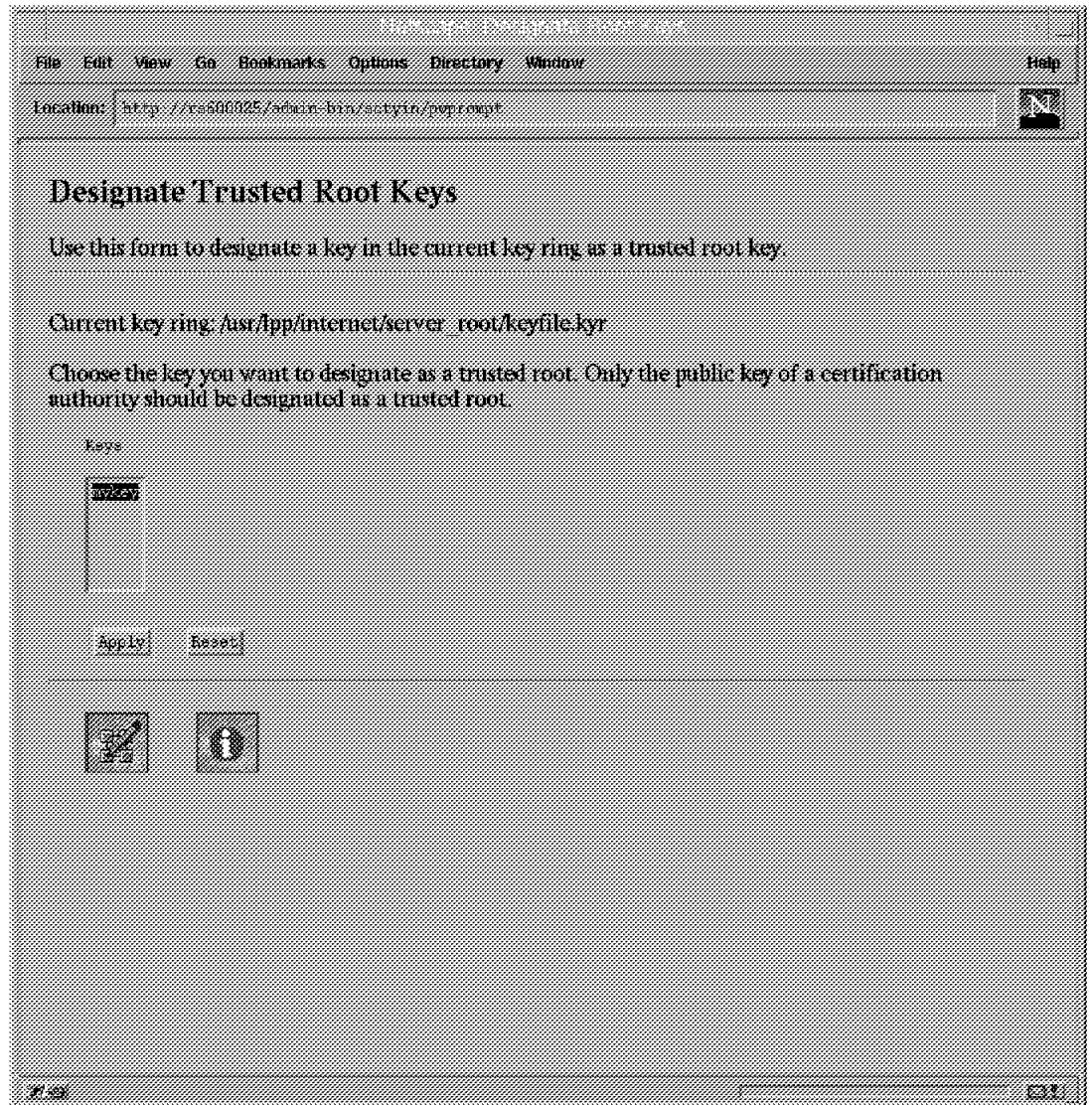


Figure 47. Designate Trusted Root Key, AIX

24. Click on **Apply**. A confirmation page will be displayed.
25. Return to the Configuration page and select **Key Management** again.
26. Enter the key ring password and select **Manage Keys**.
27. Select the key you have created previously to be the default, as shown in Figure 48 on page 94:

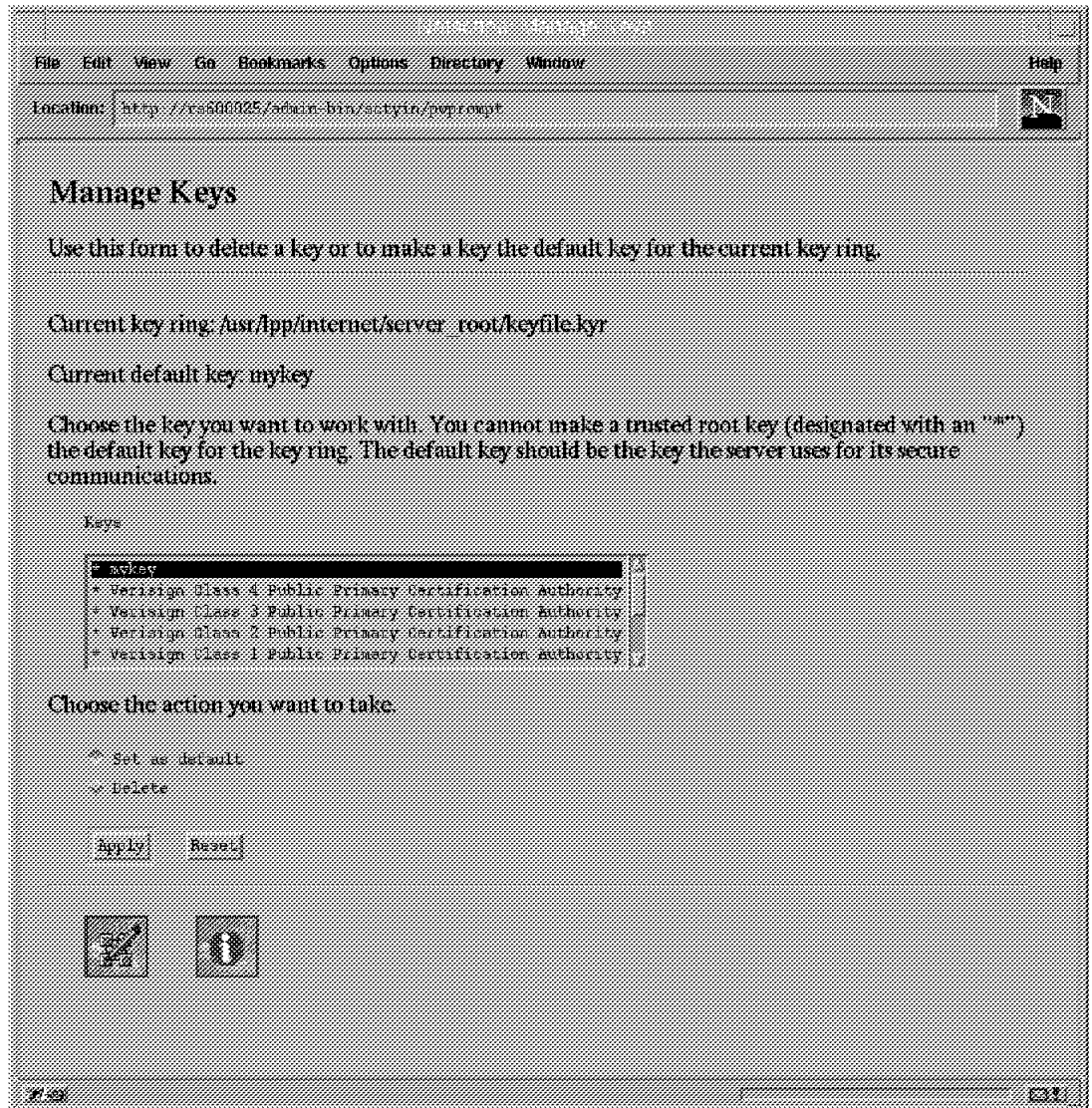


Figure 48. Select Default Key, AIX

28. Click on **Apply**. A confirmation page will be displayed.
29. Stop and restart ICSS using the following commands from an AIX command prompt:

```
stopsrc -s httpd
startsrc -s httpd
```

#### 4.1.3.3 Configure the Net.Commerce Server

Follow the instructions below to configure Net.Commerce:

1. Start your Web browser, if it is not running already.
2. Disable all caching (memory and file) and also disable the use of proxy and/or SOCKS servers.
3. Start the **Administration Web Server**, a separate ICSS foreground process which is used for Net.Commerce configuration, by entering the following AIX commands:

```
cd /usr/lpp/NetCommerce/server/bin
./start_admin_server
```



4. Go to URL `http://<your_hostname>:4444` to access the Net.Commerce Administration server pages, where `<your_hostname>` is the TCP/IP domain name of your server, and 4444 is the TCP/IP port at which the server is listening.
5. Log on with a user ID of `webadmin` and a password of `webibm` which are the default values for the ICSS administrator. You will be presented with the Net.Commerce Server Manager page as shown in the figure below:

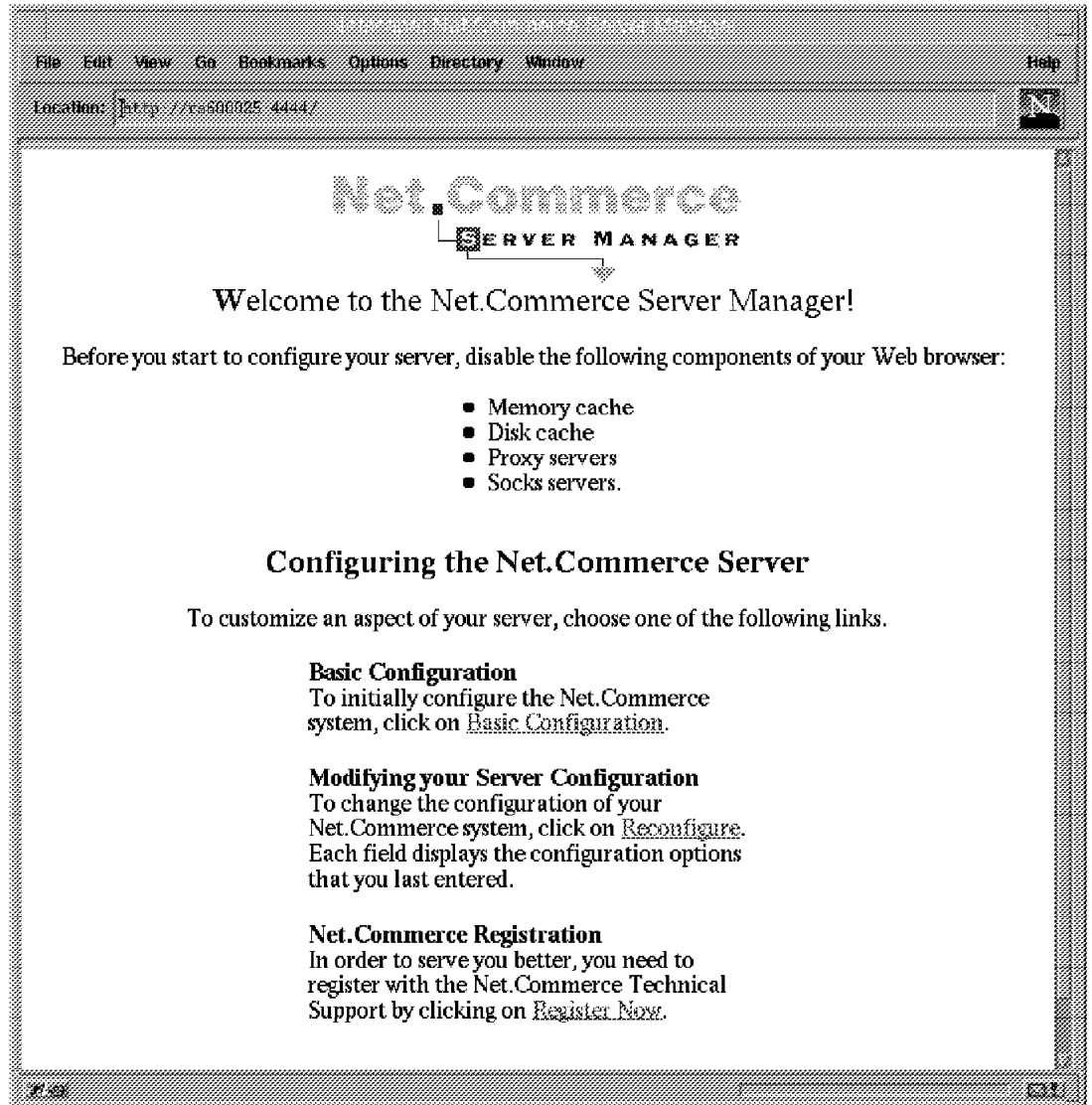


Figure 49. Net.Commerce Server Manager, AIX

6. Scroll down and click on **Basic Configuration**.
7. Scroll down and click on **First Step System Configuration** in the lower left hand corner.
8. Log on again with the same user ID and password as before.
9. Scroll down the System Configuration page and make sure all entries are correct. The table below lists the settings used for our scenario:

Parameter	Value
Host Name	rs600025.itso.ral.ibm.com
Port Number	8178
Database Name	mserver
Database Administrator	inst1
Administrator Password	inst1
Log File Location	/usr/lpp/NetCommerce/logs
Number of Processes	2

10. Click on **Submit** to update system configuration. If you are asked to log on again, do so with the same user ID and password as before. A confirmation page will be displayed.
11. Click on **Next Step Access Control**.
12. Scroll down the Access Control page and make sure all entries are correct. The table below lists the settings used for our scenario:

Parameter	Value
Net.Commerce Macro Path	/usr/lpp/NetCommerce/macro/en_US
Web Server Admin Program	/usr/lpp/internet/server_root/cgi-bin/htadm
Password File	/usr/lpp/NetCommerce/password/netc.pwd
Web Server Configuration File	/etc/httpd.conf
Web Server HTML Path	/usr/lpp/internet/server_root/pub
Web Server CGI-BIN Path	/usr/lpp/NetCommerce/cgi-bin

13. Click on **Submit** to update access control. A confirmation page will be displayed.
14. Stop and restart ICSS for changes to become effective.
 

**Note:** Make sure you stop the ICSS service, not the Net.Commerce Administration Web server.
15. Click on **Next Step Database Management**.
16. Click on **Start Database Manager**. A confirmation page will be displayed.
17. Click on **Database Management**.
18. Scroll down and check **Regular Creation**, then click on **Create** to create the default Net.Commerce database. This may take a few minutes to complete.
 

**Note:** If the database creation fails from within Net.Commerce Server Manager, try to create it manually by logging in as the DB2 instance owner and running the `/usr/lpp/NetCommerce/bin/mstblcrt` command file.
19. Click on **Next Step Server Control**.
20. Click on **Enable** to create an entry in the `/etc/inittab` file to run the `/etc/rc.netc` which allows you to start the DB2 and Net.Commerce servers automatically when the system starts. A confirmation page will be displayed.

21. Click on **Server Control**.
22. Click on **Startup** to start the Net.Commerce server. A confirmation page will be displayed.  
**Note:** If the server fails to start, log in as the DB2 instance owner and try to start the server manually.
23. Stop the Administration Web Server using the following AIX commands:  

```
cd /usr/lpp/Net.Commerce/server/bin
./stop_admin_server
```
24. Shut down and restart the system.

#### 4.1.4 Verification of the Installation

There are several checks to verify a successful installation of Net.Commerce. For the sake of simplicity and completeness, we have summarized them in this section. We indicate for each check where during the installation process it would be appropriate to be performed.

1. Check that the DB2 environment will be initialized correctly for the DB2 instance owner. The home directory for the DB2 instance owner should contain a .profile file which should contain the following line:

```
. sqllib/profile
```

This should be done after DB2 has been installed.

2. Check that ICSS, DB2 and Net.Commerce servers will be loaded at system start. The /etc/inittab file should contain the following entries:

```
rchttpd:2:wait:/etc/rc.httpd > /dev/console 2>script # Start HTTP daemon
rcnetc:2:once:/etc/rc.netc ##IBM Net.Commerce Server
```

This should be done after all components have been installed.

3. Check that ICSS security is working by pointing your browser at the following URL: `https://<your_hostname>/Frntpage.html`

That should cause the browser to notify you that the server has sent a security certificate and that you are about to enter SSL secure mode.

This should be done after ICSS security has been configured.

4. Check that the following files have been created and/or updated with Net.Commerce specific information:

```
/etc/mserver.conf
/etc/httpd.conf
/usr/lpp/NetCommerce/password/netc.pwd
/usr/lpp/NetCommerce/http/en_US/db2www.ini
/etc/rc.netc
```

Figure 50 on page 98 illustrates the relationship of the Net.Commerce components and the configuration files:

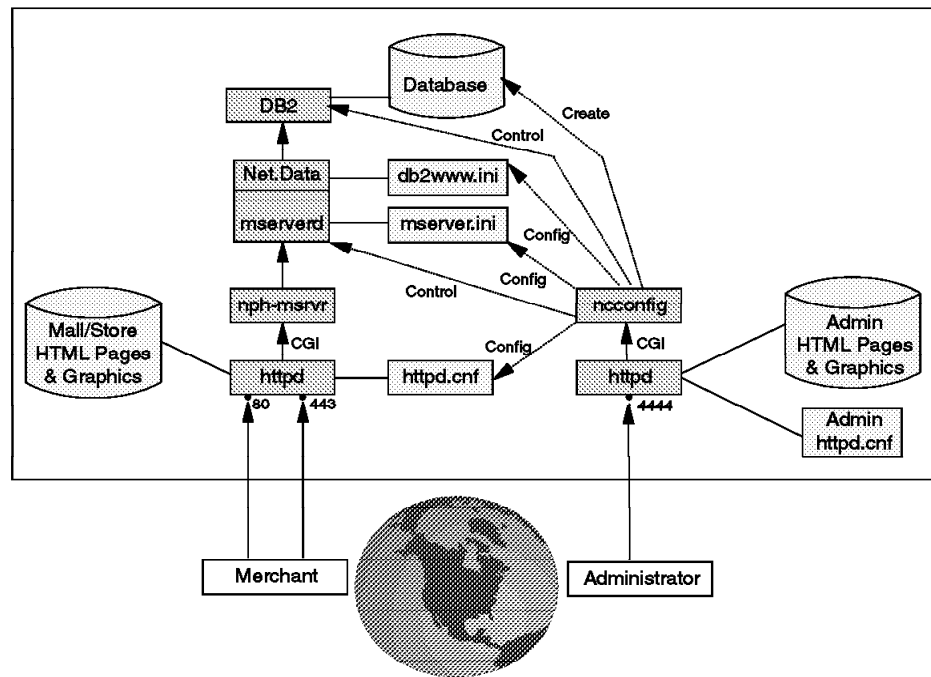


Figure 50. Net.Commerce Configuration Files

The following example lists a successful `/etc/mserver.conf` file for the single Net.Commerce server:

```

MS_HOSTNAME rs600025.itso.ra1.ibm.com
MS_PORT 8178
MACRO_PATH /usr/lpp/NetCommerce/macro/en_US
MS_HTADMIN /usr/lpp/internet/server_root/cgi-bin/htadm
MS_HTPASS /usr/lpp/NetCommerce/password/netc.pwd
MS_LOGPATH /usr/lpp/NetCommerce/logs
WWW_CONFIG /etc/httpd.conf
WWW_HTML_PATH /usr/lpp/internet/server_root/pub
WWW_CGIBIN_PATH /usr/lpp/NetCommerce/cgi-bin
MS_DBNAME demoma11
MS_DBINST inst1
MS_DBPASS kliiodeholapdfmc
MS_CONFIG http://rs600025.itso.ra1.ibm.com:4444/admin/nccconfig
MS_NUM_CHILD 2
NC_INST_PATH /usr/lpp/NetCommerce
NC_DB_SCRIPT democrt
NC_DB_SCRIPT_DIR /usr/lpp/NetCommerce/bin
MS_TRANS_COUNT 1000
DB_RETRY_LIMIT 0
DB_RETRY_INTERVAL 60
DB_CODESET IBM-850
DB_TERRITORY US
MERCHANT_KEY pfifkjaclogkacpepcifpdfhmgkfake
NC_DBADMIN inst1
NC_DBADMINPWD kliiodeholapdfmc
NC_LANG en_US
MS_HTML_MAX 1000000
    
```

Figure 51. AIX `mserver.conf` File for Single System

This should be done after Net.Commerce has been configured.

5. Check that the Net.Commerce-specific configuration for ICSS is working by pointing your browser at the following URL:

`http://<your_hostname>/demomall/basemall.htm`

The example below lists the Net.Commerce-specific entries in the `/etc/httpd.conf` file:

```
##IBM Net.Commerce PROTECT directives.
Protect /msprotect/* {
ServerId Private_Authorization
AuthType Basic
GetMask All@(*)
PutMask All@(*)
PostMask All@(*)
Mask All@(*)
PasswdFile /usr/lpp/NetCommerce/password/netc.pwd
}
### END OF NET.COMMERCE BLOCK

##IBM Net.Commerce EXEC directives.
Exec /msprotect/* /usr/lpp/NetCommerce/cgi-bin/*
Exec /cgi-bin/* /usr/lpp/NetCommerce/cgi-bin/*
### END OF NET.COMMERCE BLOCK

##IBM Net.Commerce PASS directives.
Pass /storemgr/* /usr/lpp/NetCommerce/html/en_US/ncadmin/storemgr/*
Pass /sitemgr/* /usr/lpp/NetCommerce/html/en_US/ncadmin/sitemgr/*
Pass /ncsamples/* /usr/lpp/NetCommerce/html/en_US/ncsample/*
Pass /ncsample/* /usr/lpp/NetCommerce/html/en_US/ncsample/*
Pass /ncagif/* /usr/lpp/NetCommerce/html/en_US/ncadmin/gif/*
Pass /ncahelp/* /usr/lpp/NetCommerce/html/en_US/ncadmin/help/*
Pass /ncadmin/* /usr/lpp/NetCommerce/html/en_US/ncadmin/*
Pass /ncerror/* /usr/lpp/NetCommerce/html/en_US/ncerror/*
Pass /ncbooks/* /usr/lpp/NetCommerce/html/en_US/ncbooks/*
Pass /te_html/* /usr/lpp/NetCommerce/html/en_US/ncadmin/teeditor/te_html/*
Pass /demomall/* /usr/lpp/NetCommerce/html/en_US/demomall/*
### END OF NET.COMMERCE BLOCK
```

Figure 52. AIX `httpd.conf` File

This should be done after Net.Commerce has been configured.

6. Check that Net.Data is configured properly. The example below lists the Net.Commerce specific entries in the `/usr/lpp/NetCommerce/http/en_US/db2www.ini` file:

```

MACRO_PATH /usr/lpp/NetCommerce/macro/en_US/ncsample;
           /usr/lpp/NetCommerce/macro/en_US/demomall;
           /usr/lpp/NetCommerce/macro/en_US
DB2INSTANCE inst1
BIND_FILE /usr/lpp/NetCommerce/bin/d2wsq1.bnd
INCLUDE_PATH /usr/lpp/NetCommerce/macro/en_US/ncsample;
             /usr/lpp/NetCommerce/macro/en_US;
             /usr/lpp/NetCommerce/html/en_US/ncadmin/teditor;
             /usr/lpp/NetCommerce/html/en_US
HTML_PATH /usr/lpp/NetCommerce/html/en_US
EXEC_PATH /usr/lpp/NetCommerce/macro/en_US/ncsample
ENVIRONMENT (DTW_SQL) /usr/lpp/NetCommerce/lib/dtwsqlshr.o
              ( IN DATABASE, LOGIN, PASSWORD, TRANSACTION SCOPE, ALIGN)
ENVIRONMENT (DTW_DEFAULT) /usr/lpp/NetCommerce/lib/dtwfuncshr.o ( )
ENVIRONMENT (DTW_APPLET) /usr/lpp/NetCommerce/lib/dtwapplshr.o ( )
ENVIRONMENT (DTW_ODBC) /usr/lpp/NetCommerce/lib/dtwodbcshr.o
              ( IN DATABASE, LOGIN, PASSWORD, TRANSACTION SCOPE, ALIGN)
ENVIRONMENT (DTW_REXX) /usr/lpp/NetCommerce/lib/dtwrexxshr.o ( OUT RETURN_CODE )
ENVIRONMENT (DTW_WEBREG) /usr/lpp/NetCommerce/lib/dtwregshr.o ( OUT RETURN_CODE)
ENVIRONMENT (DTW_FILE) /usr/lpp/NetCommerce/lib/dtwffishr.o ( OUT RETURN_CODE)
ENVIRONMENT (DTW_PERL) /usr/lpp/NetCommerce/lib/dtwperlshr.o ( OUT RETURN_CODE)

```

Figure 53. AIX db2www.ini File

This should be done after Net.Commerce has been configured.

7. Check that Net.Commerce macros, CGI-BIN programs and the connection to the database are working properly by registering as a shopper from the following URL:

[http://<your\\_hostname>/cgi-bin/nph-msrvr;/register/form](http://<your_hostname>/cgi-bin/nph-msrvr;/register/form)

8. Check that the Net.Commerce administration tools and JavaScript are working properly by pointing your browser at the following URL:

[http://<your\\_hostname>/ncadmin](http://<your_hostname>/ncadmin)

Log on as user admin with password admin. The Net.Commerce Administrator page will be displayed as shown below:

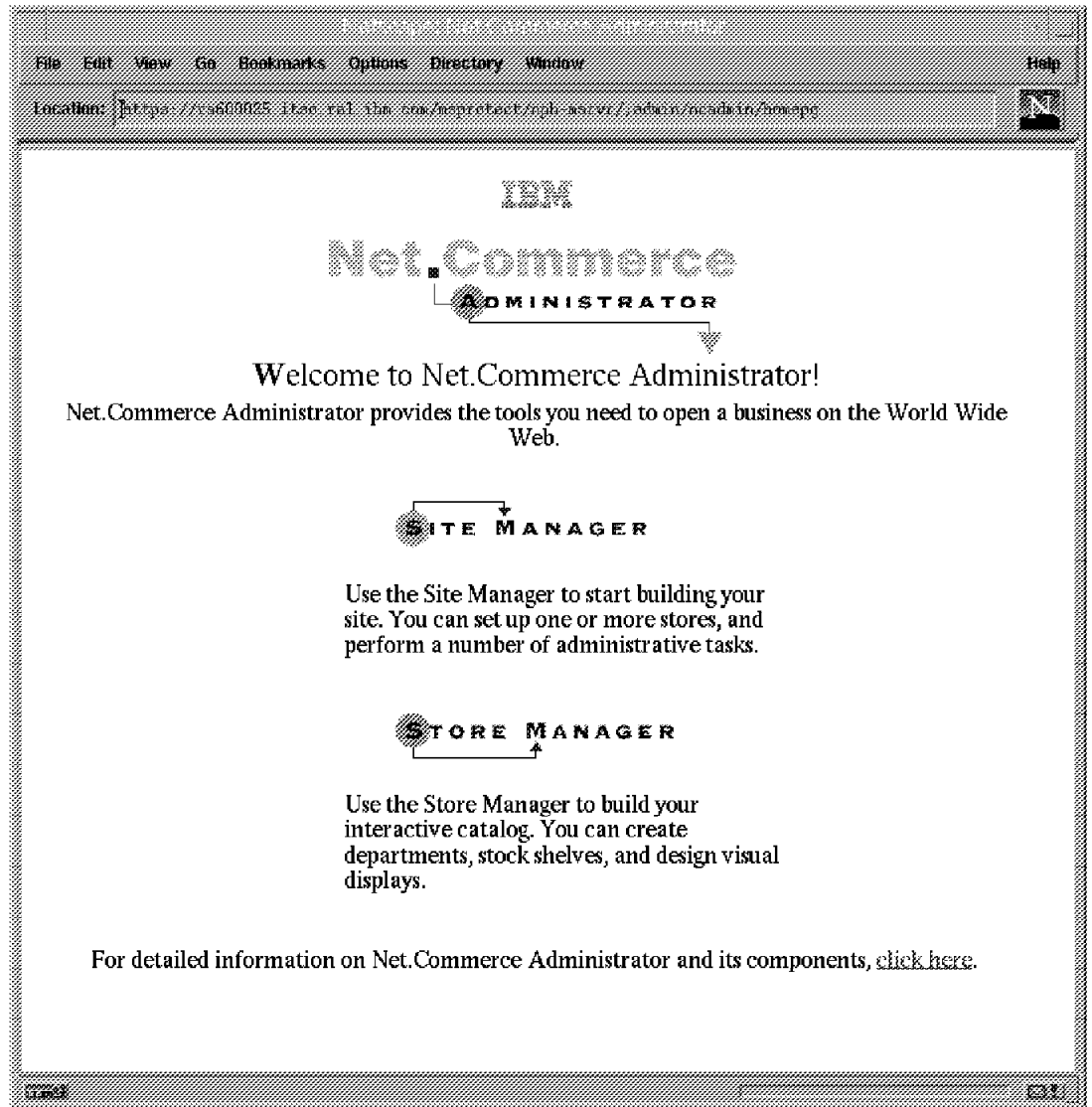


Figure 54. Net.Commerce Administrator, AIX

9. Check that Java and the database connection are working properly.
  - a. Click on **Site Manager**.
  - b. Scroll down and click on **Template Designer**.
  - c. Select the **Load** button.

While the Java applet initializes, it displays status information in the gray bar below the white input area. Check that database information is being retrieved.

#### 4.1.5 Installing the Demo Mall

This section describes the steps necessary to install the demo mall in our scenario.

1. Start the Net.Commerce Administration Web server.
2. Start your Web browser and go to URL `http://<your_hostname>:4444` to access the Net.Commerce Administration server pages.

3. Click on **Reconfigure**.
  4. Click on **Server Control**.
  5. Log on with a user ID of webadmin and a password of webibm. Use this ID and password whenever you are asked to log in again.
  6. Scroll down and click on **Shut Down** to stop the Net.Commerce server.
- Note:** You must stop Net.Commerce server before you go ahead and change the system configuration.
7. Check that the server has actually been stopped by issuing the following command from an AIX command prompt:

```
ps -ef | grep mserver
```

If there are problems, stop the Net.Commerce server manually by killing the parent mserverd process using the kill command.

8. Click on **Reconfigure** again.
9. Click on **System Configuration**.
10. Scroll down and change the database name from mserver to demomall.
11. Click on **Submit** to update system configuration.
12. Copy the netccur file from the /usr/lpp/NetCommerce/bin directory to the /home/inst1/sqllib/function directory.
13. Log in as the DB2 instance owner and change to the /usr/lpp/NetCommerce/html/en\_US/demomall/data directory.
14. Run the democrt command file. This creates the demo mall database. It may take a few minutes to complete.
15. Open a text editor window and load the Net.Commerce Fix Pak #1 readme file, readme\_aixfp1.txt.
16. In the AIX command window where inst1 is still logged in, start a DB2 command line interface by entering db2.
17. Enter the following DB2 command:  

```
connect to demomall
```
18. Copy and paste the following lines from the readme file in the editor into the DB2 command line; press Enter after each one:

```
db2 "CREATE FUNCTION formatcur (float, char(30), char(1)) RETURNS  
char(200) NOT FENCED EXTERNAL NAME 'netccur!formatcur' NOT  
VARIANT NULL CALL NO SQL PARAMETER STYLE DB2SQL LANGUAGE C NO  
EXTERNAL ACTION"  
  
db2 "CREATE FUNCTION formatcurd (float, char(30)) RETURNS char(200)  
NOT FENCED EXTERNAL NAME 'netccur!formatcurd' NOT VARIANT NULL  
CALL NO SQL PARAMETER STYLE DB2SQL LANGUAGE C NO EXTERNAL ACTION"
```

Figure 55. Adding User Defined Functions to the Net.Commerce Database on AIX

19. Enter the following DB2 commands to end the command line interface:  

```
connect reset  
terminate
```
20. Exit from the inst1 user.



21. Restart the Net.Commerce server using either the Server Control page or the manual command (/usr/lpp/NetCommerce/bin/mserverd). Make sure the server is back up.
22. Stop the Administration Web Server.

The demo mall should now be ready to use. Connect to it by using the URL `http://<your_hostname>/demomall/basemall.htm` and start shopping. Figure 56 shows the demo mall home page:

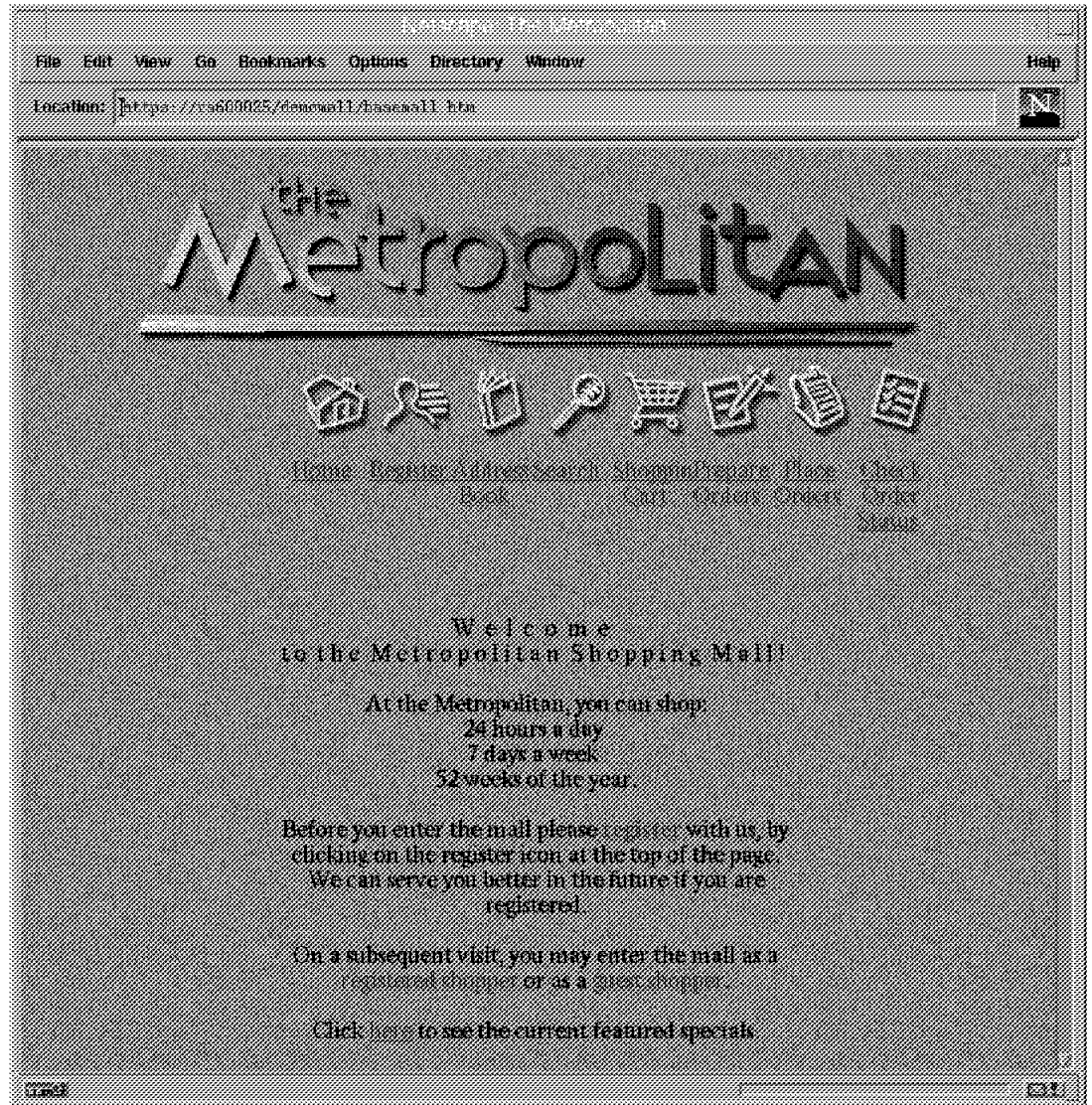


Figure 56. Net.Commerce Demo Mall Home Page, AIX

#### 4.1.6 Uninstalling the System

The following section explains how Net.Commerce and its related components can be removed from your system.

1. Start the Net.Commerce Administration Web Server.
2. Start your Web browser and access the Administration Web server.

3. Click on **Reconfigure**, then click on **Server Control**.
4. Click on **Disable** to delete the entry in the /etc/inittab file to autostart Net.Commerce at startup.
5. Stop all components.
6. Make sure you save all files, data, information and databases that you may need again or on another system.
7. From SMIT, go to **Software Installation & Maintenance** and select **Remove Software**.
8. From the list, select all Net.Commerce, DB2, Internet an IPF components.
9. Click on **OK** to remove the software.
10. Shut down and restart your system.
11. Remove the remaining files and directory trees:
  - /usr/lpp/internet directory tree
  - /usr/lpp/NetCommerce directory tree
  - /home/inst1 directory tree
12. Delete user IDs and groups that have been associated with Net.Commerce components.

---

## 4.2 Installation and Basic Configuration on Windows NT

This section describes the installation and basic configuration of the Net.Commerce system on multiple Windows NT systems.

### 4.2.1 Preparation

We generally recommend that you install the latest available Service Pak for Windows NT to update your operating system to the most actual service level. This should minimize (but cannot eliminate) the risk of system failures due to problems related to the base operating system. (At the time this book was published, that was Service Pak 5 for NT 3.51 and Service Pak 3 for NT 4.0.)

As described in 1.2, "Net.Commerce Overview" on page 2, all required software components (DB2, ICSS, Net.Commerce) are part of your Net.Commerce software package. For the specific system requirements, please refer to 3.1.5.2, "Windows NT System Requirements" on page 46 and 3.3.1, "Database Space Requirements" on page 49.

For our scenarios, we used Net.Commerce Version 1.1 which is a Fix Pak to Version 1.0. You can download the Fix Pak #1 for Net.Commerce from the following URL:

<http://www.internet.ibm.com/commercepoint/net.commerce/upgrades.html>

Before you start the installation of Net.Commerce, you have to create a user account for DB2:

**Note:** This account ID must not exceed 8 characters or DB2 will fail to install.

1. From the Administrative Tools window, open **User Manager for Domains**.
2. On the menu bar, click on **User** and select **New User**. A panel will appear as shown in Figure 57 on page 105.

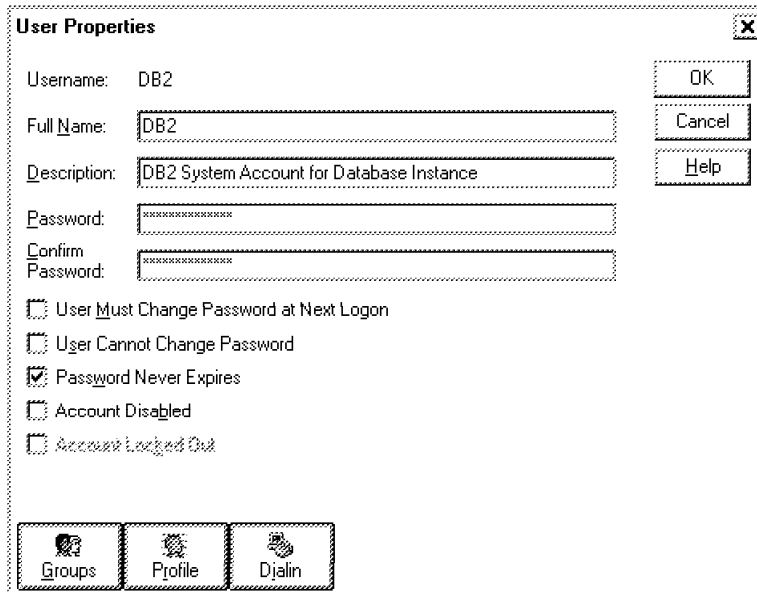


Figure 57. Creating a Windows NT Account for DB2

3. Fill in the user name. In our case it is DB2 which is the default database instance name for DB2.
4. Fill in a password for the user, in our case it is also DB2.  
**Note:** Windows NT passwords are case-sensitive.
5. Deselect User Must Change Password At Next Logon.
6. Optionally, select **Password Never Expires**.
7. Click on **Groups**.
8. Select **Administrators** from the Not member of list and click the **<-Add** button, then click **OK**.
9. Click **Add** in the New User window. The user definition will be added to the system.
10. Click **Close** in the New User window, then exit from the User Manager.
11. Log off the system and log on again with the user ID you just created. The following figure shows the Windows NT shut down/log off dialog.

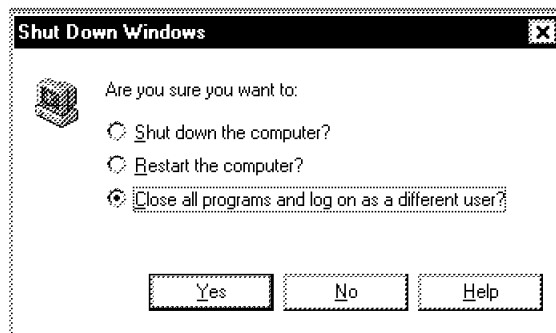


Figure 58. Windows NT Shut Down/Log Off Dialog

## 4.2.2 Installation of the Files on the Net.Commerce Server

On this system, you need to install the following components:

1. DB2 client to access the database on the database server
2. ICSS to allow shoppers to access your system and to configure Net.Commerce
3. Net.Commerce server

Normally, you would install all components from the Net.Commerce CD-ROM using the integrated setup program. However, since our scenario required an installation that was different from the defaults, we installed each component separately. The table below illustrates the differences between the default installation and our setup:

<i>Table 5. Component Installation (Net.Commerce Server, NT)</i>	
<b>ITSO Scenario</b>	<b>Default Installation</b>
DB2 Client Application Enabler 2.1.2 with ODBC support	DB2 Server 2.1.2
Net.Commerce 1.1 (with Fix Pak #1)	Net.Commerce 1.0

We also used Lotus Approach to access the Net.Commerce database and therefore needed to have ODBC support installed.

### 4.2.2.1 Installation of DB2 Client Application Enabler

Follow the instructions below to install the necessary DB2 components.

1. On your Net.Commerce CD-ROM, change to the directory Db2ClientI386EnDisk1.
2. Start the Setup program by double-clicking on the program icon.
3. Click **Next** on the Welcome window.
4. On the Setup window, choose **Full** to install all client components. This is shown in the figure below:

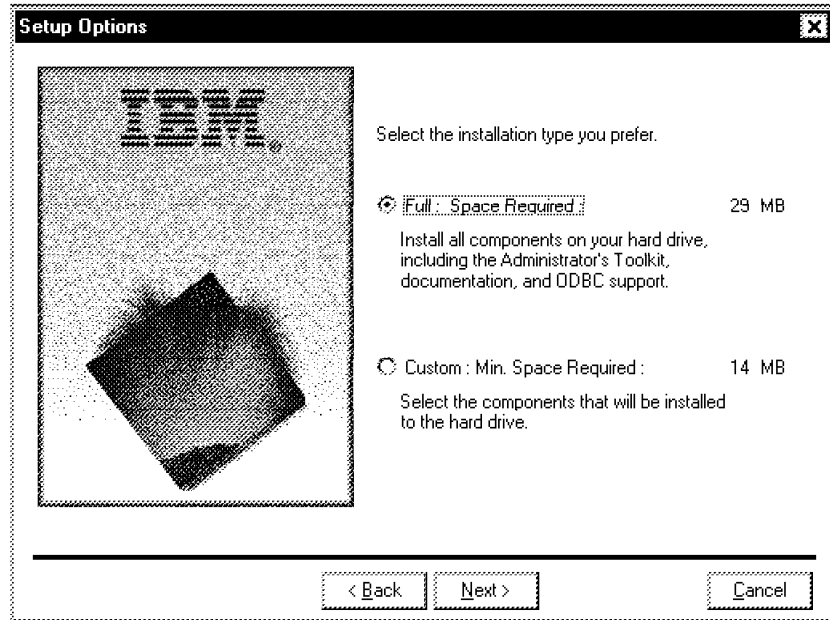


Figure 59. DB2 Client Setup Options, NT

5. Click **Next** and select the destination drive and directory for the files, then click **Install** to start the installation.
6. When the installation has finished, restart your workstation by clicking on **Finish** on the Setup Complete window.

#### 4.2.2.2 Installation of Internet Connection Secure Server

The installation of Net.Commerce also includes the installation of ICSS, so there is no need to run a separate setup program for that component.

**Note:** We tried to use ICSS 4.2 because it offers, among a series of new features, a better performance than ICSS 4.1 which ships with the Net.Commerce product, but ICSS 4.2 is not compatible with Net.Commerce 1.1.

#### 4.2.2.3 Installation of Net.Commerce

If Net.Commerce 1.0 has not been installed before, you can install the Fix Pak #1 right away. To do so, unzip the files to a temporary directory using utilities such as PKZIP or WINZIP.

If Net.Commerce 1.0 is already installed and configured, please follow the instructions in the Readme file for Fix Pak #1 to have the Fix Pak installed.

The following section describes the installation of Net.Commerce 1.1 using the Fix Pak #1 files.

1. Change to the directory where you extracted the Fix Pak #1 files and start the Setup program.
2. Click **Next** on the Welcome window.
3. Click **Yes** to accept the license agreement.
4. Choose **Net.Commerce: Build M29** and **Internet Connection Secure Server** on the Components Selection window which is shown below, then click **Next**.

**Note:** This Net.Commerce setup includes the installation of ICSS 4.1.

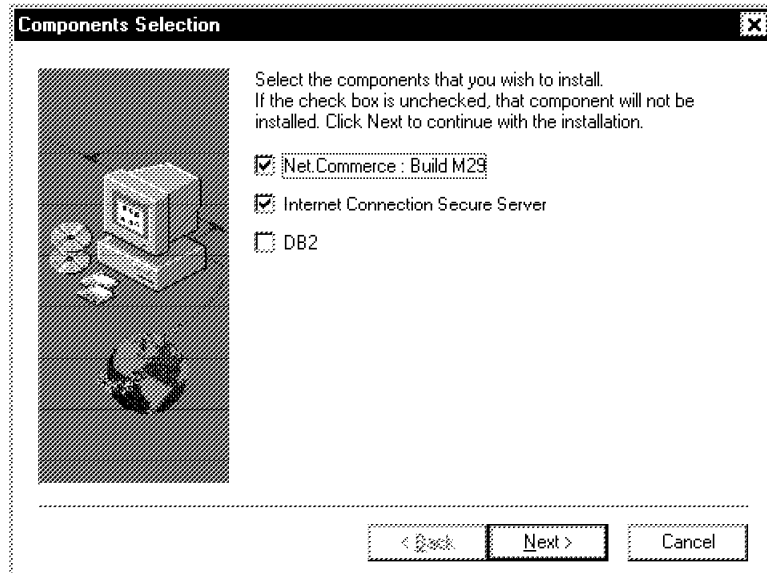


Figure 60. Net.Commerce Fix Pak #1 Component Selection, NT

5. Choose the appropriate documentation on the Documentation Selection window, then click **Next**.
6. Choose to install the Net.Commerce Sample Store on the Samples Selection window, then click **Next**.
7. Select the destination drive and directory for the files, then click **Next** to start the installation.
8. Click on **OK** after the installation of ICSS has completed. The process continues with the installation of Net.Commerce.
9. When the installation has finished, restart your workstation by clicking on **Finish** on the Setup Complete window. (Ignore the IBM Net.Commerce folder that pops up in the upper left-hand corner.)

### 4.2.3 Installation of the Files on the Database Server

On this system, you need to install the DB2 server to manage the Net.Commerce database.

Normally, you would install all components from the Net.Commerce CD-ROM using the integrated setup program. However, since our scenario required an installation that was different from the defaults, we installed each component separately. The table below illustrates the differences between the default installation and our setup:

<i>Table 6. Component Installation (Database Server, NT)</i>	
<b>ITSO Scenario</b>	<b>Default Installation</b>
DB2 Server 2.1.2 with ODBC support	DB2 Server 2.1.2

We also used Lotus Approach to access the Net.Commerce database and therefore needed to have ODBC support installed.

### 4.2.3.1 Installation of DB2 Server

Follow the instructions below to install the necessary DB2 components:

1. On your Net.Commerce CD-ROM, change to the directory Db2ServerI386EnDisk1.
2. Start the Setup program by double-clicking on the program icon.
3. Click **Next** on the Welcome window.
4. On the DB2 Setup window, choose **Server** to install a DB2 server system. This is shown in the figure below:

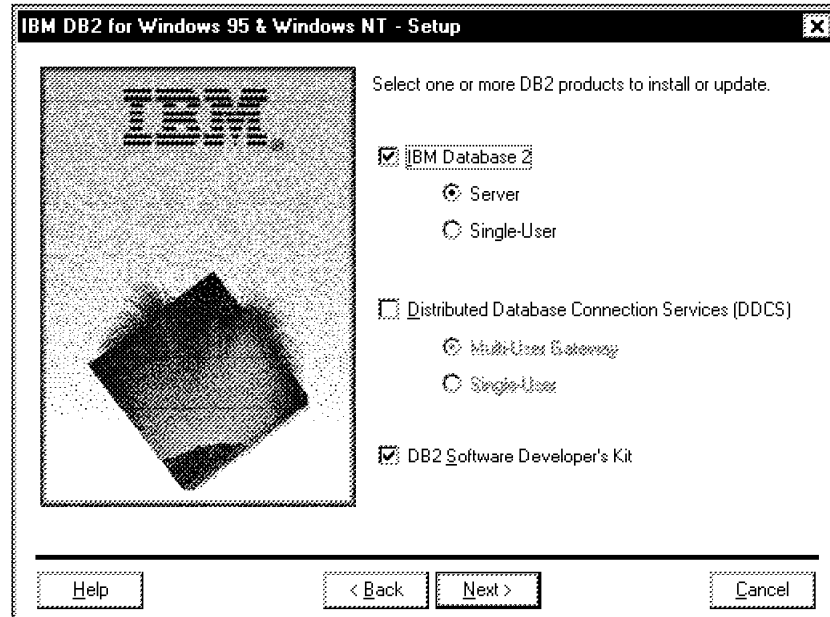


Figure 61. DB2 Server Setup Options, NT

5. Click **Next** and choose the appropriate option in the Licensing window.

**Note:** If you click on Try and Buy, you may use your DB2 system for a period of 60 days after which you will have to purchase a license key. Databases that you have created during that time will not be lost but cannot be used again until you install the license key.

6. On the Setup window, choose **Full** to install all server components.
7. Click **Next** and select the destination drive and directory for the files, then click **Install** to start the installation.
8. When the installation has finished, restart your workstation by clicking on **Finish** on the Setup Complete window.

### 4.2.4 Basic Configuration of the Database Server

The following configuration tasks are required on this system:

1. Select the DB2 services to start automatically at system start.
2. Configure DB2 environment variable for communications.
3. Configure TCP/IP services for DB2.
4. Create the Net.Commerce database.

#### 4.2.4.1 Set DB2 Startup Configuration

To start the DB2 services automatically when the server starts, perform the following steps:

1. From the Control Panel Window, open **Services**.
2. Select **DB2 - DB2** and click on **Startup...**
3. On the Service window, select **Automatic** in the Startup Type box, then click **OK**. Figure 62 shows the service configuration windows:

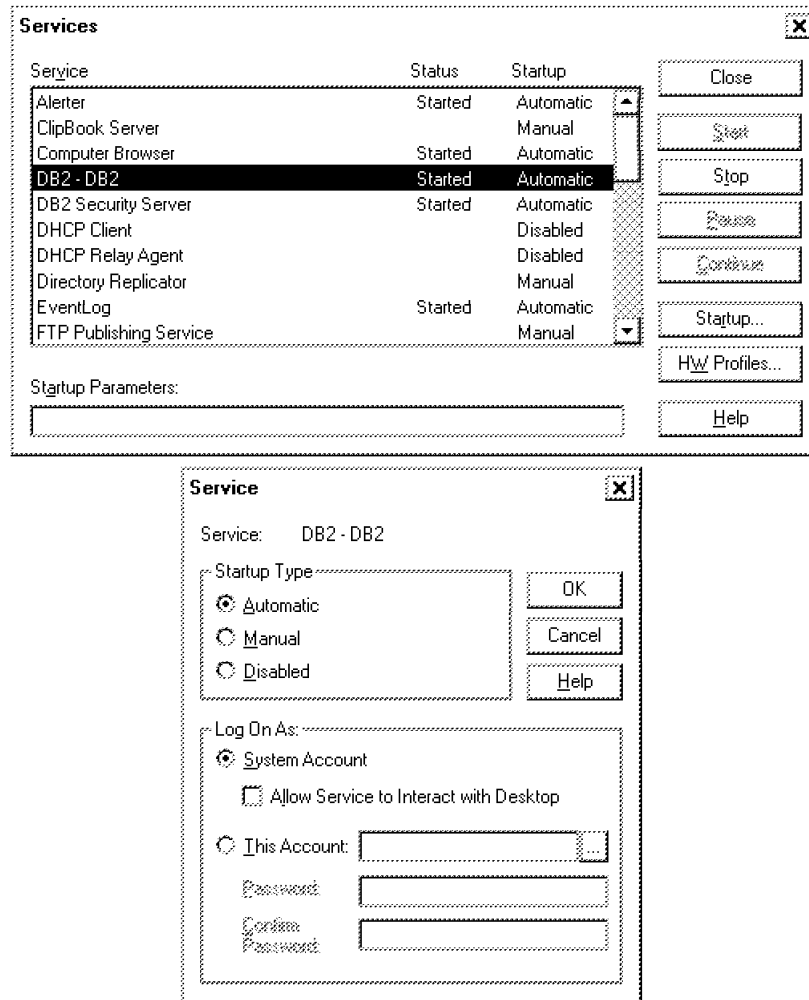


Figure 62. Setting DB2 Services Startup Configuration, NT

4. Do the same with the DB2 Security Server service.
5. Click **Close** to exit Services.

#### 4.2.4.2 Set DB2 Communication Environment

In order to tell the DB2 server what communication protocols to use for client connections, you have to configure the DB2COMM system environment variable. Do this by following the steps below:

1. From the Control Panel Window, open **System**.
2. Click on the **Environment** tab.



3. Click on the **DB2INSTANCE** variable in the System Variables box. The name and current setting of this variable will appear in the Variable and Value fields in the lower third of the window.
4. Change DB2INSTANCE to DB2COMM.
5. Change DB2 to TCPIP.
6. Click on **Set** to add the new environment variable DB2COMM. This is shown in the figure below.

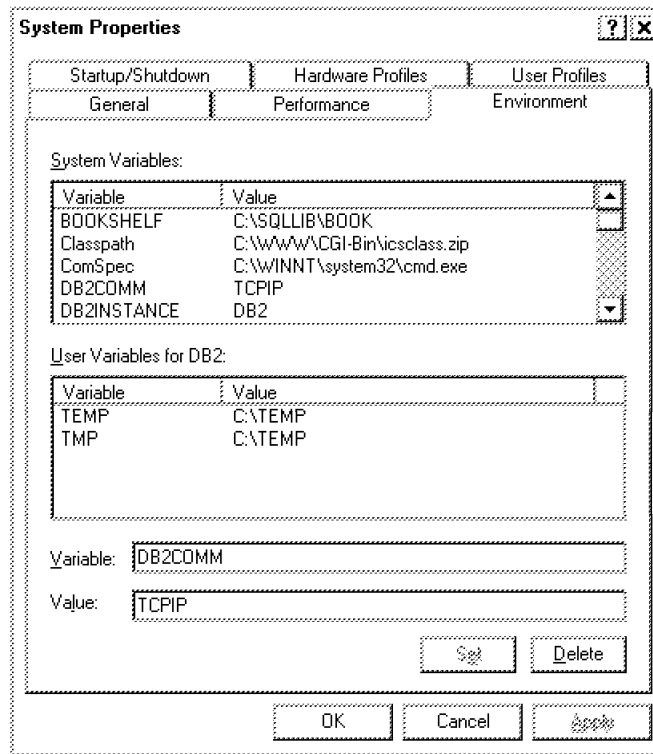


Figure 63. Setting DB2 Communication Environment Variable, NT

7. Click on **OK** to exit the System window.
8. Shut down and restart the system.

#### 4.2.4.3 Configure TCP/IP Ports and Services for DB2

In order to enable DB2 communication over TCP/IP, you have to create a port and service definition for each DB2 instance that you want to accept client connections. These definitions are stored in a flat ASCII file that you have to manually edit in the following way:

1. Open **MS DOS Command Prompt** from the Programs folder.
2. Edit the services file in the winntsystem32driversetc directory using, for instance, the Notepad editor.
3. Append the following lines to the services file:

```
netcomm      11111/tcp      # DB2 service for remote clients
netcommi     11112/tcp      # DB2 interrupt service
#
```

<b>netcomm</b>	The name of the DB2 TCP/IP service that clients will use to communicate to this server instance. You can choose this name but it must not be the same as an already existing one in the services file.
<b>11111/tcp</b>	The number of the TCP port where the DB2 server will listen for client connections. Make sure the port number is between 11000 and 19000 and is not one already defined in the services file.

The specification of a DB2 interrupt service is optional. If you add it, make sure that the port number is exactly one higher than the number you selected for the DB2 service.

**Note:** You will configure corresponding parameters on the Net.Commerce server later. Make sure that the values you entered here are matched by the Net.Commerce server configuration.

4. Open **DB2 Command Window** from the DB2 for Windows NT folder.
5. Enter the following command to check the current DB2 configuration:  

```
db2 get database manager configuration | more
```

Watch for the value of the Service name (svcname) parameter.
6. Update the svcname parameter with the name of the DB2 service that you specified before, using the following command:  

```
db2 update database manager configuration using svcname=netcomm
```

where netcomm is the name of the TCP/IP service for DB2.
7. Type exit to close the DB2 Command window.
8. Shut down and restart the system.

#### 4.2.4.4 Creating the Net.Commerce Default Database

1. Copy the following files from the Net.Commerce server to a temporary directory on the database server:  

```
ibmNetCommercebinmstblcrt.cmd  
ibmNetCommercebindb2wsql.bnd  
ibmNetCommercebindb2link.bnd  
ibmNetCommercebinmsdbtax.bnd  
ibmNetCommercebinbrowser.ixf  
ibmNetCommercebinstatcode.ixf  
ibmNetCommercebintaxcgry.ixf  
ibmNetCommercebintaxprcode.ixf
```
2. Start **DB2 Command Window** from the DB2 for Windows NT folder.
3. Change to the temporary directory and run the mstblcrt command file. This creates the Net.Commerce database. It may take a few minutes to complete.

### 4.2.5 Basic Configuration of the Net.Commerce Server

The following configuration tasks are required on this system:

1. Select the ICSS service to start automatically at system start.
2. Create security keys and certificates for ICSS.
3. Configure the Net.Commerce server.

#### 4.2.5.1 Set ICSS Startup Configuration

1. From the Control Panel window, open **Services**.
2. Select **Internet Connection Secure Server** and click on **Startup....**
3. On the Service window, select **Automatic** in the Startup Type box, then click **OK**. Figure 64 shows the service configuration windows:

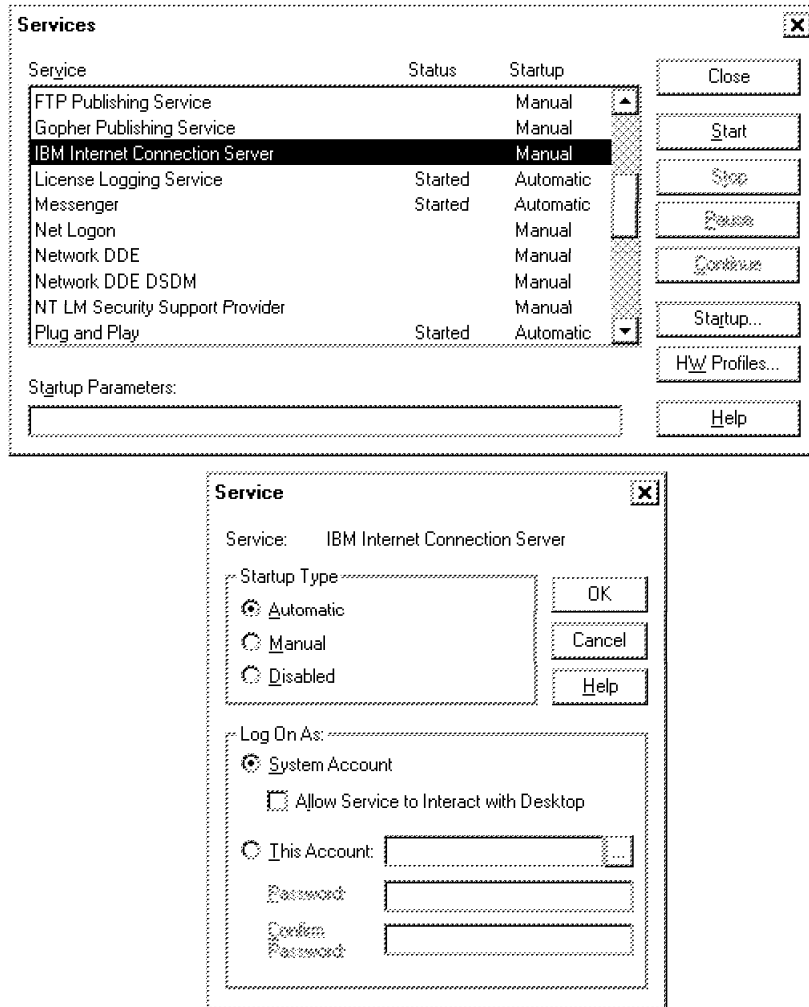


Figure 64. Setting ICSS Service Startup Configuration, NT

4. Click **Close** to exit Services.

#### 4.2.5.2 Configure and Enable ICSS Security

You need to request a signed certificate from a certification authority and create a key ring file that contains the certificate that will be signed. This process is discussed in 4.1.3.1, "The Certification Process" on page 86.

For the detail information about how to create the key ring file and the certificate you can see *IBM Net.Commerce: Installation and Operation Guide*, GC09-2373.

For our scenario, it was sufficient to use self-signed certificates, so we describe that process below.

**Note:** This process is identical to AIX and Windows NT with the exception of the location of the key ring and certificate files.

1. Start your Web browser if it is not running already.
2. Disable all caching (memory and file) and also disable the use of proxy and/or SOCKS servers.
3. Start ICSS if it is not running already.
4. Go to URL `http://<your_hostname>/Frntpage.html` to access the ICSS configuration pages on your Net.Commerce server, where `<your_hostname>` is the TCP/IP domain name of your server.
5. Click on **Configuration and Administration Forms**.
6. Log on with a user ID of `webadmin` and a password of `webibm` which are the default values for the ICSS administrator.
7. Scroll down to Security and click on **Create Keys**.
8. Select **Other** and **Apply**.
9. Change the length of the encryption key to the largest available size. Also enter the name of the key and the key ring file. This is shown in Figure 65:

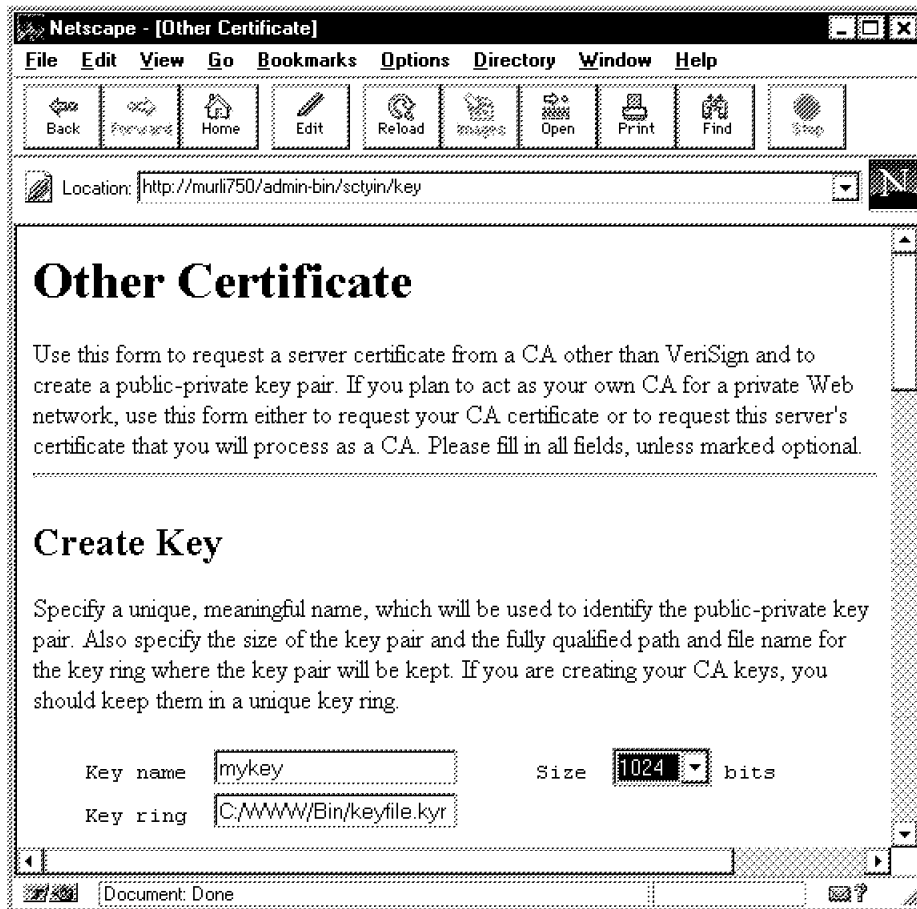


Figure 65. Creating Encryption Keys, NT

10. Scroll down and enter the key ring password and also check **Automatic login**, as shown in Figure 66 on page 115:

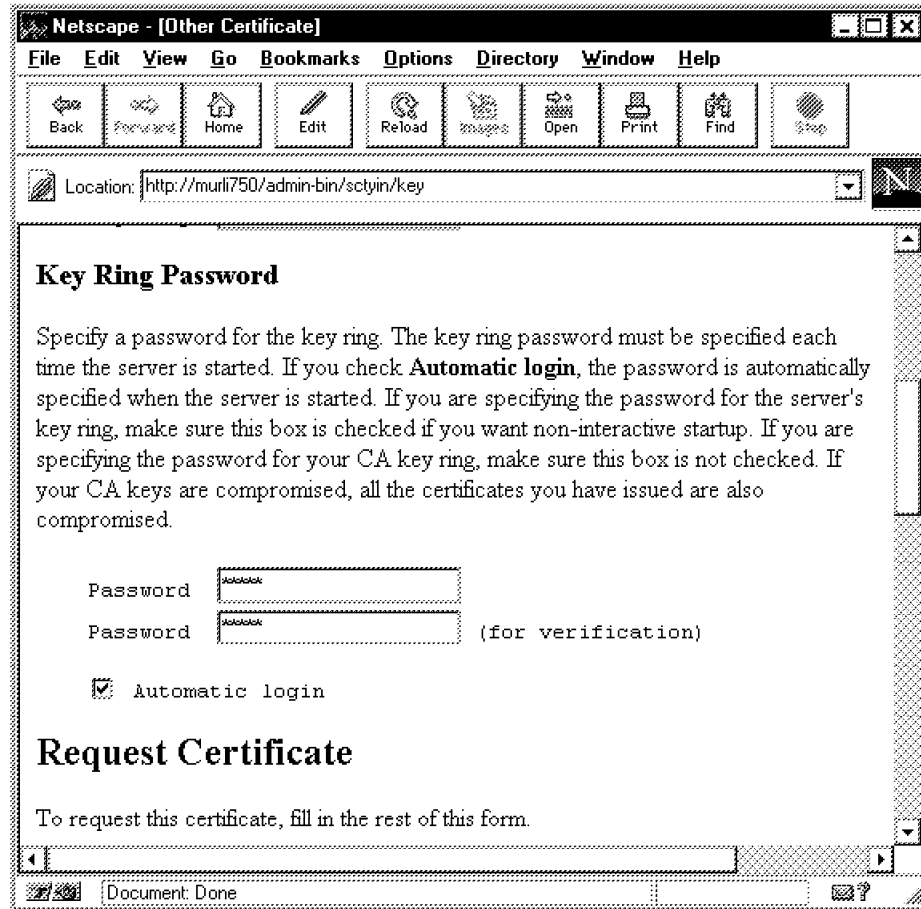


Figure 66. Key Ring Password

11. Scroll down and enter the information about your server as shown in Figure 67 on page 116. This will be included in the certificate.

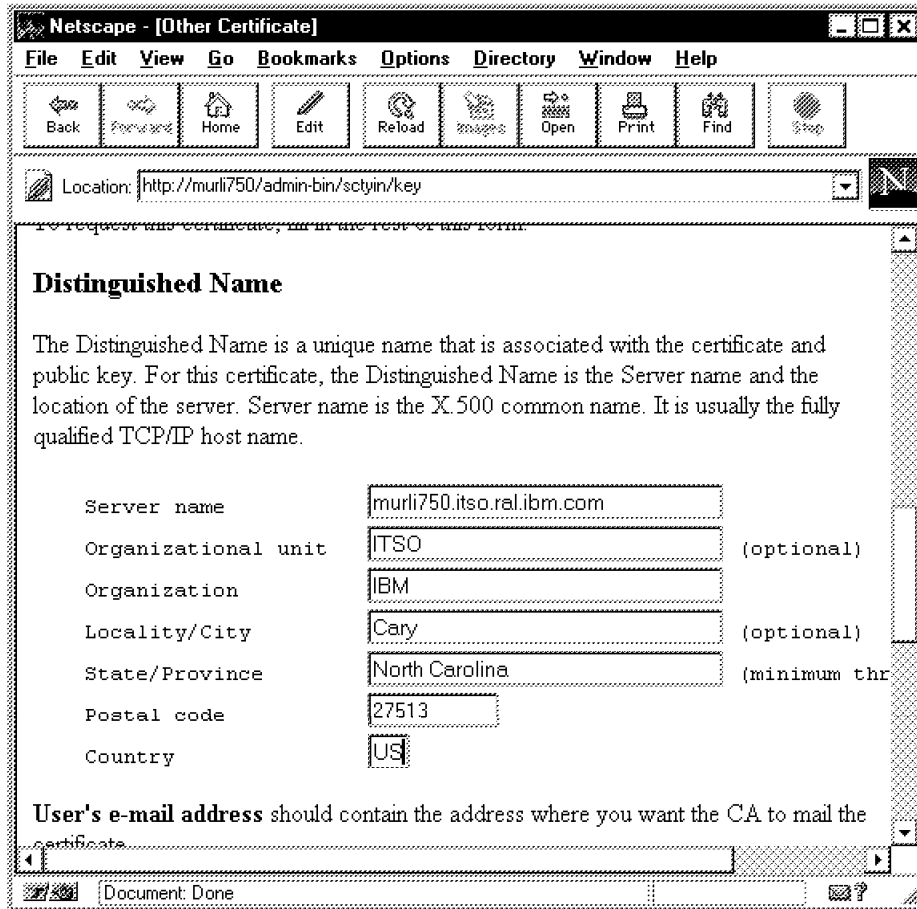


Figure 67. Key System Information, NT

12. Scroll down and enter an administrator e-mail address. Also select **Don't mail** and enter the name of the file for the certificate request, as shown in Figure 68 on page 117:

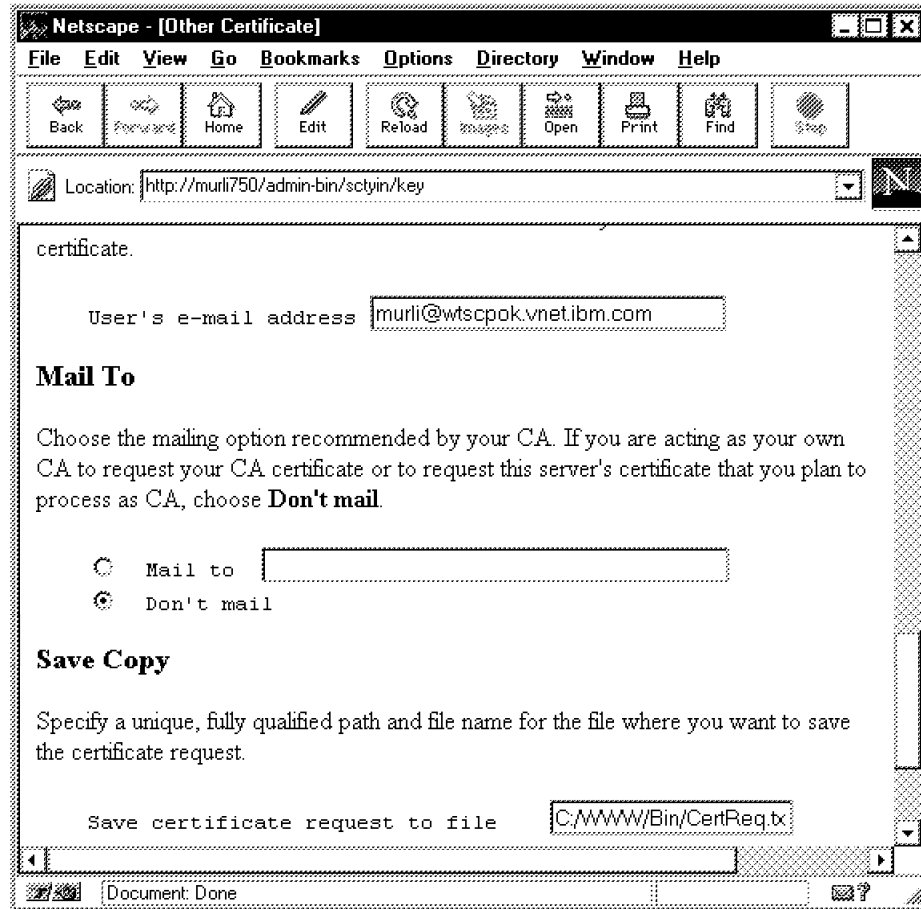


Figure 68. Certificate Request Information

13. Scroll down and click on **Apply**. A confirmation page will be displayed.
14. Return to the Configuration page and select **Security Configuration**.
15. Make sure that both S-HTTP and SSL connection are enabled.
16. Scroll down and select a key ring file to be the current one, as shown in Figure 69 on page 118:

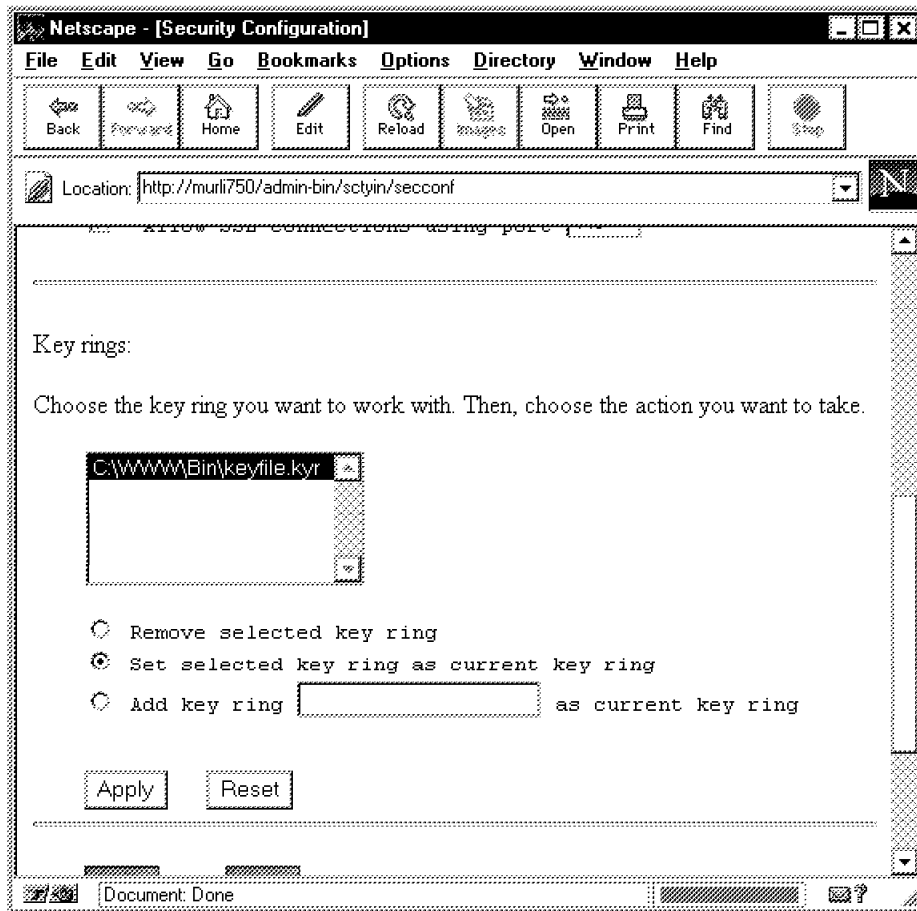


Figure 69. Current Key Ring File, NT

17. Click on **Apply**. A confirmation page will be displayed.
18. Return to the Configuration page and select **Receive Certificate**.
19. Enter the name of the certificate request and key ring files as well as the key ring password, as shown in Figure 70 on page 119:



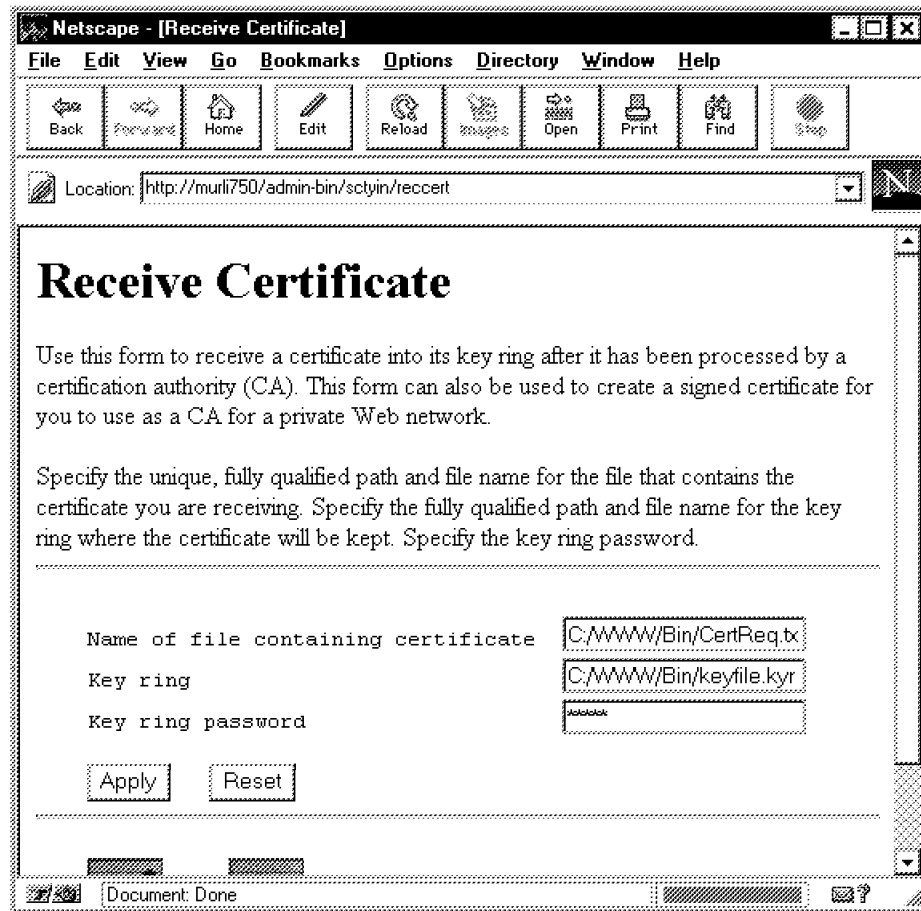


Figure 70. Receive Certificate, NT

20. Click on **Apply**. A confirmation page will be displayed.
21. Return to the Configuration page and select **Key Management**.
22. Enter the key ring password and select **Designate Trusted Root Keys**.
23. Select a key to be designated as trusted root key, as shown in Figure 71 on page 120:

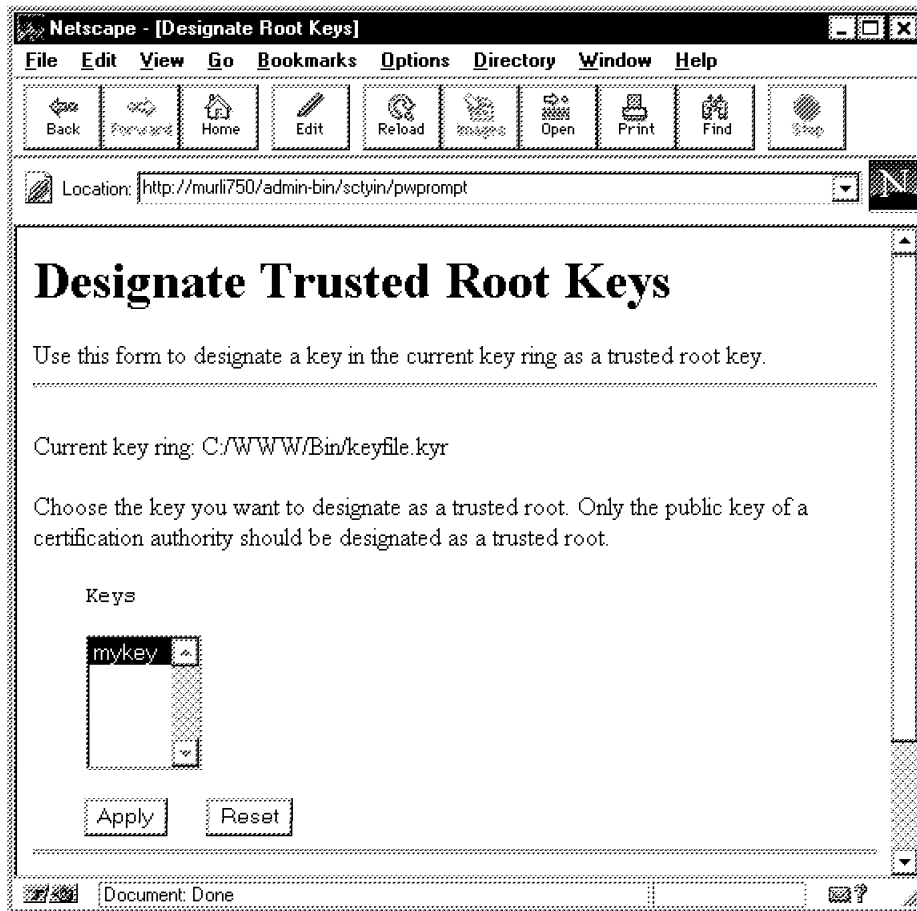


Figure 71. Designate Trusted Root Key, NT

24. Click on **Apply**. A confirmation page will be displayed.
25. Return to the Configuration page and select **Key Management** again.
26. Enter the key ring password and select **Manage Keys**.
27. Select the key you have created previously to be the default, as shown in Figure 72 on page 121:

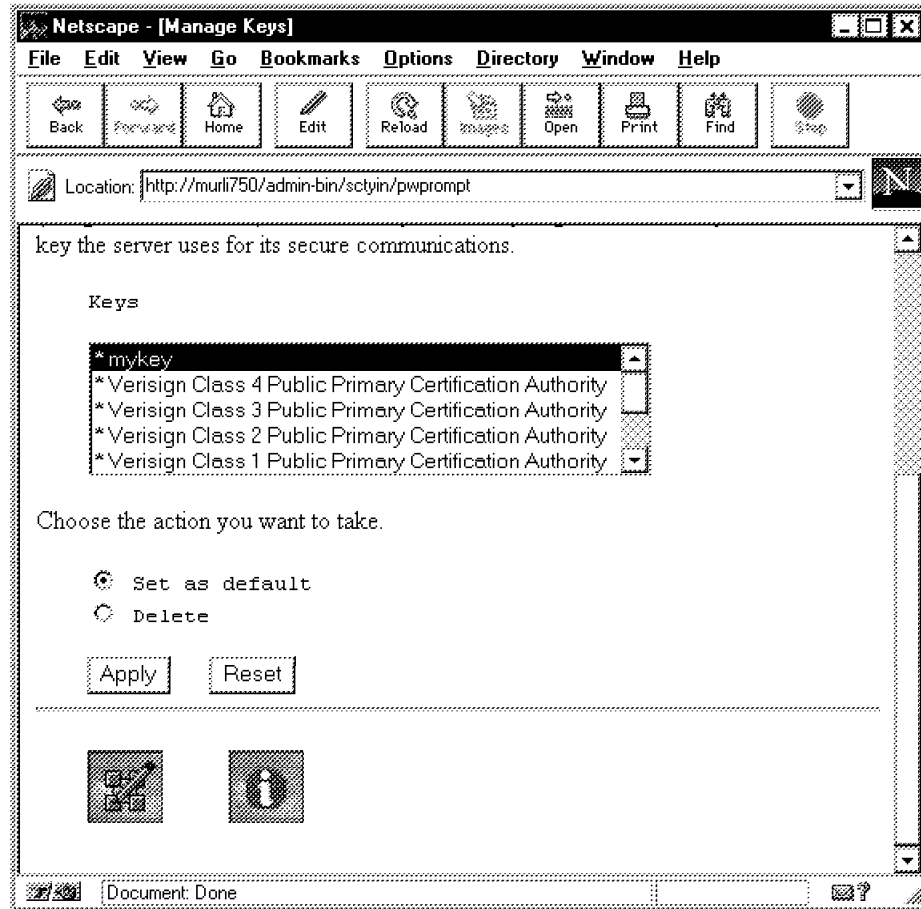


Figure 72. Select Default Key, NT

28. Click on **Apply**. A confirmation page will be displayed.
29. Stop and restart ICSS.

#### 4.2.5.3 Configure the Net.Commerce Server

Follow the instructions below to configure Net.Commerce:

1. Start your Web browser if it is not running already.
2. Disable all caching (memory and file) and also disable the use of proxy and/or SOCKS servers.
3. From the IBM Net.Commerce Server and Administrator window, start the **Administration Web Server**, a separate ICSS foreground process which is shown in the figure below:

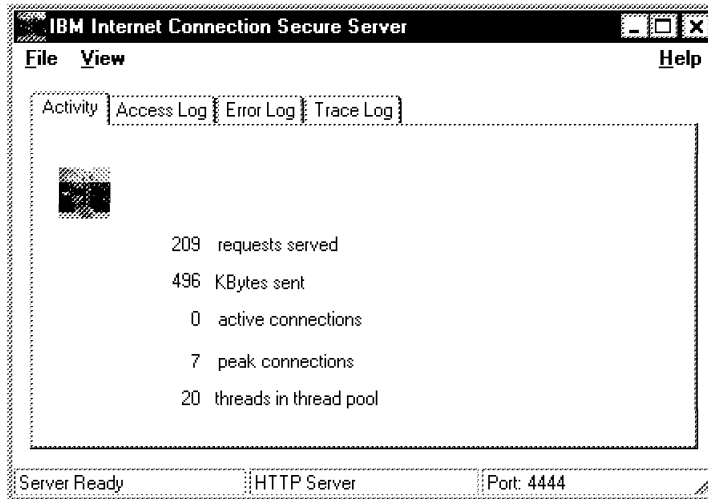


Figure 73. Net.Commerce Administration Web Server

4. Go to URL [http://<your\\_hostname>:4444](http://<your_hostname>:4444) to access the Net.Commerce Administration server pages, where <your\_hostname> is the TCP/IP domain name of your server, and 4444 is the TCP/IP port at which the server is listening.
5. Log on with a user ID of webadmin and a password of webibm which are the default values for the ICSS administrator. You will be presented with the Net.Commerce Server Manager page as shown in the figure below:

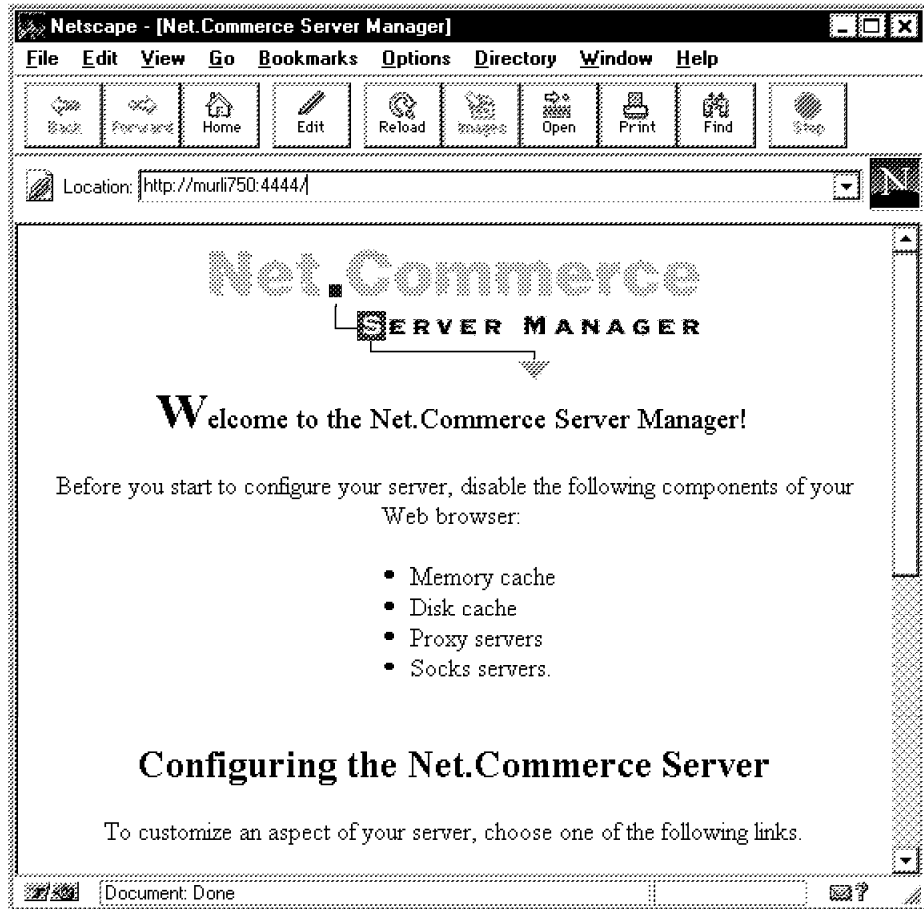


Figure 74. Net.Commerce Server Manager, NT

6. Scroll down and click on **Basic Configuration**.
7. Scroll down and click on **First Step System Configuration** in the lower left-hand corner.
8. Log on again with the same user ID and password as before.
9. Scroll down the System Configuration page and make sure all entries are correct. The table below lists the settings used for our scenario:

Parameter	Value
Host Name	murli750.itso.ral.ibm.com
Port Number	8178
Database Name	mserver
Database Administrator	DB2
Administrator Password	DB2
Log File Location	C:\IBMNetCommercelogs
Number of Processes	2

10. Click on **Submit** to update system configuration. If you are asked to log on again, do so with the same user ID and password as before. A confirmation page will be displayed.

11. Click on **Next Step Access Control**.
12. Scroll down the Access Control page and make sure all entries are correct. The table below lists the settings used for our scenario:

<b>Parameter</b>	<b>Value</b>
Net.Commerce Macro Path	C:IBMNetCommercemacroen_US
Web Server Admin Program	C:WWWbinhtadm.exe
Password File	C:IBMNetCommercepasswordnetc.pwd
Web Server Configuration File	C:WINNThttpd.cnf
Web Server HTML Path	C:WWWHTML
Web Server CGI-BIN Path	C:WWWCGI-Bin

13. Click on **Submit** to update access control. A confirmation page will be displayed.
14. Stop and restart ICSS for changes to become effective.
 

**Note:** Make sure you stop the ICSS service, not the Net.Commerce Administration Web server. Use the Control Panel to stop and restart the ICSS service.
15. Click on **Next Step Database Management**.
16. Click on **Start Database Manager**. A confirmation page will be displayed.
 

**Note:** Do not attempt to create the database from this page as it has already been created at the database server.
17. Click on **Next Step Server Control**.
18. Click on **Enable** to create a Windows NT service definition for the Net.Commerce server which allows you to start the server automatically when the system starts. A confirmation page will be displayed.
19. Click on **Server Control** to get back to the Server Control page.
20. Click on **Startup** to start the Net.Commerce server. A confirmation page will be displayed.
21. Click on **Server Machine (Optional)**.
22. Fill in the necessary information to communicate with the database server. Use values that correspond to the DB2 server configuration as used in 4.2.4.3, "Configure TCP/IP Ports and Services for DB2" on page 111. The following table lists the configuration parameters that were used for our scenario:

<b>Parameter</b>	<b>Value</b>
Service Name	netcomm
TCP/IP Port	11111
Remote Server Name	wtr05192.itso.ral.ibm.com
Remote Note Name	wtr05192
Remote Database Name	mserver

23. Click on **Submit** to update the Server Machine configuration. A confirmation page will be displayed which may indicate a DB2 error 1063. Do not worry about that because mserver.ini will be updated correctly despite that problem. This step also updates the DB2 client configuration.
24. Stop the Administration Web Server.
25. Shut down and restart the system.

#### 4.2.6 Verification of the Installation

There are several checks to verify a successful installation of Net.Commerce. For the sake of simplicity and completeness, we have summarized them in this section. We indicate for each check where during the installation process it would be appropriate to be performed.

1. Check the DB2 logs in the Db2log directory for possible installation problems.
2. Check, using the System program from the Control Panel folder, that the following environment variables have been set for DB2:

```
DB2INSTANCE
DB2PATH
```

This should be done after DB2 has been installed (client and server).

3. Check, from a DB2 command line window, if a DB2 instance has been successfully created. Enter the following command to list DB2 instances:  
db2ilist

If no instance is displayed, create one using the following command:

```
db2icrt DB2
```

This should be done after DB2 has been installed (client and server).

4. Check, using the Services program from the Control Panel folder, that the following Windows NT services have been created for DB2:

```
DB2 Security Server (server and client)
DB2 - <instance_name> (server only)
```

where <instance\_name> is the value of the DB2 instance variable (and also the name of the user ID you have created for DB2). This should be done after DB2 has been installed (client and server).

5. Check, using the Services program from the Control Panel folder, that the Windows NT Internet Connection Server service has been created for ICSS. This should be done after Net.Commerce and ICSS have been installed.

6. Check that ICSS security is working by pointing your browser to the URL:  
[https://<your\\_hostname>/Frntpage.html](https://<your_hostname>/Frntpage.html)

That should cause the browser to notify you that the server has sent a security certificate and that you are about to enter SSL secure mode.

This should be done after ICSS security has been configured.

7. Check that the following files have been created and/or updated with Net.Commerce specific information:

```
WINNTmserver.ini
WINNThttpd.cnf
IBMNetCommercepasswordnetc.pwd
WWWHTMLdb2www.ini
```

## WINNTsystem32driversetcservices

Figure 50 on page 98 illustrates the relationship of the Net.Commerce components and the configuration files.

The following example lists a successful mserver.ini file for the Net.Commerce server using a separate database server:

```
MS_HOSTNAME murli750.itso.ral.ibm.com
MS_PORT 8178
MACRO_PATH C:\IBM\NetCommerce\macro\en_US
MS_HTADMIN C:\WWW\bin\htadm.exe
MS_HTPASS C:\IBM\NetCommerce\password\netc.pwd
MS_LOGPATH C:\IBM\NetCommerce\logs
WWW_CONFIG C:\WINNT\httpd.cnf
WWW_HTML_PATH C:\WWW\HTML
WWW_CGIBIN_PATH C:\WWW\CGI-Bin
MS_DBNAME mserver
MS_DBINST DB2
MS_DBPASS igkekddnkapdfmc
MS_CONFIG http://murli750.itso.ral.ibm.com:4444/admin/ncconfig
NC_DB_SRVNAME netcomm 1
NC_DB_SRVPORT 11111 2
NC_REMOTE_HOST wtr05192.itso.ral.ibm.com 3
MS_NUM_CHILD 2
NC_INST_PATH C:\IBM\NetCommerce
NC_DB_SCRIPT mstblcrt.cmd
NC_DB_SCRIPT_DIR C:\IBM\NetCommerce\bin
MS_TRANS_COUNT 1000
DB_RETRY_LIMIT 0
DB_RETRY_INTERVAL 60
DB_CODESET IBM-850
DB_TERRITORY US
MERCHANT_KEY pfifkjaclogkacpepcifpdfhlmgkfake
NC_DBADMIN DB2
NC_DBADMINPWD igkekddnkapdfmc
NC_DBNODE wtr05192 4
NC_LANG en_US
MS_HTML_MAX 1000000
```

Figure 75. NT mserver.ini File for Multiple Systems

The following is an explanation to the differences between a single and a multiple system configuration file that are highlighted in the figure above:

- 1** : Name of the DB2 TCP/IP service.
- 2** : Name of the TCP port for communication between Net.Commerce and the database server.
- 3** : Fully qualified name for the database machine.
- 4** : DB2 node name for the database machine (maximum 8 characters).

This should be done after Net.Commerce has been configured.

8. Check, using the Services program from the Control Panel folder, that the Windows NT Net.Commerce Server service has been created for Net.Commerce. This should be done after Net.Commerce has been configured.
9. Check that the Net.Commerce-specific configuration for ICSS is working by pointing your browser at the following URL:

[http://<your\\_hostname>/demomall/basemall.htm](http://<your_hostname>/demomall/basemall.htm)

The example below lists the Net.Commerce-specific entries in the WINNThttpd.cnf file:



```
##IBM Net.Commerce PROTECT directives.
Protect /msprotect/* {
ServerId      Private_Authorization
Authtype      Basic
GetMask       All@(*)
PutMask       All@(*)
PostMask      All@(*)
Mask          All@(*)
PasswdFile    C:\IBM\NetCommerce\password\netc.pwd
}
### END OF NET.COMMERCE BLOCK

##IBM Net.Commerce EXEC directives.
Exec          /msprotect/*      C:\IBM\NetCommerce\cgi-bin\*
Exec          /cgi-bin/*       C:\WWW\CGI-Bin\*
### END OF NET.COMMERCE BLOCK

##IBM Net.Commerce PASS directives.
Pass /storengr/* C:\IBM\NetCommerce\html\en_US\ncadmin\storengr\*
Pass /sitemgr/* C:\IBM\NetCommerce\html\en_US\ncadmin\sitemgr\*
Pass /ncsamples/* C:\IBM\NetCommerce\html\en_US\ncsample\*
Pass /ncsample/* C:\IBM\NetCommerce\html\en_US\ncsample\*
Pass /ncagif/* C:\IBM\NetCommerce\html\en_US\ncadmin\gif\*
Pass /ncahelp/* C:\IBM\NetCommerce\html\en_US\ncadmin\help\*
Pass /ncadmin/* C:\IBM\NetCommerce\html\en_US\ncadmin\*
Pass /ncerror/* C:\IBM\NetCommerce\html\en_US\ncerror\*
Pass /ncbooks/* C:\IBM\NetCommerce\html\en_US\ncbooks\*
Pass /te_html/* C:\IBM\NetCommerce\html\en_US\ncadmin\teditor\te_html\*
Pass /demomall/* C:\IBM\NetCommerce\html\en_US\demomall\*
### END OF NET.COMMERCE BLOCK
```

Figure 76. Windows NT httpd.cnf File

This should be done after Net.Commerce has been configured.

10. Check that Net.Data is configured properly. The example below lists the Net.Commerce specific entries in the WWWHTMLdb2www.ini file:

```
ENVIRONMENT (DTW_SQL)      DTWSQL
( IN_DATABASE, LOGIN, PASSWORD, TRANSACTION_SCOPE, SHOWSQL, ALIGN)
ENVIRONMENT (DTW_DEFAULT) DTWFUNC ( )
ENVIRONMENT (DTW_APPLET)  DTWJAVA ( )
ENVIRONMENT (DTW_ODBC)    DTWODBC
( IN_DATABASE, LOGIN, PASSWORD, TRANSACTION_SCOPE, SHOWSQL, ALIGN)
MACRO_PATH C:\IBM\NetCommerce\macro\en_US\ncsample;
C:\IBM\NetCommerce\macro\en_US\demomall;
C:\IBM\NetCommerce\macro\en_US
INCLUDE_PATH C:\IBM\NetCommerce\macro\en_US\ncsample;
C:\IBM\NetCommerce\macro\en_US;
C:\IBM\NetCommerce\html\en_US\ncadmin\teditor;
C:\IBM\NetCommerce\html\en_US
EXEC_PATH C:\IBM\NetCommerce\macro\en_US\ncsample
BIND_FILE C:\IBM\NetCommerce\bin\d2wsq1.bnd
HTML_PATH C:\WWW\HTML
DB2INSTANCE DB2
DB2PATH C:\SQLLIB
```

Figure 77. Windows NT db2www.ini File

This should be done after Net.Commerce has been configured.

11. Check that Net.Commerce macros, CGI-BIN programs and the connection to the database are working properly by registering as a shopper from the following URL:

[http://<your\\_hostname>/cgi-bin/nph-msrvr/register/form](http://<your_hostname>/cgi-bin/nph-msrvr/register/form)

12. Check that the Net.Commerce administration tools and JavaScript are working properly by pointing your browser to the following URL:

http://<your\_hostname>/ncadmin

Log on as user admin with password admin. The Net.Commerce Administrator page will be displayed as shown below:



Figure 78. Net.Commerce Administrator

13. Check that Java and the database connection are working properly.
  - a. Click on **Site Manager**.
  - b. Scroll down and click on **Template Designer**.
  - c. Select the **Load** button.

While the Java applet initializes, it displays status information in the gray bar below the white input area. Check that database information is being retrieved.

#### 4.2.7 Installing the Demo Mall

This section describes the steps necessary to install the demo mall in our scenario.

1. On the Net.Commerce server system, start the Net.Commerce Administration Web server.
2. Start your Web browser and go to URL `http://<your_hostname>:4444` to access the Net.Commerce Administration server pages.
3. Click on **Reconfigure**.

4. Click on **Server Control**.
5. Log on with a user ID of webadmin and a password of webibm. Use this ID and password whenever you are asked to log in again.
6. Scroll down and click on **Shut Down** to stop the Net.Commerce server.  
**Note:** You must stop Net.Commerce server before you go ahead and change the system configuration. If you have multiple Net.Commerce servers, perform these steps on each of them.
7. Check from the Services program that the server has actually been stopped. If there are problems, stop the Net.Commerce service from there.
8. Click on **Reconfigure** again.
9. Click on **System Configuration**.
10. Scroll down and change the database name from mserver to demomall.
11. Click on **Submit** to update the system configuration.
12. Click on **Reconfigure** again.
13. Click on **Server Machine (Optional)**.
14. Scroll down and verify that the database name is now demomall. If it is not, change it to that new name.
15. Click on **Submit** to update the Server Machine configuration. A confirmation page will be displayed which may indicate a DB2 error 1063. Do not worry about that because mserver.ini will be updated correctly despite that problem. This step also updates the DB2 client configuration.
16. On the database server system, copy all files from the IbmNetCommercehtmlen\_USdemomalldata directory on the Net.Commerce server to the same temporary directory on the database server that you used to store the files for the mserver database creation (see 4.2.4.4, "Creating the Net.Commerce Default Database" on page 112), in our case SQLLIBINnetcomm.
17. Edit the democrt.cmd file and remove the path information for the .IXF files (lines 402, 408, 414, 420) and the .BND files (lines 606 to 608). Save the file.
18. Copy the netccur.dll file from the ibmNetCommercebin directory on the Net.Commerce server to the SQLLIBbin directory on the database server.
19. Create a Windows NT command file in the temporary directory on the database server, we called it MSADFUNC.CMD, containing the following information:

```
REM This command file adds two user defined functions (UDFs)
REM to the Net.Commerce database
REM
@echo.
echo Connecting to the database
db2 "CONNECT TO demomall"

echo adding function formatcur()
db2 "CREATE FUNCTION formatcur (float, char(30), char(1)) RETURNS
char(200) NOT FENCED EXTERNAL NAME 'netccur!formatcur' NOT
VARIANT NULL CALL NO SQL PARAMETER STYLE DB2SQL LANGUAGE C NO
EXTERNAL ACTION"

echo adding function formatcurd()
db2 "CREATE FUNCTION formatcurd (float, char(30)) RETURNS char(200)
NOT FENCED EXTERNAL NAME 'netccur!formatcurd' NOT VARIANT NULL
CALL NO SQL PARAMETER STYLE DB2SQL LANGUAGE C NO EXTERNAL ACTION"

echo resetting connection
db2 "CONNECT RESET"
```

Figure 79. MSADFUNC Command File

**Note:** You can copy and paste the DB2 commands from the Readme.txt file of the Net.Commerce Fix Pak #1.

20. Start a **DB2 Command Window** from the DB2 for Windows NT folder.
21. Change to the temporary directory and run the democrt command file, then run the msadfunc command file. This creates the demo mall database. It may take a few minutes to complete.
22. On the Net.Commerce server system, restart the Net.Commerce server using either the Server Control page or the Services program. Make sure the server is back up.
23. Stop the Administration Web Server.

The demo mall should now be ready to use. Connect to it by using the URL [http://<your\\_hostname>/demomall/basemall.htm](http://<your_hostname>/demomall/basemall.htm)

and start shopping. Figure 80 on page 131 shows the demo mall home page:

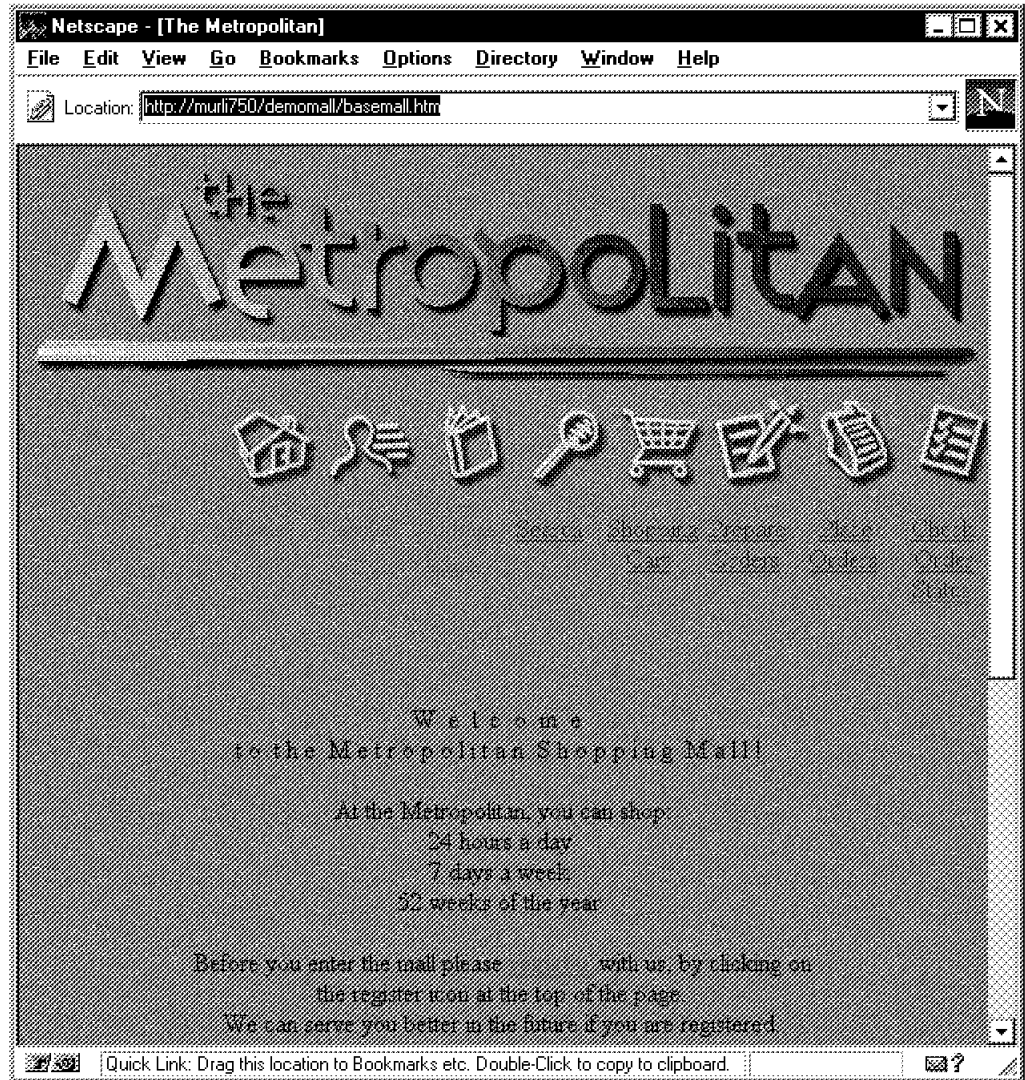


Figure 80. Net.Commerce Demo Mall Home Page, NT

#### 4.2.8 Configure ODBC Support for the Net.Commerce Database

This section describes the steps to configure ODBC access to the Net.Commerce database on the Net.Commerce server system, where DB2 client is running. Similar steps are required to enable ODBC access at the database server, if that is desired.

1. Start the DB2 services on the database server, if they are not running already.
2. From the DB2 for Windows NT window, start **CLI-ODBC Administrator**.
3. A list of local database aliases is displayed that should contain the name of the Net.Commerce database (mserver), as shown in the figure below:

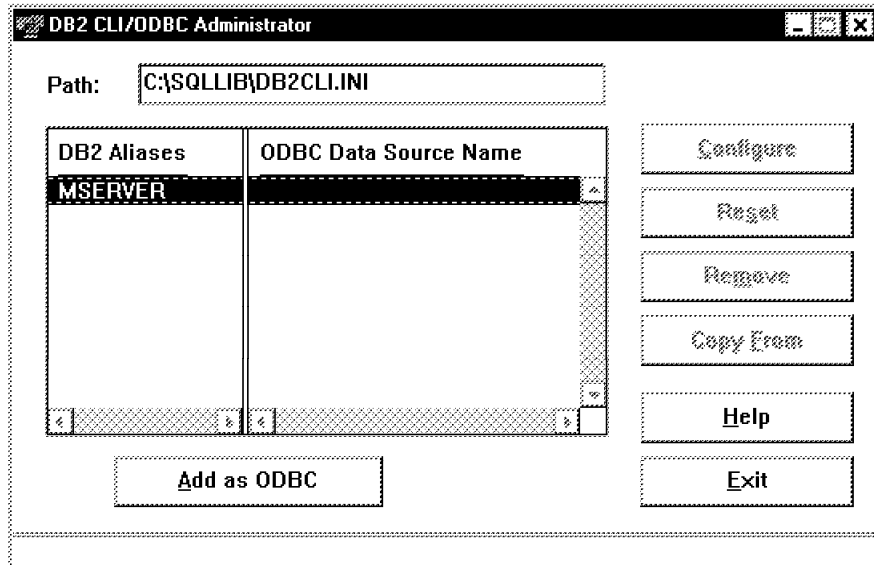


Figure 81. DB2 Client ODBC Administrator

4. Click on **Add as ODBC** to add the definition to the system.
5. No further configuration is required, so just exit CLI-ODBC Administrator.
6. Open the **ODBC** program from the Control Panel folder and click on the **ODBC Drivers** tab. The IBM DB2 ODBC driver should be listed, as shown in the figure below:

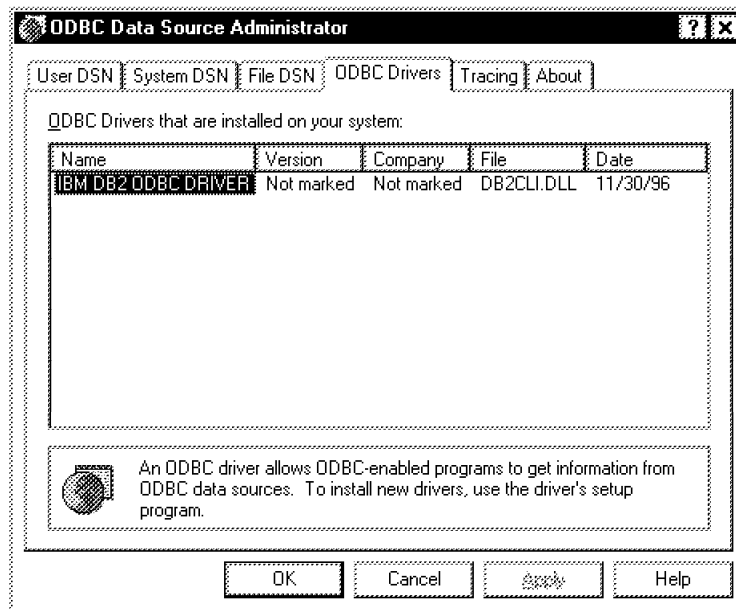


Figure 82. Listing Windows NT ODBC Drivers

7. Click on the **User DSN** tab. The mserver database should be listed there, as shown in the figure below:

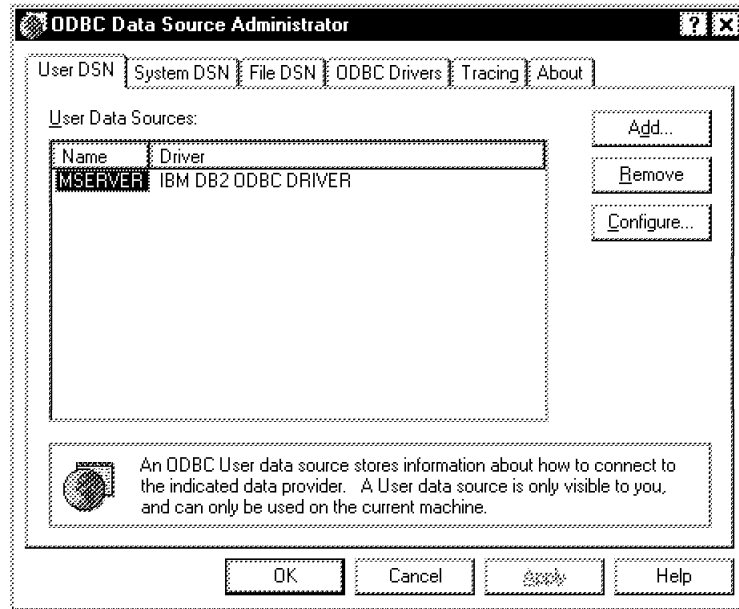


Figure 83. Listing Windows NT ODBC User Data Sources

8. Exit ODBC Data Source Administrator.

## 4.2.9 Basic Performance Tuning

As a first step to achieve better performance with your Net.Commerce system than what would normally be the case when the system defaults are used, we suggest that you do the following:

### 4.2.9.1 System Hardware

Using a multi-processor system will provide better performance because Windows NT can scale running processes over multiple CPUs. For instance, Windows NT 3.51 has been reported to scale well with up to four CPUs.

To reduce the load on a single Net.Commerce system, consider the option of having the database on a separate system as we have shown in our scenario. Refer to 3.2.2, "Using a Multiple System Configuration" on page 48 for a more detailed discussion on multi-system configurations.

### 4.2.9.2 Base Operating System

Set a minimum page file size of 64 MB for the Net.Commerce and database servers. Increase this value if other applications are also running at any server. In that case, we also recommend using more RAM than 64 MB which is considered the minimum for reasonable server performance.

If you are using a Windows NT server for the Net.Commerce system, tune the server service to maximize throughput for network applications. This optimizes server memory for distributed applications that perform their own memory caching, such as DB2 and ICSS.

### 4.2.9.3 Internet Connection Secure Server

You can obtain information about performance tuning and capacity planning for ICSS from the following URL:

<http://www.ics.raleigh.ibm.com>

**Note:** Some performance tips such as, for instance, turning off access logging, placing frequently accessed pages in memory cache, or not using security, may not be desired or are not applicable to a Net.Commerce system for the following reasons:

- You may want to maintain access logs for problem determination.
- Since most of the Net.Commerce pages are dynamic, they cannot be cached. A special cache for those pages will be available with Net.Commerce Version 2.
- You cannot use Net.Commerce without ICSS security.

### 4.2.9.4 DB2 Server

Placing the database log files on a different physical drive may provide some performance gains. Also, running a re-org job against the database every month may keep database server performance at a satisfactory level. For a more detailed discussion on DB2 performance tuning, please refer to the online manuals.

### 4.2.9.5 Net.Commerce Server

The MS\_NUM\_CHILD parameter in the mserver.ini file specifies how many mserverd processes will be run when the Net.Commerce server is started. To increase performance and to meet heavy loads, this number should be increased. Development experience has shown that this parameter scales well up to a value of eight; then no more performance gains can be achieved due to system overhead. As a rule of thumb, calculate 32-64 MB of RAM per two processes of mserverd.

## 4.2.10 Uninstalling the System

The following section explains how Net.Commerce and its related components can be removed from your system.

1. Start the Net.Commerce Administration Web Server.
2. Start your Web browser and access the Administration Web server.
3. Click on **Reconfigure**, then click on **Server Control**.
4. Click on **Disable** to delete the Net.Commerce Server service definition.
5. Stop all components.
6. Make sure you save all files, data, information and databases that you may need again or on another system.
7. To uninstall Net.Commerce, open the **Uninstall IBM Net.Commerce Server** program from the IBM Net.Commerce Server and Administrator folder. Follow the instructions on the screen.
8. To uninstall ICSS, open the **Uninstall IBM Internet Connection Server** program from the IBM Internet Connection Server folder. Follow the instructions on the screen.
9. To uninstall DB2, open the **Uninstall DB2 Products** program from the DB2 for Windows NT folder. Follow the instructions on the screen.



10. Shut down and restart your system.
11. Remove the remaining files and directory trees:
  - Ibm directory tree
  - SQLLIB directory tree
  - Db2log directory tree
  - WINNThttpd.cnf file
  - WINNTmserver.ini file
12. Delete user IDs and groups that have been associated with Net.Commerce components.

---

### 4.3 Using Lotus Approach to Access the Net.Commerce Database

We have explained in previous sections that maintaining data in the database via the Net.Commerce Administrator may be a cumbersome task. We have therefore decided to use Lotus Approach to access the Net.Commerce database and manipulate data.

**Note:** Accessing the Net.Commerce database with Lotus Approach, or any other ODBC-based frontend, for that matter, may have some security implications as far as the data that is being accessed is concerned. Please see 3.3.2, "Accessing Data and the Database" on page 50 for more information on database considerations.

To install Lotus Approach 97 (from Lotus SmartSuite 97), you need the Lotus Approach/SmartSuite CD-ROM. To connect to the database, you need the DB2 ODBC driver from the Net.Commerce setup CD-ROM.

1. Install the DB2 ODBC driver as mentioned in 4.2.2.1, "Installation of DB2 Client Application Enabler" on page 106 and 4.2.3.1, "Installation of DB2 Server" on page 109.
2. Define the Net.Commerce database to be accessed via ODBC as described in 4.2.8, "Configure ODBC Support for the Net.Commerce Database" on page 131.
3. Install Lotus Approach. A default installation of the product is sufficient for our purpose.
4. Start Lotus Approach.
5. Select **Open** from either the wizard, the File menu, or from the smarticon bar.
6. Select **ODBC Data Sources** as the file type, and you should see the databases you added as ODBC.
7. Double-click on a database and enter the user ID and password for the DB2 instance owner. Select to open the database in **Share** mode.
8. Double-click on **<instance>@<database>** to get a list of tables. Figure 84 on page 136 shows the PRODUCT table from the demomall database viewed by Lotus Approach, and the list tables dialog box:

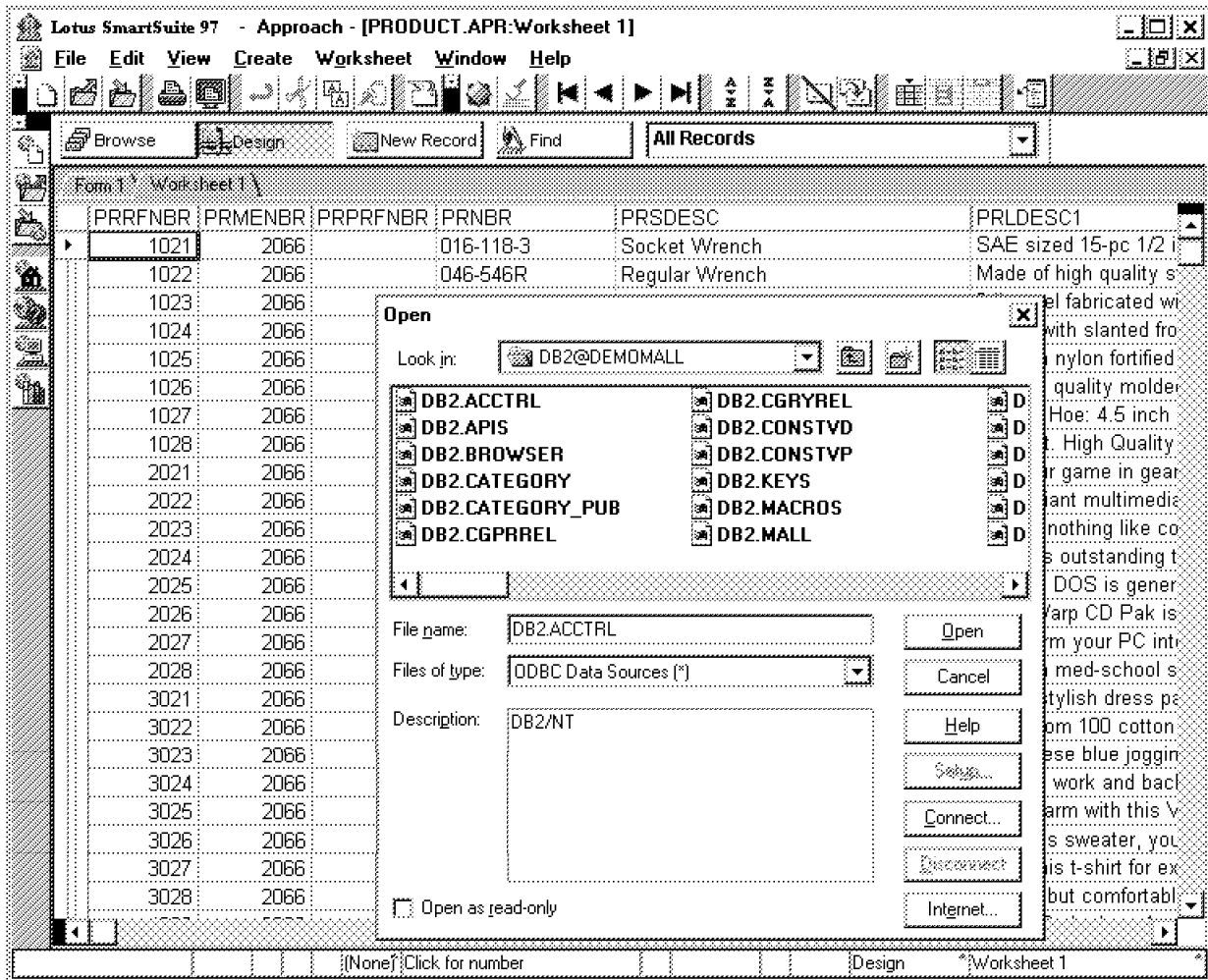


Figure 84. Opening Net.Commerce Databases with Lotus Approach

9. Select a table you want to open. If you are not sure what you're doing and you don't want to change anything in the database, select **Open as Read-only**. Otherwise all changes you make will be transferred to the database immediately.

Before making changes, make sure you have a good knowledge of the structure of the Net.Commerce database. If not, you can damage the integrity of the information in the database which will lead to very unpleasant results.

For an overview of the tables in the Net.Commerce database, use the *Net.Commerce Customization Guide and Reference*.

All data will remain in the database; the design details are stored in a separate .APR file. Joins and design changes you make to keep the overview in the forms or in the worksheet file are kept there.

---

## Chapter 5. Integrating Net.Commerce with CICS

In this chapter we take a look at how Net.Commerce can interface with a typical legacy application written for a CICS environment on a mainframe.

---

### 5.1 General Concept

In our environment, we cannot access the CICS application on the mainframe directly from Net.Commerce. Our mainframe runs on MVS, and our CICS is at CICS/ESA Version 3.3.

To integrate the Net.Commerce system with the existing CICS environment we therefore consider the external access interface that the CICS provides to do this, which means we will use either the External Call Interface (ECI) or the External Presentation Interface (EPI).

#### 5.1.1 External Presentation Interface (EPI)

The EPI allows a non-CICS application program to be viewed as a 3270 application by a CICS server system to which it is connected. When a non-CICS application starts a transaction in a CICS server, 3270 data stream and events are passed between the server and the application; the application can then manipulate this 3270 stream as it wants.

CICS/ESA (Version 4.1 with PTfs UN90142 and UN90143 or later) supports EPI calls from both NT and OS/2 Clients.

#### 5.1.2 External Call Interface (ECI)

The ECI allows a non-CICS application to call a CICS program on a CICS server. The CICS program cannot perform terminal I/O but can access and update all other CICS resources. The same CICS program can also be called by another CICS program using EXEC CICS LINK. Data is exchanged between the two programs by means of a COMMAREA.

CICS/ESA (Version 3.3 or later) supports ECI calls from NT, OS/2, Windows 95, Windows 3.1, and Mac clients.

---

### 5.2 Platform Implementations

We distinguish two platforms on which a Net.Commerce V1.1 system can be installed:

1. AIX
2. Windows NT

#### 5.2.1 AIX Platform

A CICS client for AIX currently cannot communicate directly with a CICS/ESA server, but it can communicate with a CICS server for AIX, Windows NT or OS/2.

### 5.2.1.1 Possible Scenario 1 for AIX

We assume that the target legacy application is a 3270 application.

The following software components are required on AIX:

- Net.Commerce system with DB2 and ICSS.
- SNA Server for AIX to communicate with CICS/ESA via SNA APPC.
- CICS Server for AIX that supports both EPI and ECI; this CICS server will be configured to route the request to the CICS/ESA.
- CICS client for AIX (part of the server package).

A scenario like this is illustrated in Figure 85:

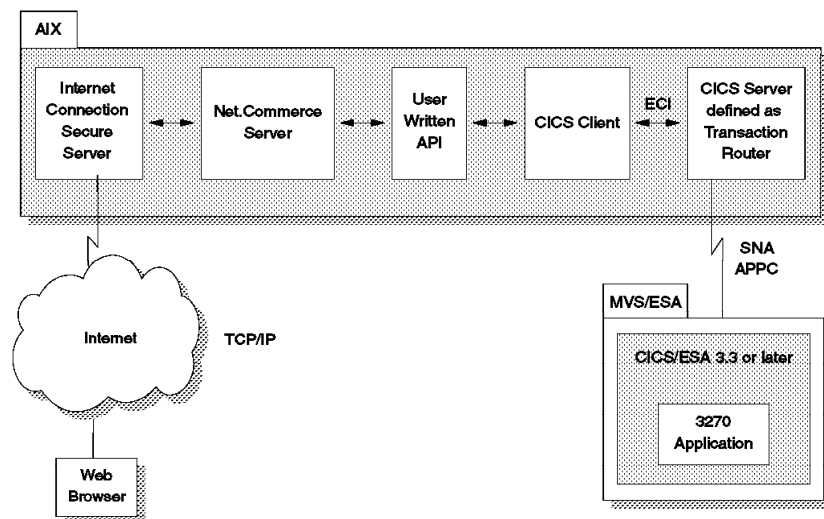


Figure 85. CICS Scenario 1 on AIX

### 5.2.1.2 Possible Scenario 2 for AIX

We assume that the target legacy application has been written to be called by EXEC CICS LINK with COMMAREA option.

The following software components are required on AIX:

- Net.Commerce system with DB2 and ICSS.
- SNA Server for AIX to communicate with CICS/ESA via SNA APPC.
- CICS Server for AIX that supports both EPI and ECI; this CICS server will be configured to route the request to CICS\ESA.
- CICS client for AIX (part of the server package).

A scenario like this is illustrated in Figure 86 on page 139:

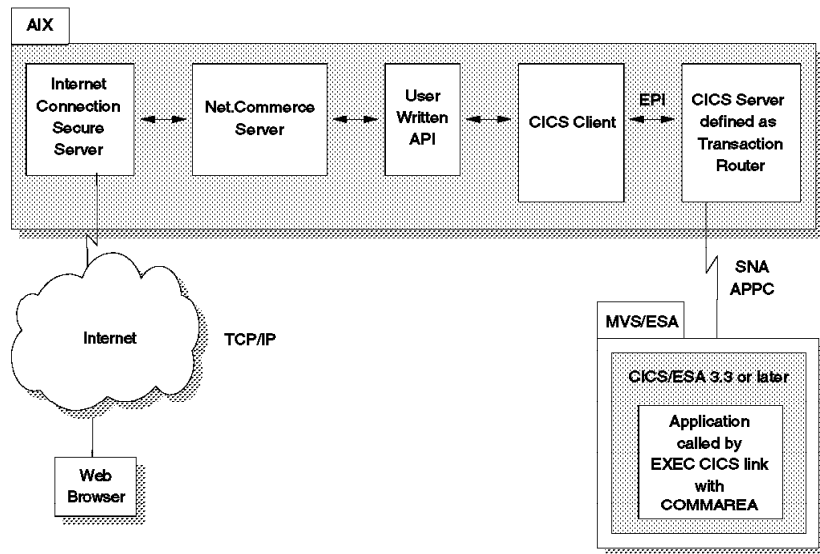


Figure 86. CICS Scenario 2 on AIX

## 5.2.2 Windows NT Platform

A CICS client for NT can communicate directly with a CICS/ESA server, and it can also communicate with a CICS server for AIX, Windows NT or OS/2.

### 5.2.2.1 Possible Scenario 1 for Windows NT

We assume that the target legacy application is a 3270 application and that the CICS/ESA is at Version 3.3, so it does not support EPI.

The following software components are required on Windows NT:

- Net.Commerce system with DB2 and ICSS.
- IBM Communications Server for NT (or IBM Personal Communications or Microsoft SNA Server) to communicate with CICS/ESA via SNA APPC.
- CICS Server for NT that supports both the EPI and the ECI; this CICS server will be configured to route the request to CICS\ESA.
- CICS client for NT (part of the server package).

A scenario like this is illustrated in Figure 87 on page 140:

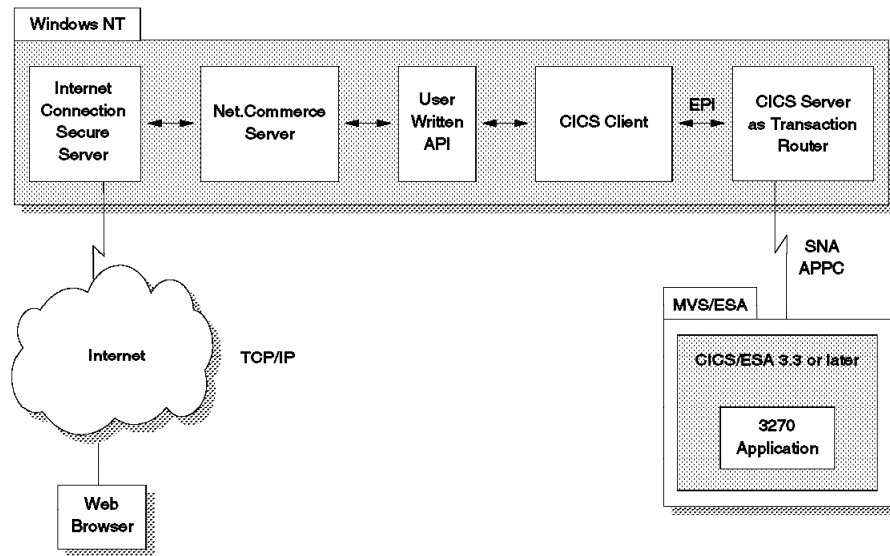


Figure 87. CICS Scenario 1 on Windows NT

### 5.2.2.2 Possible Scenario 2 for Windows NT

We assume that the target legacy application is a 3270 application and that the CICS/ESA (Version 4.1 with PTFs) does support EPI.

The following software components are required on Windows NT:

- Net.Commerce system with DB2 and ICSS
- IBM Communications Server for NT (or IBM Personal Communications or Microsoft SNA Server) to communicate with CICS/ESA via SNA APPC
- CICS client for NT

A scenario like this is illustrated in Figure 88 on page 141:

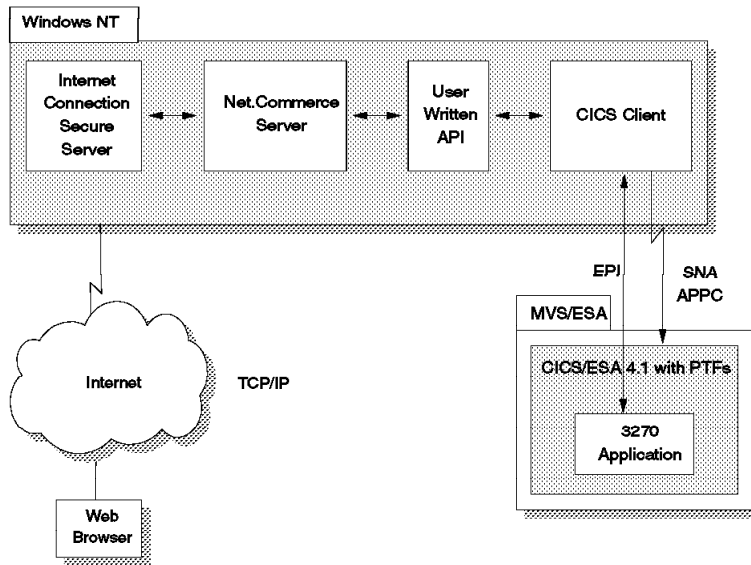


Figure 88. CICS Scenario 2 on Windows NT

### 5.2.2.3 Possible Scenario 3 for Windows NT

We assume that the target legacy application is an application written to be called by EXEC CICS LINK with COMMAREA option.

The following software components are required on Windows NT:

- Net.Commerce system with DB2 and ICSS
- IBM Communications Server for NT (or IBM Personal Communications or Microsoft SNA Server) to communicate with CICS/ESA via SNA APPC
- CICS client for NT

A scenario like this is illustrated in Figure 89 on page 142:

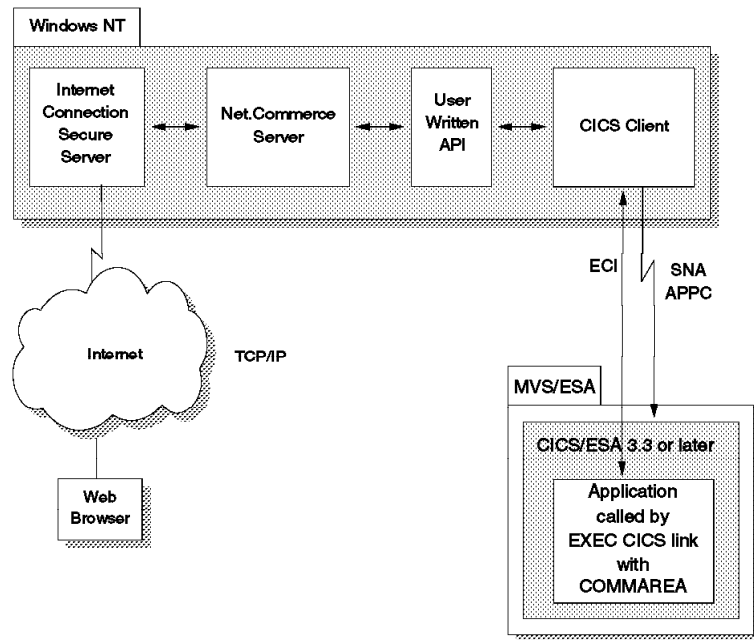


Figure 89. CICS Scenario 3 on Windows NT

---

### 5.3 System Environment

We have chosen to implement the NT Architecture Scenario 1, but we have added an additional system to provide a multiple machine configuration for Net.Commerce for a better load distribution. This is illustrated in Figure 90 on page 143:



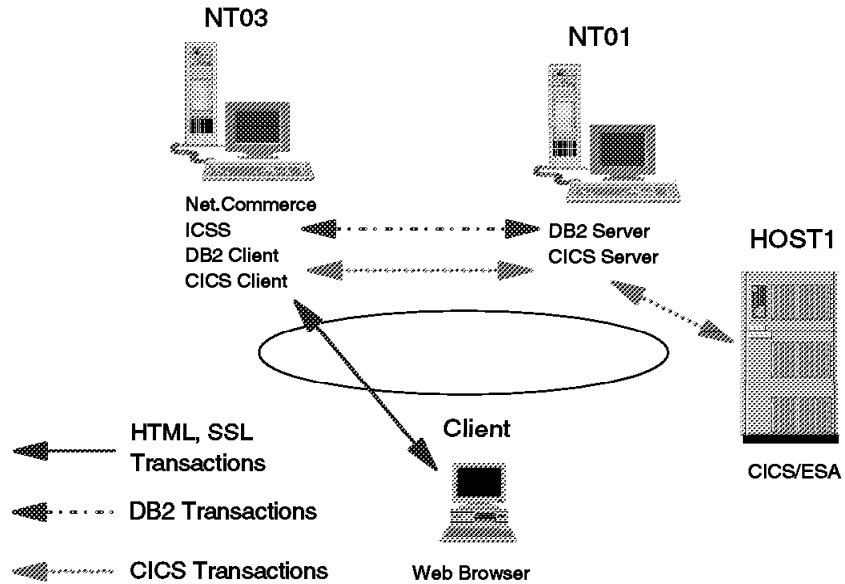


Figure 90. System Environment for Net.Commerce and CICS Integration

The following list shows all the software that we need:

- One Windows NT system which we call NT03 and has the following software installed:
  1. NT Operating System Version 4.0
  2. Net.Commerce system with FixPak #1 which in our case means Net.Commerce server with ICSS Version 4.1 and DB2 Client for NT
  3. IBM CICS client for NT Version 2.0.1, shipped with IBM Transaction Server for NT Version 4.0
  4. IBM CICS Internet Gateway (for testing purposes only), shipped with IBM Transaction Server for NT Version 4.0
- Another NT system which we call NT01, has the following software installed:
  1. NT Operating System Version 4.0
  2. IBM DB2 Server for NT Version 2.1.2
  3. IBM Transaction Server for NT Version 4.0
  4. IBM Personal Communications for NT Version 4.1
- One host, we call it HOST01, which has the following software installed:
  1. OS/390 V1R2
  2. VTAM V4R4
  3. NCP V7R4
  4. CICS/ESA V3R3

### 5.3.1 Additional System Requirements

IBM Transaction Server for NT imposes additional hardware requirements on the database system in our scenario:

- 128 MB memory minimum (192 MB recommended)
- 200 MB page file minimum
- 250 MB disk space for the Transaction Server for NT files (including documentation, SFS log and data volumes, and one CICS region)
- 25 MB disk space for Personal Communications files
- 300 MB disk space for the Microsoft Visual C++ files (including documentation)

For regular Net.Commerce system requirements, please refer to 3.1.5.2, "Windows NT System Requirements" on page 46.

**Note:** IBM Transaction Server for Windows NT also requires Service Pak 2 or later for NT 4.0.

---

## 5.4 The Application Scenario

We chose the demomall application scenario, and in particular we wanted to change the registration task. Our purpose was not to build a real life case but to demonstrate how it is possible to call a CICS/ESA application from Net.Commerce.

We change the demo mall registration form so that there is one more field called CICS USER. This field is associated with the column SHCHAQUE in the SHOPPER table. (Net.Commerce in Version 1.1 does not allow you to associate a new field of the registration form with a customer column as for example SHFIELD1 or SHFIELD2.)

Our API function ensures that the values for SALNAME, SAADDR1, SACITY, SASTATE, SACNTRY and SAZIPC are not null. Then, if the field SCHAQUE that for us is CICS USER is not blank, it calls the transaction CESN which is only defined on the CICS server on NT01. The CESN transaction is the typical CICS logon transaction. (We have to call it on the NT server because it is not defined on our CICS/ESA.)

If the logon is successful (the password is the Net.Commerce password that the shopper is asked to provide), it calls another transaction EPI1 that is a routed transaction defined on the CICS Transaction Server for NT and routed to CICS/ESA on HOST1. (On the host this transaction is defined as CEDA.) The API checks that the CEDA transaction has been successfully called. No further processing is done on the mainframe.

---

## 5.5 Mainframe Definitions

In this section, we show the definitions of VTAM and CICS on the mainframe as they relate to our scenario.

## **5.5.1 VTAM/NCP Definitions**

In our case, VTAM was set up for automatic configuration, so no static switched major node and cross domain resource manager definitions were required for PUs and independent LUs. Therefore, we include displays from NetView NCCF to show the active configuration parameters of the used resources.

### **5.5.1.1 VTAM Start Options**

Figure 91 on page 146 shows the start options for VTAM subarea 18.

```

IST1188I ACF/VTAM V4R4   STARTED AT 06:36:30 ON 05/15/97
IST1349I COMPONENT ID IS 5695-11701-401
IST1348I VTAM STARTED AS INTERCHANGE NODE
IST1189I ALSREQ = NO
IST1189I ASIRFMSG = OLUSSCP
IST1189I AUTHLEN = YES
IST1189I AUTOTI = 0
IST1189I BNDYN = ***NA***
IST1189I BSCMDRS = (STATS,INOPS)
IST1189I CACHETI = 8
IST1189I CDRSCTI = 480S
IST1189I CDSREFER = 1
IST1189I CMPMIPS = 100
IST1189I CNMTAB = ISTMGCOO
IST1189I CONFIG = IO
IST1189I CPCDRSC = NO
IST1189I CSALIMIT = NOLIMIT
IST1189I DATEFORM = MDY
IST1189I DIRTIME = 691200S
IST1189I DLRTCB = 32
IST1189I DSPLYMAX = 65535
IST1189I DYNADJCP = YES
IST1189I DYNDLGMD = NONE
IST1189I DYNMODTB = NONE
IST1189I ENCRYPTN = NO
IST1189I ESIRFMSG = ALLSSCP
IST1189I FSIRFMSG = OLUSSCP
IST1189I HNTSIZE = 4080
IST1189I HOSTSA = 18
IST1189I HPR = (RTP,ANR)
IST1189I HPRPST = MEDIUM
IST1189I HPRPST = NETWRK
IST1189I INITDB = ALL
IST1189I IOINT = 0
IST1189I IOPURGE = 300S
IST1189I ISTCOSDF = INDLU
IST1189I LIST = IO
IST1189I MAXLOCAT = 5000
IST1189I MAXSSCPS = 10
IST1189I MIHTMOUT = 1800
IST1189I MSGMOD = NO
IST1189I MXSSCPRU = 4096
IST1189I NCPBUFSZ = 512
IST1189I NMVTLOG = NPDA
IST1189I NODETYPE = NN
IST1189I NSRTSIZE = *BLANKS*
IST1189I OSIEVENT = PATTERNS
IST1189I OSITOP0 = LINES
IST1189I OSRTSIZE = 43
IST1189I PIUMAXDS = 200
IST1189I PPOLOG = YES
IST1189I RESUSAGE = 100
IST1189I SAWMAXDS = 100
IST1189I SDLCMDRS = (STATS,INOPS)
IST1189I SIRFMSG = ALLSSCP
IST1189I SMEAUTH = DISCARD
IST1189I SORDER = APPN
IST1189I SRCOUNT = 10
IST1189I SSCPDYN = YES
IST1189I SSCPNAME = RAI
IST1189I SSDTMOUT = 30
IST1189I STRGR = ***NA***
IST1189I SUPP = NOSUP
IST1189I TNSTAT = CNSL,TIME=60
IST1189I UPDDDELAY = 60S
IST1189I VERIFYCP = NONE
IST1189I VFYREDTI = OFF
IST1189I VRTGCPCP = YES
IST1189I WARM = NO
IST1189I XNETALS = YES
IST314I END

APPNCOS = NONE
ASYDE = TERM
AUTORTRY = AUTOCAP
BN = NO
BNORD = ***NA***
BSCTMOUT = 286
CDRDYN = YES
CDSERVR = NO
CINDXSIZ = 8176
CMPVTAM = 0
COLD = YES
CONNTYPE = APPN
CPCP = YES
CSA24 = NOLIMIT
DIRSIZE = 0
DISCNTIM = (15,0)
DSPLYDEF = 100
DSPLYWLD = FULLWILD
DYNASSCP = YES
DYNLU = YES
ENCRPREF = NONE
ENHADDR = NO
FLDTAB = ISTMSFLD
GWSSCP = YES
HOSTPU = ISTPUS18
HOTIOTRM = 0
HPRPST = LOW
HPRPST = HIGH
HSRTSIZE = 9973
INOPDUMP = OFF
IOMSGLIM = 2147483647
IRNSTRGE = 0
LIMINTCP = ***NA***
MAINTLVL = *BLANKS*
MAXLURU = 6144
MAXSUBA = 15
MSGLEVEL = BASE
MXSAWBUF = 10000
MXSUBNUM = 511
NETID = USIBMRA
NODELST = *BLANKS*
NQNM0DE = NAME
NUMTREES = 100
OSIMGMT = YES
OSITOP0 = ILUCDRSC
PDTRCBUF = 2
PLUALMSG = NOSUPP
PSSTRACE = NORB
ROUTERES = 128
SAWMXQPK = 0
SECLVLCP = ***NA***
SLUALMSG = NOSUPP
SNAPREQ = 1000
SRCHRED = OFF
SRTIMER = 30S
SSCPID = 18
SSCPORD = PRIORITY
SSEARCH = YES
STRMNPS = ***NA***
SWNORDER = CPNAME
TRANSLAT = (0,1,2,3,4,5,6,7)
USSTAB = *BLANKS*
VFYRED = YES
VRTG = YES
VTAMEAS = 32001
XCFINIT = ***NA***

240S
60S
480S
120S
1
1
2
2
3

```

Figure 91. VTAM Options for SA18

The following is an explanation to the points highlighted in the example above:

- 1** DYNLU=YES and CDRDYN=YES allow dynamic definition of independent LUs.
- 2** The combination of HOSTSA and NODETYPE=NN defines this VTAM as an interchange node, providing both subarea and APPN functions.
- 3** In VTAM, the SSCPNAME is also used as the APPN CP name.

#### **5.5.1.2 Dynamic Definition of Switched PUs and Dependent LUs**

A device coming in to VTAM over a LAN is considered to be a switched device and must have a switched major node to define the PU and dependent LUs associated with it. VTAM provides a configuration services XID exit, ISTECCS, which among other things allows dynamic definition of switched devices. The sample exit shipped with VTAM will intercept the XID from an incoming switched device and based on the IDBLK and IDNUM or CPNAME generate a dynamic PU definition. The name generated in our example is done by the name generation table in the exit and bases the definition on a predefined model PU.

For example, in our scenario, the workstation uses an IDBLK of 05D. Based on definitions coded in the exit for an '05D' IDBLK, the exit uses a PU model called PUMOD05D, an LU model called LUMOD05D, and uses the letter "W" as a name prefix to build a PU definition. The model PU and LU definitions are defined in a VTAM by activating a model major node (Figure 92 on page 148). The PU name is defined using the prefix "W" and the IDNUM.

```

***** 00950000
* MODEL FOR IDBLK X'05D' - PC 3270 EMULATION * 01050000
***** 01150000
SWMODEL VBUILD TYPE=MODEL
PUMOD05D PU ADDR=13, X
          DISCNT=NO, X
          CONNTYPE=APPN, X
          MAXDATA=1033, X
          PUTYPE=2
*
LU6205D LU LOCADDR=0,DLOGMOD=DSIL6MOD
LUMOD05D LU LOCADDR=2, X
          PACING=0, X
          DLOGMOD=D4C32XX3, X
          MODETAB=ISTINCLM, X
          USSTAB=US327X, X
          VPACING=0
***** 00950000
* MODEL FOR IDBLK X'017' - PC 3270 EMULATION * 01050000
***** 01150000
PUMOD017 PU ADDR=C1, X01200000
          ANS=CONT, X01250000
          PUTYPE=2, 01300000
LUMOD017 LU LOCADDR=2, X01350000
          MODETAB=ISTINCLM, X01400000
          USSTAB=ISTINCLM, X01450000
          PACING=1, 01500000
***** 01550000
* MODEL FOR IDBLK X'056' - AS/400 * 01650000
***** 01750000
PUMOD056 PU ADDR=01, X01800000
          ANS=CONT, X01850000
          PUTYPE=2, 01900000
LUMOD056 LU LOCADDR=2, X01950000
          MODETAB=ISTINCLM, X02000000
          USSTAB=ISTINCLM, X02050000
          PACING=7, 02100000
LUMODP56 LU LOCADDR=2,DLOGMOD=M2SDLCQ,MODETAB=AMODETAB
*
***** 01550000
* MODEL FOR IDBLK X'012' - DEC * 01650000
***** 01750000
PUMOD012 PU ADDR=C1, +
          MAXDATA=521, +
          MAXOUT=7,PASSLIM=7, +
          PUTYPE=2
LUMODA12 LU LOCADDR=0,DLOGMOD=SNASVCMG,MODETAB=MTAPP
LUMODB12 LU LOCADDR=0,DLOGMOD=SNASVCMG,MODETAB=MTAPP
LUMODC12 LU LOCADDR=2,DLOGMOD=M2SDLCQ,MODETAB=AMODETAB
LUMODC13 LU LOCADDR=3,DLOGMOD=M2SDLCQ,MODETAB=AMODETAB
LUMODC14 LU LOCADDR=4,DLOGMOD=M2SDLCQ,MODETAB=AMODETAB
***** 00950000
* * 01000000
* MODEL FOR IDBLK X'061' - PC 3270 EMULATION * 01050000
* * 01100000
***** 01150000
PUMOD061 PU ADDR=C1, X01200000
          ANS=CONT, X01250000
          PUTYPE=2, 01300000
LUMOD061 LU LOCADDR=2, X01350000
          MODETAB=ISTINCLM, X01400000
          USSTAB=ISTINCLM, X01450000
          PACING=1, 01500000

```

Figure 92. Models VTAMLST Entry

### 5.5.1.3 PU Configuration for the Windows NT System

In our scenario the workstation uses an IDNUM of 05253 so the PU name is W05253. This is configured on the workstation as shown in Figure 109 on page 158. The following screen shows the active SNA PU configuration for our CICS server on NT:

```
NCCF          N E T V I E W      RAIAN MURLI    05/21/97 17:28:27 A
C RAIAN      DISPLAY NET,ID=W05253,SCOPE=ALL
  RAIAN      IST097I DISPLAY ACCEPTED
' RAIAN
IST075I NAME = W05253          , TYPE = PU_T2.1
IST486I STATUS= ACTIV--LX-, DESIRED STATE= ACTIV
IST1043I CP NAME = WTR05253, CP NETID = USIBMRA , DYNAMIC LU = YES
IST1589I XNETALS = YES
IST1105I RESOURCE STATUS TGN CP-CP TG CHARACTERISTICS
IST1106I W05253 AC/R  21 YES  982D0000000000000000000017100808080
IST1482I HPR = ANR - OVERRIDE = N/A - CONNECTION = NO
IST136I SWITCHED SNA MAJOR NODE = ISTDSWMN
IST081I LINE NAME = J0006013, LINE GROUP = EGO6L02 , MAJNOD = RA6NCPX
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST355I LOGICAL UNITS:
IST080I W0525302 ACT/S---X- W0525303 ACTIV---X- W0525304 ACTIV---X-
IST080I W0525305 ACTIV---X-
IST080I WTR05253 ACT/S----Y 2
IST314I END
-----
```

Figure 93. Display of PU W05253 from VTAM

More information on the VTAM configuration services XID exit can be found in *VTAM V4R4 Customization*, LY43-0075.

### 5.5.1.4 Dynamic Definition of Independent LUs

CP-CP sessions with VTAM require both cross-domain resource (CDRSC) definitions and adjacent CP (ADJCP) definitions to represent the partner CP. The VTAM start option DYNADJCP=YES allows dynamic definitions of adjacent CPs. Dynamically defined adjacent CPs are put in the VTAM major node ISTADJCP.

Independent LUs (ILUs) attaching to VTAM through a type 2.1 PU can be dynamically defined as cross-domain resources (CDRSCs). The PU becomes the adjacent link station (ALS) for the ILU. Dynamic CDRSCs are put in the VTAM major node ISTCDRDY (**1**).

In the VTAM start options we have coded DYNLU=YES and CDRDYN=YES to allow dynamic CDRSC definitions to be created for independent LUs. Adjacent CPs are independent LUs and are defined in the same way. See Figure 91 on page 146 for a complete listing of the VTAM options for this scenario.

Our workstation PU, W05253, is the adjacent link station for our independent LU, WTR05253 (**2**). On the workstation, this is configured as shown in Figure 106 on page 156. The following screen shows the active SNA independent LU 6.2 configuration for our CICS server on NT:

```

NETVIEW      RAIAN MURLI      05/21/97 17:30:13 A
C RAIAN      DISPLAY NET,ID=WTR05253,SCOPE=ALL
RAIAN        IST097I DISPLAY ACCEPTED
' RAIAN
IST075I NAME = USIBMRA.WTR05253 , TYPE = ADJACENT CP
IST486I STATUS= ACT/S----Y, DESIRED STATE= ACTIV
IST1447I REGISTRATION TYPE = NO
IST977I MDLTAB=***NA*** ASLTAB=***NA***
IST1333I ADJLIST = ***NA***
IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
IST934I DLOGMOD=CPSVCMG USS LANGTAB=***NA***
IST597I CAPABILITY-PLU ENABLED ,SLU ENABLED ,SESSION LIMIT NONE
IST231I CDRSC MAJOR NODE = ISTCDRDY 1
IST1184I CPNAME = USIBMRA.WTR05253 - NETSRVR = ***NA***
IST1044I ALSLIST = ISTAPNPU
IST082I DEVTYPE = INDEPENDENT LU / CDRSC
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000005, SESSION REQUESTS = 0000000000
IST206I SESSIONS:
IST1081I ADJACENT LINK STATION = W05253 2
IST634I NAME STATUS SID SEND RECVR TP NETID
IST635I RAIAC ACTIV-S F2A3256035CF5564 0006 0006 0 0 USIBMRA
IST635I RAIAC ACTIV/SV-S F2A3256035CF5563 0001 0001 0 0 USIBMRA
IST635I RAI ACTIV/CP-S F2A3256035CF5562 05C0 0001 0 0 USIBMRA
IST635I RAIAC ACTIV/SV-P F86FE1644B8A79E8 0000 0001 0 0 USIBMRA
IST635I RAI ACTIV/CP-P F86FE1644B8A6DBC 0001 05C3 0 0 USIBMRA
IST924I -----
IST075I NAME = USIBMRA.WTR05253 , TYPE = DIRECTORY ENTRY
IST1186I DIRECTORY ENTRY = REGISTERED EN
IST1184I CPNAME = USIBMRA.WTR05253 - NETSRVR = USIBMRA.RAI
IST314I END

```

Figure 94. Display of WTR05253 from VTAM

The following screen shows the active APPN configuration for our CICS server on NT:

```

DISPLAY NET,ID=ISTADJCP,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST075I NAME = ISTADJCP , TYPE = ADJCP MAJOR NODE
IST486I STATUS= ACTIV , DESIRED STATE= ACTIV
IST1100I ADJACENT CONTROL POINTS FROM MAJOR NODE ISTADJCP
IST1102I NODENAME NODETYPE CONNECTIONS CP CONNECTIONS NATIVE
IST1103I USIBMRA.WTR05253 EN 1 1 *NA*

```

Figure 95. ISTADJCP Display

The following screen shows the active CDRSC major node for dynamically defined CDRSCs. Our CICS server on NT is represented by the WTR05253 entry:

```

CNMKWIND OUTPUT FROM DIS ISTCDRDY LIN
DISPLAY NET,ID=ISTCDRDY,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST075I NAME = ISTCDRDY , TYPE = CDRSC SEGMENT
IST486I STATUS= ACTIV , DESIRED STATE= ACTIV
IST478I CDRSCS:
IST483I RAK ACT/S----Y, CDRM = RAI , NETID = USIBMRA
IST483I RAP ACT/S----Y, CDRM = RAI , NETID = USIBMRA
IST483I RAB ACT/S----Y, CDRM = RAI , NETID = USIBMRA
IST483I WTR05253 ACT/S----Y, CDRM = ***NA***, NETID = USIBMRA
IST1500I STATE TRACE = OFF
IST314I END

```

Figure 96. ISTCDRDY Display



The following screen shows the active VTAM APPN directory entry for our CICS server on NT:

```
* RAIAN      D NET,DIRECTRY,ID=WTR05253
  RAIAN      IST097I  DISPLAY  ACCEPTED
' RAIAN
IST350I     DISPLAY TYPE = DIRECTORY
IST1186I    DIRECTORY ENTRY = REGISTERED EN
IST1184I    CPNAME = USIBMRA.WTR05253 - NETSRVR = USIBMRA.RAI
IST314I     END
```

Figure 97. VTAM APPN Directory Entry for WTR05253

### 5.5.1.5 CICS Application Major Node

The following screen shows the application major node for the CICS system on subarea 18:

```
VBUILD TYPE=APPL
RAIAC      APPL ACBNAME=RAIAC,MODETAB=MQMTAB,EAS=20,SONSCIP=YES,          *00083000
           AUTH=(ACQ,SPO,VPACE,PASS),PARSESS=YES,VPACING=5
```

Figure 98. CICS Application Major Node

Note that the LU62APPC log mode is an entry in this table. This is the name that must be configured on the NT system, as shown in Figure 113 on page 160.

### 5.5.1.6 Independent LU 6.2 Configuration for the CICS/ESA Application

The following screen shows the active SNA independent LU 6.2 configuration for our CICS application on the host, having sessions with the NT system:

```
NCCF                N E T V I E W      RAIAN MURLI    05/21/97 17:39:39 A
C RAIAN             DISPLAY NET,ID=RAIAC,SCOPE=ALL
  RAIAN             IST097I  DISPLAY  ACCEPTED
' RAIAN
IST075I             NAME = USIBMRA.RAIAC      , TYPE = APPL
IST486I             STATUS= ACT/S            , DESIRED STATE= ACTIV
IST1447I            REGISTRATION TYPE = CDSERVR
IST977I             MDLTAB=***NA*** ASLTAB=***NA***
IST861I             MODETAB=MQMTAB  USSTAB=***NA*** LOGTAB=***NA***
IST934I             DLOGMOD=***NA*** USS LANGTAB=***NA***
IST1632I            VPACING = 5
IST597I             CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
IST231I             APPL      MAJOR NODE = RAIBCICS
IST654I             I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I            STATE TRACE = OFF
IST271I             JOBNAME = CICS18  , STEPNAME = CICS18  , DSPNAME = ISTE4476
IST1050I            MAXIMUM COMPRESSION LEVEL - INPUT = 0  , OUTPUT = 0
IST1633I            ASRCVLM = 1000000
IST1634I            DATA SPACE USAGE: CURRENT =          0 MAXIMUM =          0
IST171I             ACTIVE SESSIONS = 0000000004, SESSION REQUESTS = 0000000000
IST206I             SESSIONS:
IST634I             NAME      STATUS      SID          SEND RECVR TP NETID
IST635I             WTR05253  ACTIV/SV-S  F86FE1644B8A79E8 0001 0000  0  0 USIBMRA
IST635I             RAKTX011  ACTIV-S     F8D3D1640B69A874 0016 0012  0  0 USIBMRA
IST635I             WTR05253  ACTIV-P     F2A3256035CF5564 0002 0002  0  0 USIBMRA
IST635I             WTR05253  ACTIV/SV-P  F2A3256035CF5563 0001 0001  0  0 USIBMRA
IST314I             END
-----
```

Figure 99. Display of CICS Application RAIAC from VTAM

### 5.5.1.7 Service Mode Definition for LU 6.2

The following screen shows the MQMTAB SNA service mode table entry for the LU 6.2 sessions between CICS/ESA and the NT server:

```

MQMTAB MODETAB
*****
* VTAM MODE TABLE ENTRY FOR APPC SESSIONS.
*****
LU62APPC  MODEENT LOGMODE=LU62APPC,                X
          FMPROF=X'13', TSPROF=X'07',              X
          PRIPROT=X' B0', SECPR0T=X' B0',          X
          COMPROT=X'50B5', RUSIZES=X'8989',        X
          PSERVIC=X'06020000000000000002F00',    X
          TYPE=X'00'
          MODEEND
          END

```

Figure 100. Mode Table Entry for LU62APPC

The RUSIZES parameter specifies the maximum length, in bytes, of the request units (RUs) that the primary LU (PLU) and secondary LU (SLU) can send to each other. Using the table in the Logon Mode Table section of *VTAM Resource Definition Reference*, SC31-8377, you can translate RUSIZES=X'8989' to RUSIZE=4096 for both PLU and SLU. This is the size you have to configure for the NT system, as shown in Figure 114 on page 161.

## 5.5.2 CICS/ESA Definitions

In order to establish a communication between your CICS server on NT and the CICS/ESA on the host, a connection and a session profile need to be created in CICS/ESA that hold the characteristics of the LU 6.2 connections.

1. Log on to CICS on the host.
2. Enter CEDA to start the CEDA transaction, shown on the screen below:

```

ENTER ONE OF THE FOLLOWING

ADd
ALter
APpend
CHeck
COpy
DEFine
DElete
DIsply
EXpand
InstaLL
Lock
Move
REMove
REName
UNlock
USerdefine
View

APPLID=RAIAC

PF 1 HELP      3 END      6 CRSR      9 MSG      12 CNCL

```

Figure 101. CICS Configuration Using the CEDA Transaction

3. Enter DEF C to create a connection profile. On the screen below, we show the definitions of our profile, NTCO, located in the ITSCSAMP group:

```

OVERTYPE TO MODIFY                                CICS RELEASE = 0330
CEDA DEFINE
Connection      : NTCO
Group           : ITSCSAMP
Description     ==> MARTIN'S NT CONNECTION
CONNECTION IDENTIFIERS
Netname        ==> WTR05253
INDsys         ==>
REMOTE ATTRIBUTES
REMOTESystem   ==>
REMOTENAME     ==>
CONNECTION PROPERTIES
Accessmethod   ==> Vtam           Vtam | IRc | INdirect | Xm
Protocol       ==> Appc          Appc | Lu61
Singlesess    ==> No            No | Yes
DATAstream    ==> User          User | 3270 | SCs | STRfield | Lms
RECORDformat  ==> U             U | Vb
OPERATIONAL PROPERTIES
+ AUTOconnect  ==> No           No | Yes | All
INService     ==> Yes          Yes | No
SECURITY
SECURITYname   ==>
ATTACHSEC     ==> Local        Local | Identify | Verify | Persistent
| Mixidpe
BINDPassword   ==>
BINDSecurity   ==> No          PASSWORD NOT SPECIFIED
No | Yes
APPLID=RAIAC
PF 1 HELP 2 COM 3 END                6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Figure 102. CICS Connection Profile Configuration

The value for Netname must match the name of the independent LU 6.2 for the NT system, as shown in Figure 94 on page 150 (VTAM) and Figure 106 on page 156 (Personal Communications).

**Note:** The + indicates that this information is actually displayed on multiple screens that you can scroll using the PF7 and PF8 keys.

4. Enter DEF S to create a session profile. On the screen below, we show the definitions of our profile, NTSE, located in the ITSCSAMP group:

```

OVERTYPE TO MODIFY                                CICS RELEASE = 0330
CEDA DEFINE
Sessions      : NTSE
Group         : ITSCSAMP
Description   ==> MARTINS NT SESSION
SESSION IDENTIFIERS
Connection    ==> NTCO
SESSName     ==>
NETnameq     ==>
MOdename     ==> LU62APPC
SESSION PROPERTIES
Protocol      ==> Appc           Appc | Lu61
MAXimum      ==> 010 , 005      0-999
RECEIVEPfx   ==>
RECEIVECount ==>                1-999
SENDPfx      ==>
SENDCount    ==>                1-999
SENDSIZE     ==> 04096          1-30720
+ RECEIVESize ==> 04096          1-30720
SESSPriority ==> 000             0-255
Transaction  :
OPERATOR DEFAULTS
OPERId       :
OPERPriority : 000              0-255
OPERRs1     : 0                 0-24,...
OPERSecurity : 1                 1-64,...
+ PRESET SECURITY
USERId      ==>
OPERATIONAL PROPERTIES
Autoconnect ==> No              No | Yes | All
INservice   :                   No | Yes
Buildchain  ==> Yes             Yes | No
USERArealen ==> 000             0-255
IOarealen   ==> 00000 , 00000  0-32767
RELreq      ==> No              No | Yes
+ DIScreq   ==> No              No | Yes
NEPclass    ==> 000             0-255
RECOVERY
RECOVOption ==> Sysdefault      Sysdefault | Clearconv | Releasesess
| Uncondrel | None
RECOVNotify ==> None            None | Message | Transaction
APPLID=RAIAC

PF 1 HELP 2 COM 3 END          6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Figure 103. CICS Session Profile Configuration

The value for MOdename must match the values in the logmode entry on the host, shown in Figure 100 on page 152, and the Mode name on the NT system, as shown in Figure 113 on page 160.

The values for MAXimum must match the values for PLU mode session limit and Minimum contention winner sessions, as shown in Figure 113 on page 160.

The values for SENDSize and RECEIVESize must match the values for RUSIZES in Figure 100 on page 152 and Maximum RU size in Figure 114 on page 161.

**Note:** The + indicates that this information is actually displayed on multiple screens that you can scroll using the PF7 and PF8 keys.

5. Exit CICS by entering CESF LOGOFF.
6. Restart the CICS/ESA region.

## 5.6 Configuring the SNA Connection with IBM Personal Communications

In this section, we show the definitions of Personal Communications as they relate to our scenario.

For the following explanations, we assume that you are basically familiar with IBM Personal Communications and SNA APPC. You should already have IBM Personal Communications installed with the 3270 Emulation and the Communications API components selected. You should also have the IBM LLC2 protocol driver installed that comes with Personal Communications. That driver implements the IEEE 802.2 LAN interface over which your NT server communicates with the host.

**Note:** We will explain our configuration for this scenario which is an APPN environment. It is meant to help you understand this scenario and as a general guidance to other installations. There are many other configurations possible, and your SNA environment may be different. Please check with your system administrator about that.

For details on the installation and configuration of Personal Communications, please refer to the product documentation.

**Note:** You can, of course, use IBM Communications Server for Windows NT or the Client Access Feature thereof, or Microsoft SNA Server, to set up your SNA connections. We used Personal Communications because it imposes less demand on the system than a server, and it provides terminal emulation which is necessary for the CICS/ESA configuration as well as problem determination.

### 5.6.1 Configure the SNA Node

Follow the instructions below to create an SNA node definition with Personal Communications:

1. From the IBM Personal Communications folder, open the **Administrative and PD Aids** folder, then start **SNA Node Operations** as shown in the figure below:

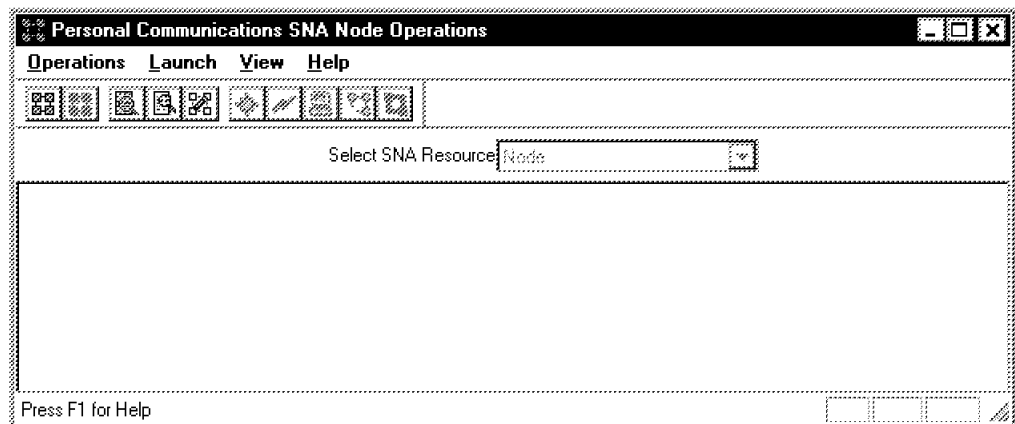


Figure 104. SNA Node Operations

Cancel out of the Software Registration window if it is displayed; you can register later.

- From the menu bar, click on **Launch** and select **SNA Node Configuration** which is shown below:

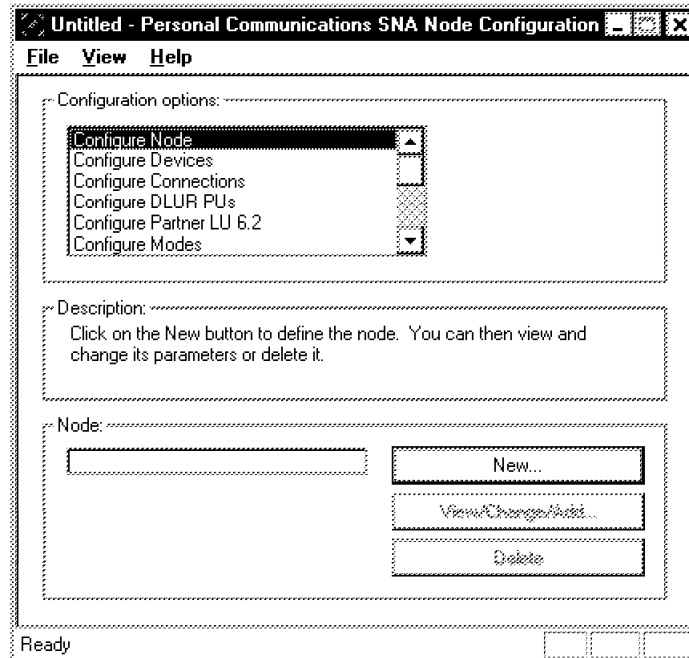


Figure 105. SNA Node Configuration

- Select **Configure Node** in the Configure options box, then click on **New...**
- On the Basic tab of the Node Definition window, fill in the appropriate information. Our configuration is shown in the figure below:

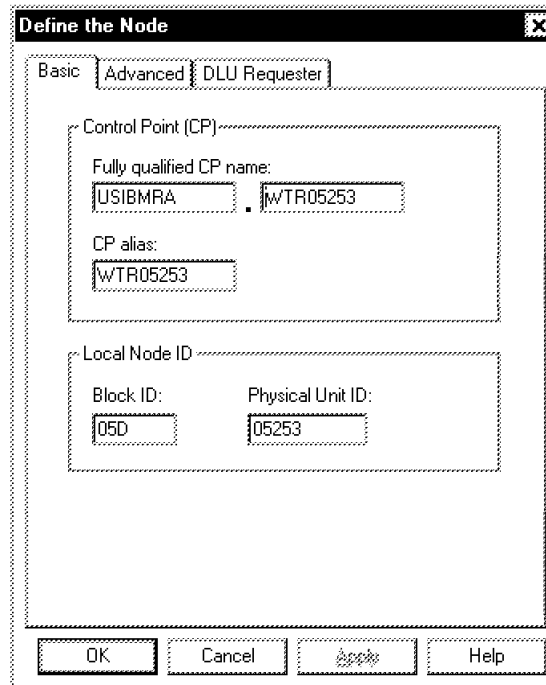


Figure 106. Node Definition

- a. The fully qualified CP name consists of a NETID portion (matches that parameter in VTAM start options as shown in Figure 91 on page 146) and the CPNAME, which is also used as an independent LU 6.2, as shown in Figure 94 on page 150.
  - b. The Block ID and Physical Unit ID are used by VTAM to dynamically define a PU, as explained in 5.5.1.2, “Dynamic Definition of Switched PUs and Dependent LUs” on page 147.
5. On the Advanced tab, check **with network node server** if your VTAM is also configured as APPN network node. In our case it was, and it was also set up for automatic configuration.
  6. We did not use DLUR configuration, so click on **OK**.
  7. Select **Configure Devices** in the Configure options box and also select **LAN** in the DLC box, then click on **New....**
  8. The Define a LAN Device window is displayed which is shown below:

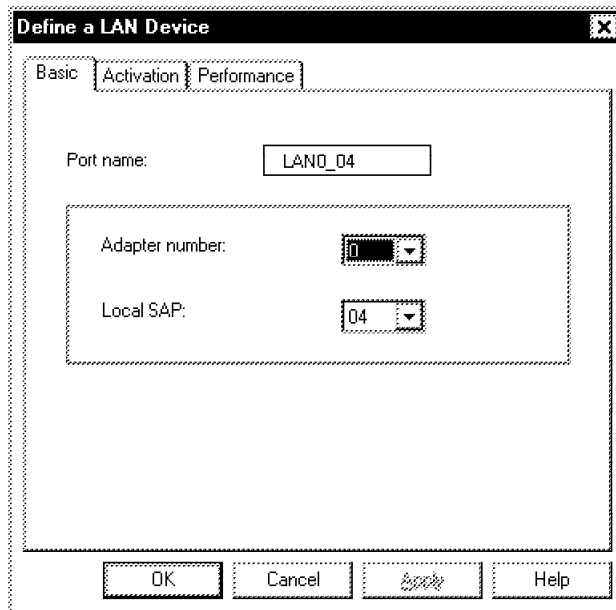


Figure 107. Define a LAN Device

Click on **OK**.

9. Select **Configure Connections** in the Configure options box and also select **LAN** in the DLC box, then click on **New....**
10. On the Basic tab of the Define a LAN Connection window, enter the appropriate information. We used the values shown below:

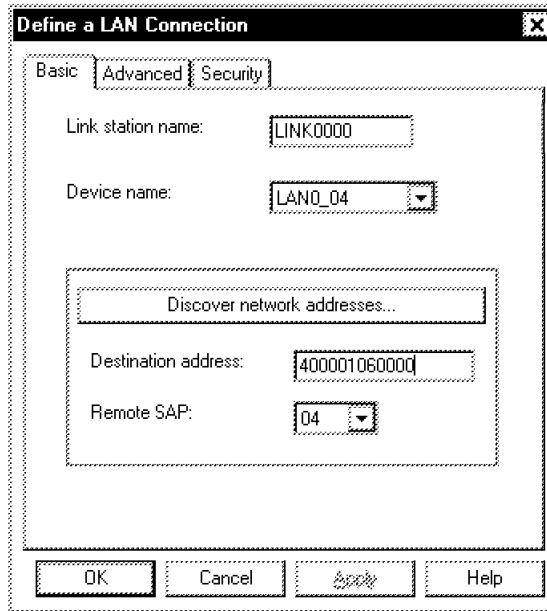


Figure 108. Define a LAN Connection - Basic

11. On the Advanced tab, check **Solicit SSCP sessions** if you also want to run terminal emulation sessions. Our configuration is shown below:

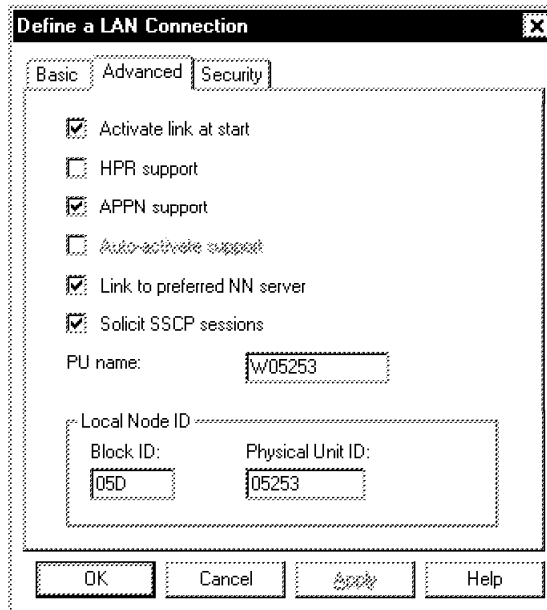


Figure 109. Define a LAN Connection - Advanced

The PU name is used to identify dependent LUs, such as terminals or printers, to VTAM, as shown in Figure 93 on page 149.

12. On the Security tab, enter the Adjacent CP name and type of the partner node, in our case VTAM configured as APPN network node, as shown below:



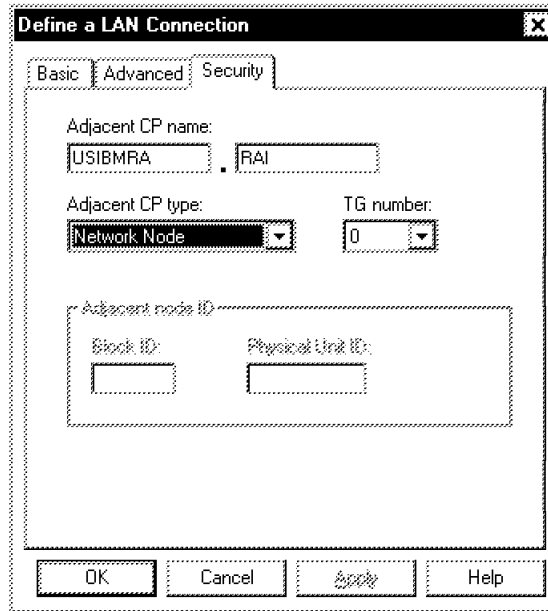


Figure 110. Define a LAN Connection - Security

13. Click on **OK**.
14. Select **Configure Partner LU 6.2** in the Configure options box, then click on **New....**
15. On the Basic tab of the Define a Partner LU 6.2 window, enter the appropriate information about the partner LU and its owning Control Point. We used the values shown below:

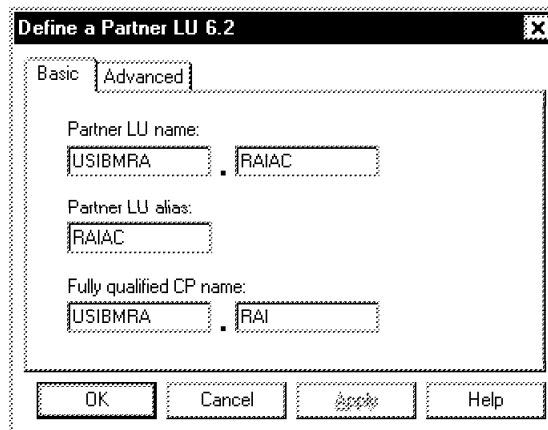


Figure 111. Define a Partner LU 6.2 - Basic

- a. USIBMRA matches the NETID parameter in VTAM start options, as shown in Figure 91 on page 146.
  - b. RAI matches the SSCPNAME parameter in VTAM start options.
  - c. RAIAC is the LU name of CICS on the host, as shown in Figure 98 on page 151.
16. On the Advanced tab, enter the Maximum LL record size as defined by the maximum RU size of the service mode which will be configured in Figure 114 on page 161. Our configuration is shown below:

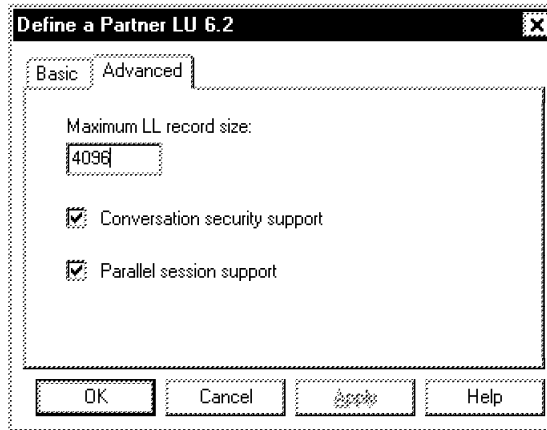


Figure 112. Define a Partner LU 6.2 - Advanced

17. Click on **OK**.
18. Select **Configure Modes** in the Configure options box, then click on **New....**
19. On the Basic tab of the Define a Mode window, enter the name of the service mode to be used for your LU 6.2 communication. This must match the definition in the service mode on the host. Figure 100 on page 152 shows the parameters for the log mode LU62APPC on the host. That logmode entry is in a log mode table called MQMTAB. This table is used by the CICS host LU, as shown in Figure 98 on page 151.

We used the values shown below:

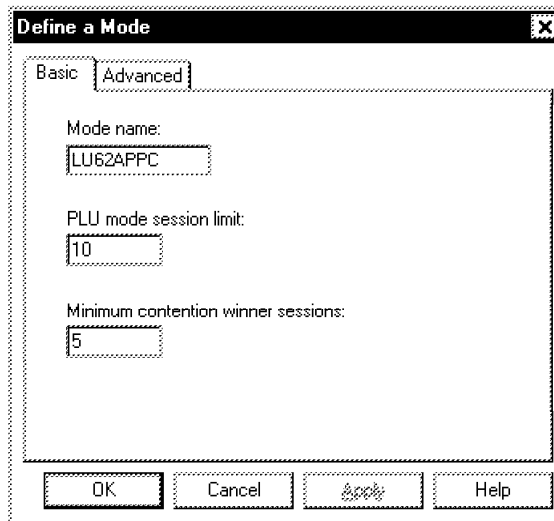


Figure 113. Define a Mode - Basic

20. On the Advanced tab, select the appropriate Class of Service name and Maximum RU size. This must match the definition of RU sizes in the service mode on the host, as shown in Figure 100 on page 152. Our configuration is shown below:

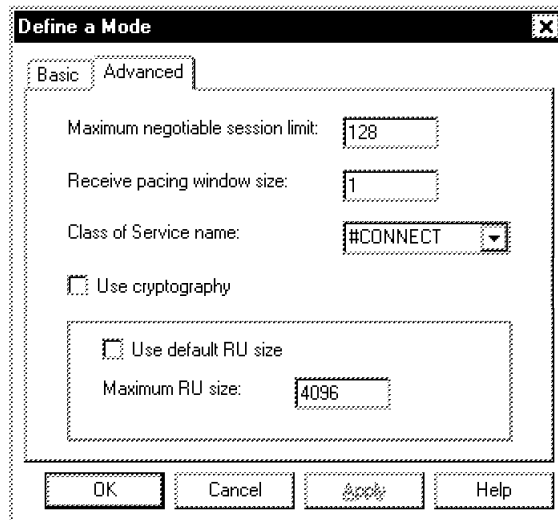


Figure 114. Define a Mode - Advanced

21. Click on **OK**.
22. From the menu bar, click on **File** and select **Save As**. Save the configuration as `hostcics`.
23. Exit SNA Node Configuration.
24. From the menu bar, click on **Operations** and select **Start Node**.
25. Double-click on `hostcics` to use the previously defined configuration.
26. In the Select SNA Resources window, select **Connections** to check if your system has successfully connected to the host. If the State of LINK0000 does not show active, call your system administrator to verify that all of the following is true:
  - All of your configuration parameters are valid and correct.
  - The LAN is operational.
  - Your system is connected to the LAN and can communicate over the LAN.
  - VTAM and NCP are up and also the LAN interface of NCP.
  - All VTAM and NCP definitions are correct and the required resources are active.

## 5.6.2 Configure 3270 Terminal Emulation

Follow the instructions below to create a 3270 emulator definition with Personal Communications:

1. Make sure the SNA Node is started.
2. From the IBM Personal Communications folder, start the **Start or Configure Session** program.
3. Click on **OK** to start configuration. The Customize Communications window is displayed which is shown below:

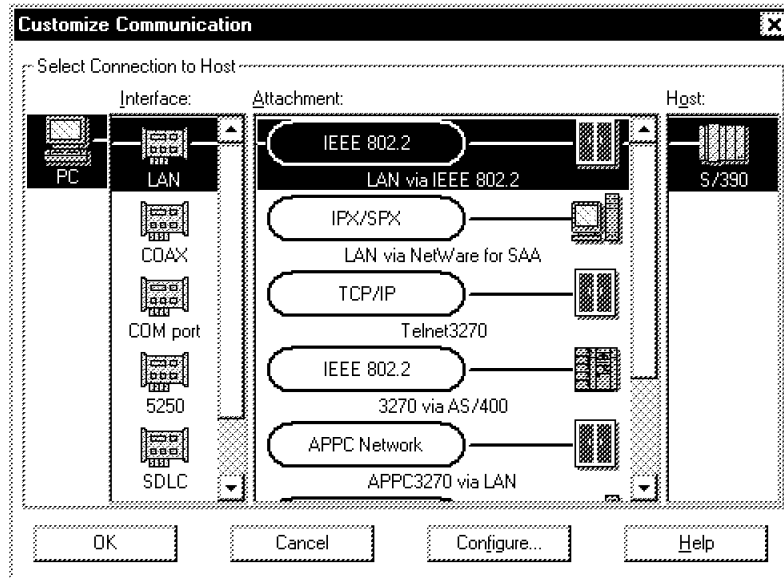


Figure 115. Customize Communication Window

4. Select **LAN via IEEE 802.2**, then click on **Configure...**. Make the appropriate selections for your 3270 terminal emulation session. Our configuration is shown below:

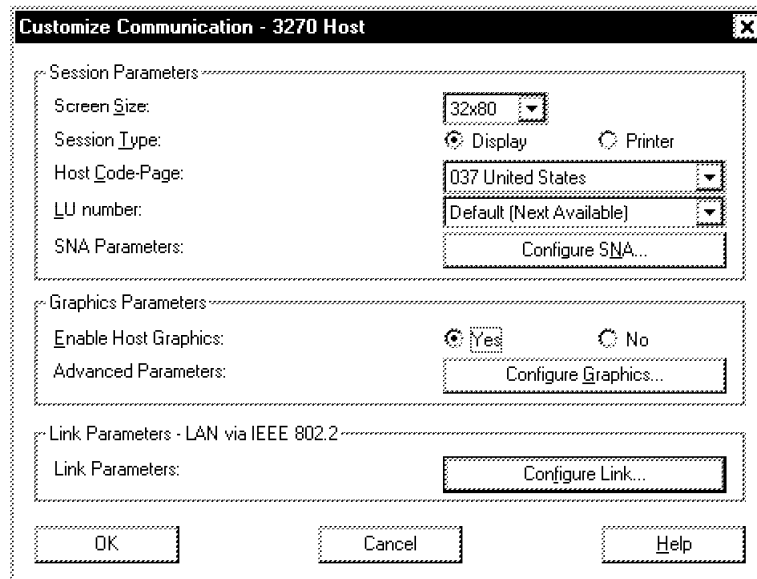


Figure 116. Customize 3270 Terminal Session

5. Click on **Configure Link...**. The Configure Local System window is displayed which is shown in Figure 117 on page 163. You should recognize some of the configuration parameters from the previous SNA node configuration. The Personal Communicationsprivatehostcics.PCG file should be displayed as the active configuration file in the SNA Node Configuration File box.

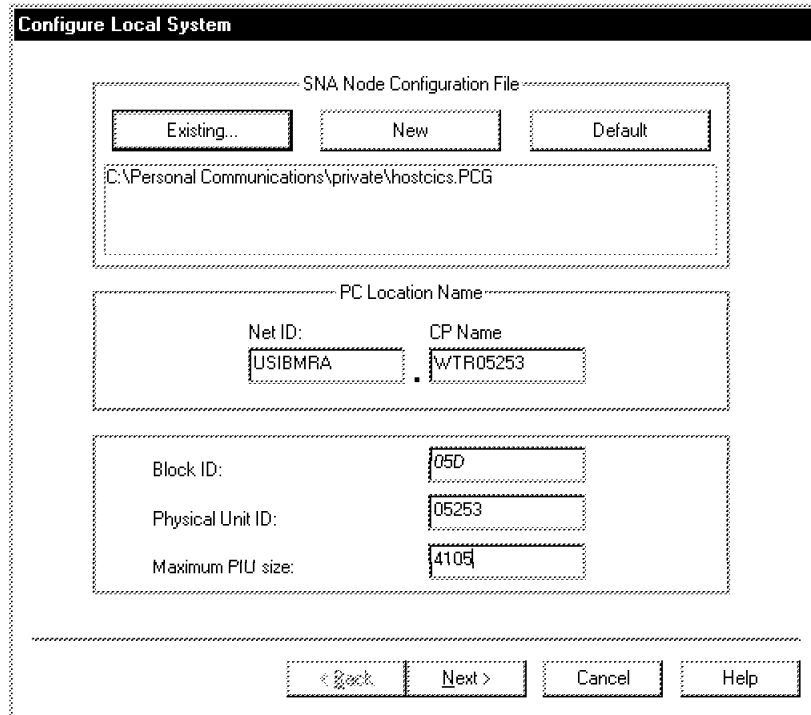


Figure 117. Configure Local System

6. Click on **Existing...**, then double-click on **hostcics** to activate all configuration parameters from the SNA node configuration.

**Note:** If you miss this step, your LAN destination address will be overwritten and the communication will fail to start. (This appears to be a bug in Personal Communications because all parameters should already be loaded.)

7. Click on **Next** twice, then click on **Finish**.
8. Double-click on **hostcics** again to save the configuration.
9. Click on **OK** to confirm the existing configuration file to be overwritten.
10. Click on **OK** in the Customize Communications - 3270 Host window.
11. Click on **OK** in the Customize Communications window. Your 3270 terminal emulation session should start successfully.

---

## 5.7 Installing and Configuring the Transaction Systems

For each step we report our configuration. For detailed information on the different steps we suggest that you read the appropriate manual. To set up this CICS scenario, you need to perform the following tasks:

- Check system requirements
- Install IBM Transaction Server and configure it on NT01
- Configure IBM Transaction Server as transaction router
- Test the Transaction Server configuration
- Install IBM CICS Client on NT03
- Configure IBM CICS Client on NT03

- Configure IBM CICS Internet Gateway on NT03 (only for testing)
- Test the CICS client configuration
- Configure Net.Commerce
- Write the API/DLL
- Write the HTML exception macro

### 5.7.1 Installing and Configuring IBM Transaction Server for Windows NT

For more information about how to install and configure CICS, please refer to the documentation shipped with the product which is in HTML format. You only need to install CICS and configure it with a DCE configuration for RPC only and a File Manager for LOCAL SFS. The following procedure installs and configures a CICS for Windows NT server and client on your systems. All the code for this installation is provided with the CICS part of the Transaction Server.

To install and run a CICS server for Windows NT, perform the following steps:

1. Log in as administrator. For this scenario, use the DB2 instance user ID.
2. Insert the Transaction Server CD-ROM in the CD-ROM drive. If you are installing from a network drive, connect to that drive.
3. Change to the SERVERS<nls> directory, where <nls> is the directory for the language you want to use with the server, in our case US.
4. Select **Next** on the Welcome panel to get to the Select Product to Install panel:

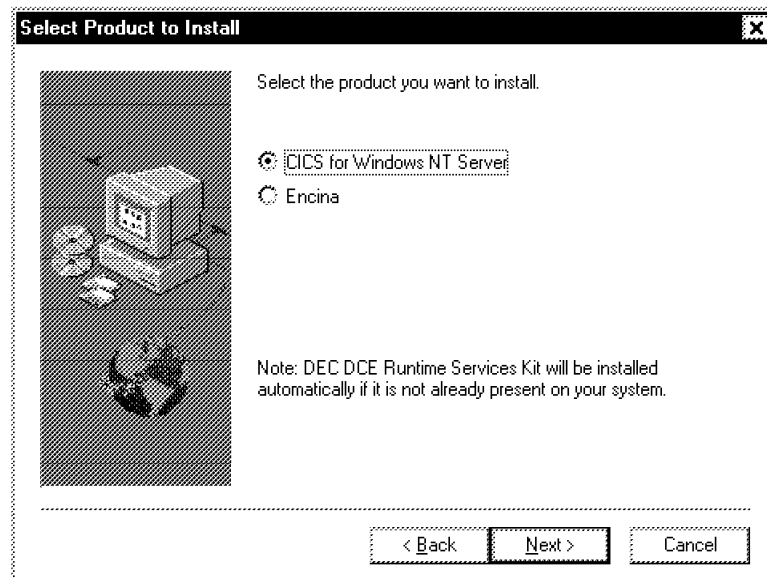


Figure 118. Select Product To Install Panel

5. To install CICS, select the **CICS for Windows NT** radio button, then click on **Next**. The Select Bundle to Install panel is displayed.

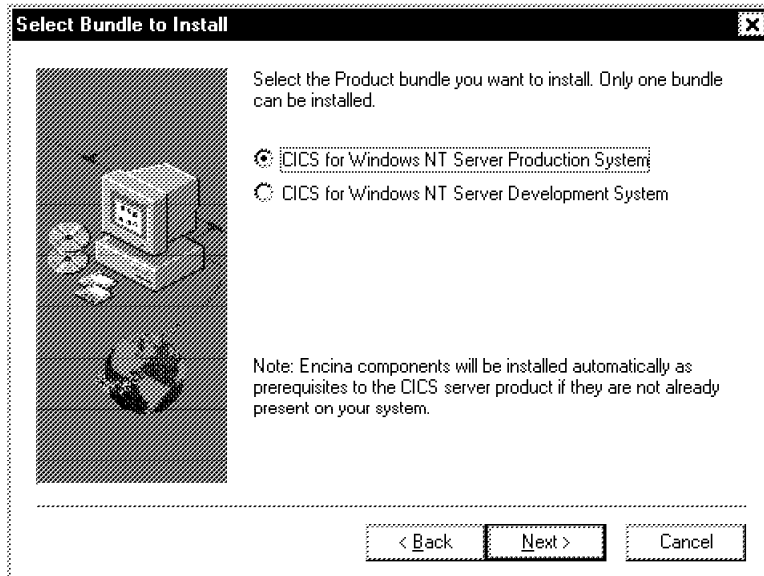


Figure 119. Select Production Environment To Install

6. Choose **Production System**, then click on **Next**.
7. On the following panel, select a **Full** setup and click on **Next**.

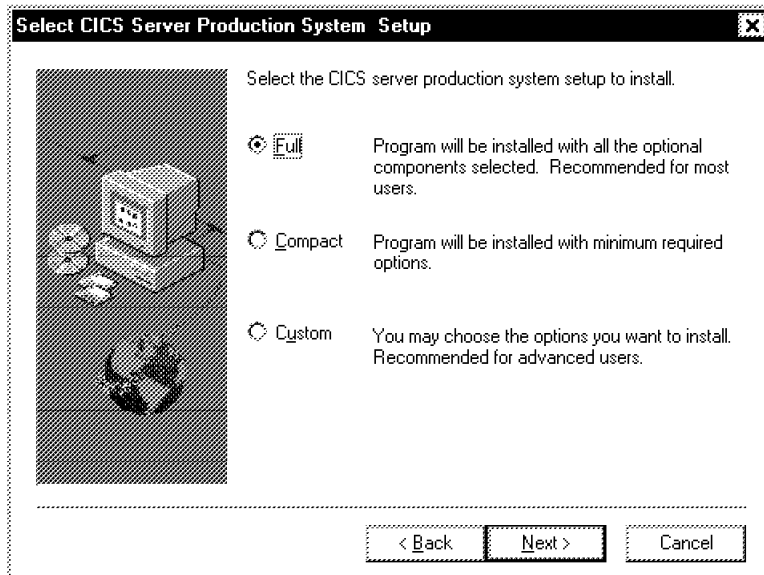


Figure 120. CICS Server Setup Type

8. The Select Server Destination Directory panel is displayed, as shown in the figure below:

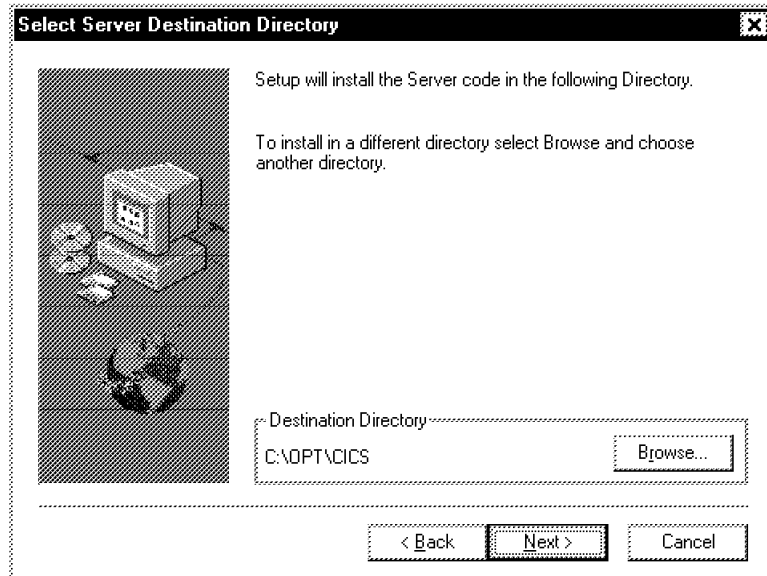


Figure 121. Select Server Destination Directory Panel

9. Check if the destination directory is correct and change it if it is not. Then click on **Next** to proceed.
10. When you're asked whether you want to add a program group, select **Yes**.
11. The Check Setup Information panel is displayed, as shown in the figure below:

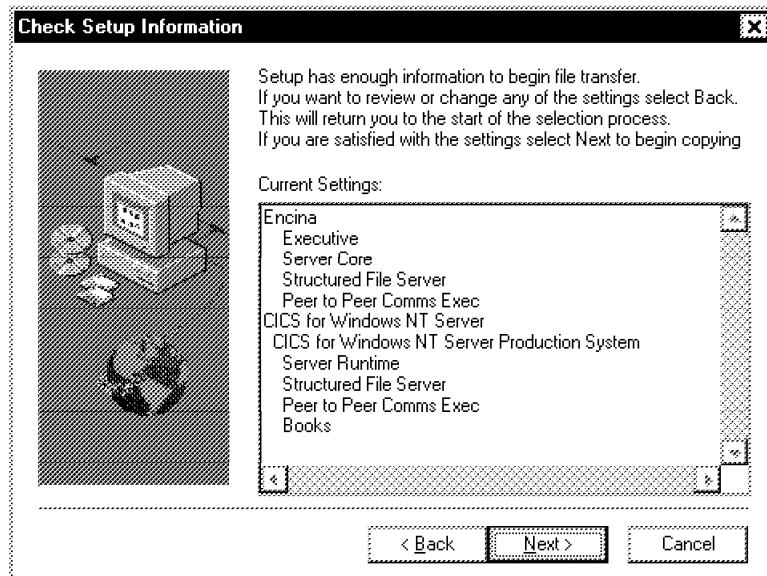


Figure 122. CICS Server Setup Information Panel

- If you are satisfied that the listed components are the ones you want to install, click on **Next**.
12. Setup installs the CICS for Windows NT Server software. To cancel the installation, select **Cancel**. If you decide you do not wish to cancel installation, select **Resume**.



13. A product license enrollment panel is displayed. If you have a production license for CICS and you want to enroll it now, select **Yes**. If you do not wish to enroll now, or you are just evaluating the CICS product, select **No**.
14. When the installation is complete, the Product Installation Complete panel is displayed. Restart your machine now before configuring the CICS system.

## 5.7.2 IBM Transaction Server Basic Configuration

To configure a CICS region and a local CICS client, perform the following steps:

1. Before using the Administration Utility to start your CICS region, we need to create a DCE configuration for the RPC-only environment. To do this, at a command prompt, enter:

```
cicscp -v -l c:tmpcicscp.log create dce -R
```

where c:tmp is a directory on the system.

2. Open the Control Panel and select the **System** icon. Add the following two environment variables to the system.
  - CICS\_HOSTS = <your\_hostname>
  - ENCINA\_BINDING\_FILE = varcics\_serversserverbindings
3. From the CICS Server for Windows NT folder, open the **Administration Utility**, then select the **Subsystem** pull-down menu.
4. Choose the **New** item, followed by CICS Region (CICSNT01) option.
5. Deselect the **Use DCE Server to locate the servers**, as shown in Figure 123:

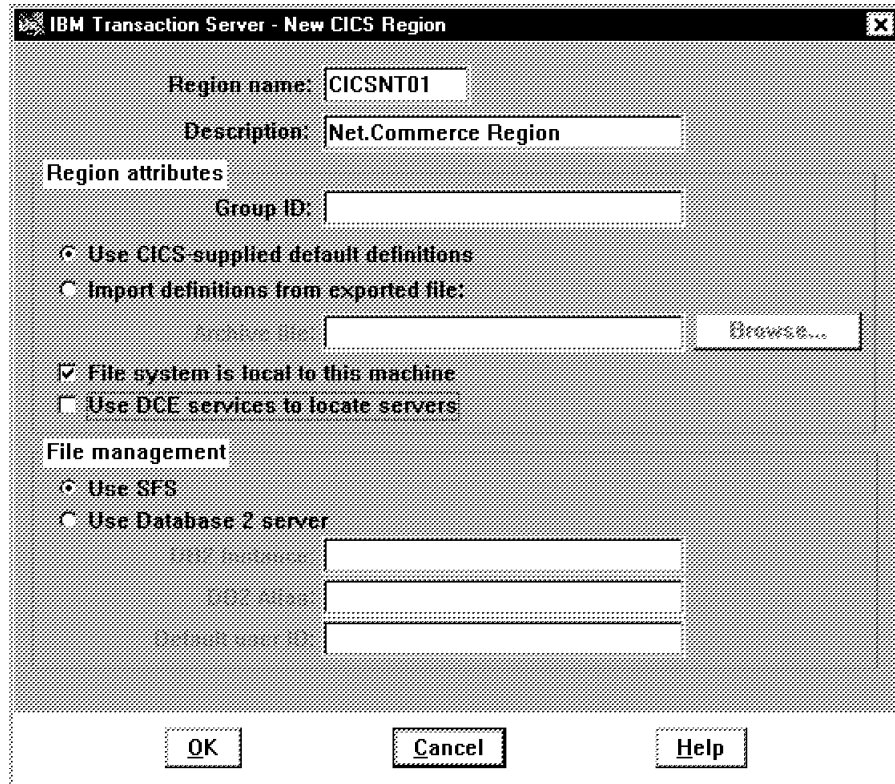


Figure 123. IBM Transaction Server New CICS Region (CICSNT01)

6. Select the **OK** button. This starts the configuration process for both the SFS and the CICS server. It also starts the SFS server. This process can take a few minutes to complete.
7. Before starting your CICS region, configure the local CICS client. To do this, add an entry in the Listener Definitions and modify the CICSLOCALI.INI file.  
From the Administration Utility, select your CICS region and choose the **Subsystem** pull-down menu.
8. Select **Resources** and then select the **Listener** option.
9. Choose the **Listeners** pull-down menu and select **New**.
10. Enter the name of the region (CICSNT01) in the Listener Name field, select **Named Pipe** in the Protocol field from the drop-down menu and enter CICSAA in the Named pipe name field. Because the systems are stopped, you can select the **Permanent** button. The figure below shows our configuration parameters:

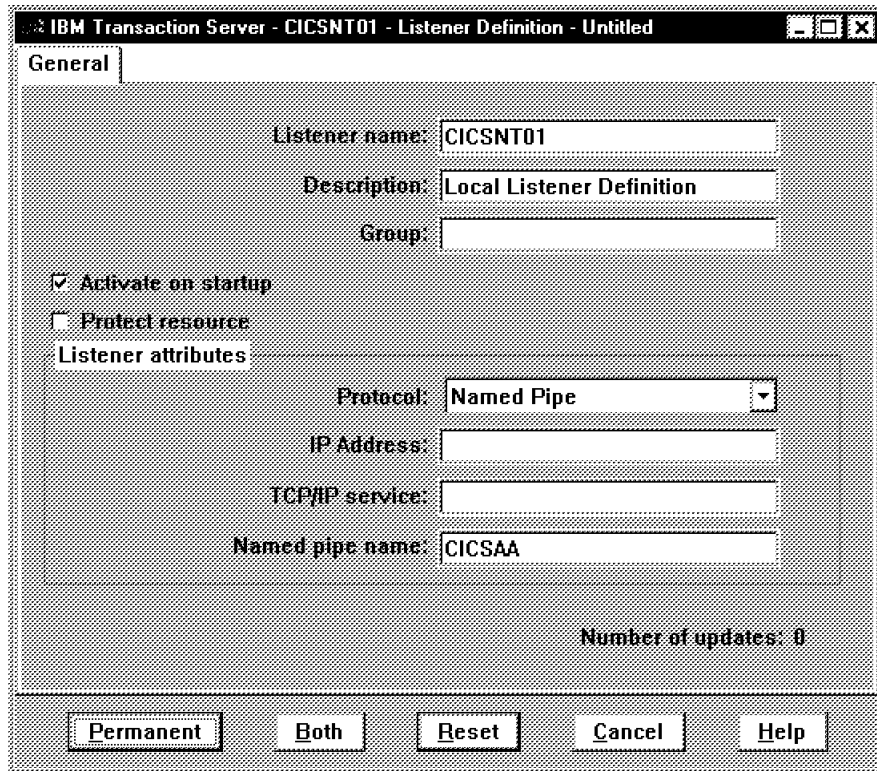


Figure 124. Local Listener Definition CICSNT01

11. Edit the OPTCICSBINCICSLCLI.INI file. These are the changes to the file:

```

Server = CICSNT01           ; Arbitrary, unique name for the server
Description = Named Pipe Support for local client ;
Protocol = LOCALCLI       ; Matches with a Driver section below
NetName = 00.00.00.00     ; Dummy
NamedPipeName = CICSAA    ; Must match servers named pipe
    
```

Figure 125. Configuring Local CICS Client in Definition File (Server)

Comment out any existing but not used definitions in this file.

This step completes the setup for the local client.

12. To verify the configuration, start the CICS region. From the Administration Utility, highlight the CICS region (CICSNT01), choose the **Subsystem** pull-down menu and select the **Start** option. When the figure below appears select the **Cold** option and then click on **OK**.

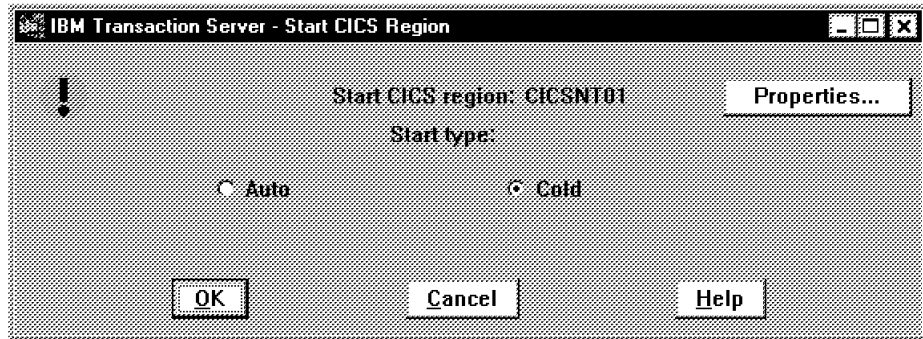


Figure 126. Start CICS NT Region

Once your SFS server and CICS region are up and running, you should see that new status in the Administration Utility, as shown in Figure 127:

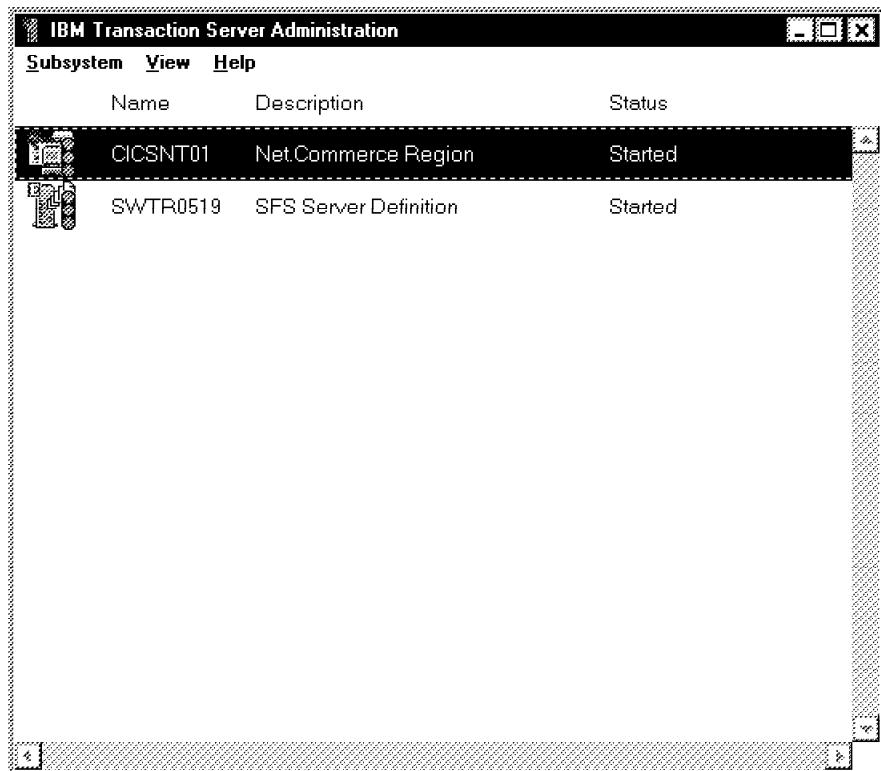


Figure 127. IBM Transaction Server Administration Utility

13. Now start your local client, by selecting **Start Local Terminal** from the CICS for Windows NT folder. If all has gone well you will have a CICS screen where you can enter one of the CICS-supplied transactions, for example CEMT, as shown in the figure below:

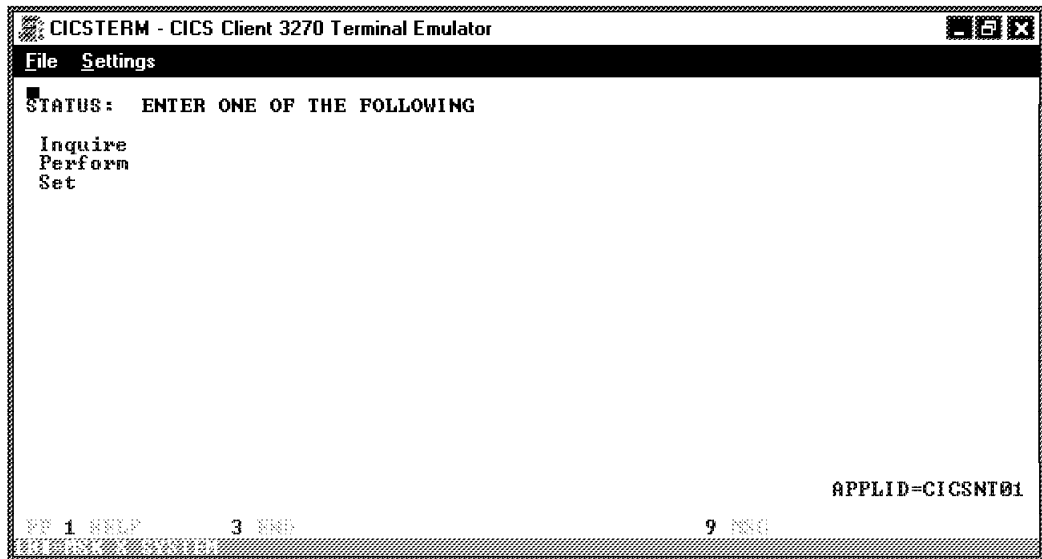


Figure 128. IBM Transaction Server Local Terminal

14. To log off from your terminal, clear the screen (F3) and type exit.
15. To stop the CICS region, select your region from the Administration Utility, and from the Subsystem pull-down menu select either **Stop** or **Stop All**. This forces the region to stop which may take a few minutes. The process is complete when the machine beeps.

### 5.7.3 Configuring IBM Transaction Server As a Transaction Router

To configure your CICS server as a transaction router, follow these steps:

1. Change the region properties by highlighting the CICS region and clicking on the right mouse button. Then select **Properties** from the context menu.

The figure below shows the properties that we added to our region, CICSNT01.

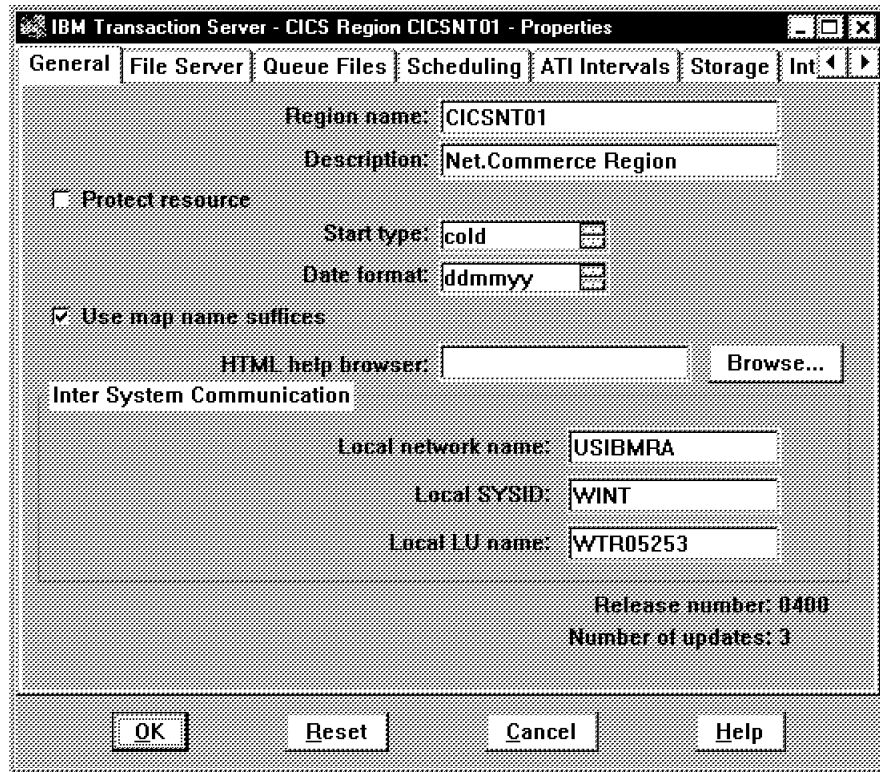


Figure 129. Properties of the Region CICSNT01

On the General panel, we added the following:

- Local network name = USIMBRA
- Local SYSID = WINT
- Local LU name = WTR05253

2. Click on **OK**.

3. Create a listener definition by selecting the **Resources** option from the region's context menu and then selecting **Listener**.

The figure below shows the data that we used to create a listener definition:

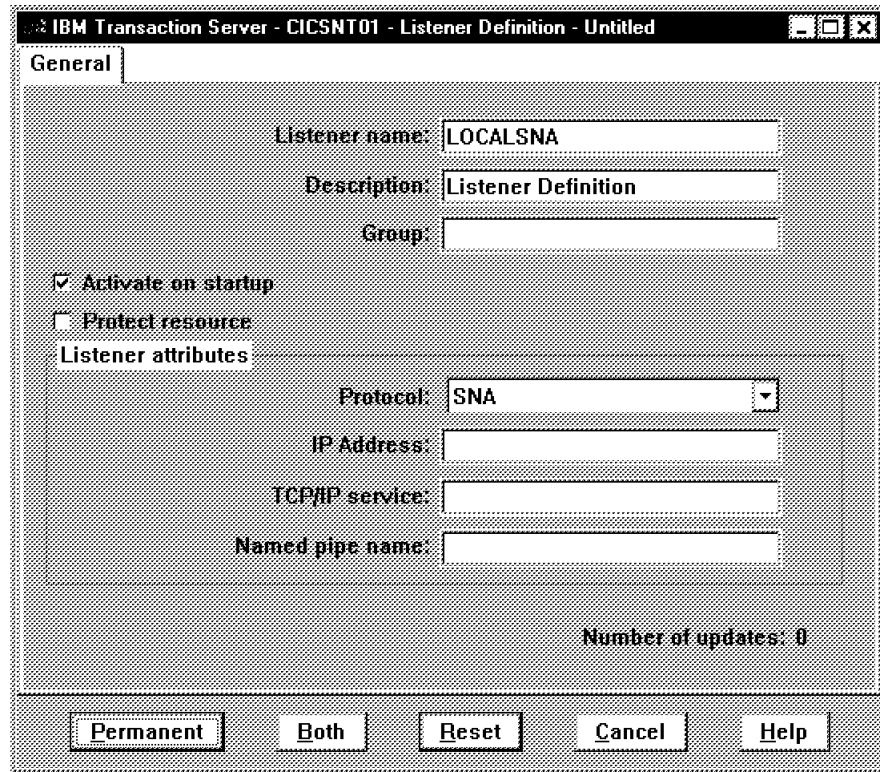


Figure 130. Listener Definition LOCALSNA

We added the following:

- Listener name = LOCALSNA
  - Protocol = SNA
4. Click on **Permanent**.
  5. Create a communication definition by selecting the **Communications** option from the region's context menu.

The following two figures show the data that we used to create a Communication definition:

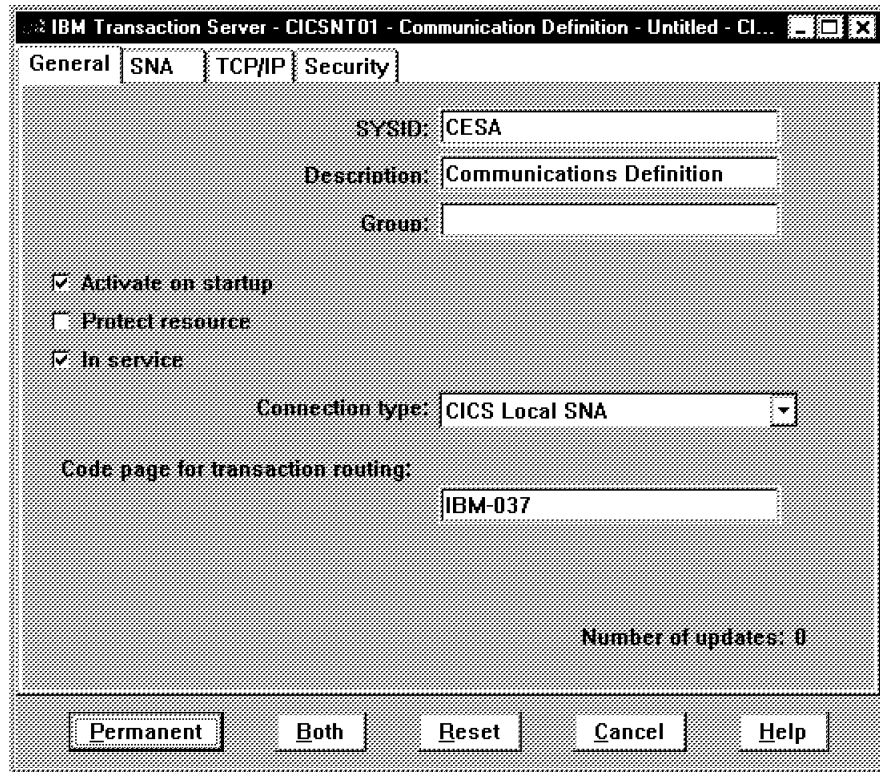


Figure 131. Communication Definition CESA - General

On the General panel, we added the following:

- SYSID = CESA
- Connection type = CICS Local SNA
- Code page for transaction routing = IBM-037

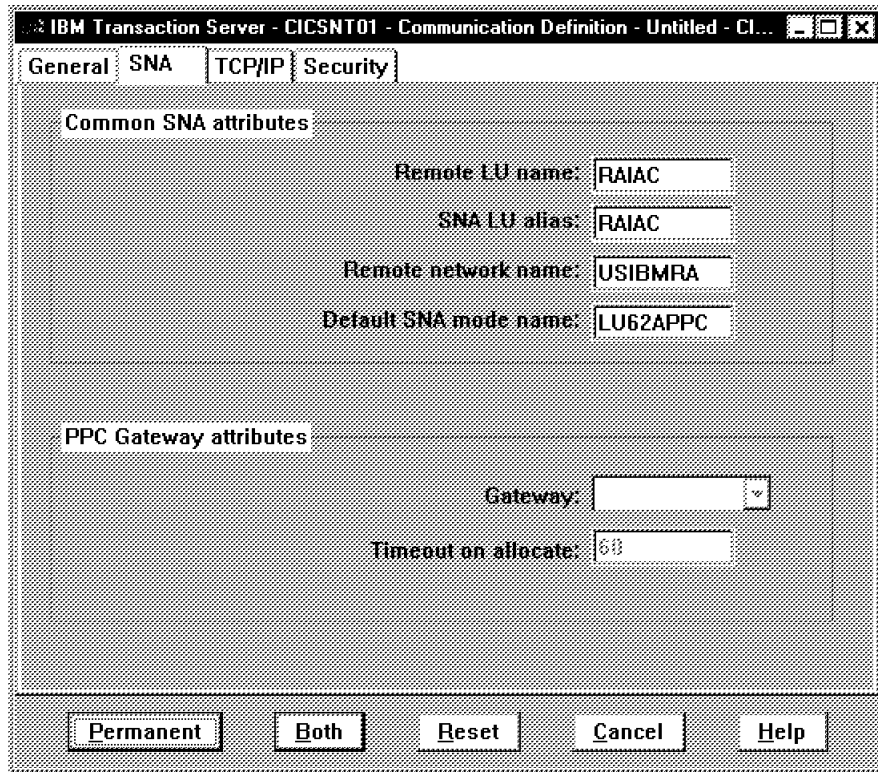


Figure 132. Communication Definition CESA - SNA

On the SNA panel, we added the following:

- Remote LU name = RAIAC
- Remote network name = USIMBRA
- Default SNA mode name = LU62APPC

6. Click on **Permanent**.

7. Create a transaction definition by selecting the **Resources** option from the region's context menu and then selecting **Transaction**.

The following three figures show the data that we used to create a transaction definition.



The screenshot shows a dialog box titled "IBM Transaction Server - CICSNT01 - Transaction Definition - Untitled". It has four tabs: "General", "Transaction", "Context", and "Security". The "General" tab is selected. The fields are as follows:

- Transaction name: EPI1
- Description: Transaction Definition
- Group: (empty)
- Activate on startup
- Protect resource
- Enabled
- Dynamic transaction routing
- Remote transaction section:
  - Remote system ID: CESA (dropdown menu)
  - Remote transaction ID: CEDA
- SNA mode name: LU62APPC
- SNA TPN profile: (empty)
- Number of updates: 0

At the bottom, there are five buttons: "Permanent", "Both", "Reset", "Cancel", and "Help".

Figure 133. Transaction Definition EPI1 - General

On the General panel, we added the following:

- Transaction name = EPI1
- Remote system ID = CESA
- Remote transaction ID = CEDA
- SNA mode name = LU62APPC

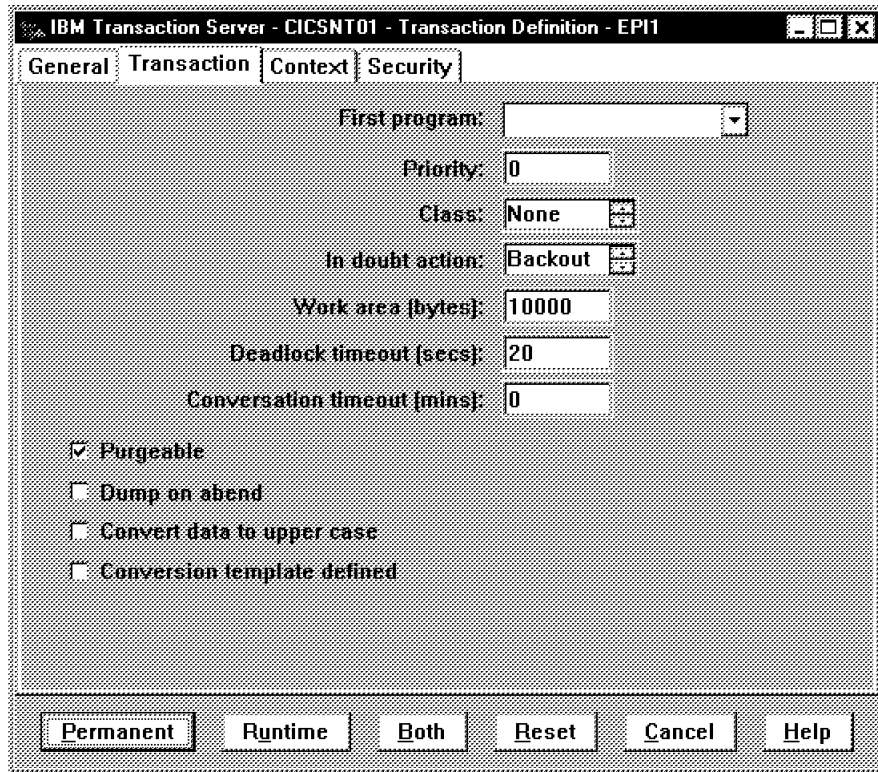


Figure 134. Transaction Definition EPI1 - Transaction

On the Transaction panel, we added the following:

- Work area (bytes) = 10000
- Deadlock timeout (secs) = 20

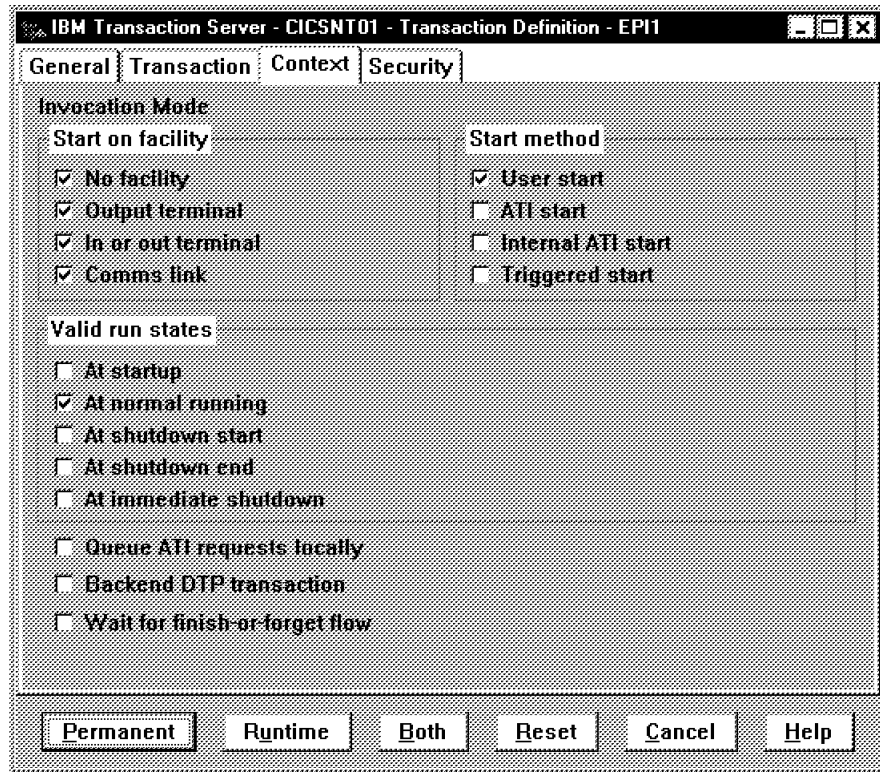


Figure 135. Transaction Definition EPI1 - Context

On the Context panel, we de-selected the following options:

- ATI start
- Internal ATI start
- Triggered start

8. Click on **Permanent**.

#### 5.7.4 Testing the Transaction Server Configuration

To test the configuration, start your local client by selecting **Start Local Terminal** from the CICS for Windows NT folder. If all has gone well you will have a CICS screen where you can enter EPI1 and see what happens. Ideally, you should see a display similar to the one shown in the figure below:

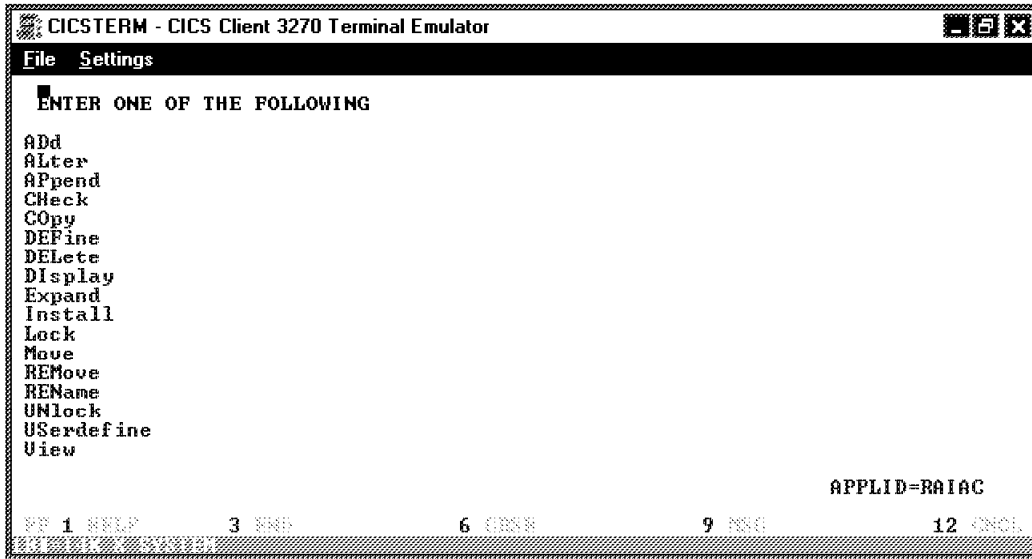


Figure 136. Testing The Transaction Router from a Local Terminal

### 5.7.5 Creating Users

We need to create users for the CESN transaction. Our API will supply user IDs and passwords from the registration macro which will then be checked against existing CICS users.

1. Select the **Resources** option from the region's context menu and then select **User**.

The following two figures show what data we added to create a CICS user.

IBM Transaction Server - CICSNT01 - User Definition - Untitled

General Security/DCE

User name: EPIUSER

Description: User Definition

Group:

Activate on startup

Protect resource

Priority: 0

Operator ID: WNT

User trace file:

Number of updates: 0

Permanent Both Reset Cancel Help

Figure 137. User Definition EPIUSER - General

On the General panel, we added the following:

- User name = EPIUSER

IBM Transaction Server - CICSNT01 - User Definition - Untitled

General Security/DCE

DCE Information

CICS Password: \*\*\*\*\*  None

DCE principal:

Security information

Resource level security key list:

none

Transaction level security key list:

1

Permanent Both Reset Cancel Help

Figure 138. User Definition EPIUSER - Security/DCE

On the Security/DCE panel, we added the following:

- CICS Password = \*\*\*\*\*

**Note:** The password literally consists of seven asterisk characters.

2. Add more users if you wish.

## 5.7.6 Creating a Listener Definition for TCP/IP Client

We need this definition because from the NT03 machine we call CICS on NT01 via TCP/IP.

1. Select the **Resources** option from the region's context menu and then select **Listener**.

The figure below shows what data we needed to create a listener definition.

The screenshot shows a dialog box titled "IBM Transaction Server - CICSNT01 - Listener Definition - Untitled". The "General" tab is selected. The fields are filled as follows: Listener name: CICSCLIENT01, Description: Listener Definition, Group: (empty). Under "Listener attributes", Protocol is set to TCP/IP, IP Address is wtr05253.itso.ral.ibm.com, TCP/IP service is cicsuser01, and Named pipe name is (empty). The "Activate on startup" checkbox is checked, and "Protect resource" is unchecked. The "Number of updates" is 0. At the bottom, there are buttons for "Permanent", "Both", "Reset", "Cancel", and "Help".

Figure 139. Listener Definition for TCP/IP Clients

We added the following:

- Listener name = CICSCLIENT01
- Protocol = TCP/IP
- IP Address = wtr05253.itso.ral.ibm.com
- TCP/IP service = cicsuser01

2. You also need to add the following entry to the winntsystem32driversetscservices file:

```
cicsuser01      64444/tcp      # CICS Client connections
```

### 5.7.7 Installing IBM CICS Client for NT

Follow these steps to install the CICS client code on the Net.Commerce server system:

1. Insert the Transaction Server for NT CD-ROM and change to the clientsI86\_CLNTCLIENTSWNT directory.
2. Type install to start the software installer which is shown in the figure below:

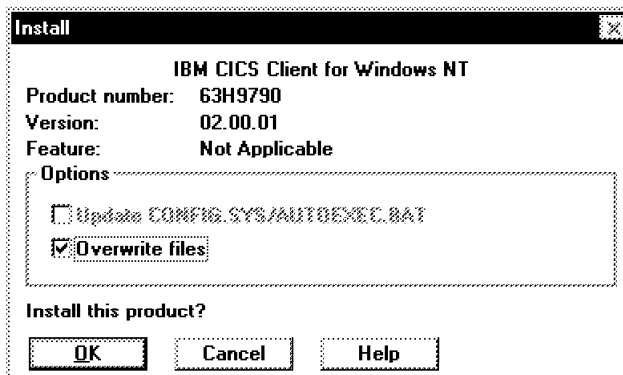


Figure 140. CICS Client Install Panel

3. Select the CICS Client components to install and a destination drive and directory, as shown in the figure below:

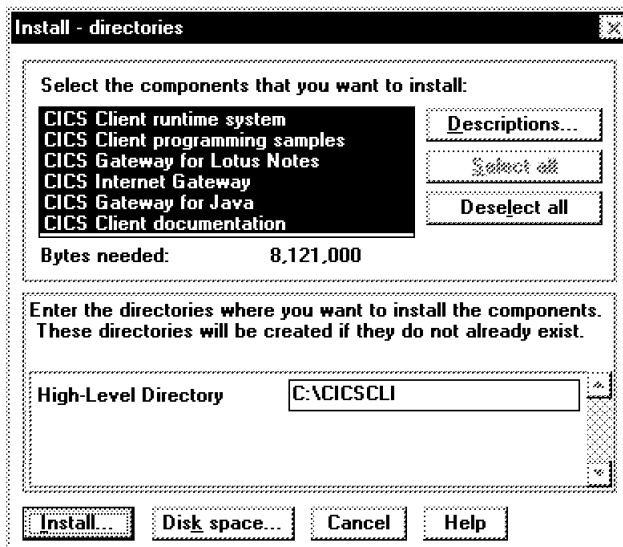


Figure 141. CICS Client Components

4. Select the language, common application, path update, and service registry options, as shown in the figure below:

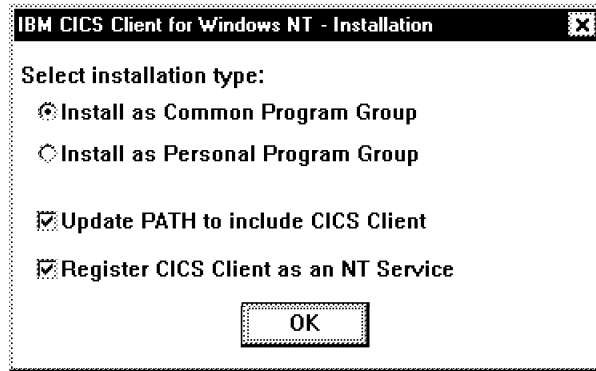


Figure 142. CICS Client Installation Options

### 5.7.8 Configuring the IBM CICS Client

For more detailed information on how to configure an IBM CICS client refer to the *CICS Clients: Administration*, SC33-1792, which is also automatically installed with the product:

For the CICS client configuration, perform the following steps:

1. Make a copy the files CICSCLI.INI and CICSCLIB.INI and CICSKEY.INI in the CICSCLIBIN directory.
2. Add three system variables so that the CICS client can locate the .INI files. These system variables are:
  - CICSCLIB=CICSCLIBINCICSCLIB.INI
  - CICSKEY=CICSCLIBINCICSKEY.INI
  - CICSCLI=CICSCLIBINCICSCLI.INI
3. Modify only the CICSCLI.INI file to tell the CICS client to communicate with your CICS server on NT01. You can do this by copying the sample server configuration parameters:

```
Server = CICSNT01           ; Name for the server
Description = TCP/IP Server ; Arbitrary description for the server
Protocol = TCP/IP          ; Matches with a Driver section below
NetName = wtr05253.itso.ral.ibm.com ; The server's TCP/IP address
Port = 64444               ; change the default TCP/IP CICS port
```

Figure 143. Configuring CICS Server in Definition File (Client)

Comment out any existing but not used parameters.

4. Add the following entry to the winntsystem32driversetc services file:
 

```
cicsuser01      64444/tcp      # CICS Client connections
```

The CICS client is now ready for use.



### 5.7.9 Testing the CICS Client Configuration from a CICS Terminal

You can test the communication to your CICS server by doing the following:

1. Start the client from the IBM CICS Client for Windows NT folder.
2. Start a CICS Terminal from the IBM CICS Client for Windows NT folder.
3. Enter EPI1 to start the routed transaction to the CICS/ESA via the Transaction Server for NT. This should bring up a screen as shown in the figure below:

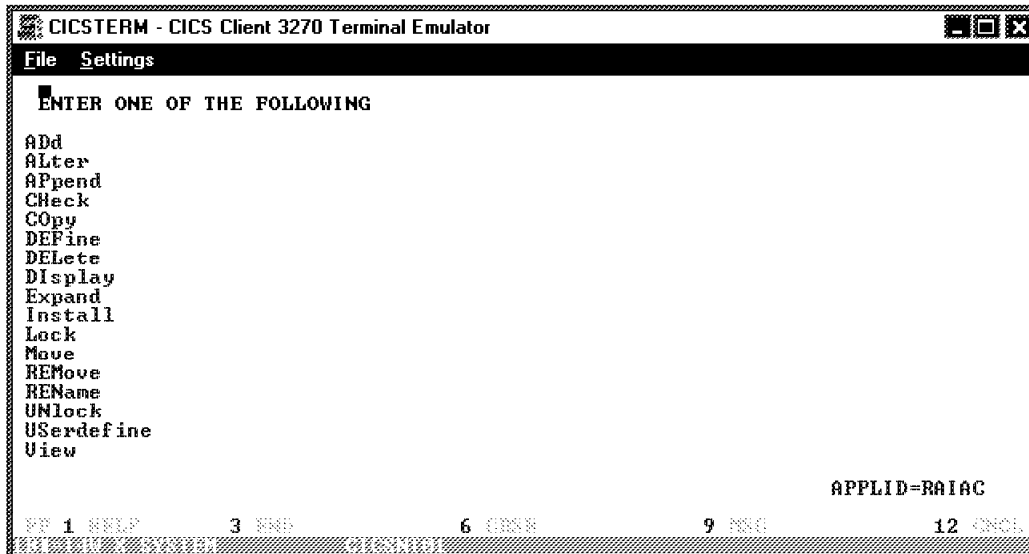


Figure 144. Routed Transaction EPI1 from Client CICS Terminal

4. Do the same with the logon transaction CESN.

### 5.7.10 Configuring IBM CICS Internet Gateway on NT03

For more detailed Information on how to configure IBM CICS Internet Gateway, please refer to the documentation in HTML format that is automatically installed with the product.

**Note:** This step is performed for testing purposes only.

We install the CICS Internet Gateway because we have to call an application via EPI. Not all the applications are callable via EPI. (Refer to *IBM Transaction Server for Windows NT - Application Programming Reference*, SC33-1887 for details.) If we are able to display our 3270 CICS application in a Web browser by means of the the CICS Internet Gateway we can be sure that we can call our CICS application via EPI.

For the CICS Internet Gateway configuration, perform the following steps:

1. Add the following sections to the ICSS configuration file, winnhttpd.cnf:

```
#### SESSION ADDED FOR CICS INETERNET GATEWAY
Protect /cig-admin/* PROT-ADMIN
#### END SESSION ADDED FOR CICS INTETRNET GATEWAY

####SESSION ADDED FOR CICS INTERNET GATEWAY
Exec      \cig-bin\*      D:\cicscli\cig\cgi\*
Exec      \cig-admin\*  D:\cicscli\cig\admin\*
####END CICS INTERNET GATEWAY SESSION

####SESSION ADDED FOR CICS INTERNET GATEWAY
Pass      /cig/*        D:\cicscli\cig\html\*
####END CICS INTERNET GATEWAY SESSION
```

Figure 145. HTTPD.CNF Entries for CICS Internet Gateway

2. Restart your system before you start the CICS Internet Gateway for the first time.

### 5.7.11 Testing the CICS Client Configuration from a Web Browser

You can test the communication to your CICS server as well as the EPI interface by doing the following:

1. Start the CICS Client from NT Services.
2. Start the CICS Internet Gateway from NT Services.
3. Start the Internet Connection Secure Server from NT Services.
4. Point your browser to the following URL:  
[http://<your\\_hostname>/cig/cigstart.htm](http://<your_hostname>/cig/cigstart.htm)

The following page will be displayed:



Figure 146. Configuration Page of the CICS Internet Gateway

5. Click on **Different ways to start a transaction** and scroll down to point 4.
6. Type CICSNT01 in the Specify CICS System field, and type EPI1 in the Transaction And Data field, then click on the **Click here to perform the transaction** button. A page similar to the one shown below appears:

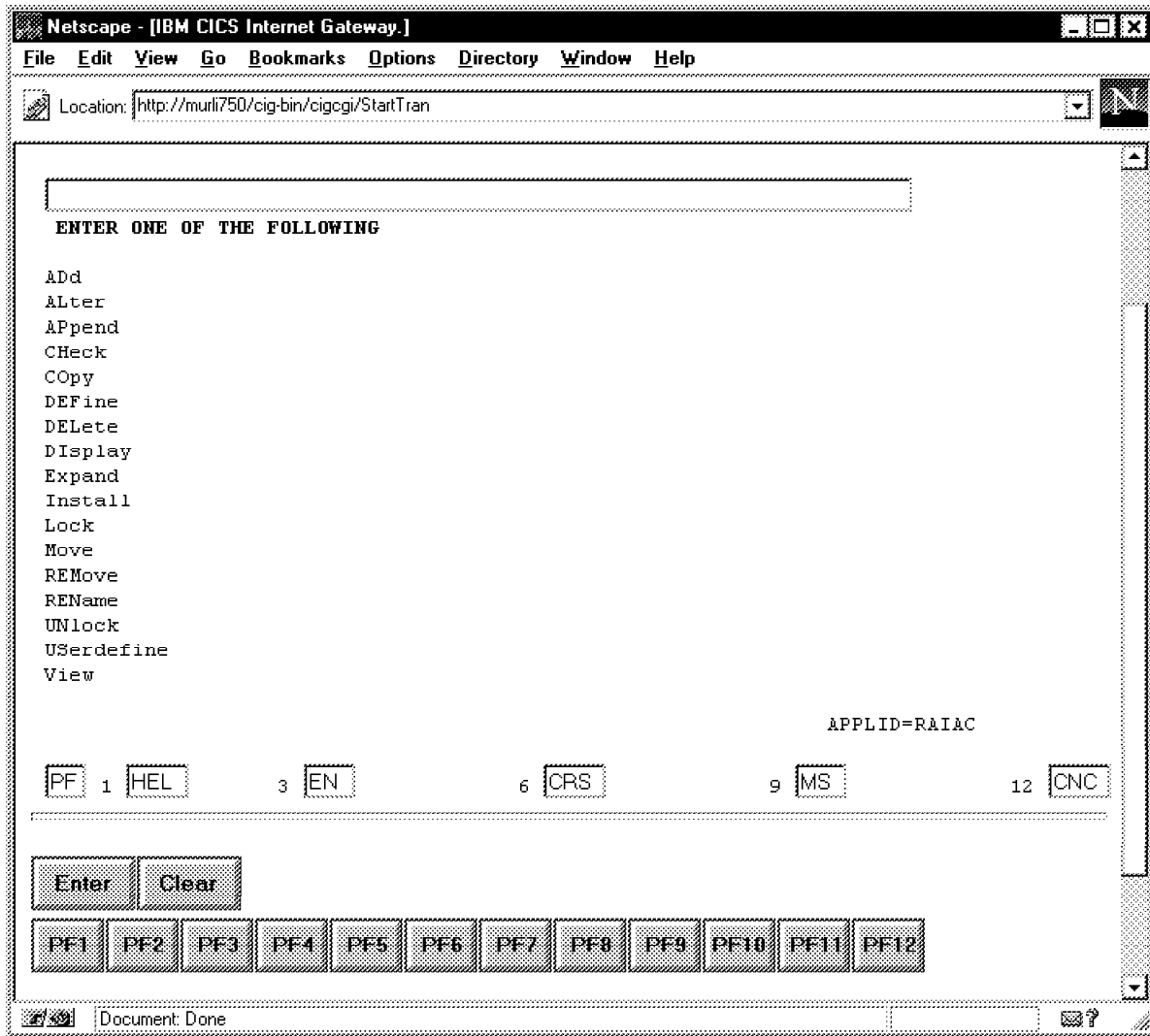


Figure 147. Routed Transaction EPI1 From Web Browser

7. Try the same with the logon transaction CESN.

## 5.8 Configuring Net.Commerce to Interface with CICS

For more detailed information about how to customize Net.Commerce System refer to *IBM Net.Commerce: Customization Guide and Reference*, SC09-2370.

### 5.8.1 Using Your Own APIs with the System

To customize Net.Commerce to use your own API/DLL, do the following:

1. Decide which API task to customize, for example AUDIT\_REG.
2. Understand which commands call that task, in our case ;register/modify and ;register/new. register/new command& register/new command register/modify command& register/modify command
3. Understand the default behavior of the commands:
  - a. ;register/modify?url=url&shlpwd=pass&shlpwder=ver\_pass
    - Modifies the shopper's registration record as follows:

- 1) Checks the required parameters.
- 2) Calls the AUDIT\_REG API task to check the remaining parameters. The default API function ensures that the values of the following parameters from table SHADDR are valid: SALNAME, SADDR1, SACITY, SASTATE, SACNTRY, SAZIPC.
- 3) Modifies the registration record.

Upon successful completion, the command calls the URL specified.

- Exception conditions:
  - If a required parameter is missing or incorrect, it calls the BAD\_REG\_MODIFY exception task.
  - If the AUDIT\_REG API task returns a non-zero return code, it calls the AUDIT\_REG API exception task.
- Parameter values:
  - url: The URL that is called when the command successfully completes
  - pass: The shopper's password
  - ver\_pass: The verification password

b. ;register/new?url=url&shlogid=log\_id&shlpwsd=pass&shlpswder=ver\_pass

- Creates a registration information for the shopper as follows:
  - 1) Checks the required information parameters.
  - 2) Calls the AUDIT\_REG API task to check the remaining parameters. The default API function ensures that the values of the following parameters from table SHADDR are valid: SALNAME, SADDR1, SACITY, SASTATE, SACNTRY, SAZIPC .
  - 3) Updates the registration record.

Upon successful completion, the command calls the URL specified.

- Exception conditions:
  - If a required parameter is missing or incorrect, it calls the BAD\_REG\_MODIFY exception task.
  - If the AUDIT\_REG API task returns a non-zero return code, it calls the AUDIT\_REG API exception task.
- Parameter values:
  - url: The URL that is called when the command successfully completes
  - log\_id: The logon user ID that the shopper chooses
  - pass: The shoppers password
  - ver\_pass: The verification password

#### 4. Understand the AUDIT\_REG API:

##### a. API signature:

```
int Calling , struct APIErrorStruct * Error, const NVPairMapStruct* NVP
```

- ##### b. Default behavior:
- The default function returns a return code of 0 if the following fields from table SHADDR are not blank, and a return code of 1

if one or more are blank: SALNAME, SAADDR1, SACITY, SASSTATE, SACNTRY, SAZIPC.

c. Error codes and extra parameters:

- 190: A required parameter was not provided. The default macro displays a message that indicates that a required fields was not provided.
  - Extra Parameters: field = field, is one of the fields that the API function checks. This is the field that was not provided.
5. Write your own DLL. Our DLL is called legacyCics.dll in the directory D:CICSSOURCE.
  6. Decide which function of the DLL is to be called by Net.Commerce. Our function is GotoCics.
  7. Decide which exception macro Net.Commerce should use if the function returns a non-zero return code. Our exception macro is mall\_err\_legacy\_cics.d2w and is in the directory C:IBMNETCOMMERCEMACROEN\_USDEMOMALL.
  8. Use the Task Management as explained in *IBM Net.Commerce Basics: Open for Business*, SC09-2372, to configure Net.Commerce to use the new DLL and the new exception macro.

## 5.8.2 Using Your own Macros with the System

To customize Net.Commerce to use your own macros, do the following:

1. Decide which macro task to customize, in our case REG\_UPD and REG\_NEW.
2. Understand which commands call the task ;register/form. register/form command& register/form command
3. Understand the default behavior of the command:
  - The ;register/form command, for shoppers who log on as registered shoppers, calls the REG\_UPD task to display their registration page. For shoppers who are not logged on as registered shoppers, it calls the REG\_NEW task to display a blank registration page.
  - Exception Conditions: None
  - Parameter Values: None
4. The macros that we override are msregform.d2w (for REG\_NEW task) and msregupdform.d2w (for REG\_UPD task) both of which are in the C:IBMNETCOMMERCEMACROEN\_USDEMOMALL directory. Because we changed only the field CICS USER in the page and because the macros have mall scope we do not change the name of the macros and therefore do not use the Task Management utility.

These two macros also include four .INC files in the directory C:IBMNETCOMMERCEMACROEN\_USNCSAMPLE:

- msregadd.inc
- msregform.inc
- msreglabels.inc
- msvarmap.inc

### 5.8.3 Running the Application without Net.Commerce

These are the instructions on how to run the application without Net.Commerce for testing purposes. On the NT03 machine, perform the following steps:

1. From a command prompt type d: and then cd cicssource.
2. Type copy legcics.c.test legcics.c.
3. Compile the DLL, legacyCics.dll, by typing nmake makecics.mak.

**Note:** To compile the DLL and the following .exe, Microsoft Visual C++ 4.2 is required.

4. Compile the testcics.exe by typing nmake testcics.mak.
5. Modify the Path environment variable by adding d:cicssource.
6. Start the application by typing testcics.
7. You should see on the screen when you are successful:

```
return code is 0
```

If you aren't successful, you should see:

```
return code is 1 (or whatever)
```

### 5.8.4 Running the Application With Net.Commerce

These are the instructions on how to run the application with the Net.Commerce system. On the NT03 machine, perform the following steps:

1. From a command prompt type d: and then cd cicssource
2. Type copy legcics.c.good legcics.c .
3. Compile the legacyCics.dll by typing nmake makecics.mak.

**Note:** To compile the DLL, Microsoft Visual C++ 4.2 is required.

4. Modify the Path environment variable by adding d:cicssource.
5. Start Net.Commerce server, DB2 and ICSS.
6. Start the CICS client.

**Note:** It is assumed that the Transaction Server and CICS/ESA are up and running.

7. Copy the files \*.inc by typing copy \*.inc  
c:\ibm\netcommerce\macro\en\_us\ncsample.
8. Copy the exception macro file by typing copy mall\_err\_reg\_legacycics.d2w  
c:\ibm\netcommerce\macro\en\_us\demomall.
9. Copy the Registration form macro file by typing copy mallregform.d2w  
c:\ibm\netcommerce\macro\en\_us\demomall.
10. Copy the Udate Registration form file by typing copy mallregupd.d2w  
c:\ibm\netcommerce\macro\en\_us\demomall.
11. Create a directory c:ibmnetcommercehtmlen\_usdemomallregistration  
(if not already created).
12. Create a directory  
c:ibmnetcommercehtmlen\_usdemomallregistrationCICS (if not already

created). In this directory the application will put some HTML error files called CicsUser.htm where CicsUser is the value of the field CICS USER in the registration form.

13. Use the Task Management tool of Net.Commerce by filling in the following information:

- DLL Name: d:cicssourcelegacyCics.dll
- API Name: GotoCics
- Exception Macro: demomallmall\_err\_reg\_legacyCics.d2w

14. From a Web browser, begin the registration process:

- Type `http://<your_hostname>/demomall/basemall.htm` to get to the demo mall home page.
- Scroll down and click on **Register**.
- Fill in the fields as shown in the figure below and click on **Submit Registration**.

The screenshot shows a web browser window with the URL `http://wtd05252.itso.ibm.com/cgi-bin/nph-msvr/register/form`. The page is titled "Personal Information" and "Contact Information". The form fields are as follows:

Field	Value
Shopper's Login ID	NANO
Password	XXXXXXXXXX
Verify Password	XXXXXXXXXX
CICS User	EPIUSER
Title	Mr
Last Name	NANO
First Name	
Middle Name	
Company Name	
Contact Information	
Address (line 1)	ADDR
Address (line 2)	
Address (line 3)	
City	CITY
State/Province	STATE
Zip/Postal Code	6789
Country	COUNTRY

Figure 148. Registration Form for CICS Scenario

- The figure below shows that the registration was successful.



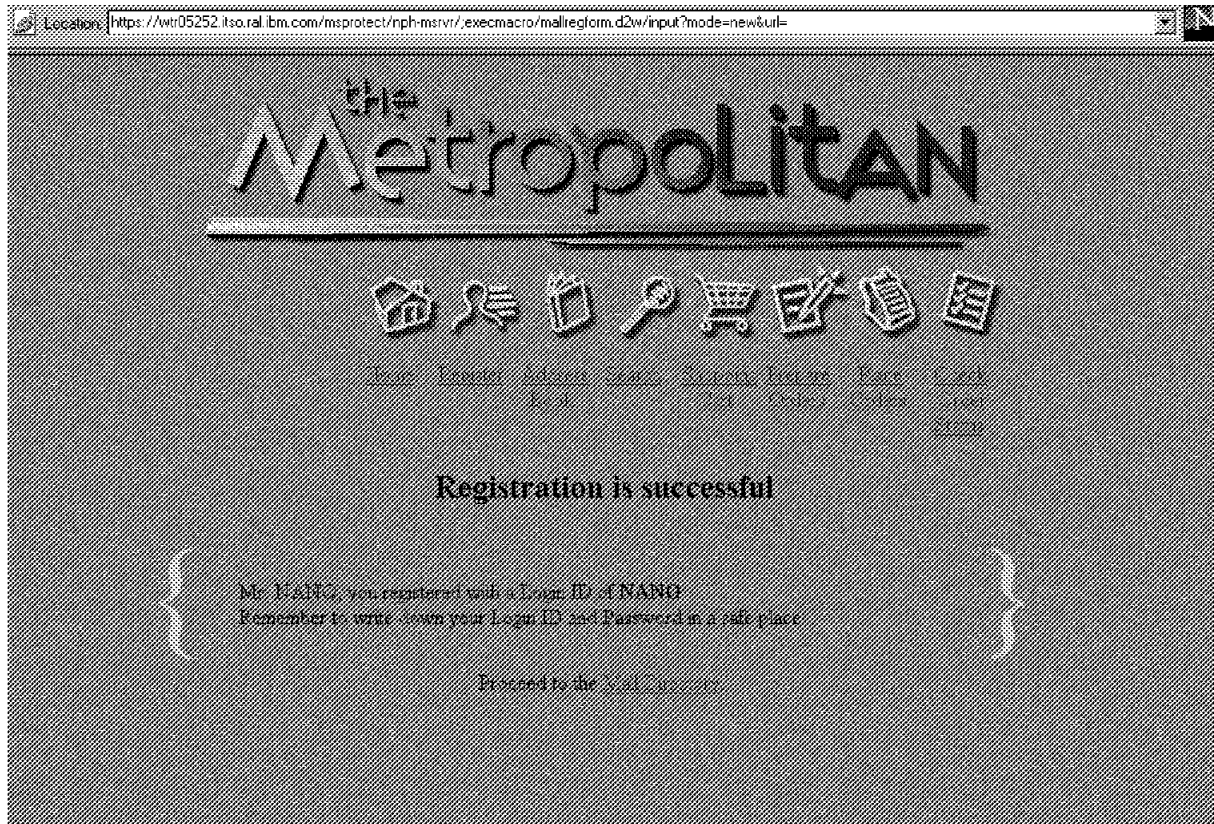


Figure 149. Registration Successful with CICS

### 5.8.5 What Happens in Case of a CICS Error?

This section shows simple error management in case of CICS errors. The CICS error is that the user has no account on the Transaction Server:

1. Fill in the fields as shown in the figure below and click on **Submit Registration**.

The screenshot shows a web browser window with the URL <https://w105252.itso.ral.ibm.com/cgi-bin/nph-misrvi/register/form>. The page contains two main sections: 'Personal Information' and 'Contact Information'. The 'Personal Information' section includes fields for Shopper's Login ID (NAN01), Password, Verify Password, CICS User (CICSNANO), Title (Mr), Last Name (NAN01), First Name, Middle Name, and Company Name. The 'Contact Information' section includes fields for Address (line 1) (ADDR1), Address (line 2), Address (line 3), City (CITY1), State/Province (STATE1), Zip/Postal Code (47758), and Country (COUNTRY1).

Figure 150. Registration Form for CICS Error Scenario

2. The figure below shows that there has been an error. That means the API returned a value other than 0 so that Net.Commerce has called the exception macro.

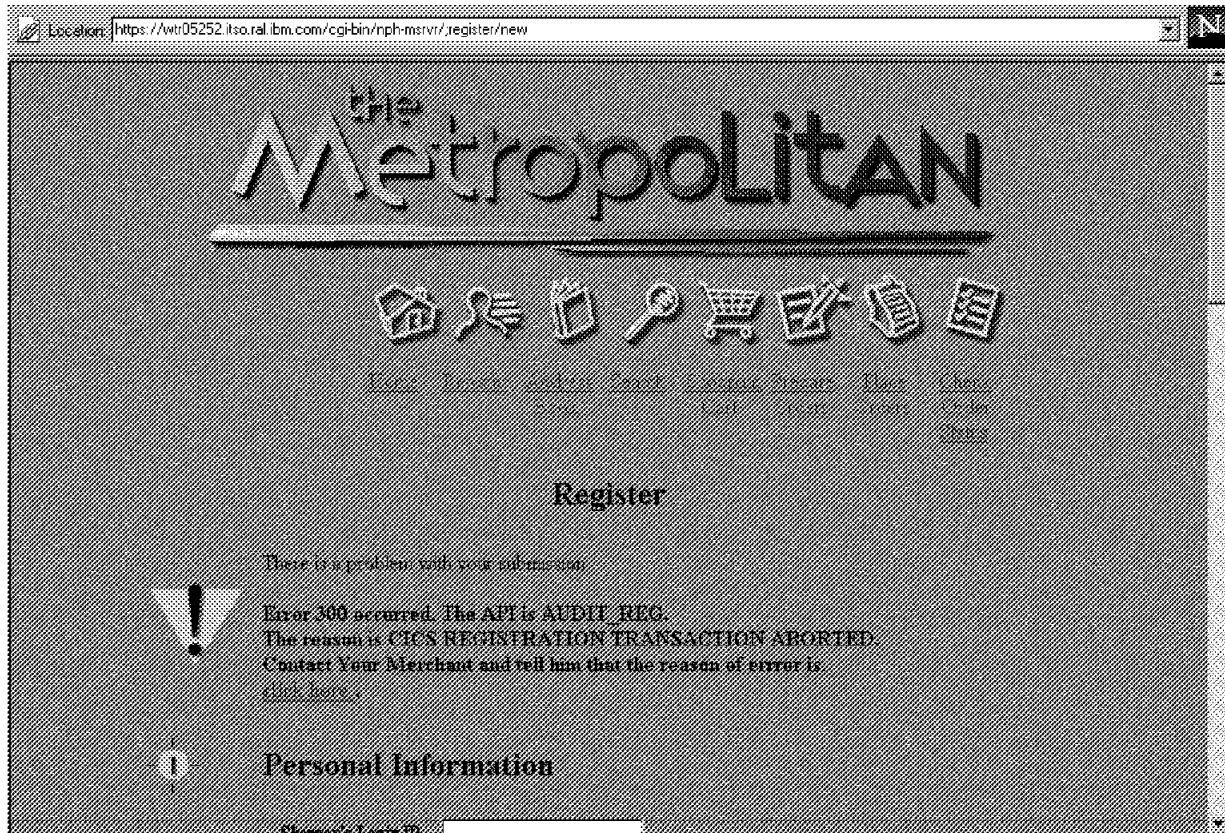


Figure 151. Exception Macro for CICS

3. If you press **click here** you will see the reason of the CICS error.

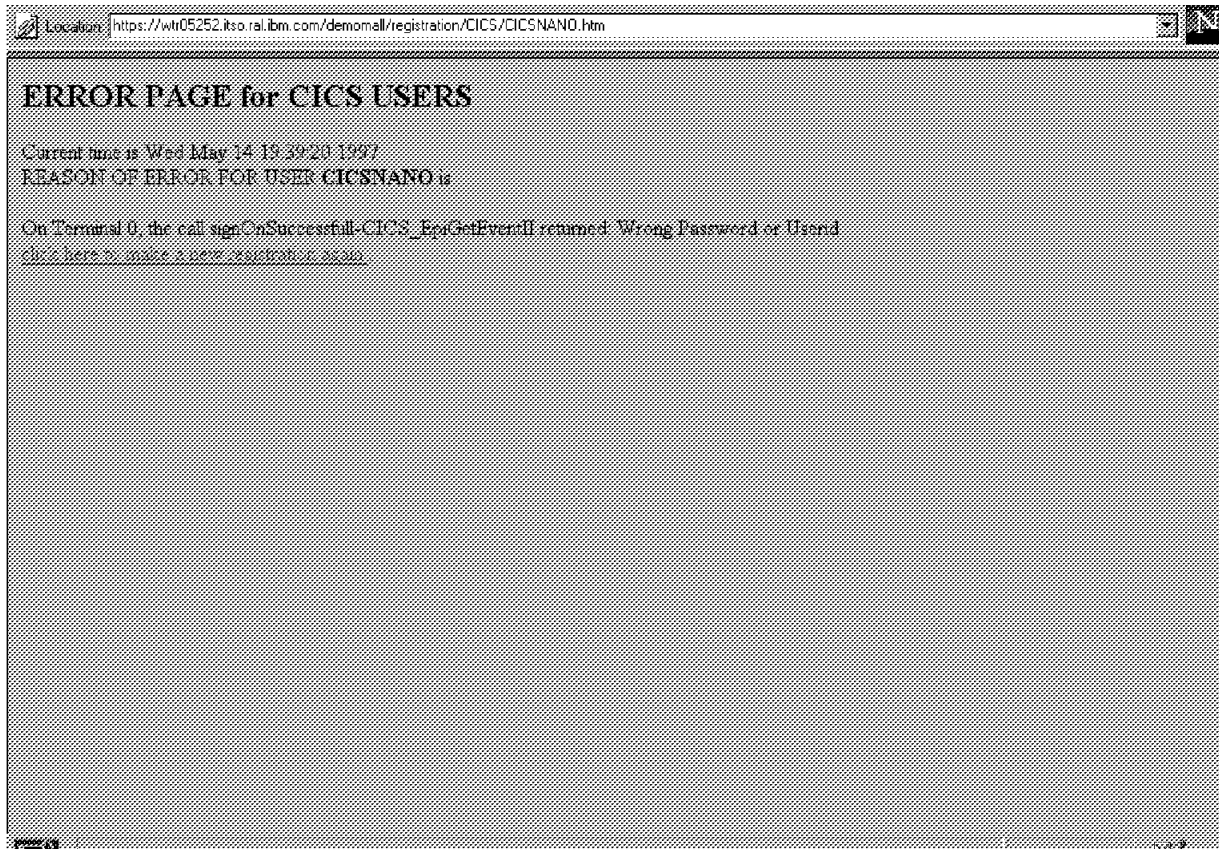


Figure 152. CICS User Error Page

4. If you select **click here to make new registration again** you will get back to the registration form.

---

## 5.9 The CICS API Program and Customized Net.Commerce Macros

This section lists the program code for the API and the associated macros to facilitate a sample integration of Net.Commerce and CICS.

### Notes:

1. The source files can be found on the diskette that comes with this publication.
2. In order to fit the code listings on the page, some lines have been wrapped.

### 5.9.1 Program Source Code

The following files constitute the source code of the legacyCics.DLL API which is called by Net.Commerce to interact with CICS. There are C source files (.C), header files (.H) and the make file for use with the Microsoft Visual C++ compiler (.MAK).

### 5.9.1.1 LEGCICS Function

The example below lists the LEGCICS.C file.

```

/*      This file contains the function GotoCics the Net.Commerce will call
      when it will Execute AUDIT_REG TASK */
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <assert.h>
#include <time.h>

#if defined(WIN32)
#include <windef.h>
#include <stdarg.h>
#include <winbase.h>
#endif

#include <nvpair_util.h> /*include for Net.Commerce*/
#include <commands.h> /*include for Net.Commerce*/
#include "epifunc.h"

void printHtmlError( char* aBuffer, const char * aCicsUser, int aCallingCommand,
                    const char* aUrl);

#if defined(WIN32)
__declspec( dllexport )
#endif

int GotoCics(int CallingCommand, struct APIErrorStruct* Error, struct NVPairMapStruct*
            NVP)
{
    char* userPageHtml;
    FILE *filePointer;
    short rc=0;
    const char* iLogid;
    const char* iPasswd;
    const char* iCicsUser;
    const char* iSalName;
    const char* iSaAddr1;
    const char* iSaCity;
    const char* iSaState;
    const char* iSaCntry;
    const char* iSaZipc;
    const char* iUrl;

    char information[2000];
    char tmpUser[80];

    iLogid=NVP_GetValue(NVP, "shlogid");
    iPasswd=NVP_GetValue(NVP, "shlpswd");
    iCicsUser = NVP_GetValue(NVP, "shschaans");
    iSalName = NVP_GetValue(NVP, "salname");
    iSaAddr1 = NVP_GetValue(NVP, "saaddr1");
    iSaCity = NVP_GetValue(NVP, "sacity");
    iSaState = NVP_GetValue(NVP, "sastate");
    iSaCntry = NVP_GetValue(NVP, "sacntry");
    iSaZipc = NVP_GetValue(NVP, "sazipc");
    iUrl=NVP_GetValue(NVP, "cancel_url");

    /*my check*/
    filePointer=fopen("d:\\cicssource\\testcics.txt", "w");
    fprintf(filePointer,"schaans = %s, salname= %s, saaddr1= %s, sacity= %s,
        sastate= %s, sacntry= %s, sazipc+%s \n", iCicsUser,iSalName,iSaAddr1,
        iSaCity,iSaState,iSaCntry,iSaZipc );
    fprintf(filePointer,"url = %s",iUrl);
    fprintf(filePointer,"login = %s,pwd = %s \n", iLogid, iPasswd);
    fclose(filePointer);

    /*end of my check*/

    // checking the default fields
    if (( *iSalName == '\0' )||
        ( *iSaAddr1 == '\0' )||
        ( *iSaCity == '\0' )||
        ( *iSaState == '\0' )||

```

```

        ( *iSaCity == '\0' )||
        ( *iSaZipc == '\0' )

    {
        AES_AppendNVPString(Error, "error_code", "290");
        AES_AppendNVPString(Error, "API", "AUDIT_REG");
        AES_AppendNVPString(Error, "REASON" , "REQUIRED PARAMETERS MISSING");
        return 1;
    }

    if (iCicsUser == '\0')
        return 0;
    else
        {
            rc=epiGoToCics(information,iCicsUser,iPasswd);

            if (rc!=0)
                { strcpy(tmpUser,iCicsUser);
                  userPageHtml=strcat(tmpUser,".htm");
                  AES_AppendNVPString(Error, "error_code",
                                      "300");
                  AES_AppendNVPString(Error, "API",
                                      "AUDIT_REG");
                  AES_AppendNVPString(Error, "REASON" ,
                                      "CICS REGISTRATION
                                      TRANSACTION ABORTED");
                  AES_AppendNVPString(Error, "USERPAGE" ,
                                      userPageHtml);
                  AES_AppendNVPString(Error, "HOST" ,
                                      "/demomall/registration/CICS");
                  printHtmlError(information,iCicsUser,
                                CallingCommand,iUrl);
                }
            return rc;
        }
    } //end else CicsUser==
return rc;

} // end function

void printHtmlError( char* aBuffer, const char * aCicsUser, int aCallingCommand,
                    const char* aUrl)
{
    /* In case of error the function write a File aCicsUser.htm so
    the responsible for Net.commerce Administrator can see the error*/
    FILE* f;
    char fileDirectory[80];
    char relFileName[80];
    char* fileName;
    char* relHtmlFileName;
    struct tm when;
    time_t now;
    time( &now );
    when = *localtime( &now );

    strcpy(fileDirectory,"c:\\IBM\\NetCommerce\\html\\En_us\\demomall\\
    registration\\CICS\\");
    strcpy(relFileName,aCicsUser);
    relHtmlFileName=strcat(relFileName,".htm");
    fileName=strcat(fileDirectory,relHtmlFileName);
    f=fopen(fileName,"w");
    fprintf(f,"<html> \n");
    fprintf(f,"<head> \n");
    fprintf(f,"</head> \n");
    fprintf(f,"<h2> ");
    fprintf(f,"ERROR PAGE for CICS USERS \n");
    fprintf(f,"</h2> \n");
    fprintf(f,"<body> \n");
    fprintf(f, "Current time is %s\n", asctime( &when ) );
    fprintf(f,"<BR>\n");
    fprintf(f,"REASON OF ERROR FOR USER <B>%s</B> is ",aCicsUser);
    fprintf(f,"<BR>\n");
    fprintf(f,"<P>\n");
    fprintf(f,"%s \n",aBuffer);
    fprintf(f,"<BR>\n");
    fprintf(f,"<A HREF=");

```

```

fprintf(f,"%c",''');
fprintf(f,"/cgi-bin/nph-msrvr/register/form");
fprintf(f,"%c",''');

if (aCallingCommand == _REGISTER_NEW_)
{
    fprintf(f,"> click here to make a new registration again </A>. \n");
}
else if (aCallingCommand == _REGISTER_MODIFY_)
{
    fprintf(f,"> click here to Update your registration again</A>. \n");
}
fprintf(f,"<BR>\n");
if (*aUrl == '\0'){
else
{
    fprintf(f,"<A HREF=");
    fprintf(f,"%c",''');
    fprintf(f,"%s",aUrl);
    fprintf(f,"%c",''');
    fprintf(f,"> click here return to your original page </A>. \n");
    fprintf(f,"<BR>\n");
}
fprintf(f,"</body> \n");
fprintf(f,"</html> \n");
fclose(f);
}

```

The example below lists the LEGCICS.H file.

```

#ifdef _LEGCICS_H
#define _LEGCICS_H
int GotoCics(int CallingCommand, struct APIErrorStruct* Error, struct NVPairMapStruct*
NVP);

#endif

```

### 5.9.1.2 EPIFUNC Function

The example below lists the EPIFUNC.C file.

```

#include "auxiliar.h"
#include <stdio.h>
#include "epifunc.h"
#include <cics_epi.h>
#include "cesn.h"
#include "epil.h"

short stopEpi(char * aInformation);
short startEpi(char * aInformation);

short epiGoToCics(char * aInformation,const char* aUserId, const char* aPassWord)
{
/* This Function is the calls all The Transaction*/
short rc=0;
/*term_index_t          termIndex;          */

rc= startEpi(aInformation);
if (rc!=0)
{
    stopEpi(aInformation);
    return rc;
}

rc=doCESN(aInformation,aUserId,aPassWord);
if (rc==0)
    rc=doEPI1(aInformation);

stopEpi(aInformation);

```

```

        return rc;
    }

short stopEpi(char * aInformation)
{
    /* this function terminates the EPI call*/
    short rc=0;
    char* name;

    /*TERMINATE EPI*/

    name ="stopEpi-CICS_EpiTerminate";
    rc = CICS_EpiTerminate();

    if (rc!=CICS_EPI_NORMAL)
    {
        response (name, rc, 0,aInformation);
        return 1;
    }
}

short startEpi(char * aInformation)
{
    /* this function initialize the EPI call*/

    short rc=0;
    char* name;

    /*INITIALIZE EPI*/
    name ="startEpi-CICS_EpiInitialize";
    rc = CICS_EpiInitialize(CICS_EPI_VERSION_100);

    if (rc!=CICS_EPI_NORMAL)
    {
        //response (name, rc, CICS_EPI_TERM_INDEX_NONE,aInformation);
        response (name, rc, 0,aInformation);
        return rc;
    }
}

```

The example below lists the EPIFUNC.H file.

```

#ifndef _EPIFUNC_H
#define _EPIFUNC_H

short epiGoToCics(char * aInformation,const char* aUserId, const char* aPassWord);

#endif

```

### 5.9.1.3 EPI1 Transaction

The example below lists the EPI1.C file.

```

/*      This file contains the code for the EPI1 transaction
      The EPI1 Transaction is the a Transaction defined on the
      CICSNT01 Server and routed to the CICE/ESA(CESA); in the CICS/ESA
      the Transaction Name is CESA*/

#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <process.h>
#include <cics3270.h>
#include "cesn.h"
#include "auxiliar.h"

void CICSEXIT callBackEPI1Proc (cics_ushort_t aIdx);

short getLastEPI1Event(term_index_t aTermIndex,char* aInformation);

short getEventCedaScreen(term_index_t aTermIndex,char* aInformation);

```



```

short doEPI1 (char* aInformation)
{
    /* this function execute the EPI1 Transaction */

    char*                dataPointer;
    DataStream           testData;
    short                rc=0;
    short                rterm=0;
    char*                name;
    term_index_t         termIndex;
    char                 server[70];
    CICS_EpiDetails_t    details;

    /* Starting a Terminal */
    strcpy(server,"CICSNT01");
    name ="doEPI1-CICS_EpiAddTerminal";
    rc = CICS_EpiAddTerminal (NULL,
                             server,
                             NULL,
                             NULL,
                             (CICS_EpiNotify_t)&callBackEPI1Proc,
                             &details,
                             &termIndex);

    if (rc!=CICS_EPI_NORMAL)
    {
        response (name, rc, termIndex, aInformation);
        return rc;
    }

    /*STARTING A TRANSACTION*/
    name ="doEPI1-CICS_EpiStartTran";

    testData.AIDKey = DFHENTER;
    testData.Byte0  = 0x20;
    testData.Byte1  = 0x20;
    memset (testData.TransactionCode, 0, CICS_EPI_TRANSID_MAX);
    memcpy (testData.TransactionCode, "EPI1",CICS_EPI_TRANSID_MAX);
    memset (testData.Data, 0, BUFFER_SIZE);
    dataPointer = &testData.AIDKey;

    rc = CICS_EpiStartTran (termIndex,
                           NULL,
                           dataPointer,
                           CICS_EPI_TRANSID_MAX+3);

    if (rc!=CICS_EPI_NORMAL)
    {
        response (name, rc, termIndex,aInformation);
    }

    /*getting the screen of the CEDA Transaction and Ending The transaction with PF3*/
    rc=getEventCedaScreen(termIndex,aInformation);
    if (rc==0)
        /*Getting The last Event*/
        rc=getLastEPI1Event(termIndex, aInformation);

    /*DELETE TERMINAL*/
    name ="doEPI1-CICS_EpiDelTerminal";
    rterm = CICS_EpiDelTerminal (termIndex);
    if (rterm!=CICS_EPI_NORMAL)
    {
        return 1;
    }

    return rc;
}

short getLastEPI1Event(term_index_t aTermIndex,char* aInformation)
{
    char cicsScreen[BUFFER_SIZE];
    short rc=0;
    char* name ="getLastEvent-CICS_EpiGetEvent";

```

```

CICS_EpiEventData_t  eventData;

    memset(cicsScreen,0,BUFFER_SIZE);
eventData.Data=cicsScreen;
eventData.Size =BUFFER_SIZE;
rc = CICS_EpiGetEvent (aTermIndex,
                      CICS_EPI_WAIT,
                      &eventData);

    if (rc!=CICS_EPI_NORMAL)
    {
        response (name, rc, aTermIndex, aInformation);
        return rc;
    }

    name ="getLastEvent-CICS_EpiGetEventII";
    memset(cicsScreen,0,BUFFER_SIZE);
eventData.Data=cicsScreen;
eventData.Size =BUFFER_SIZE;
rc = CICS_EpiGetEvent (aTermIndex,
                      CICS_EPI_WAIT,
                      &eventData);

    if (rc!=CICS_EPI_NORMAL)
    {
        response (name, rc, aTermIndex, aInformation);
        return rc;
    }

    if (eventData.Event!=CICS_EPI_EVENT_END_TRAN)
    {
        otherResponse(name,aTermIndex,"Wrong Event",aInformation);
        return 1;
    }

    return rc;
}

short getEventCedaScreen(term_index_t aTermIndex, char* aInformation)
{
    //char output[BUFFER_SIZE];
    char cicsScreen[BUFFER_SIZE];
    short rc=0;
    int index=0;
    int size=0;
    //int sizeScreen=0;
    char* name ="getEventCedaScreen-CICS_EpiGetEvent";
    CICS_EpiEventData_t  eventData;
    char*  dataPointer;
    DataStream  testData;

    //first EpiGetEvent
    memset(cicsScreen,0,BUFFER_SIZE);
eventData.Data=cicsScreen;
eventData.Size =BUFFER_SIZE;
rc = CICS_EpiGetEvent (aTermIndex,
                      CICS_EPI_WAIT,
                      &eventData);

    if (rc!=CICS_EPI_ERR_MORE_EVENTS)
    {
        response (name, rc, aTermIndex, aInformation);
        return 1;
    }

    if (eventData.Event!=CICS_EPI_EVENT_SEND)
    {
        otherResponse(name,aTermIndex,"Wrong Event",aInformation);
        return 1;
    }

    size = eventData.Size;
    if (size <=0)
    {
        otherResponse(name,aTermIndex,"Wrong Size",aInformation);
        return 1;
    }
}

```

```
    }

    if((rc=isText(cicsScreen,"Lock")==1)
    {
        otherResponse(name,aTermIndex,"Not a CEDA Screen",aInformation);
        return 1;
    }

    //memcpy(output,cicsScreen,BUFFER_SIZE);
    //sizeScreen=size;

    //second Epi GetEvent
    name ="getEventCedaScreen-CICS_EpiGetEventII";
    memset(cicsScreen,0,BUFFER_SIZE);
    eventData.Data=cicsScreen;
    eventData.Size =BUFFER_SIZE;
    rc = CICS_EpiGetEvent (aTermIndex,
                          CICS_EPI_WAIT,
                          &eventData);

    if (rc!=CICS_EPI_NORMAL)
    {
        response (name, rc, aTermIndex, aInformation);
        return 1;
    }
    if (eventData.Event!=CICS_EPI_EVENT_CONVERSE)
    {
        otherResponse(name,aTermIndex,"Wrong Event",aInformation);
        return 1;
    }
    //End transaction with PF3
    name ="doEPI1-CICS_EpiReply";
    testData.AIDKey = DFHPF3;
    testData.Byte0 = 0x20;
    testData.Byte1 = 0x20;
    memset (testData.TransactionCode, 0, CICS_EPI_TRANSID_MAX);
    memcpy (testData.TransactionCode, "EPI1",CICS_EPI_TRANSID_MAX);
    memset (testData.Data, 0, BUFFER_SIZE);
    //memcpy (testData.Data, output, sizeScreen);
    dataPointer = &testData.AIDKey;
    rc = CICS_EpiReply (aTermIndex,
                      dataPointer,
                      CICS_EPI_TRANSID_MAX+3);

    if (rc!=CICS_EPI_NORMAL)
    {
        response (name, rc, aTermIndex,aInformation);
        return 1;
    }
    return 0;
}

void CICSEXIT callBackEPI1Proc (cics_ushort_t aidx)
{
    /* Empty No user Exit procedure*/
}

```

The example below lists the EPI1.H file.

```
#ifndef _EPI1_H
#define _EPI1_H
#include "auxiliar.h"
short doEPI1 (char* aInformation);

#endif

```

### 5.9.1.4 CESN Transaction

The example below lists the CESN.C file.

```

/*      This file contains the code for the CESN transaction
      The CESN Transaction is the SIGN-ON Transaction defined on the
      CICSNT01 Server */

#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <process.h>
#include <cics3270.h>
#include "cesn.h"
#include "auxiliar.h"

void CICSEXIT callBackCESNProc (cics_ushort_t aIdx);
short setText(char* aScreen, const char * aLabel, const char *aName);
short getEventPwdScreen(term_index_t aTermIndex, const char* aUserId,
                        const char* aPassWord, char* aInformation);
short signOnSuccessful(term_index_t aTermIndex, char* aInformation);

short doCESN (char* aInformation, const char* aUserId, const char* aPassWord)
{
    /* this function execute the CESN Transaction */

    char*          dataPointer;
    DataStream     testData;
    short          rc=0;
    short          rterm=0;
    char*          name;
    term_index_t   termIndex;
    char           server[70];
    CICS_EpiDetails_t details;

    /* Starting a Terminal */
    strcpy(server,"CICSNT01");
    name ="doCESN-CICS_EpiAddTerminal";
    rc = CICS_EpiAddTerminal (NULL,
                             server,
                             NULL,
                             NULL,
                             (CICS_EpiNotify_t)&callBackCESNProc,
                             &details,
                             &termIndex);

    if (rc!=CICS_EPI_NORMAL)
    {
        response (name, rc, termIndex, aInformation);
        return rc;
    }

    /*STARTING A TRANSACTION*/
    name ="doCESN-CICS_EpiStartTran";

    testData.AIDKey = DFHENTER;      /* ENTER key */
    testData.Byte0 = 0x20;           /* Buffer address byte 0 cursor Address*/
    testData.Byte1 = 0x20;           /* Buffer address byte 1 */
    memset (testData.TransactionCode, 0, CICS_EPI_TRANSID_MAX);
    /*to clean*/
    memcpy (testData.TransactionCode, "CESN", CICS_EPI_TRANSID_MAX);
    /* Transaction CESN */
    memset (testData.Data, 0, BUFFER_SIZE); /* Clear data area */

    dataPointer = &testData.AIDKey; /* Point to beginning of the datastream */

    rc = CICS_EpiStartTran (termIndex,
                           NULL,
                           dataPointer,
                           CICS_EPI_TRANSID_MAX+3);

    if (rc!=CICS_EPI_NORMAL)
    {
        response (name, rc, termIndex, aInformation);
        return rc;
    }
}

```

```

    }

    rc=getEventPwdScreen(termIndex,aUserId, aPassWord, aInformation);
    if (rc==0)
        rc=signOnSuccessfull(termIndex,aInformation);

    name ="doCESN-CICS_EpiDelTerminal";
    rterm = CICS_EpiDelTerminal (termIndex);
    if (rterm!=CICS_EPI_NORMAL)
    {
        return 1;
    }

    return rc;
}

short setText(char* aScreen, const char * aLabel, const char *aName)
{
/*This function check the screen for the Label and update the screen with aName*/
    char* p;
    int i;
    i=strlen(aName);
    if ((p=strstr(aScreen,aLabel))==NULL)
        return 1;
    memcpy((p+14),aName,strlen(aName));
    return 0;
}

short getEventPwdScreen(term_index_t aTermIndex,const char* aUserId,
                        const char* aPassWord,char* aInformation)
{
/* This function controls that the CESN transaction has displayed the right screen
and prepare the screen to input aUserId and aPassword */
    short rc=0;
    CICS_EpiEventData_t  eventData;
    char* name;
    int size;
    cics_char_t          cesnWrapper[30];
    cics_ushort_t lenId,lenPwd;
    cics_ushort_t cicslen;
    char                cicsScreen[BUFFER_SIZE];

    memset(cicsScreen,0,BUFFER_SIZE);
    eventData.Data=cicsScreen;
    eventData.Size =BUFFER_SIZE;

    /* getting the first Event*/
    name ="getEventPwdScreen-CICS_EpiGetEvent";
    rc = CICS_EpiGetEvent (aTermIndex,
                          CICS_EPI_WAIT,
                          &eventData);

    if (rc!=CICS_EPI_ERR_MORE_EVENTS)
    {
        response (name, rc, aTermIndex, aInformation);
        return 1;
    }

    if (eventData.Event!=CICS_EPI_EVENT_SEND)
    {
        otherResponse(name,aTermIndex,"Wrong Event",aInformation);
        return 1;
    }

    size = eventData.Size;
    if (size <=0)
    {
        otherResponse(name,aTermIndex,"Size <0",aInformation);
        return 1;
    }

/*check if is the CESN Screen*/
    if((rc=isText(cicsScreen,"Userid"))==1)
    {
        otherResponse(name,aTermIndex,"NOT a CESN SCREEN",aInformation);
        return 1;
    }
}

```

```

    }
    if((rc=isText(cicsScreen,"Password")==1)
    {
        otherResponse(name,aTermIndex,"Not a CESN SCREEN",aInformation);
        return 1;
    }

/* getting the second Event*/

    name ="getEventPwdScreen-CICS_EpiGetEventII";
    memset(cicsScreen,0,BUFFER_SIZE);
    eventData.Data=cicsScreen;
    eventData.Size =BUFFER_SIZE;
    rc = CICS_EpiGetEvent (aTermIndex,
                          CICS_EPI_WAIT,
                          &eventData);

    if (rc!=CICS_EPI_NORMAL)
    {
        response (name, rc, aTermIndex, aInformation);
        return 1;
    }
    if (eventData.Event!=CICS_EPI_EVENT_END_TRAN)
    {
        otherResponse(name,aTermIndex,"Wrong Event",aInformation);
        return 1;
    }

    lenId=strlen(aUserId);
    lenPwd=strlen(aPassWord);
    cicslen=lenId+lenPwd+13;
    name ="getEventPwdScreen-CICS_EpiStartTransII";
    memset(cesnWrapper,0,30);
    sprintf(cesnWrapper,"%c%c%cCESN%c%c%c%s%c%c%c%s",
           0x27,0x00,0x00,
           0x11,0x2b,0x21,aUserId,
           0x11,0x21,0x5e,aPassWord);
    rc=CICS_EpiStartTran(aTermIndex,NULL,(cics_ubyte_t*)cesnWrapper,
                       cicslen);
    if (rc!=CICS_EPI_NORMAL)
    {
        response (name, rc, aTermIndex,aInformation);
        return rc;
    }

    return rc;
}

short signOnSuccessfull(term_index_t aTermIndex, char* aInformation)
{
/* This function checks if the SignOn was successfull*/
    char    cicsScreen[BUFFER_SIZE];
    short rc;
    short rEvent=0;
    CICS_EpiEventData_t  eventData;

    /*getting the first Event after the transaction*/
    char* name ="signOnSuccessfull-CICS_EpiGetEvent";
    memset(cicsScreen,0,BUFFER_SIZE);
    eventData.Data=cicsScreen;
    eventData.Size =BUFFER_SIZE;
    rc = CICS_EpiGetEvent (aTermIndex,
                          CICS_EPI_WAIT,
                          &eventData);

    if ((rc!=CICS_EPI_NORMAL))
    {
        response (name, rc, aTermIndex, aInformation);
        //return rc;
    }
    if (eventData.Event!=CICS_EPI_EVENT_SEND)
    {
        otherResponse(name,aTermIndex,"Wrong Event",aInformation);
        //return rc;
    }
}

```

```

    }
    /*getting the second Event after the transaction*/
    name ="signOnSuccessfull-CICS_EpiGetEventII";
    memset(cicsScreen,0,BUFFER_SIZE);
    eventData.Data=cicsScreen;
    eventData.Size =BUFFER_SIZE;
    rc = CICS_EpiGetEvent (aTermIndex,
                          CICS_EPI_WAIT,
                          &eventData);
    if ((rc!=CICS_EPI_NORMAL)&&(rc!=CICS_EPI_ERR_MORE_EVENTS))
    {
        response (name, rc, aTermIndex, aInformation);
        //return rc;
    }

    /*check if in the screen there is a word "complete"*/
    if((rc==isText(cicsScreen,"complete"))==1)
    {
        otherResponse(name,aTermIndex,"Wrong Password or Userid",aInformation);
    }

    /* Getting Additional Events */
    name="signOnSuccessfull-AdditionalEvents";
    while(eventData.Event!=CICS_EPI_EVENT_END_TRAN)
    {
        memset(cicsScreen,0,BUFFER_SIZE);
        eventData.Data=cicsScreen;
        eventData.Size =BUFFER_SIZE;
        rEvent = CICS_EpiGetEvent (aTermIndex,
                                  CICS_EPI_WAIT,
                                  &eventData);
        if ((rEvent!=CICS_EPI_NORMAL)&&(rEvent!=CICS_EPI_ERR_MORE_EVENTS))
        {
            response (name, rEvent, aTermIndex, aInformation);
            return rEvent;
        }
    }

    return rc;
}

void CICSEXIT callBackCESNProc (cics_ushort_t aIdx)
{
    /* Empty No user Exit procedure*/
}

```

The example below lists the CESN.H file.

```

#ifndef _CESN_H
#define _CESN_H
#include "auxiliar.h"
short doCESN (char* aInformation, const char* aUserId, const char* aPassWord);

#endif

```

### 5.9.1.5 AUXILIAR Function

The example below lists the AUXILIAR.C file.

```

/* This file contains auxiliary function*/

#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <process.h>
#include "auxiliar.h"

/*****AUXILIAR FUNCTIONS*****/
void response (char *aCall, short rc, short aIndex, char* aInformation)
{
    char          *p;
}

```

```

switch (rc)
{
    default:                p = "!!!Unknown!!!";          break;
    case CICS_EPI_NORMAL:   p = "CICS_EPI_NORMAL";          break;
    case CICS_EPI_ERR_NOT_INIT: p = "CICS_EPI_ERR_NOT_INIT";      break;
    case CICS_EPI_ERR_BAD_INDEX: p = "CICS_EPI_ERR_BAD_INDEX";    break;
    case CICS_EPI_ERR_FAILED: p = "CICS_EPI_ERR_FAILED";      break;
    case CICS_EPI_ERR_VERSION: p = "CICS_EPI_ERR_VERSION";        break;
    case CICS_EPI_ERR_IS_INIT: p = "CICS_EPI_ERR_IS_INIT";      break;
    case CICS_EPI_ERR_NO_SYSTEMS: p = "CICS_EPI_ERR_NO_SYSTEMS";    break;
    case CICS_EPI_ERR_MORE_SYSTEMS: p = "CICS_EPI_ERR_MORE_SYSTEMS"; break;
    case CICS_EPI_ERR_MAX_TERMS: p = "CICS_EPI_ERR_MAX_TERMS";    break;
    case CICS_EPI_ERR_SYSTEM: p = "CICS_EPI_ERR_SYSTEM";          break;
    case CICS_EPI_ERR_TRAN_ACTIVE: p = "CICS_EPI_ERR_TRAN_ACTIVE";  break;
    case CICS_EPI_ERR_NO_DATA: p = "CICS_EPI_ERR_NO_DATA";        break;
    case CICS_EPI_ERR_TTI_ACTIVE: p = "CICS_EPI_ERR_TTI_ACTIVE";    break;
    case CICS_EPI_ERR_ATI_ACTIVE: p = "CICS_EPI_ERR_ATI_ACTIVE";    break;
    case CICS_EPI_ERR_NO_CONVERSE: p = "CICS_EPI_ERR_NO_CONVERSE"; break;
    case CICS_EPI_ERR_ATI_STATE: p = "CICS_EPI_ERR_ATI_STATE";    break;
    case CICS_EPI_ERR_NO_EVENT: p = "CICS_EPI_ERR_NO_EVENT";      break;
    case CICS_EPI_ERR_MORE_EVENTS: p = "CICS_EPI_ERR_MORE_EVENTS"; break;
    case CICS_EPI_ERR_MORE_DATA: p = "CICS_EPI_ERR_MORE_DATA";    break;
    case CICS_EPI_ERR_WAIT: p = "CICS_EPI_ERR_WAIT";              break;
    case CICS_EPI_ERR_SERVER_DOWN: p = "CICS_EPI_ERR_SERVER_DOWN"; break;
    case CICS_EPI_ERR_SENSE_CODE: p = "CICS_EPI_ERR_SENSE_CODE";   break;
    case CICS_EPI_ERR_SECURITY: p = "CICS_EPI_ERR_SECURITY";      break;
    case CICS_EPI_ERR_IN_CALLBACK: p = "CICS_EPI_ERR_IN_CALLBACK"; break;
    case CICS_EPI_ERR_NULL_PARM: p = "CICS_EPI_ERR_NULL_PARM";    break;
} /* endswitch */
printErr ( aCall, p,aInformation);

if (rc == CICS_EPI_ERR_FAILED)
{
    error (aIndex,aInformation);
} /* endif */
} /* Response */

void otherResponse (char *aCall,short aIndex,char* aReason, char* aBuffer)
{
    sprintf(aBuffer,"On Terminal %d, the call %s returned: %s \n",aIndex,
        aCall,aReason);
}

void error (short aIndex,char* aInformation)
{
    CICS_EpiSysError_t sysErr;
    CICS_EpiGetSysError (aIndex, &sysErr);
    printCicsErr(sysErr,aInformation);
}

short isText(char* aScreen, const char * aText)
{
    char* p;
    if ((p=strstr(aScreen,aText))==NULL)
        return 1;
    return 0;
}

int printCicsErr(CICS_EpiSysError_t aSysErr, char* aBuffer)
{
    return sprintf(aBuffer, "SysErr: Cause=%lu, Value=%lu, Msg=%s \n",
        aSysErr.Cause, aSysErr.Value, aSysErr.Msg );
}

int printErr(char* aCall, char* p, char* aBuffer)
{
    return sprintf(aBuffer,"%s returned: %s \n", aCall, p);
}

```

The example below lists the AUXILIAR.H file.



```

#ifndef __AUXILIAR_H
#define __AUXILIAR_H

#include <stdio.h>
#include <string.h>
#include <cics_epi.h>

#define BUFFER_SIZE 2000 /* Datastream buffer size */
#define LIST_SIZE 10 /* Used by EpiListSystems - maximum */
/* number of systems connectable to */
typedef unsigned short term_index_t; /* Variable type to hold a Terminal*/

typedef struct /* Format of 3270 datastream */
{
    char AIDKey;
    char Byte0;
    char Byte1;
    char TransactionCode[CICS_EPI_TRANSID_MAX];
    char Data[BUFFER_SIZE];
} DataStream;

/*****AUXILIAR FUNCTIONS*****/
void response (char *aCall, short rc, short aIndex, char* aInformation);
void otherResponse (char *aCall,short aIndex,char* aReason, char* aBuffer);
void error (short aIndex,char* aInformation);
short isText(char* aScreen, const char * aText);
int printCicsErr(CICS_EpiSysError_t aSysErr, char* aBuffer);
int printErr(char* aCall, char* p, char* aBuffer);
#endif

```

### 5.9.1.6 MAKE File

The example below lists the MAKECICS.MAK file.

```

CFLAGS= /nologo /MT /W3 /GX /Zi /O2 /D "WIN32" /D "_X86_" /D "DEBUG" /D "CICS_W32"
        /YX /c
CPP= cl
ERASE= del
COPY = copy

INCLUDE001=c:\ibm\netcommerce\adt;
INCLUDE002=$(INCLUDE001);d:\cicscli\include;
INCLUDE=$(INCLUDE);$(INCLUDE002)
LINK32=link

LINK32_FLAGS= kerne132.lib user32.lib gdi32.lib winspool.lib cmd1g32.lib\
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib\
odbc32.lib db2api.lib db2cli.lib oldnames.lib d:\cicscli\lib\cclwin32.lib /nologo
        /subsystem:windows /dll\
/incremental:no /machine:I386\
/out:"legacyCics.dll" /implib:"legacyCics.lib"

all: legacyCics.dll

legacyCics.dll: nvpair_util.obj legcics.obj epil.obj epifunc.obj cesn.obj
        auxiliar.obj
        $(LINK32) nvpair_util.obj legcics.obj epil.obj epifunc.obj cesn.obj
        auxiliar.obj \
        $(LINK32_FLAGS)

nvpair_util.obj : c:\ibm\netcommerce\adt\nvpair_util.c
        $(CPP) $(CFLAGS) /TP c:\ibm\netcommerce\adt\nvpair_util.c

legcics.obj : legcics.c
        $(CPP) $(CFLAGS) legcics.c

epifunc.obj : epifunc.c
        $(CPP) $(CFLAGS) epifunc.c

epil.obj : epil.c
        $(CPP) $(CFLAGS) epil.c

```

```
cesn.obj : cesn.c
          $(CPP) $(CFLAGS) cesn.c

auxiliar.obj : auxiliar.c
             $(CPP) $(CFLAGS) auxiliar.c
```

## 5.9.2 Macro Definitions

The following files constitute the source code of the Net.Commerce (Net.Data) macros that have to be used in conjunction to the legacyCics.DLL API.

### 5.9.2.1 Register Form

The example below lists the MALLREGFORM.D2W file.

```
%{=====
The sample Templates, HTML and Macros are furnished by IBM as simple
examples to provide an illustration. These examples have not been
thoroughly tested under all conditions. IBM, therefore, cannot guarantee
imply reliability, serviceability or function of these programs. All
programs contained herein are provided to you "AS IS".

The sample Templates, HTML and Macros may include the names of individuals,
companies, brands and products in order to illustrate them as completely as
possible. All of these are names are fictitious and any similarity to the names
and addresses used by an actual business enterprise is entirely coincidental.

(C) Copyright IBM Corp. 1995, 1996.
=====}%}

%define {
    url_in = url ? "$(url)" : ""
}

%SQL(MallInfo){

    select mhthead, mhtfoot, mhtbase
    from mall

    %SQL_REPORT{
        %ROW{
            @DTW_assign( MallHeader, V_mhthead)
            @DTW_assign( MallFooter, V_mhtfoot)
            @DTW_assign( MallBase, V_mhtbase)
        }
    }

    %SQL_MESSAGE{
        default: { %} :continue %}

    }

%SQL(Get_Name) {
    select shlogid, satitle, safname, samname, salname
    from shopper, shaddr
    where (
        sanick = '$(SESSION_ID)' and shlogid='$(SESSION_ID)' and
        (shshtyp = 'R' or shshtyp = 'A')
        and sashnbr=shrfnbr and saadrflg='P'
    )

    %SQL_REPORT {

        %ROW{
            %if (V_satitle != "NULL")
                $(V_satitle).
            %endif
            %if (V_safname != "NULL")
                $(V_safname)
            %endif
            %if (V_samname != "NULL")
                $(V_samname)
            %endif
        }
    }
}
```

```
        %endif
        %if (V_salname != "NULL")
            $(V_salname),
        %endif
        you registered with a Login ID of <B>$(V_shlogid)</B>.<BR>
        Remember to write down your Login ID and Password in a safe place.</td>

    %}

%SQL_MESSAGE{
    default: {
        ERROR
    }
}

%}

%HTML_INPUT{

<HTML>

%EXEC_SQL(MallInfo)

<HEAD>
    <META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
    <TITLE>Registration Information</TITLE>
</HEAD>

<BODY BACKGROUND="/ncsample/back1.gif">

    %include "$(MallHeader)"

    <p align=center><font size=5><strong>Registration is successful</strong></font></p>

    <CENTER>
        <TABLE WIDTH=80%>
            <TR>
                <TD align=center width=50></td>
                <TD>
                    %EXEC_SQL(Get_Name)
                </TD>
                <TD align=center width=50></td>
            </TR>
        </TABLE>

        %if (url_in == "")
            Proceed to the <A HREF="/cgi-bin/nph-msrvr;/execmacro/mall_dir.d2w/report">
                Mall Directory</A>.
        %else
            <FORM ACTION="$(url_in)">
                <INPUT TYPE="submit" VALUE="Continue">
            </FORM>
        %endif
    </CENTER>

    %include "$(MallFooter)"

</BODY>

</HTML>
%}

%HTML_REPORT{

<HTML>

%EXEC_SQL(MallInfo)

<HEAD>
    <META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
    <TITLE>Registration Information</TITLE>
</HEAD>
```

```

<BODY BACKGROUND="/ncsample/back1.gif">

%include "$(MallHeader)"

<CENTER>

<FORM method=POST action="/cgi-bin/nph-msrvr/register/new">
  <INPUT Type = "hidden" Name="url" Value="/msprotect/nph-msrvr/execmacro/
    mallregform.d2w/input?mode=new&url=$(url_in)" >
  <INPUT Type = "hidden" Name="mode" Value="new" >
  <INPUT Type = "hidden" Name="cancel_url" Value="$(url_in)" >

  <p align=center><font size=5><strong>New Registration</strong></font></p>

  <TABLE WIDTH=80%>

    <TR>
      <td align=center width=50></td>
      <td>To register as a new shopper, fill in the form below.
        Items listed in <b>bold</b> must not be left blank.
        All information will be kept strictly confidential by the mall and store
        owners.
        We will not communicate this information to others as part of a mailing
        list.</td>
      <td align=center width=50>
      </td>
    </TR>

    %INCLUDE "/ncsample/msregadd.inc"
    %INCLUDE "/ncsample/msregform.inc"

    <TR>
      <TD WIDTH=40><IMG SRC="/ncsample/4.gif"></TD>
      <TD WIDTH=460 VALIGN=MIDDLE><FONT SIZE=+2><B>Submit Your Registration</B>
      </FONT></TD>
    </TR>

    <TR>
      <TD></TD>
      <TD>
        You are now ready to submit your registration for approval.
        To submit your registration, click the Submit Registration button.
        To cancel your registration, click the back arrow button below.
        <TABLE CELLPADDING=10>
          <TR>
            <TD>
              <IMG SRC="/ncsample/hoption.gif">
            </TD>
            <TD>
              <INPUT TYPE="submit" VALUE="Submit Registration">
            </TD>
          </TR>
          <TR>
            <TD>
              <IMG SRC="/ncsample/hoption.gif">
            </TD>
            <TD>
            </TD>
          </TR>
        </TABLE>
      </TD>
    </TR>

    %if (url_in == "")
      <A HREF="$(MallBase)"><IMG SRC="/ncsample/cancel.gif" border=0></A>
    %else
      <A HREF="$(url_in)"><IMG SRC="/ncsample/cancel.gif" border=0></A>
    %endif
  </TD>
</TR>
</TABLE>
</TD>
</TR>
</TABLE>
</CENTER>

%include "$(MallFooter)"

```

```
</BODY>
</HTML>
%}
```

### 5.9.2.2 Register Update

The example below lists the MALLREGUPD.D2W file.

```
%{=====
The sample Templates, HTML and Macros are furnished by IBM as simple
examples to provide an illustration. These examples have not been
thoroughly tested under all conditions. IBM, therefore, cannot guarantee
imply reliability, serviceability or function of these programs. All
programs contained herein are provided to you "AS IS".

The sample Templates, HTML and Macros may include the names of individuals,
companies, brands and products in order to illustrate them as completely as
possible. All of these are names are fictitious and any similarity to the names
and addresses used by an actual business enterprise is entirely coincidental.

(C) Copyright IBM Corp. 1995, 1996.
=====}%

%define {
    url_in = url ? "$(url)" : ""
    mode_in = "modify"
}%

%SQL(MallInfo){

    select mhthead, mhtfoot, mhtbase from mall

    %SQL_REPORT{
        %ROW{
            @DTW_assign( MallHeader, V_mhthead)
            @DTW_assign( MallFooter, V_mhtfoot)
            @DTW_assign( MallBase, V_mhtbase)
        }
    }

    %SQL_MESSAGE{
        default: { %} :continue %}
    }

%SQL (DO_UPDATE) {

    with selshaddr as
        (select saeph1st, satitle, salname, safname, samname, sarepcom, saphone1,
            saphone2, safax, saaddr1, saaddr2, saaddr3, sacity, sastate,
            sacntry, sazipc, saemail1, saemail2, sadphtyp, saephtyp, sabtcall,
            sashnbr
            from shaddr
            where sanick='$(SESSION_ID)' and saadrflg='P'),

    selshopper as
        (select saeph1st, satitle, salname, safname, samname, sarepcom, saphone1,
            saphone2, safax, saaddr1, saaddr2, saaddr3, sacity, sastate,
            sacntry, sazipc, saemail1, saemail2, sadphtyp, saephtyp, sabtcall,
            shlogid, shlpswd, shshtyp, shcomm, shchaans, shrfnbr
            from shopper, selshaddr
            where shrfnbr=sashnbr and shlogid='$(SESSION_ID)' )

    select saeph1st, satitle, salname, safname, samname, sarepcom, saphone1,
        saphone2, safax, saaddr1, saaddr2, saaddr3, sacity, sastate, sacntry,
        sazipc, saemail1, saemail2, sadphtyp, saephtyp, sabtcall, shlogid,
        shlpswd, shshtyp, shcomm, shchaans, sdgendr, sdage, sdincom, sdmstat,
        sdhnbr, sdcomp, sdintrs
    from shopdem, selshopper
    where shrfnbr=sdshnbr
```

```

%SQL_REPORT{
  %ROW{
    <form method=POST action="/cgi-bin/nph-msrvr/register/modify">
    <INPUT Type = "hidden" Name="url" Value="/cgi-bin/nph-msrvr/
      ;execmacro/mallregupd.d2w/input?mode=modify&url=${url_in}" >
    <INPUT Type = "hidden" Name="mode" Value="modify" >
    <INPUT Type = "hidden" Name="cancel_url" Value="${url_in}" >

    <p align=center><font size=5><strong>Change Registration</strong></font></p>
    <CENTER>
    <TABLE WIDTH=80%>

    <TR>
    <td align=center width=50></td>
    <td>To change your registration information, simply update the form below.
    Items listed in <b>bold</b> must not be left blank.
    All information will be kept strictly confidential by the mall and store
    owners.
    We will not communicate this information to others as part of a mailing
    list.</td>

    <td align=center width=50>
    </td></tr>

    %INCLUDE "/ncsample/msvarmap.inc"
    %INCLUDE "/ncsample/msreglabels.inc"
    %INCLUDE "/ncsample/msregform.inc"

    <TR>
    <TD WIDTH=40><IMG SRC="/ncsample/4.gif"></TD>

    <TD WIDTH=460 VALIGN=MIDDLE><FONT SIZE=+2><B>Submit Your Changes</B>
    </FONT></TD>

    </TR><TR>

    <TD></TD><TD>
    You are now ready to change your registration.
    To change your registration, click the Change Registration button.
    To cancel your changes, click the back arrow button below.
    <TABLE CELLPADDING=10>
    <TR>
    <TD><IMG SRC="/ncsample/hoption.gif"></TD><TD>
    <INPUT TYPE="submit" VALUE="Change Registration">
    </TD>
    </TR>
    <TR>
    <TD><IMG SRC="/ncsample/hoption.gif"></TD><TD>
    </FORM>
    %if (url_in == "")
    <A HREF="${MallBase}"><IMG SRC="/ncsample/cancel.gif" border=0></A>
    %else
    <A HREF="${url_in}"><IMG SRC="/ncsample/cancel.gif" border=0></A>
    %endif

    </TD>
    </TABLE>
    </TD></TR>
    </TABLE>

    </CENTER>

    %}

  %}

%}

%HTML_INPUT{
<HTML>

```

```
%EXEC_SQL(MallInfo)

<HEAD>
  <META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
  <TITLE>Mall Registration</TITLE>
</HEAD>

<BODY BACKGROUND="/ncsample/back1.gif">

  %include "$(MallHeader)"

  <p align=center><font size=5><strong>Registration is changed</strong></font></p>

  <CENTER>
    <TABLE WIDTH=80%>
      <TR>
        <TD align=center width=50></td>
        <TD>
          Changes have been made successfully to your account.<BR>
          If you changed your Password, please remember to write them down in a safe
          place.
        </TD>
        <TD align=center width=50></td>
      </TR>
    </TABLE>

    %if (url_in == "")
      Proceed to the <A HREF="/cgi-bin/nph-msrvr;/execmacro/mall_dir.d2w/report">
      Mall Directory</A>.
    %else
      <FORM ACTION="$(url_in)">
      <INPUT TYPE="submit" VALUE="Continue">
      </FORM>
    %endif
  </CENTER>

  %include "$(MallFooter)"

</BODY>

</HTML>
%}

%HTML_REPORT{

<HTML>

%EXEC_SQL(MallInfo)

<HEAD>
  <META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
  <TITLE>Registration Information</TITLE>
</HEAD>

<BODY BACKGROUND="/ncsample/back1.gif">

  %include "$(MallHeader)"

  %EXEC_SQL(DO_UPDATE)

  %include "$(MallFooter)"

</BODY>

</HTML>

%}
```

### 5.9.2.3 Exception Macro

The example below lists the MALL\_ERR\_REG\_LEGACYCICS.D2W file.

```

%{=====
The sample Templates, HTML and Macros are furnished by IBM as simple
examples to provide an illustration. These examples have not been
thoroughly tested under all conditions. IBM, therefore, cannot guarantee
imply reliability, serviceability or function of these programs. All
programs contained herein are provided to you "AS IS".

The sample Templates, HTML and Macros may include the names of individuals,
companies, brands and products in order to illustrate them as completely as
possible. All of these are names are fictitious and any similarity to the names
and addresses used by an actual business enterprise is entirely coincidental.

(C) Copyright IBM Corp. 1995, 1996.
=====}%

%define {
mode_in = mode ? "$(mode)" : "new"
cancel = cancel_url ? "$(cancel_url)" : ""
%}

%SQL(MallInfo){

  select mhthead, mhtfoot, mhtbody
  from mall

  %SQL_REPORT{

    %ROW{
      @DTW_assign( MallHeader, V_mhthead )
      @DTW_assign( MallFooter, V_mhtfoot )
      @DTW_assign( MallBase, V_mhtbody )
    }
  }

  %SQL_MESSAGE{
    100 : {
      %} : continue
    default: { %} :continue %}

  %}

%HTML_REPORT{

<HTML>

<HEAD>
  <META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
  <TITLE>
    Registration Information
  </TITLE>
</HEAD>

%EXEC_SQL(MallInfo)

<BODY BACKGROUND="/ncsample/back1.gif">

%include "$(MallHeader)"

  <center>

    %if (mode_in == "new")
      <p align=center><font size=5><strong>Register</strong></font></p>
    %else
      <p align=center><font size=5><strong>Change Registration</strong></font></p>
    %endif

    %INCLUDE "/ncsample/msreglabels.inc"

    <form method=POST action="/cgi-bin/nph-msrvr/register/$(mode_in)">
      %if (mode_in == "new")
        <INPUT Type = "hidden" Name="url" Value="/msprotect/nph-msrvr/

```



```

                                ;execmacro/mallregform.d2w/input?mode=$(mode_in)" >
%else
  <INPUT Type = "hidden" Name="url" Value="/msprotect/nph-msrvr/
                                ;execmacro/mallregupd.d2w/input?mode=$(mode_in)" >
%endif
<INPUT TYPE="hidden" NAME="mode" VALUE="$(mode_in)">

<TABLE WIDTH=80% border=0>
  <TR>

    <td align=center width=85>
      
    </td>

    <td>
      There is a problem with your submission:<P>
      <B>
        %if (error_code == "4")
          There was a problem saving your password. Please try again.<BR>
          If the problem persists, please notify customer service.
        %elif (error_code == "160")
          The information in the <i><a href="#$(field)">$(msg)</a></i> field is
          invalid.
        %elif (error_code == "170")
          The <i><a href="#shlpswd">verify password</A></i> did not match your
          password.
        %elif (error_code == "180")
          Someone is already using this <i><a href="#shlogid">Login ID</a></i>.
          Please choose another.
        %elif (error_code == "190")
          The <i><a href="#$(field)">$(msg)</a></i> field must be filled in.
        %elif (error_code == "200")
          You have already registered.
        %elif (error_code == "210")
          This error is to be fixed by the server team.
        %elif (error_code == "270")
          The passwords can only consist of alphabetic (a-z) and numeric (0-9)
          characters .
        %elif (error_code == "280")
          Shopper Login IDs can only consist of alphabetic (a-z) and numeric (0-9)
          characters.
        %elif (error_code == "290")
          Error $(error_code) occurred. The API is $(API). The reason is $(REASON);
        %elif (error_code == "300")
          Error $(error_code) occurred. The API is $(API).<BR>
          The reason is $(REASON).<BR>
          Contact Your Merchant and tell him that the reason of
          error is.<BR>
          <A HREF="$(HOST)/$(USERPAGE)"> click here </A>.
        %else
          An undefined error occurred. Please contact the mall administrator.
          The error code is $(error_code).
        %endif
      </B>
    </TD>
  </TR>

  %INCLUDE "/ncsample/msregform.inc"

  <TR>
    <TD WIDTH=40>
      <IMG SRC="/ncsample/4.gif">
    </TD>
    %if (mode_in == "new")
      <TD WIDTH=460 VALIGN=MIDDLE>
        <FONT SIZE=+2><B>Submit Your Registration</B></FONT>
      </TD>
    %else
      <TD WIDTH=460 VALIGN=MIDDLE>
        <FONT SIZE=+2><B>Submit Your Changes</B></FONT>
      </TD>
    %endif
  </TR>

</TR>

```

```

<TD></TD>
<TD>
%if (mode_in == "new")
  You are now ready to submit your registration for approval.
  To submit your registration, click the Submit Registration button.
  To cancel your registration, click the back arrow button below.<P>
%else
  You are now ready to change your registration.
  To change your registration, click the Change Registration button.
  To cancel your changes, click the back arrow button below.<P>
%endif
</TD>
</TR>

<TR>
<TD></TD>
<TD>
  <TABLE CELLPADDING=10>
    <TR>
      <TD>
        <IMG SRC="/ncsample/hoption.gif">
      </TD>
      <TD>
        %if (mode_in == "new")
          <INPUT TYPE="submit" VALUE="Submit Registration">
        %else
          <INPUT TYPE="submit" VALUE="Change Registration">
        %endif
      </TD>
    </TR>
  </TABLE>
  <TR>
    <TD>
      <IMG SRC="/ncsample/hoption.gif">
    </TD>
  </TR>
</FORM>
  <TD>
    %if (cancel != "")
      <A HREF="$(url_in)"><IMG SRC="/ncsample/cancel.gif" border=0></A>
    %else
      <A HREF="$(MailBase)"><IMG SRC="/ncsample/cancel.gif" border=0></A>
    %endif
  </TD>
</TR>
</TABLE>
</TD>
</TR>

</TABLE>

</CENTER>

%include "$(MailFooter)"

</BODY>
</HTML>
%}

```

### 5.9.3 Include Files

These macros also include four .INC files to be copied into the directory IBMNETCOMMERCEMACROEN\_USNCSAMPLE:

- msregadd.inc
- msregform.inc
- msreglabels.inc
- msvarmap.inc

### 5.9.3.1 MSREGADD Include File

The example below lists the MSREGADD.INC file:

```
@DTW_assign(l_shlogid, "Shopper's Login ID")
@DTW_assign(l_shlpswd, "Password")
@DTW_assign(l_shlpswdver, "Verify Password")
@DTW_assign(l_shchaans, "CICS User")
@DTW_assign(l_satitle, "Title")
@DTW_assign(l_salname, "Last Name")
@DTW_assign(l_samname, "Middle Name")
@DTW_assign(l_safname, "First Name")
@DTW_assign(l_sarepcom, "Company Name")
@DTW_assign(l_saaddr1, "Address (line 1)")
@DTW_assign(l_saaddr2, "Address (line 2)")
@DTW_assign(l_saaddr3, "Address (line 3)")
@DTW_assign(l_sacity, "City")
@DTW_assign(l_sastate, "State/Province")
@DTW_assign(l_sazipc, "Zip/Postal Code")
@DTW_assign(l_sacntry, "Country")
@DTW_assign(l_saphone1, "Daytime Phone Number")
@DTW_assign(l_saphone2, "Evening Phone Number")
@DTW_assign(l_sadphtyp, "Daytime Phone Type")
@DTW_assign(l_saephyp, "Evening Phone Type")
@DTW_assign(l_safax, "Fax Number")
@DTW_assign(l_saeph1st, "Phone Unlisted?")
@DTW_assign(l_saemail1, "E-mail or URL (1)")
@DTW_assign(l_saemail2, "E-mail or URL (2)")
@DTW_assign(l_shcomm, "Preferred Method of Communication")
@DTW_assign(l_sabtcall, "Best Time to Call")
@DTW_assign(l_sdage, "Age")
@DTW_assign(l_sdgendr, "Gender")
@DTW_assign(l_sdmstat, "Marital Status")
@DTW_assign(l_sdcomp, "Employer")
@DTW_assign(l_sdhhnbr, "Number of Household Members")
@DTW_assign(l_sdincom, "Annual Household Income")
@DTW_assign(l_sdintrs, "Interests/Hobbies")
```

### 5.9.3.2 MSREGFORM Include File

The example below lists the MSREGFORM.INC file:

```
<TR><TD>&nbsp;</TD></TR>

<TR>
<TD WIDTH=40><IMG SRC="/ncsample/1.gif"></TD>

<TD VALIGN=MIDDLE><FONT SIZE=+2><B>Personal Information</B></FONT></TD>
</TR><TR>

<TD></TD><TD>

<TABLE CELLSPACING=10>
<TR>
<TD>
<B><FONT SIZE=-1>$(l_shlogid)</FONT></B></TD><TD>
<A NAME="shlogid">
%if (mode_in == "modify")
<font size="+2"><B>$(shlogid)</B></font>
<INPUT Type = "hidden" Name="shlogid" Value="$(shlogid)">
%else
<INPUT TYPE="text" NAME="shlogid" VALUE="$(shlogid)" SIZE="20" MAXLENGTH="30">
%endif
</TD>
</TR>

<TR>
<TD>
<B><FONT SIZE=-1>$(l_shlpswd)</FONT></B>
</TD>
<TD>
<A NAME="shlpswd">
<INPUT TYPE="password" NAME="shlpswd" VALUE="$(shlpswd)" SIZE="20" MAXLENGTH="16">
```

```

</TD>
<TD>
  <B><FONT SIZE=-1>$(1_shlpswvder)</FONT></B>
</TD>
<TD>
  <A NAME="shlpswvder">
    <INPUT TYPE="password" NAME="shlpswvder" VALUE="$(shlpswvder)" SIZE="20" MAXLENGTH="16">
  </TD>
</TR>

<TR>

<TD><FONT SIZE=-1>$(1_shchaans)</FONT></TD><TD>
<A NAME="shchaans">
<INPUT TYPE="text" NAME="shchaans" VALUE="$(shchaans)" SIZE="20" MAXLENGTH="20">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_satitle)</FONT></TD><TD>
<A NAME="satitle">
<SELECT NAME="satitle">
  @DTW_assign(sel1, "")
  @DTW_assign(sel2, "")
  @DTW_assign(sel3, "")
  @DTW_assign(sel4, "")
  %if (satitle == null)
  @DTW_assign(sel1, "SELECTED")
  %elif (satitle == "Mr")
  @DTW_assign(sel1, "SELECTED")
  %elif (satitle == "Mrs")
  @DTW_assign(sel2, "SELECTED")
  %elif (satitle == "Ms")
  @DTW_assign(sel3, "SELECTED")
  %elif (satitle == "Dr")
  @DTW_assign(sel4, "SELECTED")
  %endif
<OPTION $(sel1) VALUE="Mr">Mr
<OPTION $(sel2) VALUE="Mrs">Mrs
<OPTION $(sel3) VALUE="Ms">Ms
<OPTION $(sel4) VALUE="Dr">Dr
</SELECT>
</TD>

<TD><B><FONT SIZE=-1>$(1_salname)</FONT></B></TD><TD>
<A NAME="salname">
<INPUT TYPE="text" NAME="salname" VALUE="$(salname)" SIZE="20" MAXLENGTH="30">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_safname)</FONT></TD><TD>
<A NAME="safname">
<INPUT TYPE="text" NAME="safname" VALUE="$(safname)" SIZE="20" MAXLENGTH="30">
</TD>

<TD><FONT SIZE=-1>$(1_samname)</FONT></TD><TD>
<A NAME="samname">
<INPUT TYPE="text" NAME="samname" VALUE="$(samname)" SIZE="20" MAXLENGTH="30">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_sarepcom)</FONT></TD><TD>
<A NAME="sarepcom">
<INPUT TYPE="text" NAME="sarepcom" VALUE="$(sarepcom)" SIZE="20" MAXLENGTH="30">
</TD>
</TR>
</TABLE>
</TD></TR><TR>

<!------->

<TD WIDTH=40><IMG SRC="/ncsample/2.gif"></TD>
<TD WIDTH=460 VALIGN=MIDDLE><FONT SIZE=+2><B>Contact Information</B></FONT></TD>
</TR><TR>

<TD></TD><TD>

<TABLE CELLSPACING=10>
<TR>

```

```

<TD><B><FONT SIZE=-1>$(1_saaddr1)</FONT></B></TD><TD>
<A NAME="saaddr1">
<INPUT TYPE="text" NAME="saaddr1" VALUE="$(saaddr1)" SIZE="40" MAXLENGTH="50">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_saaddr2)</FONT></TD><TD>
<A NAME="saaddr2">
<INPUT TYPE="text" NAME="saaddr2" VALUE="$(saaddr2)" SIZE="40" MAXLENGTH="50">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_saaddr3)</FONT></TD><TD>
<A NAME="saaddr3">
<INPUT TYPE="text" NAME="saaddr3" VALUE="$(saaddr3)" SIZE="40" MAXLENGTH="50">
</TD></TR>
</TABLE>

<TABLE CELLSPACING=10>
<TR>

<TD><B><FONT SIZE=-1>$(1_sacity)</FONT></B></TD><TD>
<A NAME="sacity">
<INPUT TYPE="text" NAME="sacity" VALUE="$(sacity)" SIZE="20" MAXLENGTH="30">
</TD>

<TD><B><FONT SIZE=-1>$(1_sastate)</FONT></B></TD><TD>
<A NAME="sastate">
<INPUT TYPE="text" NAME="sastate" VALUE="$(sastate)" SIZE="20" MAXLENGTH="20">
</TD></TR><TR>

<TD><B><FONT SIZE=-1>$(1_sazipc)</FONT></B></TD><TD>
<A NAME="sazipc">
<INPUT TYPE="text" NAME="sazipc" VALUE="$(sazipc)" SIZE="12" MAXLENGTH="20">
</TD>

<TD><B><FONT SIZE=-1>$(1_sacntry)</FONT></B></TD><TD>
<A NAME="sacntry">
<INPUT TYPE="text" NAME="sacntry" VALUE="$(sacntry)" SIZE="20" MAXLENGTH="30">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_saphone1)</FONT></TD><TD>
<A NAME="saphone1">
<INPUT TYPE="text" NAME="saphone1" VALUE="$(saphone1)" SIZE="12" MAXLENGTH="30">
</TD>

<TD><FONT SIZE=-1>$(1_saphone2)</FONT></TD><TD>
<A NAME="saphone2">
<INPUT TYPE="text" NAME="saphone2" VALUE="$(saphone2)" SIZE="12" MAXLENGTH="30">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_sadphtyp)</FONT></TD><TD>
<A NAME="sadphtyp">
<SELECT NAME="sadphtyp">
  @DTW_assign(sel1, "")
  @DTW_assign(sel2, "")
  %if (sadphtyp == null)
  @DTW_assign(sel1, "SELECTED")
  %elif (sadphtyp == "PHN")
  @DTW_assign(sel1, "SELECTED")
  %elif (saephtyp == "TTY")
  @DTW_assign(sel2, "SELECTED")
  %endif
<OPTION $(sel1) VALUE="PHN">PHN
<OPTION $(sel2) VALUE="TTY">TTY
</SELECT>
</TD>

<TD><FONT SIZE=-1>$(1_saephtyp)</FONT></TD><TD>
<A NAME="saephtyp">
<SELECT NAME="saephtyp">
  @DTW_assign(sel1, "")
  @DTW_assign(sel2, "")
  %if (saephtyp == null)
  @DTW_assign(sel1, "SELECTED")
  %elif (saephtyp == "PHN")

```











```

@DTW_assign(l_saemail2, "E-mail or URL (2)")
@DTW_assign(l_shcomm, "Preferred Method of Communication")
@DTW_assign(l_sabtcall, "Best Time to Call")
@DTW_assign(l_sdage, "Age")
@DTW_assign(l_sdgendr, "Gender")
@DTW_assign(l_sdmstat, "Marital Status")
@DTW_assign(l_sdcomp, "Employer")
@DTW_assign(l_sdhhnbr, "Number of Household Members")
@DTW_assign(l_sdincom, "Annual Household Income")
@DTW_assign(l_sdintrs, "Interests/Hobbies")

@DTW_assign(b1_shlogid, "<BLINK><I>Shopper's Login ID</I></BLINK>")
@DTW_assign(b1_shlpswd, "<BLINK><I>Password</I></BLINK>")
@DTW_assign(b1_shlpswdver, "<BLINK><I>Verify Password</I></BLINK>")
@DTW_assign(b1_shchaans, "<BLINK><I>Cics User</I></BLINK>")
@DTW_assign(b1_satitle, "<BLINK><I>Title</I></BLINK>")
@DTW_assign(b1_salname, "<BLINK><I>Last Name</I></BLINK>")
@DTW_assign(b1_samname, "<BLINK><I>Middle Name</I></BLINK>")
@DTW_assign(b1_safname, "<BLINK><I>First Name</I></BLINK>")
@DTW_assign(b1_sarepcom, "<BLINK><I>Company Name</I></BLINK>")
@DTW_assign(b1_saaddr1, "<BLINK><I>Address (line 1)</I></BLINK>")
@DTW_assign(b1_saaddr2, "<BLINK><I>Address (line 2)</I></BLINK>")
@DTW_assign(b1_saaddr3, "<BLINK><I>Address (line 3)</I></BLINK>")
@DTW_assign(b1_sacity, "<BLINK><I>City</I></BLINK>")
@DTW_assign(b1_sastate, "<BLINK><I>State/Province</I></BLINK>")
@DTW_assign(b1_sazipc, "<BLINK><I>Zip/Postal Code</I></BLINK>")
@DTW_assign(b1_sacntry, "<BLINK><I>Country</I></BLINK>")
@DTW_assign(b1_saphone1, "<BLINK><I>Daytime Phone Number</I></BLINK>")
@DTW_assign(b1_saphone2, "<BLINK><I>Evening Phone Number</I></BLINK>")
@DTW_assign(b1_sadphtyp, "<BLINK><I>Daytime Phone Type</I></BLINK>")
@DTW_assign(b1_saephtyp, "<BLINK><I>Evening Phone Type</I></BLINK>")
@DTW_assign(b1_safax, "<BLINK><I>Fax Number</I></BLINK>")
@DTW_assign(b1_saeph1st, "<BLINK><I>Phone Unlisted?</I></BLINK>")
@DTW_assign(b1_saemail1, "<BLINK><I>E-mail or URL (1)</I></BLINK>")
@DTW_assign(b1_saemail2, "<BLINK><I>E-mail or URL (2)</I></BLINK>")
@DTW_assign(b1_shcomm, "<BLINK><I>Preferred Method of Communication</I></BLINK>")
@DTW_assign(b1_sabtcall, "<BLINK><I>Best Time to Call</I></BLINK>")
@DTW_assign(b1_sdage, "<BLINK><I>Age</I></BLINK>")
@DTW_assign(b1_sdgendr, "<BLINK><I>Gender</I></BLINK>")
@DTW_assign(b1_sdmstat, "<BLINK><I>Marital Status</I></BLINK>")
@DTW_assign(b1_sdcomp, "<BLINK><I>Employer</I></BLINK>")
@DTW_assign(b1_sdhhnbr, "<BLINK><I>Number of Household Members</I></BLINK>")
@DTW_assign(b1_sdincom, "<BLINK><I>Annual Household Income</I></BLINK>")
@DTW_assign(b1_sdintrs, "<BLINK><I>Interests/Hobbies</I></BLINK>")

@DTW_assign(msg, "unknown field")
%if (field == "shlogid")
    @DTW_assign(msg, l_shlogid)
    @DTW_assign(l_shlogid, b1_shlogid)
%elif (field == "shlpswd")
    @DTW_assign(msg, l_shlpswd)
    @DTW_assign(l_shlpswd, b1_shlpswd)
%elif (field == "shlpswdver")
    @DTW_assign(msg, l_shlpswdver)
    @DTW_assign(l_shlpswdver, b1_shlpswdver)
%elif (field == "shchaque")
    @DTW_assign(msg, l_shchaque)
    @DTW_assign(l_shchaque, b1_shchaque)
%elif (field == "shchaans")
    @DTW_assign(msg, l_shchaans)
    @DTW_assign(l_shchaans, b1_shchaans)
%elif (field == "shfield1")
    @DTW_assign(msg, l_shfield1)
    @DTW_assign(l_shfield1, b1_shfield1)
%elif (field == "shfield2")
    @DTW_assign(msg, l_shfield2)
    @DTW_assign(l_shfield2, b1_shfield2)
%elif (field == "satitle")
    @DTW_assign(msg, l_satitle)
    @DTW_assign(l_satitle, b1_satitle)
%elif (field == "salname")
    @DTW_assign(msg, l_salname)
    @DTW_assign(l_salname, b1_salname)
%elif (field == "samname")
    @DTW_assign(msg, l_samname)

```

```

@DTW_assign(l_samname, b1_samname)
%elif (field == "safname")
@DTW_assign(msg,l_safname)
@DTW_assign(l_safname, b1_safname)
%elif (field == "sarepcom")
@DTW_assign(msg,l_sarepcom)
@DTW_assign(l_sarepcom, b1_sarepcom)
%elif (field == "saaddr1")
@DTW_assign(msg,l_saaddr1)
@DTW_assign(l_saaddr1, b1_saaddr1)
%elif (field == "saaddr2")
@DTW_assign(msg,l_saaddr2)
@DTW_assign(l_saaddr2, b1_saaddr2)
%elif (field == "saaddr3")
@DTW_assign(msg,l_saaddr3)
@DTW_assign(l_saaddr3, b1_saaddr3)
%elif (field == "sacity")
@DTW_assign(msg,l_sacity)
@DTW_assign(l_sacity, b1_sacity)
%elif (field == "sastate")
@DTW_assign(msg,l_sastate)
@DTW_assign(l_sastate, b1_sastate)
%elif (field == "sazipc")
@DTW_assign(msg,l_sazipc)
@DTW_assign(l_sazipc, b1_sazipc)
%elif (field == "sacntry")
@DTW_assign(msg,l_sacntry)
@DTW_assign(l_sacntry, b1_sacntry)
%elif (field == "saphone1")
@DTW_assign(msg,l_saphone1)
@DTW_assign(l_saphone1, b1_saphone1)
%elif (field == "saphone2")
@DTW_assign(msg,l_saphone2)
@DTW_assign(l_saphone2, b1_saphone2)
%elif (field == "sadphtyp")
@DTW_assign(msg,l_sadphtyp)
@DTW_assign(l_sadphtyp, b1_sadphtyp)
%elif (field == "saephtyp")
@DTW_assign(msg,l_saephtyp)
@DTW_assign(l_saephtyp, b1_saephtyp)
%elif (field == "mecfac")
@DTW_assign(msg,l_safax)
@DTW_assign(l_safax, b1_safax)
%elif (field == "saeph1st")
@DTW_assign(msg,l_saeph1st)
@DTW_assign(l_saeph1st, b1_saeph1st)
%elif (field == "saemail1")
@DTW_assign(msg,l_saemail1)
@DTW_assign(l_saemail1, b1_saemail1)
%elif (field == "saemail2")
@DTW_assign(msg,l_saemail2)
@DTW_assign(l_saemail2, b1_saemail2)
%elif (field == "shcomm")
@DTW_assign(msg,l_shcomm)
@DTW_assign(l_shcomm, b1_shcomm)
%elif (field == "sabtcall")
@DTW_assign(msg,l_sabtcall)
@DTW_assign(l_sabtcallr, b1_sabtcall)
%elif (field == "sdage")
@DTW_assign(msg,l_sdage)
@DTW_assign(l_sdage, b1_sdage)
%elif (field == "sdgendr")
@DTW_assign(msg,l_sdgendr)
@DTW_assign(l_sdgendr, b1_sdgendr)
%elif (field == "sdmstat")
@DTW_assign(msg,l_sdmstat)
@DTW_assign(l_sdmstat, b1_sdmstat)
%elif (field == "sdcomp")
@DTW_assign(msg,l_sdcomp)
@DTW_assign(l_sdcomp, b1_sdcomp)
%elif (field == "sdhhnbr")
@DTW_assign(msg,l_sdhhnbr)
@DTW_assign(l_sdhhnbr, b1_sdhhnbr)
%elif (field == "sdincom")
@DTW_assign(msg,l_sdincom)

```

```

    @DTW_assign(l_sdincom, b1_sdincom)
%elif (field == "sdintrs")
    @DTW_assign(msg,l_sdintrs)
    @DTW_assign(l_sdinstrs, b1_sdintrs)
%endif

```

### 5.9.3.4 MSVARMAP Include File

The example below lists the MSVARMAP.INC file:

```

@DTW_assign(shlogid, V_shlogid)
@DTW_assign(shlpswd, V_shlpswd)
@DTW_assign(shlpswdver, V_shlpswd)
@DTW_assign(shchaans, V_shchaans)
@DTW_assign(satitle, V_satitle)
@DTW_assign(salname, V_salname)
@DTW_assign(samname, V_samname)
@DTW_assign(safname, V_safname)
@DTW_assign(sarepcom, V_sarepcom)
@DTW_assign(saaddr1, V_saaddr1)
@DTW_assign(saaddr2, V_saaddr2)
@DTW_assign(saaddr3, V_saaddr3)
@DTW_assign(sacity, V_sacity)
@DTW_assign(sastate, V_sastate)
@DTW_assign(sazipc, V_sazipc)
@DTW_assign(sacntry, V_sacntry)
@DTW_assign(saphone1, V_saphone1)
@DTW_assign(saphone2, V_saphone2)
@DTW_assign(sadphtyp, V_sadphtyp)
@DTW_assign(saephtyp, V_saephtyp)
@DTW_assign(safax, V_safax)
@DTW_assign(saeph1st, V_saeph1st)
@DTW_assign(saemail1, V_saemail1)
@DTW_assign(saemail2, V_saemail2)
@DTW_assign(shcomm, V_shcomm)
@DTW_assign(sabtcall, V_sabtcall)
@DTW_assign(sdage, V_sdage)
@DTW_assign(sdgendr, V_sdgendr)
@DTW_assign(sdmstat, V_sdmstat)
@DTW_assign(sdcomp, V_sdcomp)
@DTW_assign(sdhhnbr, V_sdhhnbr)
@DTW_assign(sdincom, V_sdincom)
@DTW_assign(sdintrs, V_sdintrs)

```

---

## Chapter 6. Integrating Net.Commerce with MQSeries

In this chapter we take a look at an MQSeries application and see how Net.Commerce can be integrated with an MQSeries environment. This chapter is for users who are familiar with MQSeries. A good ITSO book on MQSeries is *MQSeries Three Tier Programming Examples*, SG24-4853.

---

### 6.1 General Concept

In this scenario, unlike a CICS environment, we don't have an existing application on the mainframe. All that we have made is completely new. In fact, we don't even communicate with the mainframe.

For this scenario, we do not distinguish between an AIX and a Windows NT platform. Let us assume that the architecture, which is MQSeries, is the same on all platforms and it therefore doesn't matter on what platform we implement this scenario.

---

### 6.2 Implementation Scenarios

We install two systems:

- System 1: Net.Commerce and MQSeries, either client or server, will be installed on this machine.
- System 2: MQSeries Server will always be installed on this machine.

There are two different ways in which Net.Commerce on System 1 can communicate with MQSeries on the System 2:

1. System 1 can be an MQSeries client.
2. System 1 can be another MQSeries server.

#### 6.2.1 Possible Scenario 1

In this scenario, we install an MQSeries client (MQI client) on System 1, so we have the following software installed:

On System 1:

- Net.Commerce system with DB2 and ICSS.
- MQSeries Client
- SNA support or TCP/IP support depending on the communications environment. For more detailed information see the *Multiplatform APPC Configuration Guide*, GG24-4485, which is shipped with the MQSeries product.

On System 2:

- MQSeries Server
- SNA support or TCP/IP support

A scenario like this is illustrated in Figure 153 on page 228:

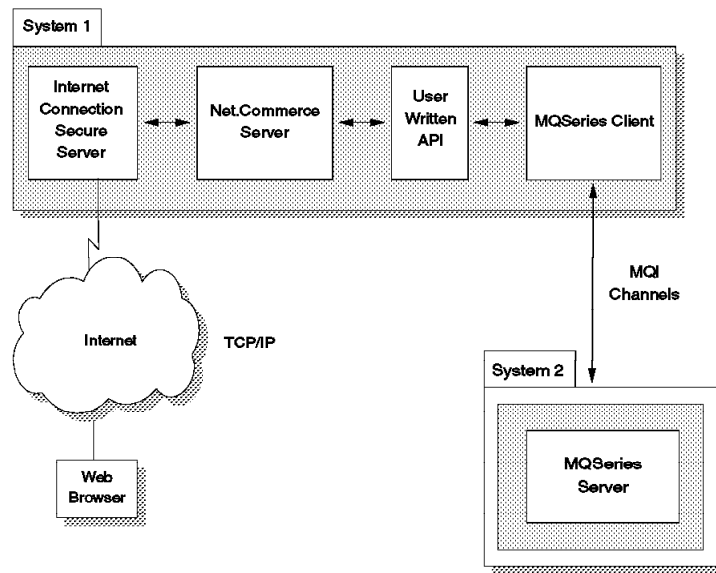


Figure 153. MQSeries Scenario 1

Remember that an MQSeries server always has a Queue Manager installed, whereas a client does not. When a client cannot connect to its server it cannot work, because the Queue Manager and the client queues reside on the server.

However, this architecture is faster for the following reasons:

1. The MQSeries client requires fewer resources than the MQSeries server.
2. The MQI channels are faster than the message channels.

## 6.2.2 Possible Scenario 2

In this scenario, we install an MQSeries server on System 1 so we have the following software installed:

On System 1:

- Net.Commerce system with DB2 and ICSS.
- MQSeries Server.
- SNA support or TCP/IP support, depending on the communications environment. For more detailed information see the *Multiplatform APPC Configuration Guide*, GG24-4485, which is shipped with the MQSeries product.

On System 2:

- MQSeries server
- SNA support or TCP/IP support

A scenario like this is illustrated in Figure 154 on page 229:

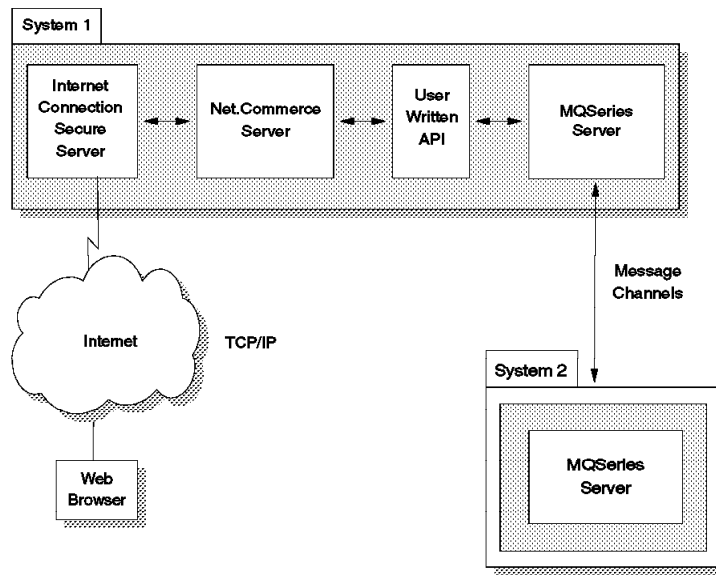


Figure 154. MQSeries Scenario 2

This architecture is less dangerous because MQSeries on System 1 is less sensitive to malfunctions of MQSeries on System 2. In this case, System 2 could be down and the Net.Commerce application will keep on working (unless the application expects some data back from System 2).

---

### 6.3 System Environment

We have chosen to implement Scenario 2, but we have added an additional system to provide a multiple machine configuration for Net.Commerce for better load distribution. This is illustrated in Figure 155 on page 230:

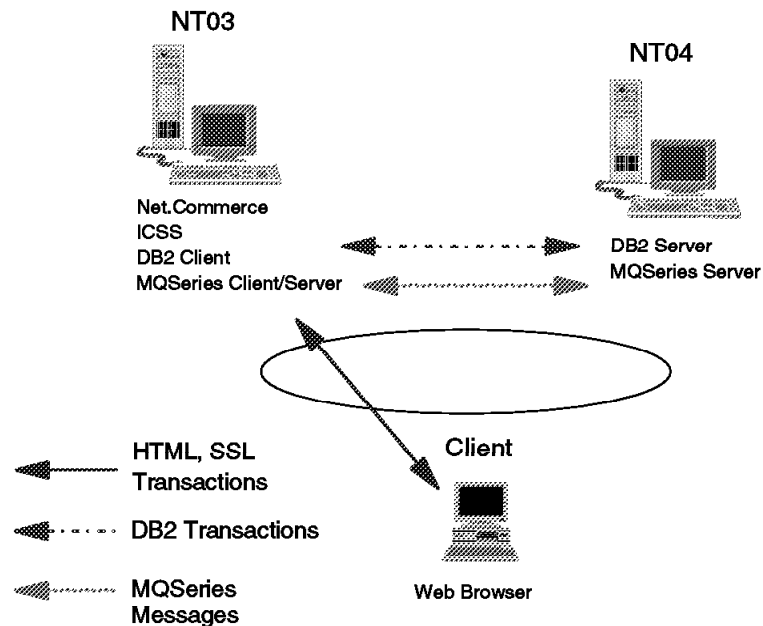


Figure 155. System Environment for Net.Commerce and MQSeries Integration

The following is the software that we need:

- One Windows NT system which we call NT03 (that corresponds to System 1) has the following software installed:
  1. NT Operating System Version 4.0
  2. Net.Commerce system with Fix Pak #1 which means in our case Net.Commerce Server with ICSS Version 4.1 and DB2 Client for NT
  3. MQSeries Server for NT Version 2.0
- Another NT system, which we call NT04 (that corresponds to System 2), with the following software installed:
  1. NT Operating System Version 4.0
  2. IBM DB2 Server for NT Version 2.1.2
  3. MQSeries Server for NT Version 2.0

### 6.3.1 Additional System Requirements

IBM MQSeries Server for NT imposes additional hardware requirements on the database and Net.Commerce systems in our scenario:

- 128 MB memory
- 128 MB page file minimum
- 65 MB of disk space for the MQSeries files (including online documentation)
- 300 MB of disk space for the Microsoft Visual C++ files (including online documentation)

For regular Net.Commerce system requirements, please refer to 3.1.5.2, “Windows NT System Requirements” on page 46.



---

## 6.4 The Application Scenario

The scenario we chose is similar to the integration with CICS, but we repeat some concepts for those who may not have read the CICS integration chapter. As in the CICS scenario, here we change the registration task in the demomall application.

We change the demo mall registration form so that there is one more field called MQSeries User. This field is associated with the column SHCHAQUE in the SHOPPER table. (Net.Commerce in Version 1.1 does not allow you to associate a new field of the registration form with a customer column, for example, SHFIELD1 or SHFIELD2.)

Our API function ensures that the values for SALNAME, SAADDR1, SACITY, SASSTATE, SACNTRY and SAZIPC are not null. Then, if the field SCHAQUE that for us is MQSeries User is not blank, it calls, by means of MQSeries, an application that resides on NT04. This application prints some data regarding the shopper to a local file.

---

## 6.5 Installing and Configuring the System

For each step we report our configuration. For detailed information on the different steps we suggest that you read the appropriate manual.

- Install MQSeries Server on NT04
- Install MQSeries Server on NT03
- Configure MQSeries Server on NT03
- Configure MQSeries Server on NT04
- Configure MQSeries on NT03 for the application
- Configure MQSeries on NT04 for the application
- Describe the MQSeries configuration on NT03
- Describe the MQSeries configuration on NT04
- Configure Net.Commerce
- Write the API/DLL
- Write the HTML EXception Macro

### 6.5.1 Installing MQSeries on NT03

Follow the instructions below to install the MQSeries server on the sender system:

1. Insert the MQSeries CD-ROM into the drive and change to the SETUP<nls> directory, where nls is the language you want to use with the server.
2. Type setup to start the setup program.
3. On the MQSeries Setup panel, check the **Update Registry** box, then click on **OK**.
4. On the installation options panel, select the components to install and the target directories, as shown in Figure 156 on page 232:

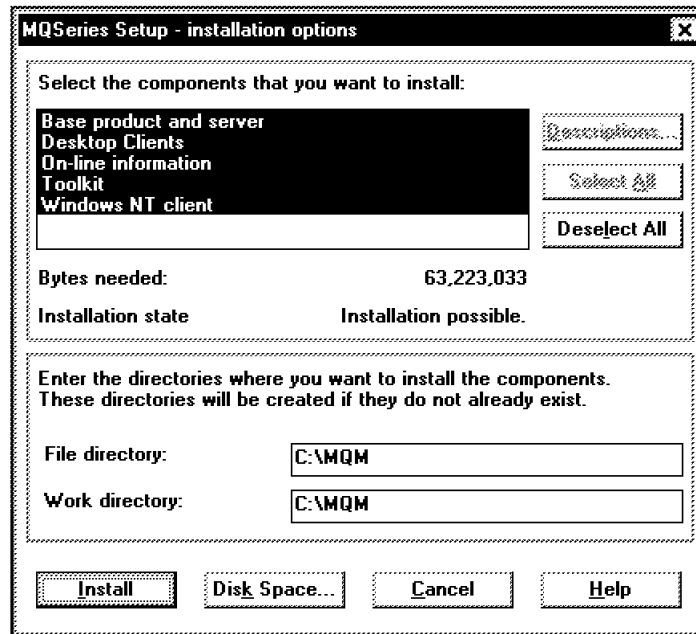


Figure 156. MQSeries Installation Options Panel

5. Click on **Install** to start the installation.
6. Restart your system when the installation has completed.

## 6.5.2 Installing MQSeries on NT04

Follow the instructions below to install the MQSeries server on the receiver system:

1. Insert the MQSeries CD-ROM into the drive and change to the SETUP<nls> directory, where nls is the language you want to use with the server.
2. Type setup to start the setup program.
3. On the MQSeries Setup panel, check the **Update Registry** box, then click on **OK**.
4. On the installation options panel, select the components to install and the target directories as shown in Figure 156:
5. Click on **Install** to start the installation.
6. Restart your system when the installation has completed.

## 6.5.3 Initial Configuration of MQSeries Server on NT03

The initial configuration of MQSeries on NT03 consists of the following steps:

1. To create a default Queue Manager, from a command prompt type `crtmqm /q WTR05252`.
2. To start the default Queue Manager, from a command prompt type `strmqm`. If you did not create the Queue Manager as default (option /q), you should type `strmqm WTR05252`.
3. To create a default MQSeries object from a command prompt type `runmqsc WTR05252< D:\MQMMQSCAMQSCOMA.TST > D:\MQSOURCE\AMQSCOMA.LOG`, where D:\MQSOURCE is your directory. Be sure that everything is okay by

checking the AMQSCOMA.LOG file; you should see at the end of the file something like the following:

```
21 MQSC commands read.  
0 commands have a syntax error.  
0 commands cannot be processed.
```

#### 6.5.4 Initial Configuration of MQSeries Server on NT04

The initial configuration of MQSeries on NT04 consists of the following steps:

1. To create a default Queue Manager, from a command prompt type `crtmqm /q WTR05305`.
2. To start the default Queue Manager, from a command prompt type `strmqm`. If you did not create the Queue Manager as default (option /q) you should have typed `strmqm WTR05305`.
3. To create a default MQSeries object, from a command prompt type `runmqsc WTR05252 < C:\MQMMQSCAMQSCOMA.TST > C:\MQSOURCE\AMQSCOMA.LOG`, where C:\MQSOURCE is your directory. Be sure that everything is okay checking the AMQSCOMA.LOG file; you should see at the end of the file something like the following:

```
21 MQSC commands read.  
0 commands have a syntax error.  
0 commands cannot be processed.
```

#### 6.5.5 Description of the MQSeries Configuration Files on NT03

Below we report the MQSeries commands that are in the file SENDER.TST which you have to create in the MQMMQSC directory.

The role of the application on the NT03 side, that is, the application that uses the Queue Manager WTR05252, is to put a message on a remote queue, which is defined as local in the Queue Manager WTR05305. So in the Queue Manager WTR05252 we have:

1. A remote queue, QRECEIVER, defined as local on the Queue Manager WTR05305, as shown in the figure below:

```
DEFINE QREMOTE('QRECEIVER') REPLACE +  
  RNAME('QRECEIVER') +  
  RQMNAME(WTR05305) +  
  XMITQ(WTR05252.TO.WTR05305) +  
  DESCR('Receiver Queue on remote system WTR05305')
```

Figure 157. MQSeries Remote Queue on NT03

2. A local transmission queue, WTR05252.TO.WTR05305, that holds the messages destined to the other system until the channel transmits them. In this way the Queue Manager can trigger the process that starts the channel program every time a message is put into the xmit queue. This is shown in the figure below:

```
DEFINE QLOCAL(WTR05252.TO.WTR05305) REPLACE +
    USAGE(xmitq) +
    TRIGGER TRIGTYPE(every) +
    INITQ(SYSTEM.CHANNEL.INITQ) +
    PROCESS(WTR05252.TO.WTR05305) +
    DESCR('Xmit queue')
```

Figure 158. MQSeries Local Transmission Queue on NT03

3. A process, WTR05252.TO.WTR05305, that tells the channel initiator what channel to start, as shown in the figure below:

```
DEFINE PROCESS(WTR05252.TO.WTR05305) REPLACE +
    USERDATA(WTR05252.TO.WTR05305) +
    DESCR('Process for starting channel')
```

Figure 159. MQSeries Channel Initiator on NT03

4. A sender channel, WTR05252.TO.WTR05305, that gets messages from the xmit queue and transmits them to the other system, as shown in the figure below:

```
DEFINE CHANNEL(WTR05252.TO.WTR05305) +
    CHLTYPE(sdr) REPLACE +
    TRPTYPE(tcp) CONNAME(9.24.104.242) +
    XMITQ(WTR05252.TO.WTR05305) +
    DESCR('Sender channel from WTR05252 to WTR05305')
```

Figure 160. MQSeries Sender Channel on NT03

where 9.24.104.242 is the TCP/IP address of the NT04 machine.

## 6.5.6 Configuring MQSeries on NT03

On NT03, from a command prompt type `runmqsc WTR05252<`  
`D:\MQSOURCE\SENDER.TST > D:\MQSOURCE\SENDER.LOG`, where `D:\MQSOURCE` is your directory. Be sure that everything is okay by checking the `SENDER.LOG` file; you should see at the end of the file something like the following:

```
4 MQSC commands read.
0 commands have a syntax error.
0 commands cannot be processed.
```

## 6.5.7 Description of the MQSeries Configuration Files on NT04

Below we report the MQSeries commands that are in the file `RECEIVER.TST` which you have to create in the `MQMMQSC` directory.

The role of the application on the NT04 side, that is the application that uses the Queue Manager `WTR05305`, is to get a message from the local queue, `QRECEIVER`. So in the Queue Manager `WTR05305` we have:

1. A local queue, `QRECEIVER`, from which the receiver application, `c:\mqsource\receiver.exe`, will get messages.
2. An initiation queue in which the Queue Manager puts a trigger message, when a message is put on the `QRECEIVER` queue. We use the

system.default.initiation.queue which is defined in the AMQSCOMA.TST file. This is shown in the figure below:

```
DEFINE QLOCAL('QRECEIVER') REPLACE +  
  TRIGGER TRIGTYPE(every) +  
  INITQ(system.default.initiation.queue) +  
  PROCESS (PROCESS.PRINT) +  
  DESCR('Take the message from other system')
```

Figure 161. MQSeries Local Queue on NT04

3. A process, PROCESS.PRINT, that contains the name of the program, c:mqsourcereceiver.exe, to be started when the trigger event occurs.

```
DEFINE PROCESS(PROCESS.PRINT) REPLACE +  
  APPLTYPE(WINDOWSNT) +  
  APPLICID('c:\mqsource\receiver.exe') +  
  DESCR('Process for application')
```

Figure 162. MQSeries Application Initiation Process on NT04

4. A receiver channel, WTR05252.TO.WTR05305, to take the message coming from the sender channel, WTR05252.TO.WTR05305, defined in the Queue Manager WTR05252.

```
DEFINE CHANNEL(WTR05252.TO.WTR05305) +  
  CHLTYPE(rcvr) REPLACE +  
  TRPTYPE(tcp) +  
  DESCR('Receiver Chane11 WTR05252.TO.WTR05305')
```

Figure 163. MQSeries Receiver Channel on NT04

## 6.5.8 Configuring MQSeries on NT04

On NT04, from a command prompt type runmqsc WTR05305< C:\MQSOURCE\RECEIVER.TST > D:\MQSOURCE\RECEIVER.LOG, where C:\MQSOURCE is your directory. Be sure that everything is okay by checking the RECEIVER.LOG file; you should see at the end of the file something like the following:

```
3 MQSC commands read.  
0 commands have a syntax error.  
0 commands cannot be processed.
```

---

## 6.6 Configuring Net.Commerce to Interface with MQSeries

For more detailed information about how to customize Net.Commerce refer to *IBM Net.Commerce: Customization Guide and Reference*, SC09-2370.

### 6.6.1 Using Your Own APIs with the System

To customize Net.Commerce to use your own API/DLLs, do the following:

1. Decide which API task to customize, for example AUDIT\_REG.
2. Understand which commands call that task, in our case ;register/modify and ;register/new. register/new command& register/new command register/modify command& register/modify command

3. Understand the default behavior of the commands:

a. `;register/modify?url=url&shlpwsd=pass&shlpswder=ver_pass`

- Modifies the shopper's registration record as follows:
  - 1) Checks the required parameters.
  - 2) Calls the AUDIT\_REG API task to check the remaining parameters. The default API function ensures that the values of the following parameters from table SHADDR are valid: SALNAME, SADDR1, SACITY, SASTATE, SACNTRY, SAZIPC.
  - 3) Modifies the registration record.

Upon successful completion, the command calls the URL specified.

- Exception conditions:
  - If a required parameter is missing or incorrect, it calls the BAD\_REG\_MODIFY exception task.
  - If the AUDIT\_REG API task returns a non-zero return code, it calls the AUDIT\_REG API exception task.
- Parameter values:
  - url: The URL that is called when the command successfully completes
  - pass: The shopper's password
  - ver\_pass: The verification password

b. `;register/new?url=url&shlogid=log_id&shlpwsd=pass&shlpswder=ver_pass`

- Creates registration information for the shopper as follows:
  - 1) Checks the required information parameters.
  - 2) Calls the AUDIT\_REG API task to check the remaining parameters. The default API function ensures that the values of the following parameters from table SHADDR are valid: SALNAME, SADDR1, SACITY, SASTATE, SACNTRY, SAZIPC.
  - 3) Updates the registration record.

Upon successful completion, the command calls the URL specified.

- Exception conditions:
  - If a required parameter is missing or incorrect, it calls the BAD\_REG\_MODIFY exception task.
  - If the AUDIT\_REG API task returns a non-zero return code, it calls the AUDIT\_\_REG API exception task.
- Parameter values:
  - url: The URL that is called when the command successfully completes.
  - log\_id: The logon user ID that the shopper chooses.
  - pass: The shopper's password.
  - ver\_pass: The verification password.

4. Understanding the API AUDIT\_REG.

a. API signature:

```
int Calling , struct APIErrorStruct * Error, const NVPairMapStruct* NVP
```

- b. Default behavior: The default function returns a return code of 0 if the following fields from table SHADDR are not blank, and a return code of 1 if one or more are blank: SALNAME, SAADDR1, SACITY, SASSTATE, SACNTRY, SAZIPC.
- c. Error codes and extra parameters:
  - 190: A required parameter was not provided. The default macro displays a message that indicates that a required field was not provided.
  - Extra parameters: field = field, is one of the fields that the API function checks. This is the field that was not provided.
5. Write your own DLL. Our DLL is called legacyMQ.dll in the directory D:MQSOURCE.
6. Decide which function of the DLL is to be called by Net.Commerce. Our function is GotoMQ.
7. Decide which exception macro Net.Commerce should use if the function returns a non-zero return code. Our exception macro is mall\_err\_legacy\_MQ.d2w and is in the directory C:IBMNETCOMMERCEMACROEN\_USDEMOMALL.
8. Use the Task Management as explained in *IBM Net.Commerce Basics: Open for Business*, SC09-2372, to let Net.Commerce take the new DLL and the new exception macro.

## 6.6.2 Using Your Macros with the System

To customize Net.Commerce to use your own macros, do the following:

1. Decide which macro task to customize, in our case REG\_UPD and REG\_NEW.
2. Understand which commands call the task ;register/form.
3. Understand the default behavior of the command:
  - The ;register/form command, for shoppers who log on as registered shoppers, calls the REG\_UPD task to display their registration page. For shoppers who are not logged on as registered shoppers, it calls the REG\_NEW task to display a blank registration page.
  - Exception conditions: None.
  - Parameter values: None.
4. The macros that we override are msregform.d2w (for REG\_NEW task) and msregupdform.d2w (for REG\_UPD task) both of which are in the C:IBMNETCOMMERCEMACROEN\_USDEMOMALL directory. Because we changed only the field MQSeries User in the page and because the macros have mall scope we do not change the name of the macros and so we do not use the Task Management utility. These two macros also include four .INC files in the directory C:IBMNETCOMMERCEMACROEN\_USNCSAMPLE:
  - msregadd.inc
  - msregform.inc
  - msreglabels.inc
  - msvarmap.inc

### 6.6.3 Running the Application without Net.Commerce

These are the instructions on how to run the application without Net.Commerce.

- On the NT03 machine (Queue Manager WTR05252), perform the following steps:
  1. From a command prompt type d: and then cd mqsource.
  2. Type copy legmq.c.test legmq.c.
  3. Compile the DLL, legacyMQ.dll, by typing nmake makemq.mak.

**Note:** To compile the DLL and the following .exe, Microsoft Visual C++ 4.2 is required.

4. Compile the testmq.exe by typing nmake testmq.mak.
5. Compile the receiver program, receiver.exe, by typing nmake makerec.mak.
6. Modify the Path environment variable by adding d:mqsource.
7. Run MQSeries by typing gosender. This .bat file contains the following two MQSeries commands:
  - strmqm (starts the MQSeries Manager WTR05252)
  - Start runmqchi (starts the channel initiator program)

- On the NT04 machine (Queue Manager WTR05305), perform the following steps:
  1. Copy the receiver.exe file to the c:mqsource directory.
  2. Run MQSeries by typing goreceiver. This .bat file contains the following three MQSeries commands:
    - strmqm (starts the MQSeries Manager WTR05252)
    - Start runmqslr -t tcp (starts a listener for the Queue Manager WTR05252)
    - Start runmqtrm (starts the trigger monitor)

- On the NT03 machine (Queue Manager WTR05252), start the application by typing testmq. If successful, you should see the following on your screen:

```
return code is 0
```

In case of an error, you should see the following:

```
return code is 1 (or whatever)
```

- On the NT04 machine (Queue Manager WTR05305), edit the file c:mqsourcmessage.log. If successful, it should look like this:

```
*****BEGINNING OF MESSAGE*****
Message received at: Tue May 13 13:10:13 1997

The message received is:
ASECCO
*****END OF MESSAGE*****
```

ASECCO is the message that we hardcoded in the file legmq.c.test.



### 6.6.3.1 Running the Application with Net.Commerce

These are the instructions on how to run the application with Net.Commerce:

- On the NT03 machine (Queue Manager WTR05252), perform the following steps:
  1. From a command prompt type `d:` and then `cd mqsource`.
  2. Type `copy legmq.c.good legmq.c`.
  3. Compile the DLL, `legacyMQ.dll` by typing `nmake makemq.mak`.  
**Note:** To compile the DLL, Microsoft Visual C++ 4.2 is required.
  4. Compile the receiver program, `receiver.exe`, by typing `nmake makerec.mak`.
  5. Modify the Path environment variable by adding `d:mqsource`.
  6. Start Net.Commerce Server, DB2, and ICSS.
  7. Copy the .inc files by typing `copy *.inc c:\ibm\netcommerce\macro\en_us\ncsample`.
  8. Copy the Exception macro file by typing `copy mall_err_reg_legacyMQ.d2w c:\ibm\netcommerce\macro\en_us\demomall`.
  9. Copy the Registration form macro file by typing `copy mallregform.d2w c:\ibm\netcommerce\macro\en_us\demomall`.
  10. Copy the Update Registration form file by typing `copy mallregupd.d2w c:\ibm\netcommerce\macro\en_us\demomall`.
  11. Create a directory `c:\ibmnetcommercehtmlen_usdemomallregistration`: (if not already created).
  12. Create a directory `c:\ibmnetcommercehtmlen_usdemomallregistrationMQ` (if not already created). In this directory the application will put some HTML error files called `MQUser.htm` where `MQUser` is the value of the field `MQUser` in the registration form.
  13. Start MQSeries by typing `strmqm` (starts the MQSeries Manager WTR05252).
  14. Run MQSeries commands by typing `gosender`. This .bat file contains the following MQSeries command:
    - Start `runmqchi` (starts the channel initiator program)
- Use the Task Management Tool of Net.Commerce by filling in the following information:
  - DLL Name: `d:mqsourcelegacyMQ.dll`
  - API Name: `GotoMQ`
  - Exception Macro: `demomallmall_err_reg_legacyMQ.d2w`
- On the NT04 machine (Queue Manager WTR05305), perform the following steps:
  1. Copy the `receiver.exe` file into the `c:mqsource` directory.
  2. Start MQSeries by typing `strmqm` (starts the MQSeries Manager WTR05305).
  3. Run MQSeries commands by typing `goreceiver`.

This .bat file contains the following two MQSeries commands:

- Enter runmq1sr -t tcp to start a listener for the messages that arrive from WTR05252).
- Start runmqtrm (starts the trigger monitor).
- From a Web browser, begin the registration process.
  - Type <http://wtr05252/demomail/basemall.htm> to get to the demo mail home page.
  - Scroll down and click on **register**.
  - Fill in the fields as shown in the figure below and click on **Submit Registration**.

The screenshot shows a web browser window with the URL <https://wtr05252.itsou.ral.ibm.com/cgi-bin/rph-msrvr/register/form>. The page is titled "1 Personal Information" and contains the following fields:

Shopper's Login ID	<input type="text" value="MAUSO"/>
Password	<input type="password"/>
Verify Password	<input type="password"/>
MQ User	<input type="text" value="MQMAUSO"/>
Title	<input type="text" value="Mr"/>
Last Name	<input type="text" value="SETE"/>
First Name	<input type="text"/>
Middle Name	<input type="text"/>
Company Name	<input type="text"/>

The page is titled "2 Contact Information" and contains the following fields:

Address (line 1)	<input type="text" value="ADDR"/>
Address (line 2)	<input type="text"/>
Address (line 3)	<input type="text"/>
City	<input type="text" value="CITY"/>
State/Province	<input type="text" value="STATE"/>
Zip/Postal Code	<input type="text" value="23869"/>
Country	<input type="text" value="COUNTRY"/>

Figure 164. Registration Form for MQSeries Scenario

- The figure below shows that the registration was successful:



Figure 165. Successful Registration with MQSeries

- On the NT04 machine (Queue Manager WTR05305), edit the file c:\mqsourcemessage.log. If registration was successful, it should look like this:

```
*****BEGINNING OF MESSAGE*****
Message received at: Wed May 14 18:17:15 1997

The message received is:
MQUser      MQMAUSO      Logid      MAUSO
LastName    SETE           Address    ADDR
City        CITY          ZipCode    23869
State       STATE        Country    COUNTRY
*****END OF MESSAGE*****
```

### 6.6.4 What Happens in the Case of an MQSeries Error

This section shows simple error management in the case of MQSeries errors. To simulate an MQSeries error on NT03 (WTR05252 Queue Manager) type `endmqm /i WTR05252`. This will stop the WTR05252 Queue Manager.

Now you are ready to simulate a failed registration process.

1. Fill in the fields as shown in the figure below and click on **Submit Registration**.

Location: https://wtr05252.lso.ral.ibm.com/cgi-bin/rph-mervr/register/loam

### 1 Personal Information

Shopper's Login ID: MAUS01

Password:  Verify Password:

MQ User: MQMAUS01

Title: Mr Last Name: SETE1

First Name:  Middle Name:

Company Name:

### 2 Contact Information

Address (line 1): ADDR1

Address (line 2):

Address (line 3):

City: CITY1 State/Province: STATE1

Zip/Postal Code: 913483 Country: COUNTRY1

Figure 166. Registration Form for MQSeries Error Scenario

2. The figure below shows that there has been an error. That means that the API returned a value other than 0 and Net.Commerce has called the exception macro.



Figure 167. Exception Macro for MQSeries

3. If you press **click here**, you will see the reason for the MQSeries error.



Figure 168. MQSeries User Error Page

4. If you press **click here to make a new registration again**, you will see the registration form again.

## 6.7 The MQSeries API Program and Customized Net.Commerce Macros

This section lists the program code for the API and the associated macros to facilitate a sample integration of Net.Commerce and MQSeries.

### Notes:

1. The source files can be found on the diskette that comes with this publication.
2. In order to fit the code listings on the page, some lines have been wrapped.

### 6.7.1 Program Source Code

The following files constitute the source code of the legacyMQ.DLL API which is called by Net.Commerce to interact with MQSeries. There are C source files (.C), header files (.H) and the make files for use with the Microsoft Visual C++ compiler (.MAK).

#### 6.7.1.1 LEGMQ Function

The example below lists the LEGMQ.C file.

```

/* This file contains the function GotoMQ the Net.Commerce will call
   when it will Execute AUDIT_REG TASK */
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <assert.h>
#include <time.h>

#if defined(WIN32)
#include <windef.h>
#include <stdarg.h>
#include <winbase.h>
#endif

#include <nvpair_util.h> /*include for Net.Commerce*/
#include <commands.h> /*include for Net.Commerce*/
#include "mqfunc.h"
#include "auxi.h"

void format(char* aMessageMQ,const char* aMQUser,const char* aLogid,const char*
           aSaName, const char* aSaAddr1,const char* aSaCity, constchar* aSaZipc,
           const char* aSaState,const char* aSaCntry);
void printHtmlError( char* aBuffer, const char * aCicsUser, int aCallingCommand,
                   const char* aUrl);

#if defined(WIN32)
__declspec( dllexport )
#endif

int GotoMQ(int CallingCommand, struct APIErrorStruct* Error, struct NVPairMapStruct*
          NVP)
{
    char* userPageHtml;
    FILE *filePointer;
    short rc=0;
    const char* iLogid;
    const char* iPasswd;
    const char* iMQUser;
    const char* iSaName;
    const char* iSaAddr1;
    const char* iSaCity;
    const char* iSaState;
    const char* iSaCntry;
    const char* iSaZipc;
    const char* iUrl;

```

```

char errInformation[SIZE_ERROR];
char messageMQ[SIZE_MESSAGE];
char tmpUser[80];

iLogid=NVP_GetValue(NVP, "shlogid");
iPasswd=NVP_GetValue(NVP, "shlpswd");
iMQUser = NVP_GetValue(NVP, "shchaans");
iSalName = NVP_GetValue(NVP, "salname");
iSaAddr1 = NVP_GetValue(NVP, "saaddr1");
iSaCity = NVP_GetValue(NVP, "sacity");
iSaState = NVP_GetValue(NVP, "sastate");
iSaCntry = NVP_GetValue(NVP, "sacntry");
iSaZipc = NVP_GetValue(NVP, "sazipc");
iUrl=NVP_GetValue(NVP, "cancel_url");
/*my check*/
filePointer=fopen("d:\\cicssource\\testcics.txt", "w");
fprintf(filePointer,"schaans = %s, salname= %s, saaddr1= %s, sacity= %s,
sastate= %s, sacntry= %s, sazipc=%s \n", iMQUser,iSalName,
iSaAddr1,iSaCity,iSaState,iSaCntry,iSaZipc );
fprintf(filePointer,"url = %s",iUrl);
fprintf(filePointer,"login = %s,pwd = %s \n", iLogid, iPasswd);
fclose(filePointer);
/*end of my check*/

// checking the default fields
if (( *iSalName == '\0' )||
( *iSaAddr1 == '\0' )||
( *iSaCity == '\0' )||
( *iSaState == '\0' )||
( *iSaCity == '\0' )||
( *iSaZipc == '\0' ))

{
AES_AppendNVPString(Error, "error_code", "290");
AES_AppendNVPString(Error, "API", "AUDIT_REG");
AES_AppendNVPString(Error, "REASON" , "REQUIRED PARAMETERS MISSING");
return 1;
}

if (iMQUser == '\0')
return 0;
else
{
/* call the MQSeries function*/
format(messageMQ,iMQUser,iLogid,iSalName,iSaAddr1,iSaCity,iSaZipc,
iSaState,iSaCntry);
rc=goToMQ(errInformation,messageMQ );
if (rc!=0)
{
strcpy(tmpUser,iMQUser);
userPageHtml=strcat(tmpUser, ".htm");
AES_AppendNVPString(Error, "error_code", "300");
AES_AppendNVPString(Error, "API", "AUDIT_REG");
AES_AppendNVPString(Error, "REASON" , "MQ REGISTRATION
PROGRAM ABORTED");
AES_AppendNVPString(Error, "USERPAGE" , userPageHtml);
AES_AppendNVPString(Error, "HOST" ,
"/demomail/registration/MQ");
printHtmlError(errInformation,iMQUser,CallingCommand,iUrl);
}
return rc;
}
}
return rc;
}

}

void printHtmlError( char* aBuffer, const char * aMQUser, int aCallingCommand,
const char* aUrl)
{
/* In case of error the function write a File aMQUser.htm so
the responsible for Net.commerce Administrator can see the error*/
FILE* f;
char fileDirectory[80];
char relFileName[80];
char* fileName;
char* relHtmlFileName;

```

```

struct tm when;
time_t now;
time( &now );
when = *localtime( &now );

strcpy(fileDirectory,"c:\\IBM\\NetCommerce\\html\\En_us\\demoma11
\\registration\\MQ\\");
strcpy(relFileName,aMQUser);
relHtmlFileName=strcat(relFileName, ".htm");
fileName=strcat(fileDirectory,relHtmlFileName);
f=fopen(fileName,"w");
fprintf(f,"<html> \n");
fprintf(f,"<head> \n");
fprintf(f,"</head> \n");
fprintf(f,"<h2> ");
fprintf(f,"ERROR PAGE for MQ USERS \n");
fprintf(f,"</h2> \n");
fprintf(f,"<body> \n");
fprintf(f, "Current time is %s\n", asctime( &when ) );
fprintf(f,"<BR>\n");
fprintf(f,"REASON OF ERROR FOR USER  %s is",aMQUser);
fprintf(f,"<BR>\n");
fprintf(f,"<P>\n");
fprintf(f,"%s \n",aBuffer);
fprintf(f,"<BR>\n");
fprintf(f,"<A HREF=");
fprintf(f,"%c",''');
fprintf(f,"/cgi-bin/nph-msrvr/register/form");
fprintf(f,"%c",''');

if (aCallingCommand == _REGISTER_NEW_)
{
    fprintf(f,"> click here to make a new registration again </A>. \n");
}
else if (aCallingCommand == _REGISTER_MODIFY_)
{
    fprintf(f,"> click here to Update your registration again</A>. \n");
}
fprintf(f,"<BR>\n");
if (*aUrl == '\0'){
else
{
    fprintf(f,"<A HREF=");
    fprintf(f,"%c",''');
    fprintf(f,"%s",aUrl);
    fprintf(f,"%c",''');
    fprintf(f,"> click here return to your original page </A>. \n");
    fprintf(f,"<BR>\n");
}
fprintf(f,"</body> \n");
fprintf(f,"</html> \n");
fclose(f);
}

void format(char* aMessageMQ,const char* aMQUser,const char* aLogid,const char*
aSa1Name, const char* aSaAddr1,const char* aSaCity,const char*
aSaZipc, const char* aSaState,const char* aSaCntry)
{
    /* this function format the Message to be sent to MQSeries*/

    sprintf(aMessageMQ,"MQUser \t %s \t Logid \t\t %s \n LastName \t %s \t Address
\t %s \n City \t\t %s \t ZipCode \t %s \n State \t %s \t Country
\t %s \n", aMQUser,aLogid,aSa1Name,aSaAddr1,aSaCity,aSaZipc,aSaState,
aSaCntry);
}

```

The example below lists the LEGMQ.H file.

```

#ifdef LEGMQ_H
#define LEGMQ_H
int GotoMQ(int CallingCommand, struct APIErrorStruct* Error, struct NVPairMapStruct*
NVP);
#endif

```



### 6.7.1.2 MQFUNC Function

The example below lists the MQFUNC.C file.

```
#include <stdio.h>
#include "sender.h"

short goToMQ(char * aErrInformation,const char* aMessageMQ)
{
/* This Function calls the Sender Code*/
short rc=0;

rc=doSender(aErrInformation,aMessageMQ);
if (rc!=0)
return rc;

return rc;
}
```

The example below lists the MQFUNC.H file.

```
#ifndef _MQFUNC_H
#define _MQFUNC_H

short goToMQ(char * aErrInformation,const char* aMessageMQ);

#endif
```

### 6.7.1.3 SENDER Function

The example below lists the SENDER.C file.

```
/* This file contains the code for the MQseries; The application
Sender PUT a message in the queue(QReceiver)*/

#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <process.h>
#include <cmqc.h>
#include "sender.h"
#include "auxi.h"

short doSender (char* aErrInformation, const char* aMessageMQ)
{
/* this function execute the MQseries PUT of the aMessageMQ*/

MQLONG compCode, reason;
MQHCONN hCon; // Connection handle
MQHOBJ hQReceiver; // Object handle for Q 1
MQOD od1 = {MQOD_DEFAULT}; // Object descriptor for Q 1
MQMD md = {MQMD_DEFAULT}; // Message descriptor
MQPMO pmo = {MQPMO_DEFAULT}; // Put message options
char buffer[SIZE_MESSAGE] ="" ; // message buffer
short rc=0;
char* name;
char qManager[20];

name="doSender-MQHCONN";
strcpy(qManager,"WTR05252");
MQCONN (qManager,&hCon,&compCode,&reason);
if (compCode!=MQRC_NONE)
{
response (name,compCode,reason,aErrInformation);
rc=1;
return (rc) ;
}
}
```

```

name="doSender-MQOPEN";
strcpy (od1.ObjectName,"QRECEIVER");
MQOPEN (hCon,&od1,MQOO_OUTPUT,&hQReceiver,&compCode, &reason);
if (compCode!=MQRC_NONE)
{
    response (name,compCode,reason,aErrInformation);
    MQDISC (&hCon, &compCode, &reason);
    response (name,compCode,reason,aErrInformation);
    rc=1;
    return (rc) ;
}

name="doSender-MQPUT";
strcpy (buffer,aMessageMQ);
MQPUT (hCon,hQReceiver,&md,&pmo,SIZE_MESSAGE,&buffer,&compCode,&reason);
if (compCode!=MQRC_NONE)
{
    response (name,compCode,reason,aErrInformation);
    MQDISC (&hCon, &compCode, &reason);
    response (name,compCode,reason,aErrInformation);
    rc=1;
    return (rc) ;
}

name="doSender-MQCLOSE";
MQCLOSE (hCon, &hQReceiver, MQCO_NONE, &compCode, &reason);
if (compCode!=MQRC_NONE)
{
    response (name,compCode,reason,aErrInformation);
    MQDISC (&hCon, &compCode, &reason);
    response (name,compCode,reason,aErrInformation);
    rc=1;
    return (rc) ;
}

name="doSender-MQDISC";
MQDISC (&hCon, &compCode, &reason);
if (compCode!=MQRC_NONE)
{
    response (name,compCode,reason,aErrInformation);
    rc=1;
    return (rc) ;
}

return rc;
}

```

The example below lists the SENDER.H file.

```

#ifndef _SENDER_H
#define _SENDER_H

short doSender (char* aErrInformation, const char* aMessageMQ);

#endif

```

#### 6.7.1.4 AUXI Function

The example below lists the AUXI.C file.

```

/* This file contains auziliary function*/

#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <process.h>
#include "auxi.h"

```

```
/******AUXILIAR FUNCTIONS******/
void response (char *aCall, MQLONG aCompCode, MQLONG aReason, char* aErrInformation)
{
    sprintf (aErrInformation,"%s ended: Completion code: %d \n      Reason: %d",
            aCall, aCompCode, aReason);
} /* response */
```

The example below lists the AUXI.H file.

```
#ifndef __AUXILIAR_H
#define __AUXILIAR_H

#include <stdio.h>
#include <string.h>
#include <cmqc.h>
#define SIZE_MESSAGE      2000
#define SIZE_ERROR        1000
/******AUXILIAR FUNCTIONS******/
void response (char *aCall, MQLONG aCompCode, MQLONG aReason, char* aErrInformation);

#endif
```

### 6.7.1.5 MAKE File (Sender)

The example below lists the MAKEMQ.MAK file.

```
CFLAGS= /nologo /MT /W3 /GX /Zi /O2 /D "WIN32" /D "_X86_" /D "DEBUG" /YX /c
CPP= cl
ERASE= del
COPY = copy

INCLUDE001=c:\ibm\netcommerce\adt;
INCLUDE002=$(INCLUDE001);d:\mqm\tools\c\include;
INCLUDE=$(INCLUDE);$(INCLUDE002)
LINK32=link

LINK32_FLAGS= kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib\
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib\
odbc32.lib db2api.lib db2cli.lib oldnames.lib d:\mqm\tools\lib\mqm.lib /nologo
/subsystem:windows /dll\
/incremental:no /machine:I386\
/out:"legacyMQ.dll" /implib:"legacyMQ.lib"

all: legacyMQ.dll

legacyMQ.dll: nvpair_util.obj legmq.obj sender.obj mqfunc.obj auxi.obj
$(LINK32) nvpair_util.obj legmq.obj sender.obj mqfunc.obj auxi.obj \
$(LINK32_FLAGS)

nvpair_util.obj : c:\ibm\netcommerce\adt\nvpair_util.c
$(CPP) $(CFLAGS) /TP c:\ibm\netcommerce\adt\nvpair_util.c

legmq.obj : legmq.c
$(CPP) $(CFLAGS) legmq.c

mqfunc.obj : mqfunc.c
$(CPP) $(CFLAGS) mqfunc.c

sender.obj : sender.c
$(CPP) $(CFLAGS) sender.c

auxi.obj : auxi.c
$(CPP) $(CFLAGS) auxi.c
```

### 6.7.1.6 RECEIVER Function

The example below lists the RECEIVER.C file.

```

/* This file contains the code for the MQseries; The application
Receiver GET the Message from the QUEUE (QReceiver)*/

#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <process.h>
#include <cmqc.h>
#include <time.h>

#define SIZE_MESSAGE 2000

void response (char *aCall, MQLONG aCompCode, MQLONG aReason);
void printMsg(const char* aMessage);

int main(void)
{
    /* this function execute the MQseries PUT of the aMessageMQ*/

    MQLONG          compCode, reason;
    MQHCONN         hCon;                // Connection handle
    MQHOBJ          hQReceiver;         // Object handle for Q 1
    MQOD            od1 = {MQOD_DEFAULT}; // Object descriptor for Q 1
    MQMD            md = {MQMD_DEFAULT}; // Message descriptor
    MQGMO           gmo = {MQGMO_DEFAULT}; // Get message options
    MQLONG          options;
    MQLONG          dataLength=SIZE_MESSAGE;
    MQLONG          messageLength=SIZE_MESSAGE;
    char            message[SIZE_MESSAGE] ="" ; // message buffer
    char*           name;
    short rc=0;
    char            qManager[20];

    name="MQCONN";
    strcpy(qManager,"WTR05305");
    MQCONN (qManager, &hCon, &compCode, &reason);
    if (compCode!=MQRC_NONE)
    {
        response (name, compCode, reason);
        rc=1;
        return rc;
    }
    name="MQOPEN";
    options = MQOO_INPUT_AS_Q_DEF;
    strcpy (od1.ObjectName, "QRECEIVER");
    MQOPEN (hCon, &od1, options, &hQReceiver, &compCode, &reason);
    if (compCode!=MQRC_NONE)
    {
        response (name, compCode, reason);
        MQDISC (&hCon, &compCode, &reason);
        response (name, compCode, reason);
        rc=1;
        return rc;
    }

    name="MQGET";
    gmo.Options = MQGMO_NO_WAIT;
    memcpy (md.MsgId, MQMI_NONE, sizeof(md.MsgId));
    memset (md.CorrelId, 0x00, sizeof(MQBYTE24));
    MQGET (hCon, hQReceiver, &md, &gmo, messageLength, message,
           &dataLength, &compCode, &reason);
    if (reason == MQRC_NO_MSG_AVAILABLE)
    {
        response (name, compCode, reason);
        MQDISC (&hCon, &compCode, &reason);
        response (name, compCode, reason);
        rc=0;
        return rc;
    }
    else

```

```

    {
        if (compCode!=MQRC_NONE)
        {
            response (name,compCode,reason);
            MQDISC (&hCon, &compCode, &reason);
            response (name,compCode,reason);
            rc=1;
            return rc;
        }
    }

    name="MQCLOSE";
    MQCLOSE (hCon, &hQReceiver, MQCO_NONE, &compCode, &reason);
    if (compCode!=MQRC_NONE)
    {
        response (name,compCode,reason);
        MQDISC (&hCon, &compCode, &reason);
        response (name,compCode,reason);
        rc=1;
        return rc;
    }

    name="MQDISC";
    MQDISC (&hCon, &compCode, &reason);
    if (compCode!=MQRC_NONE)
    {
        response (name,compCode,reason);
        rc=1;
        return (rc) ;
    }

    printMsg(message);
    return rc;
}

void response (char *aCall, MQLONG aCompCode, MQLONG aReason)
{
    FILE* f;

    f=fopen("c:\\mqsource\\err.log","w");

    fprintf (f,"%s ended: Completion code: %d \n      Reason: %d",
            aCall, aCompCode, aReason);
    fclose(f);
} /* response */

void printMsg(const char* aMessage)
{
    /* This function write the Message in a specified Format*/

    FILE* f;
    struct tm when;
    time_t now;
    time( &now );
    when = *localtime( &now );

    f=fopen("c:\\mqsource\\message.log","a+");
    fprintf(f,"*****BEGINNING OF MESSAGE*****\n");
    fprintf(f,"Message received at: %s\n", asctime( &when ));
    fprintf (f,"The message received is:\n");
    fprintf (f,"%s",aMessage);
    fprintf(f,"*****END OF MESSAGE*****\n");
    fclose(f);
}

```

### 6.7.1.7 MAKE File (Receiver)

The example below lists the MAKEREC.MAK file.

```
CFLAGS= /nologo /MT /W3 /GX /Zi /O2 /D "WIN32" /D "_X86_" /D "DEBUG" /YX /c
CPP= cl
ERASE= del
COPY = copy

INCLUDE002=$(INCLUDE001);d:\mqm\tools\c\include;
INCLUDE=$(INCLUDE);$(INCLUDE002)
LINK32=link

LINK32_FLAGS= d:\mqm\tools\lib\mqm.lib /nologo\
/incremental:no /machine:I386\

all: receiver.exe

receiver.exe: receiver.obj
    $(LINK32) receiver.obj \
    $(LINK32_FLAGS)
```

## 6.7.2 Macro Definitions

The following files constitute the source code of the Net.Commerce (Net.Data) macros that have to be used in conjunction to the legacyMQ.DLL API.

### 6.7.2.1 Register Form

The example below lists the MALLREGFORM.D2W file.

```
%{=====
The sample Templates, HTML and Macros are furnished by IBM as simple
examples to provide an illustration. These examples have not been
thoroughly tested under all conditions. IBM, therefore, cannot guarantee
imply reliability, serviceability or function of these programs. All
programs contained herein are provided to you "AS IS".

The sample Templates, HTML and Macros may include the names of individuals,
companies, brands and products in order to illustrate them as completely as
possible. All of these are names are fictitious and any similarity to the names
and addresses used by an actual business enterprise is entirely coincidental.

(C) Copyright IBM Corp. 1995, 1996.
=====}%

%define {
    url_in = url ? "${url}" : ""
}

%}

%SQL(MallInfo){

    select mhthead, mhtfoot, mhtbase
    from mall

    %SQL_REPORT{
        %ROW{
            @DTW_assign( MallHeader, V_mhthead)
            @DTW_assign( MallFooter, V_mhtfoot)
            @DTW_assign( MallBase, V_mhtbase)
        }
    }

    %}

%SQL_MESSAGE{
    default: { %} :continue %}

%}

%SQL(Get_Name) {
    select shlogid, satitle, safname, samname, salname
    from shopper, shaddr
```

```

where (
  sanick = '${(SESSION_ID)}' and shlogid='${(SESSION_ID)}' and
  (shshtyp = 'R' or shshtyp = 'A')
  and sashnbr=shrfnbr and saadrflg='P'
)

%SQL_REPORT {
  %ROW{
    %if (V_satitle != "NULL")
      $(V_satitle).
    %endif
    %if (V_safname != "NULL")
      $(V_safname)
    %endif
    %if (V_samname != "NULL")
      $(V_samname)
    %endif
    %if (V_salname != "NULL")
      $(V_salname),
    %endif
    you registered with a Login ID of <B>$(V_shlogid)</B>.<BR>
    Remember to write down your Login ID and Password in a safe place.</td>

  %}
%}

%SQL_MESSAGE{
  default: {
    ERROR
  }
%}

%}

%HTML_INPUT{
<HTML>

%EXEC_SQL(MallInfo)

<HEAD>
  <META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
  <TITLE>Registration Information</TITLE>
</HEAD>

<BODY BACKGROUND="/ncsample/back1.gif">

  %include "$(MallHeader)"

  <p align=center><font size=5><strong>Registration is successful</strong></font></p>

  <CENTER>
    <TABLE WIDTH=80%>
      <TR>
        <TD align=center width=50></td>
        <TD>
          %EXEC_SQL(Get_Name)
        </TD>
        <TD align=center width=50></td>
      </TR>
    </TABLE>

    %if (url_in == "")
      Proceed to the <A HREF="/cgi-bin/nph-msrvr/;execmacro/mall_dir.d2w/report">
        Mall Directory</A>.
    %else
      <FORM ACTION="$(url_in)">
        <INPUT TYPE="submit" VALUE="Continue">
      </FORM>
    %endif
  </CENTER>

  %include "$(MallFooter)"

```

```

</BODY>

</HTML>
%}

%HTML_REPORT{

<HTML>

%EXEC_SQL(MallInfo)

<HEAD>
  <META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
  <TITLE>Registration Information</TITLE>
</HEAD>

<BODY BACKGROUND="/ncsample/back1.gif">

  %include "$(MallHeader)"

  <CENTER>

    <FORM method=POST action="/cgi-bin/nph-msrvr/register/new">
      <INPUT Type = "hidden" Name="url" Value="/msprotect/nph-msrvr/execmacro/
        mallregform.d2w/input?mode=new&url=${url_in}" >
      <INPUT Type = "hidden" Name="mode" Value="new" >
      <INPUT Type = "hidden" Name="cancel_url" Value="${url_in}" >

      <p align=center><font size=5><strong>New Registration</strong></font></p>

      <TABLE WIDTH=80%>

        <TR>
          <td align=center width=50></td>
          <td>To register as a new shopper, fill in the form below.
            Items listed in <b>bold</b> must not be left blank.
            All information will be kept strictly confidential by the mall and store
            owners.
            We will not communicate this information to others as part of a mailing
            list.</td>

          <td align=center width=50>
          </td>
        </TR>

        %INCLUDE "/ncsample/msregadd.inc"
        %INCLUDE "/ncsample/msregform.inc"

        <TR>
          <TD WIDTH=40><IMG SRC="/ncsample/4.gif"></TD>
          <TD WIDTH=460 VALIGN=MIDDLE><FONT SIZE=+2><B>Submit Your Registration</B>
          </FONT></TD>
        </TR>

        <TR>
          <TD></TD>
          <TD>
            You are now ready to submit your registration for approval.
            To submit your registration, click the Submit Registration button.
            To cancel your registration, click the back arrow button below.
            <TABLE CELLSPACING=10>
              <TR>
                <TD>
                  <IMG SRC="/ncsample/hoption.gif">
                </TD>
                <TD>
                  <INPUT TYPE="submit" VALUE="Submit Registration">
                </TD>
              </TR>
              <TR>
                <TD>
                  <IMG SRC="/ncsample/hoption.gif">
            </TABLE>
          </TD>
        </TR>
      </FORM>
    </CENTER>
  </BODY>
</HTML>
%}

```



```

        </TD>
        <TD>
</FORM>
        %if (url_in == "")
        <A HREF="$(MallBase)"><IMG SRC="/ncsample/cancel.gif" border=0></A>
        %else
        <A HREF="$(url_in)"><IMG SRC="/ncsample/cancel.gif" border=0></A>
        %endif
    </TD>
</TR>
</TABLE>
</TD>
</TR>
</TABLE>
</CENTER>

%include "$(MallFooter)"

</BODY>

</HTML>
%}

```

### 6.7.2.2 Register Update

The example below lists the MALLREGUPD.D2W file.

```

%{=====
The sample Templates, HTML and Macros are furnished by IBM as simple
examples to provide an illustration. These examples have not been
thoroughly tested under all conditions. IBM, therefore, cannot guarantee
imply reliability, serviceability or function of these programs. All
programs contained herein are provided to you "AS IS".

The sample Templates, HTML and Macros may include the names of individuals,
companies, brands and products in order to illustrate them as completely as
possible. All of these are names are fictitious and any similarity to the names
and addresses used by an actual business enterprise is entirely coincidental.

(C) Copyright IBM Corp. 1995, 1996.
=====}%

%define {
    url_in = url ? "$(url)" : ""
    mode_in = "modify"
}%

%SQL(MallInfo){

    select mhthead, mhtfoot, mhtbase from mall

    %SQL_REPORT{
        %ROW{
            @DTW_assign( MallHeader, V_mhthead)
            @DTW_assign( MallFooter, V_mhtfoot)
            @DTW_assign( MallBase, V_mhtbase)
        }
    }

    %SQL_MESSAGE{
        default: { %} :continue %}
    }

%}

%SQL (DO_UPDATE) {

    with selshaddr as
    (select saeph1st, satitle, salname, safname, samname, sarepcom, saphone1,
        saphone2, safax, saaddr1, saaddr2, saaddr3, sacity, sastate,
        sacntry, sazipc, saemail1, saemail2, sadphtyp, saephtyp, sabtcall,
        sashnbr
    from shaddr
    where sanick='$(SESSION_ID)' and saadrflg='P'),

```

```

selshopper as
(select saeph1st, satitle, salname, safname, samname, sarepcom, saphone1,
    saphone2, safax, saaddr1, saaddr2, saaddr3, sacity, sastate,
    sacntry, sazipc, saemail1, saemail2, sadphtyp, saephtyp, sabcall,
    shlogid, shlpwd, shstyp, shcomm, shchaans, shrfnbr
from shopper, selshaddr
where shrfnbr=sashnbr and shlogid='$(SESSION_ID)' )

select saeph1st, satitle, salname, safname, samname, sarepcom, saphone1,
    saphone2, safax, saaddr1, saaddr2, saaddr3, sacity, sastate,
    sacntry, sazipc, saemail1, saemail2, sadphtyp, saephtyp, sabcall,
    shlogid, shlpwd, shstyp, shcomm, shchaans, sdgendr, sdage, sdincom,
    sdmstat, sdhnbr, sdcomp, sdintrs
from shopdem, selshopper
where shrfnbr=sdshnbr

%SQL_REPORT{

%ROW{

<form method=POST action="/cgi-bin/nph-msrvr/register/modify">
<INPUT Type = "hidden" Name="url" Value="/cgi-bin/nph-msrvr/execmacro/
    mallregupd.d2w/input?mode=modify&url=$(url_in)" >
<INPUT Type = "hidden" Name="mode" Value="modify" >
<INPUT Type = "hidden" Name="cancel_url" Value="$(url_in)" >

<p align=center><font size=5><strong>Change Registration</strong></font></p>
<CENTER>
<TABLE WIDTH=80%>

<TR>
<td align=center width=50></td>
<td>To change your registration information, simply update the form below.
    Items listed in <b>bold</b> must not be left blank.
    All information will be kept strictly confidential by the mall and store
    owners.
    We will not communicate this information to others as part of a mailing
    list.</td>

<td align=center width=50>
</td></tr>

%INCLUDE "/ncsample/msvarmap.inc"
%INCLUDE "/ncsample/msreglabels.inc"
%INCLUDE "/ncsample/msregform.inc"

<TR>
<TD WIDTH=40><IMG SRC="/ncsample/4.gif"></TD>

<TD WIDTH=460 VALIGN=MIDDLE><FONT SIZE=+2><B>Submit Your Changes</B>
</FONT></TD>

</TR><TR>

<TD></TD><TD>

You are now ready to change your registration.
To change your registration, click the Change Registration button.
To cancel your changes, click the back arrow button below.
<TABLE CELLPADDING=10>
<TR>
<TD><IMG SRC="/ncsample/hoption.gif"></TD><TD>
<INPUT TYPE="submit" VALUE="Change Registration">
</TD>
</TR>
<TR>
<TD><IMG SRC="/ncsample/hoption.gif"></TD><TD>
</FORM>

%if (url_in == "")
<A HREF="$(MallBase)"><IMG SRC="/ncsample/cancel.gif" border=0></A>
%else
<A HREF="$(url_in)"><IMG SRC="/ncsample/cancel.gif" border=0></A>
%endif

```

```
</TD>
</TABLE>
</TD></TR>
</TABLE>

</CENTER>

    %}

    %}

%}

%HTML_INPUT{

<HTML>

%EXEC_SQL(MallInfo)

<HEAD>
  <META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
  <TITLE>Mall Registration</TITLE>
</HEAD>

<BODY BACKGROUND="/ncsample/back1.gif">

  %include "$(MallHeader)"

  <p align=center><font size=5><strong>Registration is changed</strong></font></p>

  <CENTER>
    <TABLE WIDTH=80%>
      <TR>
        <TD align=center width=50></td>
        <TD>
          Changes have been made successfully to your account.<BR>
          If you changed your Password, please remember to write them down
          in a safe place.
        </TD>
        <TD align=center width=50></td>
      </TR>
    </TABLE>

    %if (url_in == "")
      Proceed to the <A HREF="/cgi-bin/nph-msrvr;/execmacro/mall_dir.d2w/report">
      Mall Directory</A>.
    %else
      <FORM ACTION="$(url_in)">
      <INPUT TYPE="submit" VALUE="Continue">
      </FORM>
    %endif
  </CENTER>

  %include "$(MallFooter)"

</BODY>

</HTML>
%}

%HTML_REPORT{

<HTML>

%EXEC_SQL(MallInfo)

<HEAD>
  <META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
  <TITLE>Registration Information</TITLE>
</HEAD>

<BODY BACKGROUND="/ncsample/back1.gif">
```

```

%include "$(MallHeader)"

%EXEC_SQL(DO_UPDATE)

%include "$(MallFooter)"

</BODY>

</HTML>

%}

```

### 6.7.2.3 Exception Macro

The example below lists the MALL\_ERR\_REG\_LEGACYMQ.D2W file.

```

%{=====
The sample Templates, HTML and Macros are furnished by IBM as simple
examples to provide an illustration. These examples have not been
thoroughly tested under all conditions. IBM, therefore, cannot guarantee
imply reliability, serviceability or function of these programs. All
programs contained herein are provided to you "AS IS".

The sample Templates, HTML and Macros may include the names of individuals,
companies, brands and products in order to illustrate them as completely as
possible. All of these are names are fictitious and any similarity to the names
and addresses used by an actual business enterprise is entirely coincidental.

(C) Copyright IBM Corp. 1995, 1996.
=====}%}

%define {
mode_in = mode ? "$(mode)" : "new"
cancel = cancel_url ? "$(cancel_url)" : ""
%}

%SQL(MallInfo){

select mhthead, mhtfoot, mhtbase
from mall

%SQL_REPORT{

%ROW{
@DTW_assign( MallHeader, V_mhthead )
@DTW_assign( MallFooter, V_mhtfoot )
@DTW_assign( MallBase, V_mhtbase )
%}
%}

%SQL_MESSAGE{
100 : {
%} : continue
default: { %} :continue %}

%}

%HTML_REPORT{

<HTML>

<HEAD>
<META HTTP-EQUIV=Expires CONTENT="Mon, 01 Jan 1996 01:01:01 GMT">
<TITLE>
Registration Information
</TITLE>
</HEAD>

%EXEC_SQL(MallInfo)

<BODY BACKGROUND="/ncsample/back1.gif">

%include "$(MallHeader)"

```

```

<center>

%if (mode_in == "new")
  <p align=center><font size=5><strong>Register</strong></font></p>
%else
  <p align=center><font size=5><strong>Change Registration</strong></font></p>
%endif

%INCLUDE "/ncsample/msreglabels.inc"

<form method=POST action="/cgi-bin/nph-msrvr/register/$(mode_in)">
  %if (mode_in == "new")
    <INPUT Type = "hidden" Name="url" Value="/msprotect/nph-msrvr/execmacro/
      mallregform.d2w/input?mode=$(mode_in)" >
  %else
    <INPUT Type = "hidden" Name="url" Value="/msprotect/nph-msrvr/execmacro/
      mallregupd.d2w/input?mode=$(mode_in)" >
  %endif
  <INPUT TYPE="hidden" NAME="mode" VALUE="$(mode_in)">

<TABLE WIDTH=80% border=0>
  <TR>

    <td align=center width=85>
      
    </td>

    <td>
      There is a problem with your submission:<P>
      <B>
        %if (error_code == "4")
          There was a problem saving your password. Please try again.<BR>
          If the problem persists, please notify customer service.
        %elif (error_code == "160")
          The information in the <i><a href="#$(field)">$(msg)</a></i> field
          is invalid.
        %elif (error_code == "170")
          The <i><a href="#shlpswd">verify password</a></i> did not match your
          password.
        %elif (error_code == "180")
          Someone is already using this <i><a href="#shlogid">Login ID</a></i>.
          Please choose another.
        %elif (error_code == "190")
          The <i><a href="#$(field)">$(msg)</a></i> field must be filled in.
        %elif (error_code == "200")
          You have already registered.
        %elif (error_code == "210")
          This error is to be fixed by the server team.
        %elif (error_code == "270")
          The passwords can only consist of alphabetic (a-z) and numeric (0-9)
          characters .
        %elif (error_code == "280")
          Shopper Login IDs can only consist of alphabetic (a-z) and numeric (0-9)
          characters.
        %elif (error_code == "290")
          Error $(error_code) occurred. The API is $(API). The reason is $(REASON);
        %elif (error_code == "300")
          Error $(error_code) occurred. The API is $(API).<BR>
          The reason is $(REASON).<BR>
          Contact Your Merchant and tell him that the reason of error is.<BR>
          <A HREF="#$(HOST)/$(USERPAGE)"> click here </A>.
        %else
          An undefined error occurred. Please contact the mall administrator.
          The error code is $(error_code).
        %endif
      </B>
    </TD>
  </TR>

  %INCLUDE "/ncsample/msregform.inc"

  <TR>
    <TD WIDTH=40>

```

```

        <IMG SRC="/ncsample/4.gif">
    </TD>
    %if (mode_in == "new")
        <TD WIDTH=460 VALIGN=MIDDLE>
            <FONT SIZE=+2><B>Submit Your Registration</B></FONT>
        </TD>
    %else
        <TD WIDTH=460 VALIGN=MIDDLE>
            <FONT SIZE=+2><B>Submit Your Changes</B></FONT>
        </TD>
    %endif
</TR>

<TR>
<TD></TD>
<TD>
    %if (mode_in == "new")
        You are now ready to submit your registration for approval.
        To submit your registration, click the Submit Registration button.
        To cancel your registration, click the back arrow button below.<P>
    %else
        You are now ready to change your registration.
        To change your registration, click the Change Registration button.
        To cancel your changes, click the back arrow button below.<P>
    %endif
</TD>
</TR>

<TR>
<TD></TD>
<TD>
    <TABLE CELLPADDING=10>
        <TR>
            <TD>
                <IMG SRC="/ncsample/hoption.gif">
            </TD>
            <TD>
                %if (mode_in == "new")
                    <INPUT TYPE="submit" VALUE="Submit Registration">
                %else
                    <INPUT TYPE="submit" VALUE="Change Registration">
                %endif
            </TD>
        </TR>
    </TD>
</TR>
<TR>
<TD>
        <IMG SRC="/ncsample/hoption.gif">
    </TD>
</TR>
</FORM>
<TD>
    %if (cancel != "")
        <A HREF="$(url_in)"><IMG SRC="/ncsample/cancel.gif" border=0></A>
    %else
        <A HREF="$(MallBase)"><IMG SRC="/ncsample/cancel.gif" border=0></A>
    %endif
</TD>
</TR>
</TABLE>
</TD>
</TR>

</TABLE>

</CENTER>

%include "$(MallFooter)"

</BODY>
</HTML>
%}

```

### 6.7.3 Include Files

These macros also include four .INC files to be copied into the directory IBMNETCOMMERCEMACROEN\_USNCSAMPLE:

- msregadd.inc
- msregform.inc
- msreglabels.inc
- msvarmap.inc

#### 6.7.3.1 MSREGADD Include File

The example below lists the MSREGADD.INC file:

```
@DTW_assign(l_shlogid, "Shopper's Login ID")
@DTW_assign(l_shlpswd, "Password")
@DTW_assign(l_shlpswdver, "Verify Password")
@DTW_assign(l_shchaans, "MQ User")
@DTW_assign(l_satitle, "Title")
@DTW_assign(l_salname, "Last Name")
@DTW_assign(l_samname, "Middle Name")
@DTW_assign(l_safname, "First Name")
@DTW_assign(l_sarepcom, "Company Name")
@DTW_assign(l_saaddr1, "Address (line 1)")
@DTW_assign(l_saaddr2, "Address (line 2)")
@DTW_assign(l_saaddr3, "Address (line 3)")
@DTW_assign(l_sacity, "City")
@DTW_assign(l_sastate, "State/Province")
@DTW_assign(l_sazipc, "Zip/Postal Code")
@DTW_assign(l_sacntry, "Country")
@DTW_assign(l_saphone1, "Daytime Phone Number")
@DTW_assign(l_saphone2, "Evening Phone Number")
@DTW_assign(l_sadphtyp, "Daytime Phone Type")
@DTW_assign(l_saephtyp, "Evening Phone Type")
@DTW_assign(l_safax, "Fax Number")
@DTW_assign(l_saeph1st, "Phone Unlisted?")
@DTW_assign(l_saemail1, "E-mail or URL (1)")
@DTW_assign(l_saemail2, "E-mail or URL (2)")
@DTW_assign(l_shcomm, "Preferred Method of Communication")
@DTW_assign(l_sabtcall, "Best Time to Call")
@DTW_assign(l_sdage, "Age")
@DTW_assign(l_sdgendr, "Gender")
@DTW_assign(l_sdmstat, "Marital Status")
@DTW_assign(l_sdcamp, "Employer")
@DTW_assign(l_sdhhnbr, "Number of Household Members")
@DTW_assign(l_sdincom, "Annual Household Income")
@DTW_assign(l_sdintrs, "Interests/Hobbies")
```

#### 6.7.3.2 MSREGFORM Include File

The example below lists the MSREGFORM.INC file:

```
<TR><TD>&nbsp;</TD></TR>

<TR>
<TD WIDTH=40><IMG SRC="/ncsample/1.gif"></TD>

<TD VALIGN=MIDDLE><FONT SIZE=+2><B>Personal Information</B></FONT></TD>
</TR><TR>

<TD></TD><TD>

<TABLE CELLSPACING=10>
<TR>
<TD>
<B><FONT SIZE=-1>$(l_shlogid)</FONT></B></TD><TD>
<A NAME="shlogid">
%if (mode_in == "modify")
<font size="+2"><B>$(shlogid)</B></font>
<INPUT Type = "hidden" Name="shlogid" Value="$(shlogid)">
%else
```

```

        <INPUT TYPE="text" NAME="shlogid" VALUE="$ (shlogid)" SIZE="20" MAXLENGTH="30">
    %endif
</TD>
</TR>

<TR>
<TD>
<B><FONT SIZE=-1>$(l_shlpswd)</FONT></B>
</TD>
<TD>
<A NAME="shlpswd">
<INPUT TYPE="password" NAME="shlpswd" VALUE="$ (shlpswd)" SIZE="20" MAXLENGTH="16">
</TD>
<TD>
<B><FONT SIZE=-1>$(l_shlpswdver)</FONT></B>
</TD>
<TD>
<A NAME="shlpswdver">
<INPUT TYPE="password" NAME="shlpswdver" VALUE="$ (shlpswdver)" SIZE="20" MAXLENGTH="16">
</TD>
</TR>

<TR>

<TD><FONT SIZE=-1>$(l_shchaans)</FONT></TD><TD>
<A NAME="shchaans">
<INPUT TYPE="text" NAME="shchaans" VALUE="$ (shchaans)" SIZE="20" MAXLENGTH="20">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(l_satitle)</FONT></TD><TD>
<A NAME="satitle">
<SELECT NAME="satitle">
    @DTW_assign(sel1, "")
    @DTW_assign(sel2, "")
    @DTW_assign(sel3, "")
    @DTW_assign(sel4, "")
    %if (satitle == null)
    @DTW_assign(sel1, "SELECTED")
    %elif (satitle == "Mr")
    @DTW_assign(sel1, "SELECTED")
    %elif (satitle == "Mrs")
    @DTW_assign(sel2, "SELECTED")
    %elif (satitle == "Ms")
    @DTW_assign(sel3, "SELECTED")
    %elif (satitle == "Dr")
    @DTW_assign(sel4, "SELECTED")
    %endif
<OPTION $(sel1) VALUE="Mr">Mr
<OPTION $(sel2) VALUE="Mrs">Mrs
<OPTION $(sel3) VALUE="Ms">Ms
<OPTION $(sel4) VALUE="Dr">Dr
</SELECT>
</TD>

<TD><B><FONT SIZE=-1>$(l_salname)</FONT></B></TD><TD>
<A NAME="salname">
<INPUT TYPE="text" NAME="salname" VALUE="$ (salname)" SIZE="20" MAXLENGTH="30">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(l_safname)</FONT></TD><TD>
<A NAME="safname">
<INPUT TYPE="text" NAME="safname" VALUE="$ (safname)" SIZE="20" MAXLENGTH="30">
</TD>

<TD><FONT SIZE=-1>$(l_samname)</FONT></TD><TD>
<A NAME="samname">
<INPUT TYPE="text" NAME="samname" VALUE="$ (samname)" SIZE="20" MAXLENGTH="30">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(l_sarepcom)</FONT></TD><TD>
<A NAME="sarepcom">
<INPUT TYPE="text" NAME="sarepcom" VALUE="$ (sarepcom)" SIZE="20" MAXLENGTH="30">
</TD>
</TR>

```



```

</TABLE>
</TD></TR><TR>

<!------->

<TD WIDTH=40><IMG SRC="/ncsample/2.gif"></TD>
<TD WIDTH=460 VALIGN=MIDDLE><FONT SIZE=+2><B>Contact Information</B></FONT></TD>
</TR><TR>

<TD></TD><TD>

<TABLE CELLSPACING=10>
<TR>

<TD><B><FONT SIZE=-1>$(1_saaddr1)</FONT></B></TD><TD>
<A NAME="saaddr1">
<INPUT TYPE="text" NAME="saaddr1" VALUE="$(saaddr1)" SIZE="40" MAXLENGTH="50">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_saaddr2)</FONT></TD><TD>
<A NAME="saaddr2">
<INPUT TYPE="text" NAME="saaddr2" VALUE="$(saaddr2)" SIZE="40" MAXLENGTH="50">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_saaddr3)</FONT></TD><TD>
<A NAME="saaddr3">
<INPUT TYPE="text" NAME="saaddr3" VALUE="$(saaddr3)" SIZE="40" MAXLENGTH="50">
</TD></TR>
</TABLE>

<TABLE CELLSPACING=10>
<TR>

<TD><B><FONT SIZE=-1>$(1_sacity)</FONT></B></TD><TD>
<A NAME="sacity">
<INPUT TYPE="text" NAME="sacity" VALUE="$(sacity)" SIZE="20" MAXLENGTH="30">
</TD>

<TD><B><FONT SIZE=-1>$(1_sastate)</FONT></B></TD><TD>
<A NAME="sastate">
<INPUT TYPE="text" NAME="sastate" VALUE="$(sastate)" SIZE="20" MAXLENGTH="20">
</TD></TR><TR>

<TD><B><FONT SIZE=-1>$(1_sazipc)</FONT></B></TD><TD>
<A NAME="sazipc">
<INPUT TYPE="text" NAME="sazipc" VALUE="$(sazipc)" SIZE="12" MAXLENGTH="20">
</TD>

<TD><B><FONT SIZE=-1>$(1_sacntry)</FONT></B></TD><TD>
<A NAME="sacntry">
<INPUT TYPE="text" NAME="sacntry" VALUE="$(sacntry)" SIZE="20" MAXLENGTH="30">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_saphone1)</FONT></TD><TD>
<A NAME="saphone1">
<INPUT TYPE="text" NAME="saphone1" VALUE="$(saphone1)" SIZE="12" MAXLENGTH="30">
</TD>

<TD><FONT SIZE=-1>$(1_saphone2)</FONT></TD><TD>
<A NAME="saphone2">
<INPUT TYPE="text" NAME="saphone2" VALUE="$(saphone2)" SIZE="12" MAXLENGTH="30">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_sadphtyp)</FONT></TD><TD>
<A NAME="sadphtyp">
<SELECT NAME="sadphtyp">
  @DTW_assign(sel1, "")
  @DTW_assign(sel2, "")
  %if (sadphtyp == null)
  @DTW_assign(sel1, "SELECTED")
  %elif (sadphtyp == "PHN")
  @DTW_assign(sel1, "SELECTED")
  %elif (saephtyp == "TTY")
  @DTW_assign(sel2, "SELECTED")

```

```

%endif
<OPTION $(sel1) VALUE="PHN">PHN
<OPTION $(sel2) VALUE="TTY">TTY
</SELECT>
</TD>

<TD><FONT SIZE=-1>$(1_saephtyp)</FONT></TD><TD>
<A NAME="saephtyp">
<SELECT NAME="saephtyp">
  @DTW_assign(sel1, "")
  @DTW_assign(sel2, "")
  %if (saephtyp == null)
    @DTW_assign(sel1, "SELECTED")
  %elif (saephtyp == "PHN")
    @DTW_assign(sel1, "SELECTED")
  %elif (saephtyp == "TTY")
    @DTW_assign(sel2, "SELECTED")
  %endif
<OPTION $(sel1) VALUE="PHN">PHN
<OPTION $(sel2) VALUE="TTY">TTY
</SELECT>
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_saeph1st)</FONT></TD><TD>
<A NAME="saeph1st">
<SELECT NAME="saeph1st">
  @DTW_assign(sel1, "")
  @DTW_assign(sel2, "")
  %if (saeph1st == null)
    @DTW_assign(sel1, "SELECTED")
  %elif (saeph1st == "0")
    @DTW_assign(sel1, "SELECTED")
  %elif (saeph1st == "1")
    @DTW_assign(sel2, "SELECTED")
  %endif
<OPTION $(sel1) VALUE="0">No
<OPTION $(sel2) VALUE="1">Yes
</SELECT>
</TD>

<TD><FONT SIZE=-1>$(1_safax)</FONT></TD><TD>
<A NAME="safax">
<INPUT TYPE="text" NAME="safax" VALUE="$(safax)" SIZE="12" MAXLENGTH="30">
</TD></TR><TR>

</TABLE>

<TABLE CELLPACING=10>
<TR>

<TD><FONT SIZE=-1>$(1_saemail1)</FONT></TD><TD>
<A NAME="saemail1">
<INPUT TYPE="text" NAME="saemail1" VALUE="$(saemail1)" SIZE="40" MAXLENGTH="254">
</TD></TR><TR>

<TD><FONT SIZE=-1>$(1_saemail2)</FONT></TD><TD>
<A NAME="saemail2">
<INPUT TYPE="text" NAME="saemail2" VALUE="$(saemail2)" SIZE="40" MAXLENGTH="254">
</TD></TR><TR>

</TABLE>

<TABLE CELLPACING=10>
<TR>

<TD><FONT SIZE=-1>$(1_shcomm)</FONT></TD><TD>
<A NAME="shcomm">
<SELECT NAME="shcomm">
  @DTW_assign(sel1, "")
  @DTW_assign(sel2, "")
  @DTW_assign(sel3, "")
  @DTW_assign(sel4, "")
  %if (shcomm == null)
    @DTW_assign(sel1, "SELECTED")
  %elif (shcomm == "P1")

```







```

@DTW_assign(l_sarepcom, "Company Name")
@DTW_assign(l_saaddr1, "Address (line 1)")
@DTW_assign(l_saaddr2, "Address (line 2)")
@DTW_assign(l_saaddr3, "Address (line 3)")
@DTW_assign(l_sacity, "City")
@DTW_assign(l_sastate, "State/Province")
@DTW_assign(l_sazipc, "Zip/Postal Code")
@DTW_assign(l_sacntry, "Country")
@DTW_assign(l_saphone1, "Daytime Phone Number")
@DTW_assign(l_saphone2, "Evening Phone Number")
@DTW_assign(l_sadphtyp, "Daytime Phone Type")
@DTW_assign(l_saephtyp, "Evening Phone Type")
@DTW_assign(l_safax, "Fax Number")
@DTW_assign(l_saeph1st, "Phone Unlisted?")
@DTW_assign(l_saemail1, "E-mail or URL (1)")
@DTW_assign(l_saemail2, "E-mail or URL (2)")
@DTW_assign(l_shcomm, "Preferred Method of Communication")
@DTW_assign(l_sabtcall, "Best Time to Call")
@DTW_assign(l_sdage, "Age")
@DTW_assign(l_sdgendr, "Gender")
@DTW_assign(l_sdmstat, "Marital Status")
@DTW_assign(l_sdcomp, "Employer")
@DTW_assign(l_sdhhnbr, "Number of Household Members")
@DTW_assign(l_sdincom, "Annual Household Income")
@DTW_assign(l_sdintrs, "Interests/Hobbies")

@DTW_assign(b1_shlogid, "<BLINK><I>Shopper's Login ID</I></BLINK>")
@DTW_assign(b1_shlpswd, "<BLINK><I>Password</I></BLINK>")
@DTW_assign(b1_shlpswdver, "<BLINK><I>Verify Password</I></BLINK>")
@DTW_assign(b1_shchaans, "<BLINK><I>Cics User</I></BLINK>")
@DTW_assign(b1_satitle, "<BLINK><I>Title</I></BLINK>")
@DTW_assign(b1_salname, "<BLINK><I>Last Name</I></BLINK>")
@DTW_assign(b1_samname, "<BLINK><I>Middle Name</I></BLINK>")
@DTW_assign(b1_safname, "<BLINK><I>First Name</I></BLINK>")
@DTW_assign(b1_sarepcom, "<BLINK><I>Company Name</I></BLINK>")
@DTW_assign(b1_saaddr1, "<BLINK><I>Address (line 1)</I></BLINK>")
@DTW_assign(b1_saaddr2, "<BLINK><I>Address (line 2)</I></BLINK>")
@DTW_assign(b1_saaddr3, "<BLINK><I>Address (line 3)</I></BLINK>")
@DTW_assign(b1_sacity, "<BLINK><I>City</I></BLINK>")
@DTW_assign(b1_sastate, "<BLINK><I>State/Province</I></BLINK>")
@DTW_assign(b1_sazipc, "<BLINK><I>Zip/Postal Code</I></BLINK>")
@DTW_assign(b1_sacntry, "<BLINK><I>Country</I></BLINK>")
@DTW_assign(b1_saphone1, "<BLINK><I>Daytime Phone Number</I></BLINK>")
@DTW_assign(b1_saphone2, "<BLINK><I>Evening Phone Number</I></BLINK>")
@DTW_assign(b1_sadphtyp, "<BLINK><I>Daytime Phone Type</I></BLINK>")
@DTW_assign(b1_saephtyp, "<BLINK><I>Evening Phone Type</I></BLINK>")
@DTW_assign(b1_safax, "<BLINK><I>Fax Number</I></BLINK>")
@DTW_assign(b1_saeph1st, "<BLINK><I>Phone Unlisted?</I></BLINK>")
@DTW_assign(b1_saemail1, "<BLINK><I>E-mail or URL (1)</I></BLINK>")
@DTW_assign(b1_saemail2, "<BLINK><I>E-mail or URL (2)</I></BLINK>")
@DTW_assign(b1_shcomm, "<BLINK><I>Preferred Method of Communication</I></BLINK>")
@DTW_assign(b1_sabtcall, "<BLINK><I>Best Time to Call</I></BLINK>")
@DTW_assign(b1_sdage, "<BLINK><I>Age</I></BLINK>")
@DTW_assign(b1_sdgendr, "<BLINK><I>Gender</I></BLINK>")
@DTW_assign(b1_sdmstat, "<BLINK><I>Marital Status</I></BLINK>")
@DTW_assign(b1_sdcomp, "<BLINK><I>Employer</I></BLINK>")
@DTW_assign(b1_sdhhnbr, "<BLINK><I>Number of Household Members</I></BLINK>")
@DTW_assign(b1_sdincom, "<BLINK><I>Annual Household Income</I></BLINK>")
@DTW_assign(b1_sdintrs, "<BLINK><I>Interests/Hobbies</I></BLINK>")

@DTW_assign(msg, "unknown field")
%if (field == "shlogid")
    @DTW_assign(msg,l_shlogid)
    @DTW_assign(l_shlogid, b1_shlogid)
%elif (field == "shlpswd")
    @DTW_assign(msg,l_shlpswd)
    @DTW_assign(l_shlpswd, b1_shlpswd)
%elif (field == "shlpswdver")
    @DTW_assign(msg,l_shlpswdver)
    @DTW_assign(l_shlpswdver, b1_shlpswdver)
%elif (field == "shchaque")
    @DTW_assign(msg,l_shchaque)
    @DTW_assign(l_shchaque, b1_shchaque)
%elif (field == "shchaans")
    @DTW_assign(msg,l_shchaans)

```

```
@DTW_assign(l_shchaans, bl_shchaans)
%elif (field == "shfield1")
  @DTW_assign(msg,l_shfield1)
  @DTW_assign(l_shfield1, bl_shfield1)
%elif (field == "shfield2")
  @DTW_assign(msg,l_shfield2)
  @DTW_assign(l_shfield2, bl_shfield2)
%elif (field == "satitle")
  @DTW_assign(msg,l_satitle)
  @DTW_assign(l_satitle, bl_satitle)
%elif (field == "salname")
  @DTW_assign(msg,l_salname)
  @DTW_assign(l_salname, bl_salname)
%elif (field == "samname")
  @DTW_assign(msg,l_samname)
  @DTW_assign(l_samname, bl_samname)
%elif (field == "safname")
  @DTW_assign(msg,l_safname)
  @DTW_assign(l_safname, bl_safname)
%elif (field == "sarepcom")
  @DTW_assign(msg,l_sarepcom)
  @DTW_assign(l_sarepcom, bl_sarepcom)
%elif (field == "saaddr1")
  @DTW_assign(msg,l_saaddr1)
  @DTW_assign(l_saaddr1, bl_saaddr1)
%elif (field == "saaddr2")
  @DTW_assign(msg,l_saaddr2)
  @DTW_assign(l_saaddr2, bl_saaddr2)
%elif (field == "saaddr3")
  @DTW_assign(msg,l_saaddr3)
  @DTW_assign(l_saaddr3, bl_saaddr3)
%elif (field == "sacity")
  @DTW_assign(msg,l_sacity)
  @DTW_assign(l_sacity, bl_sacity)
%elif (field == "sastate")
  @DTW_assign(msg,l_sastate)
  @DTW_assign(l_sastate, bl_sastate)
%elif (field == "sazipc")
  @DTW_assign(msg,l_sazipc)
  @DTW_assign(l_sazipc, bl_sazipc)
%elif (field == "sacntry")
  @DTW_assign(msg,l_sacntry)
  @DTW_assign(l_sacntry, bl_sacntry)
%elif (field == "saphone1")
  @DTW_assign(msg,l_saphone1)
  @DTW_assign(l_saphone1, bl_saphone1)
%elif (field == "saphone2")
  @DTW_assign(msg,l_saphone2)
  @DTW_assign(l_saphone2, bl_saphone2)
%elif (field == "sadphtyp")
  @DTW_assign(msg,l_sadphtyp)
  @DTW_assign(l_sadphtyp, bl_sadphtyp)
%elif (field == "saephtyp")
  @DTW_assign(msg,l_saephtyp)
  @DTW_assign(l_saephtyp, bl_saephtyp)
%elif (field == "mecfac")
  @DTW_assign(msg,l_safax)
  @DTW_assign(l_safax, bl_safax)
%elif (field == "saeph1st")
  @DTW_assign(msg,l_saeph1st)
  @DTW_assign(l_saeph1st, bl_saeph1st)
%elif (field == "saemail1")
  @DTW_assign(msg,l_saemail1)
  @DTW_assign(l_saemail1, bl_saemail1)
%elif (field == "saemail2")
  @DTW_assign(msg,l_saemail2)
  @DTW_assign(l_saemail2, bl_saemail2)
%elif (field == "shcomm")
  @DTW_assign(msg,l_shcomm)
  @DTW_assign(l_shcomm, bl_shcomm)
%elif (field == "sabtcall")
  @DTW_assign(msg,l_sabtcall)
  @DTW_assign(l_sabtcallr, bl_sabtcall)
%elif (field == "sdage")
  @DTW_assign(msg,l_sdage)
```

```

    @DTW_assign(l_sdage, bl_sdage)
%elif (field == "sdgendr")
    @DTW_assign(msg,l_sdgendr)
    @DTW_assign(l_sdgendr, bl_sdgendr)
%elif (field == "sdmstat")
    @DTW_assign(msg,l_sdmstat)
    @DTW_assign(l_sdmstat, bl_sdmstat)
%elif (field == "sdcomp")
    @DTW_assign(msg,l_sdcomp)
    @DTW_assign(l_sdcomp, bl_sdcomp)
%elif (field == "sdhhnbr")
    @DTW_assign(msg,l_sdhhnbr)
    @DTW_assign(l_sdhhnbr, bl_sdhhnbr)
%elif (field == "sdincom")
    @DTW_assign(msg,l_sdincom)
    @DTW_assign(l_sdincom, bl_sdincom)
%elif (field == "sdintrs")
    @DTW_assign(msg,l_sdintrs)
    @DTW_assign(l_sdintrs, bl_sdintrs)
%endif

```

### 6.7.3.4 MSVARMAP Include File

The example below lists the MSVARMAP.INC file:

```

@DTW_assign(shlogid, V_shlogid)
@DTW_assign(shlpswd, V_shlpswd)
@DTW_assign(shlpswdver, V_shlpswd)
@DTW_assign(shchaans, V_shchaans)
@DTW_assign(satitle, V_satitle)
@DTW_assign(salname, V_salname)
@DTW_assign(samname, V_samname)
@DTW_assign(safname, V_safname)
@DTW_assign(sarepcom, V_sarepcom)
@DTW_assign(saaddr1, V_saaddr1)
@DTW_assign(saaddr2, V_saaddr2)
@DTW_assign(saaddr3, V_saaddr3)
@DTW_assign(sacity, V_sacity)
@DTW_assign(sastate, V_sastate)
@DTW_assign(sazipc, V_sazipc)
@DTW_assign(sacntry, V_sacntry)
@DTW_assign(saphone1, V_saphone1)
@DTW_assign(saphone2, V_saphone2)
@DTW_assign(sadphtyp, V_sadphtyp)
@DTW_assign(saephtyp, V_saephtyp)
@DTW_assign(safax, V_safax)
@DTW_assign(saeph1st, V_saeph1st)
@DTW_assign(saemail1, V_saemail1)
@DTW_assign(saemail2, V_saemail2)
@DTW_assign(shcomm, V_shcomm)
@DTW_assign(sabtcall, V_sabtcall)
@DTW_assign(sdage, V_sdage)
@DTW_assign(sdgendr, V_sdgendr)
@DTW_assign(sdmstat, V_sdmstat)
@DTW_assign(sdcomp, V_sdcomp)
@DTW_assign(sdhhnbr, V_sdhhnbr)
@DTW_assign(sdincom, V_sdincom)
@DTW_assign(sdintrs, V_sdintrs)

```



## Appendix A. ITSO System Environment

This section describes the hardware and software installed to run the scenarios that have been described in 2.7, "An Example of a Small Shop" on page 36, Chapter 5, "Integrating Net.Commerce with CICS" on page 137, and Chapter 6, "Integrating Net.Commerce with MQSeries" on page 227.

Component	Single Server	Development System	Split Server A	Split Server B
System	RS/6000 PowerPC 43P-140	PS/2 9595	PC 750	PC 750
CPU	Power 604e	Pentium	Pentium	Pentium
CPU Speed	166 MHz	66 MHz	166 MHz	166 MHz
Memory	192 MB	32 MB	64 MB	64 MB
Diskspace	1 GB	1 GB	1.6 GB	1.6 GB
Operating System	AIX 4.2	Windows NT 4.0	Windows NT 4.0 + SP3	Windows NT 4.0 + SP3
Installed Software	Net.Commerce 1.1 ICSS 4.1 DB2/6000 2.1.2	Net.Commerce 1.1 ICSS 4.1 DB2/NT 2.1.2 CICS Client for NT MQSeries 2.0 for NT Microsoft Visual C++ 4.2	Net.Commerce 1.1 ICSS 4.1 DB2/NT CAE 2.1.2 CICS Client for NT CICS Internet Gateway MQSeries 2.0 for NT	DB2/NT Server 2.1.2 Transaction Server 4.0 for NT Personal Communicatons 4.1 for NT MQSeries 2.0 for NT
Hostname	rs600025	wtr05252	wtr05305	wtr05253
IP Address	9.24.104.50	9.24.104.48	9.24.104.242	9.24.104.150

### Notes:

1. Split servers A and B and the development system were used for the CICS scenario.
2. Split servers A and B and the development system were used for the MQSeries scenario.
3. Split servers A and B were used for the Ma Baker scenario.
4. Split servers A and B and the single server were used for the installation and configuration chapter.



---

## Appendix B. Special Notices

This publication is intended to help planners and system administrators who are responsible for setting up a Net.Commerce site requiring integration with existing legacy applications.

The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM Transaction Server or CICS, MQSeries, or the Net.Commerce APIs. Please refer to the documentation of those products for more information.

Especially, care should be taken when using the sample API programs that are supplied with this publication. Please follow the instructions on how to install and use those samples and make the necessary changes to the code according to your specific environment. The samples are supplied as-is which means there is no support or maintenance provided if they do not work as described or cause damage to your system.

A working knowledge of IBM Internet Connection Server, IBM Database Server, SQL and HTML is assumed. Some knowledge of World Wide Web security, CICS and MQSeries as well as the AIX and Windows NT operating systems is also assumed.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate

them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX	APPN
AS/400	CICS
CICS/ESA	CICS/6000
Cryptolope	DATABASE 2
DataJoiner	DB2
DB2/6000	HACMP/6000
IBM	MQSeries
NetView	Net.Commerce
Net.Data	OS/2
OS/390	PowerPC
PowerPC 604	PS/2
RS/6000	VTAM

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

PostScript  
Lotus Approach  
StandbyServer, Vinca  
Sales/Use Tax System  
Prologistics.Merchant, Tandata  
RSA  
Netscape Navigator  
Oracle  
PKZIP

Adobe Systems, Inc.  
Lotus Development Corp.  
Vinca Corp.  
Taxware international Inc.  
Tandata Corp.  
RSA Data Security, Inc.  
Netscape Corp.  
Oracle Corp.  
PKWARE, Inc.



---

## Appendix C. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

---

### C.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 279.

- *A Guide to the Internet Connection Servers*, SG24-4805
- *Secure Electronic Transactions: Credit Card Payment on the Web in Theory and Practice*, SG24-4978
- *Safe Surfing: How to Build a Secure World Wide Web Connection*, SG24-4564
- *MQSeries Three Tier Programming Examples*, SG24-4853
- *Multiplatform APPC Configuration Guide*, GG24-4485

---

### C.2 Product Documentation

The following documentation, which is referenced throughout this document, is shipped, among other publications, with the Net.Commerce product. It can be viewed online when installed.

- *IBM Net.Commerce: Installation and Operation Guide*, GC09-2373
- *IBM Net.Commerce Basics: Open for Business*, SC09-2372
- *IBM Net.Commerce: Customization Guide and Reference*, SC09-2370
- *IBM Internet Connection Secure Server for AIX: Up and Running!*, SC31-8203
- *IBM Internet Connection Secure Server for Windows NT: UP and Running!*, SC31-8278
- *IBM DB2 for AIX Installation and Operation Guide*, S20H-4757
- *IBM DB2 for Windows NT Installation and Operation Guide*, S33H-0312
- *IBM DB2: Administration Guide*, S20H-4580
- *IBM DB2: Application Programming Guide*, S20H-4643
- *IBM DB2: Command Reference*, S20H-4645
- *IBM DB2: SQL Reference*, S20H-4665

---

### C.3 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041

CD-ROM Title	Subscription Number	Collection Kit Number
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

---

## C.4 Other Publications

These publications are also relevant as further information sources:

- *IBM Transaction Server for Windows NT - Installation Guide*, GC33-1880
- *IBM Transaction Server for Windows NT - Administration Guide*, SC33-1881
- *IBM Transaction Server for Windows NT - Intercommunications Guide (Base)*, SC33-1882
- *CICS Family - Interproduct Communication*, SC33-0824
- *IBM Transaction Server for Windows NT - Application Programming Guide*, SC33-1888
- *IBM Transaction Server for Windows NT - Application Programming Reference*, SC33-1887
- *CICS Clients: Administration*, SC33-1792
- *CICS Clients: Gateways*, SC33-1821
- *IBM MQSeries for Windows NT*, SC33-1643
- *IBM MQSeries Clients*, GC33-1632
- *IBM MQSeries Command Reference*, SC33-1369
- *VTAM V4R4 Customization*, LY43-0075
- *VTAM Resource Definition Reference*, SC31-8377

---

## C.5 World Wide Web Resources for Information

The following URLs can be visited to find more information on electronic commerce and IBM products associated with that area:

- *Net.Data Documentation*, <http://www.software.ibm.com/data/net.data/docs/>



---

## How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL <http://www.redbooks.ibm.com>.

---

## How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

To get LIST3820s of redbooks, type one of the following commands:

```
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get BookManager BOOKs of redbooks, type the following command:

```
TOOLCAT REDBOOKS
```

To get lists of redbooks, type one of the following commands:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
```

To register for information on workshops, residencies, and redbooks, type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
```

For a list of product area specialists in the ITSO: type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
```

- **Redbooks Home Page on the World Wide Web**  
<http://w3.itso.ibm.com/redbooks>
- **IBM Direct Publications Catalog on the World Wide Web**  
<http://www.elink.ibm.link.ibm.com/pb1/pb1>

IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to [announce@webster.ibm.link.ibm.com](mailto:announce@webster.ibm.link.ibm.com) with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

---

## How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** — send orders to:

	<b>IBMMAIL</b>	<b>Internet</b>
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark
--	--	--

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States) or (+1)001-408-256-5422 (Outside USA)** — ask for:

Index # 4421 Abstracts of new redbooks  
Index # 4422 IBM redbooks  
Index # 4420 Redbooks for last six months

- **Direct Services** - send note to [softwareshop@vnet.ibm.com](mailto:softwareshop@vnet.ibm.com)

- **On the World Wide Web**

Redbooks Home Page <http://www.redbooks.ibm.com>  
"Redpieces" - online-readable excerpts from redbooks in progress <http://www.redbooks.ibm.com/redpieces>  
IBM Direct Publications Catalog <http://www.elink.ibm.link.ibm.com/pbl/pbl>

- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to [announce@webster.ibm.link.ibm.com](mailto:announce@webster.ibm.link.ibm.com) with the keyword subscribe in the body of the note (leave the subject line blank).

---

## IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity
-------	--------------	----------


---

First name	Last name
------------	-----------

---

Company

---

Address

---

City	Postal code	Country
------	-------------	---------

---

Telephone number	Telefax number	VAT number
------------------	----------------	------------

• Invoice to customer number \_\_\_\_\_

• Credit card number \_\_\_\_\_

---

Credit card expiration date

Card issued to

Signature

**We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.**



## List of Abbreviations

<b>AIX</b>	Advanced Interactive Executive	<b>ILU</b>	Independent Logical Unit
<b>API</b>	Application Programming Interface	<b>ISO</b>	International Standards Organization
<b>APPC</b>	Advanced Program-to-Program Communication	<b>ITSO</b>	International Technical Support Organization
<b>APPN</b>	Advanced Peer-to-Peer Networking	<b>JPEG</b>	Joint Photographic Experts Group
<b>AS/400</b>	Application System/400	<b>LAN</b>	Local Area Network
<b>CAE</b>	Client Application Enabler	<b>LU</b>	Logical Unit
<b>CDRSC</b>	Cross Domain Resource	<b>MD2</b>	RSA Message Digest 2 Algorithm
<b>CGI</b>	Common Gateway Interface	<b>MD5</b>	RSA Message Digest 5 Algorithm
<b>CICS</b>	Customer Information Control System	<b>MIME</b>	Multipurpose Internet Mail Extensions
<b>CP</b>	Control Point	<b>MQI</b>	Message Queue Interface
<b>DB2</b>	IBM DATABASE 2	<b>MVS</b>	Multiple Virtual Storage Operating System
<b>DCE</b>	Distributed Computing Environment	<b>NCP</b>	Network Control Program
<b>DES</b>	Digital Encryption Standard	<b>NETID</b>	SNA Network Identifier
<b>DLL</b>	Dynamic Link Library	<b>ODBC</b>	Open Database Connectivity
<b>DMZ</b>	De-Militarized Zone	<b>OS/2</b>	Operating System/2
<b>DNS</b>	Domain Name System	<b>PLU</b>	Primary Logical Unit
<b>DRDA</b>	Distributed Relational Database Architecture	<b>PU</b>	Physical Unit
<b>ECI</b>	External Call Interface	<b>RAM</b>	Random Access Memory
<b>EDI</b>	Electronic Data Interchange	<b>RC4</b>	RSA Rivest Cipher 4 Algorithm
<b>EPI</b>	External Presentation Interface	<b>RI</b>	Referential Integrity
<b>ER</b>	Entity Relationships	<b>ROM</b>	Read-only Memory
<b>GIF</b>	Graphic Interchange Format	<b>RU</b>	Request Unit
<b>HACMP</b>	High Availability Cluster Multiprocessing	<b>SET</b>	Secure Electronic Transactions
<b>HTML</b>	Hypertext Markup Language	<b>SFS</b>	Structured File Server
<b>HTTP</b>	Hypertext Transfer Protocol	<b>S-HTTP</b>	Secure Hypertext Transfer Protocol
<b>IBM</b>	International Business Machines Corporation	<b>SLU</b>	Secondary Logical Unit
<b>ICSS</b>	Internet Connection Secure Server	<b>SMIT</b>	System Management Interface Tool
<b>IDBLK</b>	Identification Block	<b>SMTP</b>	Simple Mail Transfer Protocol
<b>IDEA</b>	International Data Encryption Algorithm	<b>SNA</b>	System Network Architecture
<b>IDNUM</b>	Identification Number	<b>SQL</b>	Structured Query Language
<b>IEEE</b>	Institute of Electrical and Electronics Engineers	<b>SSCP</b>	System Service Control Point
		<b>SSL</b>	Secure Sockets Layer

<b>TCP/IP</b>	Transmission Control Protocol / Internet Protocol	<b>VTAM</b>	Virtual Telecommunications Access Method
<b>URL</b>	Uniform Resource Locator	<b>WWW</b>	World Wide Web
		<b>XID</b>	Exchange Identifier

---

## Index

### Special Characters

/etc/httpd.conf file 97, 99  
/etc/inittab file 96, 97  
/etc/mserver.conf file 97, 98  
/etc/rc.netc file 96, 97  
.profile file 97

### A

abbreviations 283  
accounting 33  
acronyms 283  
address book 28, 31, 32, 53  
administrator 50  
Adobe Photoshop 22  
advertising 20, 24  
AMQSCOMA.TST file 235  
anonymous users 31  
API error codes  
    See Net.Commerce Server  
API signature  
    See Net.Commerce Server  
APPC  
    See Personal Communications  
applicable taxes 80  
AUXI.C file 248  
AUXI.H file 249  
AUXILIAR.C file 205  
AUXILIAR.H file 206

### B

backend systems 44  
bibliography 277  
business consultant 43

### C

catalog 5, 20, 22, 24, 28, 30, 67, 78  
categories 5, 6, 17, 18, 21, 24, 25, 30, 38, 45, 55  
CESN.C file 202  
CESN.H file 205  
CGI  
    See Common Gateway Interface (CGI)  
CICS 34, 144, 159, 164, 167, 194, 231, 271  
    3270 application 137, 138, 140, 183  
    3270 data stream 137  
    additional requirements 144  
    Administration Utility 167, 168, 169, 170  
    API 178, 186, 192, 194  
    API Name 190  
    API signature 187  
    APPC 138, 139, 140, 141  
    ATI start 177

### CICS (continued)

C source files 194  
CEDA transaction 144, 152, 175  
CEMT transaction 169  
CESF transaction 154  
CESN transaction 144, 178, 183, 202  
Check Setup Information panel 166  
CICS application 151, 183  
CICS application major node 151  
CICS client 163, 181, 182, 184, 189  
CICS client for AIX 138  
CICS client for NT 139, 140, 141, 143, 164  
CICS environment 227  
CICS error 191, 193  
CICS Internet Gateway 143, 163, 183, 184  
CICS program 137  
CICS region 144, 167, 168, 169, 170  
CICS region. 154  
CICS server 137, 163, 168, 170, 182, 184, 189  
CICS server for AIX 138, 139  
CICS server for NT 139, 143, 149, 150, 151, 164, 183  
CICS server for OS/2 139  
CICS server on NT 152  
CICS Terminal 183  
CICS USER field 144, 188, 190  
CICS users 178  
CICS\_HOSTS environment variable 167  
CICS/ESA 138, 139, 140, 141, 143, 144, 152, 183, 189  
CICSCLI environment variable 182  
CICSCOL environment variable 182  
CICSKEY environment variable 182  
Code page for transaction routing 173  
COMMAREA 137, 138, 141  
communication definition 172  
Compile the DLL 189  
configuration 167, 169, 177  
connection profile 152  
Connection type 173  
DCE configuration 164, 167  
DCE server 167  
Deadlock timeout 176  
Default SNA mode name 174  
DLL Name 190  
ECI 138, 139  
ENCINA\_BINDING\_FILE environment variable 167  
environment variables 167  
EPI 138, 139, 140, 183, 184  
EPI1 transaction 175, 177, 183, 185, 198  
exception macro 190  
EXEC CICS LINK 137, 138, 141  
External Call Interface (ECI) 137  
External Presentation Interface (EPI) 137

CICS (*continued*)

File Manager 164  
 header files 194  
 httpd.cnf file 183  
 include files 216  
 installation 164, 181  
 Internal ATI start 177  
 IP Address 180  
 listener definition 168, 171, 180  
 Listener Name 168, 172, 180  
 local client 168, 177  
 Local LU name 171  
 Local network name 171  
 LOCAL SFS 164  
 Local SYSID 171  
 Local Terminal 169  
 LU62APPC log mode 151, 160  
 make file 194  
 MAXimum parameter 154  
 mode name 174, 175  
 MODename parameter 154  
 Named Pipe protocol 168  
 Netname parameter 153  
 Product Installation Complete panel 167  
 product license enrollment panel 167  
 Protocol 172, 180  
 RECEIVESize parameter 154  
 Remote LU name 174  
 Remote network name 174  
 Remote system ID 175  
 Remote transaction ID 175  
 Select Bundle to Install panel 164  
 Select Product to Install panel 164  
 Select Server Destination Directory panel 165  
 SENDSize parameter 154  
 service registry 181  
 session profile 153  
 SFS server 168, 169  
 SNA mode name 175  
 source code 194, 208  
 start 169, 177, 183  
 stop 170  
 SYSID 173  
 TCP/IP service 180  
 test 177, 183  
 transaction definition 174  
 Transaction name 175  
 transaction router 170  
 Triggered start 177  
 Work area 176  
 CICSCLI.INI file 182  
 CICSCLCOL.INI file 182  
 CICSKEY.INI file 182  
 CICSCLCLI.INI file 168  
 Client Access Feature 155  
 APPN network node 158  
 refiled-cics.CICS/ESA 155

Commerce.Point 2  
 Common Gateway Interface (CGI) 3, 11, 12, 64, 68, 100, 127  
 compile the receiver program 239  
 credit card details 28  
 credit card information 57, 67  
 credit card payment 32  
 Cryptolope  
   *See Security*  
 customer database 53  
 customization 36, 43, 44, 45, 50  
 CyberCash 32

**D**

DB2 Database 2, 8, 81, 133, 138, 139, 140, 141, 189, 227, 228, 230  
   access protection 50  
   backup 60  
   circular logging 49  
   CLI-ODBC Administrator 131  
   committed transactions 60  
   configuration 271  
   crash recovery 60  
   database instance 105  
   database log files 134  
   database manager 61  
   database recovery 59  
 Database Server 47  
 DataJoiner 50  
 DB2 client 106, 125, 131  
 DB2 Client for NT 143  
 DB2 Command Window 112  
 DB2 instance 85, 125  
 DB2 instance owner 97, 102, 135  
 DB2 instance user 164  
 DB2 interrupt service 112  
 DB2 logs 125  
 DB2 node name 126  
 DB2 ODBC driver 135  
 DB2 Security Server service 110  
 DB2 server 83, 97, 108, 112, 124  
 DB2 Server for NT 143  
 DB2 services 109, 131  
 DB2 Setup window 109  
 DB2 skills 43  
 DB2 systems 34  
 DB2COMM system environment variable 110  
 DB2INSTANCE system environment variable 111, 125  
 DB2PATH system environment variable 125  
 DRDA 50  
 DRDA-compatible 35  
 environment 85, 97  
 flat file environment 41  
 full backup 50  
 installation 83, 106, 108, 271  
 instance owner 85  
 license key 109



## DB2 Database (*continued*)

- log files 60, 61
- Net.Data skills 43
- non-recoverable database 60
- ODBC 50
- ODBC access 131
- ODBC drivers 132
- ODBC support 106, 108
- offline backup 60
- online backup 60
- performance 61, 134
- point of recovery 60
- re-org 134
- recoverable database 60
- Referential Integrity (RI) 51
- restore 60
- restore recovery 60
- retained logging 50, 61
- roll-forward recovery 60
- rolling back transactions 60
- SQL skills 43
- TCP/IP port 111, 112, 126
- TCP/IP service 111, 112, 126
- transactions 60
- uninstall 103, 134
- units of work 60
- user account 81, 104
- Windows NT services 125
- DB2COMM system environment variable
  - See DB2 Database
- DB2INSTANCE system environment variable
  - See *also* DB2 Database
  - refid-db2.DB2 instance 111
- db2www.ini file 41, 97, 99, 125, 127
- DCE
  - See CICS
- demo mall 43
- demographic data 24, 31
- DigiCash 32
- Distributed Computing Environment (DCE)
  - See CICS

## E

- e-mail 24, 34, 35, 116
- electronic commerce 1, 67, 77
- electronic contract 79
- Electronic Data Interchange (EDI) 34
- electronic shopping 77
- electronic store 23
- encryption keys 77
- Entity Relationships(ER) Model
  - See Net.Commerce database
- EPI1.C file 198
- EPI1.H file 201
- EPIFUNC.C file 197
- EPIFUNC.H file 198
- exception macro
  - See Net.Commerce Server

## External Call Interface (ECI)

See CICS

## External Presentation Interface (EPI)

See CICS

## F

- fax 34, 35, 36, 38
- Firewall
  - See Security
- First Virtual 32
- flat file 34, 38
- fulfillment process 24, 34

## G

- GIF format 22
- groups, shopper 50

## H

- High Availability Cluster Multiprocessing (HACMP/6000) 48
- httpd.cnf file 125
- httpf.cnf file 126
- Hypertext Markup Language (HTML) 6, 11, 29, 63, 68, 69, 164, 183, 190
  - dynamic page 2, 6
  - dynamic pages 21, 29
  - frames 21, 46
  - HTML editor 6
  - HTML extensions 21
  - HTML form 27, 53, 55, 64
  - HTML forms 13, 50
  - HTML skills 43
  - hyperlinks 30
  - static pages 23
  - tables 46

## I

- IBM CICS Internet Gateway 143
- IBM Communications Server for NT 139, 140, 141
- IBM Communications Server for Windows NT 155
- IBM Interactive Network Dispatcher 48
- ICSS
  - See Internet Connection Secure Server (ICSS)
- independent LUs
  - See VTAM
- Informix 50
- Internet 1, 2, 17, 20, 22, 32, 36, 43, 44, 47, 48, 60, 63, 65, 66, 71, 74, 77, 80
  - connection 24
  - Internet service provider (ISP) 65
  - security 66
  - Wild West Web 67
- Internet Connection Secure Server (ICSS) 3, 8, 47, 62, 69, 71, 78, 83, 87, 94, 96, 97, 106, 107, 113, 114, 121, 124, 125, 133, 138, 139, 140, 141, 143, 184, 189, 227, 228, 230

Internet Connection Secure Server (ICSS) (*continued*)

- access logging 134
- administration and configuration forms 87, 88
- cache, memory 134
- capacity planning 134
- certificate 89
- certificate request 89, 118
- certification authority 87
- concurrent users 64, 65
- configuration 99, 113, 126, 271
- Configuration and Administration Forms 114
- configuration file 29
- connection speed 65
- Create Keys 114
- encryption key 89, 114
- file sizes 63
- file types 63
- hits per day 63, 65
- httpd.cnf file 183
- installation 83, 271
- key management 92, 119
- key ring file 86, 90, 113, 114, 118
- key ring password 89, 91, 92, 93, 114, 119, 120
- memory cache 134
- performance 134
- private keys 87
- public keys 87
- receive certificate 91, 118
- S-HTTP 90, 117
- security 89, 134
- security certificate 87
- security configuration 88, 90, 117
- self-signed certificate 87, 88, 113
- SSL 90, 97, 117, 125
- target audience 65
- trusted root key 119
- trusted root keys 87, 92
- uninstall 103, 134
- Web server skills 43
- Windows NT service 125
- WWW transaction 64, 66

inventory 33

**J**

- Java 6, 43, 45, 46, 101, 128
- JavaScript 43, 46, 100, 127
- JPEG format 22

**L**

- LEGACYCICS.DLL file 189, 194, 208, 237
- legacyMQ.dll file 238, 244, 252
- legal implications 77
- LEGCICS.C file 195
- LEGCICS.H file 197
- LEGMQ.C file 244
- LEGMQ.H file 246

- local laws 77
- Lotus Approach 35, 50, 106, 108, 135

**M**

- MAKECICS.MAK file 207
- MAKEMQ.MAK file 249
- MAKEREC.MAK file 252
- mall 2, 4, 11, 35, 36, 49, 77
- MALL\_ERR\_LEGACY\_CICS.D2W file 188
- MALL\_ERR\_LEGACY\_MQ.D2W file 237
- MALL\_ERR\_REG\_LEGACYCICS.D2W file 214
- MALL\_ERR\_REG\_LEGACYMQ.D2W file 258
- MALLREGFORM.D2W file 208, 252
- MALLREGUPD.D2W file 211, 255
- manufacturer 32, 35, 36
- market research 18
- marketing 43, 45
- mass import utility 50
- MasterCard 72
- memory caching 133
- merchant 9, 31, 34, 35, 45, 50, 54, 78, 79
- merchants 68
- Microsoft Access 35
- Microsoft SNA Server 139, 140, 141, 155
- Microsoft Visual C++ 4.2 144, 189, 194, 238, 239, 244
- MQFUNC.C file 247
- MQFUNC.H file 247
- MQSeries 34, 144, 227, 228, 229, 230, 231, 244, 271
  - additional requirements 230
  - API 231, 235, 242, 244, 252
  - API name 239
  - C source files 244
  - channel initiator 234
  - compile the DLL 238, 239
  - compile the receiver program 238
  - default MQSeries object 232, 233
  - default Queue Manager 232, 233
  - DLL name 239
  - exception macro 239, 242
  - header files 244
  - include files 261
  - initial configuration 232, 233
  - initiation queue 234
  - local queue 234
  - local transmission queue 233
  - make files 244
  - message channels 228
  - MQI channels 228
  - MQI Client 227
  - MQSeries application 227
  - MQSeries client 227, 228
  - MQSeries environment 227
  - MQSeries error 241
  - MQSeries server 227, 228, 231, 232
  - MQSeries Setup panel 231, 232
  - MQSeries User field 231, 237
  - process 234, 235

### MQSeries (continued)

- Queue Manager 228, 233, 234, 235, 241
- receiver application 234
- receiver channel 235
- receiver system 232
- remote queue 233
- sender channel 234, 235
- sender system 231
- SNA support 227, 228
- source code 244, 252
- TCP/IP support 227, 228
- trigger event 235
- trigger message 234
- xmit queue 233, 234
- MS\_NUM\_CHILD parameter 134
- mserver.ini file 125, 126, 129, 134
- mserverd program
- MSREGADD.INC file 217, 261
- MSREGFORM.D2W file 188, 237
- MSREGFORM.INC file 217, 261
- MSREGLABELS.INC file 223, 267
- MSREGUPDFORM.D2W file 188, 237
- MSVARMAP.INC file 226, 270
- multimedia 43, 45, 64
- multiple parents 18
- multiple systems 48
  - availability: 48
  - database security 48
  - fault tolerance 48
  - intelligent routers 48
  - performance 48
- Multipurpose Internet Mail Extensions (MIME) 68

## N

- Net.Commerce Administrator 3, 8, 18, 53, 100, 128, 135
  - Administration web server 62
- Net.Commerce Site Manager 3, 4, 36, 101, 128
  - access control 4
  - mall front 4
  - shipping providers 4
  - shopper information 4
  - store records 4
  - task management 4
- Net.Commerce Store Manager 3, 5, 54
  - customer information 6
  - product categories 6
  - product information 5
  - shipping services 5
  - shopper groups 5
  - store information 5
- Net.Commerce Template Designer 3, 4, 6, 13, 21, 101, 128
- Net.Commerce APIs
  - See Net.Commerce Server
- Net.Commerce database 3, 12, 19, 32, 33, 35, 48, 49, 50, 59, 96, 100, 127, 131, 135, 136
  - access 48, 50

### Net.Commerce database (continued)

- access protection 50
- APIS table 14
- backup 60
- cleaning 49
- consistent state 60
- crash recovery 60
- creation 96, 112
- database recovery 59
- database structure 50
- database tables 52
- demo mall 101
- DEMOMALL database 102, 129, 135
- Entity Relationships(ER) Model 51
- inconsistent state 60
- log files 49, 60, 61
- MACROS table 14
- MSERVER database 50, 102, 129
- MSHIPMODE table 28
- non-recoverable database 60
- ODBC 50
  - ODBC access 131
  - offline backup 60
  - online backup 60
- ORDERPAY table 28, 57
- ORDERS table 28, 57
- ORDPAYMTHD table 28, 57
- ORLOCK column 28
- performance 61
- point of recovery 60
- recoverable database 60
- Referential Integrity (RI) 51
- restore 60, 63
- restore recovery 60
- retained logging 61
- roll-forward 63
- roll-forward recovery 60
- server 126
- SHADDR table 28, 187, 236
- SHIPTO table 28, 57
- SHOPPER table 12, 144, 231
- SHOPPING table 28, 57
- transactions 49
- unusable state 60
- usable state 60

### Net.Commerce Server 3, 47, 48, 62, 63, 83, 97, 106, 125, 131, 143, 186, 188, 189, 230, 235

- Access Control page 96, 124
- Administration Web Server 94, 97, 121, 128
- API 144, 164, 178, 186, 192, 194, 208, 231, 235, 242, 244, 252
- API error codes 188, 237
- API Name 190, 239
- API signature 187, 236
- API task 186, 235
- APIs 14, 34, 36, 43, 45, 80
- AUDIT\_REG API 186, 187, 235, 236
- BAD\_REG\_MODIFY exception task 187

Net.Commerce Server (*continued*)

- category pages 21, 30
- commands 11, 14, 186, 188, 235, 237
- components 97
- configuration 94, 121, 271
- configuration files 97, 126
- customize 186
- database machine 126
- Database Management page 96, 124
- demo mall 30, 101, 128, 144, 231
- demo mall home page 103, 130
- DLL Name 190, 239
- exception conditions 187, 188, 236, 237
- exception macro 164, 188, 189, 190, 214, 237, 239, 242
- Fix Pak #1 81, 83, 84, 86, 102, 104, 107, 130, 230
- FixPak #1 143
- installation 86, 271
- log\_id parameter 187, 236
- macro 38, 178
- macro task 188, 237
- macros 11, 13, 14, 21, 23, 27, 32, 36, 45, 64, 100, 127, 208, 216, 237, 252, 261
  - %DEFINE section 13
  - %HTML\_INPUT section 14
  - %HTML\_REPORT section 14
  - %SQL section 13
- mall home pages 28
- mserverd processes 134
- mserverd program 13
- multiple systems configuration 48
- Net.Commerce Administration Web server 101
- Net.Commerce Build M29 107
- Net.Commerce server daemon 3
- Net.Commerce server director 3, 11
- Net.Commerce Server Manager 95
- Net.Commerce Server Manager page 122
- non-UI commands 12
- nph-msrvr program 11, 12
- order form 32, 41
- order pages 21
- parameter values 187, 188, 236, 237
- pass parameter 187, 236
- performance 134
- product pages 21, 22, 30, 33, 38
- REG\_NEW macro 188, 237
- REG\_UPD macro 188, 237
- registration form 190
- registration macro 178
- registration page 31
- Server Control page 96, 124
- Server Machine page 124
- SHIPTO command 28
- SHIPTO/UPDATE macro 25
- shut down 129
- single system configuration 47
- staging server 33
- startup 97, 124

Net.Commerce Server (*continued*)

- store home pages 28
- System Configuration page 95, 123
- target audience 65
- UI commands 11
- uninstall 103, 134
- url parameter 187, 236
- ver\_pass parameter 187, 236
- Windows NT service 126
- Net.Commerce system 2, 8, 11, 66, 81, 133, 134, 138, 139, 140, 141, 143, 227, 228, 230
- Net.Commerce Version 1.0 107
- Net.Commerce Version 1.1 81, 107
- Net.Commerce Version 2 33, 50, 134
- Net.Data 11, 13, 14, 41, 43, 99, 127, 208, 252
- netc.pwd file 97, 125
- netccur file 102
- netccur.dll file 129
- network dispatcher 35
- nph-msrvr program
  - See Net.Commerce Server

## O

- ODBC 50, 131, 135
- ODBC Data Source 135
- offerings 24, 29
- online store 2, 5
- Oracle 50
- order fulfillment 23, 33
- order process 24, 27, 31, 36
- ordering process 25, 27, 35

## P

- page file size 133
- PaintShop Pro 22
- payment 28, 57, 72
- payment data 67
- payment information 25
- payment method 32
- payment process 32
- payment systems 44
- payment transaction 72
- performance 133
- Personal Communications 139, 140, 141, 143, 144, 155, 163
  - 3270 emulation 155
  - 3270 terminal emulation session 162, 163
  - adjacent CP name 158
  - APPC 155
  - APPN network node 157
  - Block ID 157
  - Class of Service name 160
  - Communications API 155
  - configuration 155, 156, 158, 160
  - Configure Local System window 162
  - CP-CP sessions 149
  - CPNAME 157

Personal Communications (*continued*)

- Customize Communications - 3270 Host window 163
- Customize Communications window 161, 163
- Define a LAN Connection window 157
- Define a LAN Device window 157
- Define a Mode window 160
- Define a Partner LU 6.2 window 159
- fully qualified CP name 157
- IEEE 802.2 LAN interface 155
- LLC2 protocol driver 155
- LU62APPC log mode 151, 160
- Maximum LL record size 159
- Maximum RU size 154, 159, 160
- NETID parameter 157, 159
- Node Definition window 156
- owning Control Point 159
- Physical Unit ID 157
- Select SNA Resources window 161
- service mode 159
- SNA Node Configuration 156, 162

photos 37

PKZIP 107

planning 36, 43

products 17, 18, 22, 23, 24, 27, 28, 29, 30, 31, 32, 34, 36, 38, 45, 49, 55, 67

Prologistics.Merchant 34

project management 43

## R

RECEIVER.C file 250

RECEIVER.EXE file 235, 238, 239

RECEIVER.LOG file 235

RECEIVER.TST file 234

Referential Integrity (RI)  
See Net.Commerce database

registered shoppers 29

registered users 31

registration 28, 31

registration process 29

## S

S-HTTP  
See Security

SAP R/3 34

Secure Electronic Transactions (SET) 8, 10, 43, 45, 72

- 1024-bit encryption 73
- 2048-bit encryption 73
- acquirer 72
- acquirer payment gateway 73
- authentication 9
- capture process 9
- cardholder 9, 72
- certification authority 73
- confidentiality 9
- cryptographic techniques 9

Secure Electronic Transactions (SET) (*continued*)

- encryption certificates 73
- integrity 9
- interoperability 9
- ISO 8583 standard 73
- issuer 73
- merchant 9, 72
- payment authorization 9
- payment gateway 9
- public key certificates 73
- public key cryptography 73
- purchaser 9
- secure transactions 9
- SET protocol 32, 79

Secure Sockets Layer (SSL)  
See Security

Security 2, 35, 43, 48, 66, 69, 76, 89, 114

- 1024-bit encryption 73
- 128-bit key 71
- 2048-bit encryption 73
- 40-bit encryption 71
- 40-bit key 78
- 40-bit keys 79
- 40-bit SSL protocol 71
- 56-bit key 77
- 56-bit keys 79
- access control 67
- accountability 68
- asymmetric key encryption methods 69, 70
- authentication 67
- Base64 encoding 68
- bastion 74
- capturing data 66
- certificate 9
- certificates 72
- certification authority 72, 73, 87
- client authentication 71
- cookies 46
- cryptographic techniques 9
- Cryptolope 18
- de-militarized zone (DMZ) 74
- DES 70, 77, 78, 79
- dictionary attack 67
- e-mail attack 67
- encryption 68
- encryption certificates 73
- encryption key 89, 114
- encryption methods 68
- encryption technologies 71
- filtering routers 74
- firewall 48, 67, 74
- hardware-based cryptographic solutions 76
- hash functions 70
- HTTP basic authentication 69
- HTTP basic security 68, 69
- IBM 4755 Cryptographic Adapter 76
- IDEA 70
- identification 67

Security (*continued*)

- inner firewall 74, 76
- integrity 68, 71
- MD2 70
- MD5 70
- non-repudiation 68, 76
- outer firewall 74, 75
- privacy 68, 71
- private keys 87
- proxy application 74, 75
- public key certificate 71
- public key certificates 73
- public key cryptography 73
- public keys 87
- public/private key encryption methods 70
- RC2 78
- RC4 70, 78
- RSA 70
- S-HTTP 69, 90, 117
- secure network 74
- secure transactions 9
- security certificate 87
- self-signed certificate 87, 113
- self-signed certificates 46
- server authentication 71
- SET protocol 9
- SmartCards 76
- spoofing 67
- SSL 2, 3, 8, 46, 69, 71, 72, 90, 117
- SSL Handshake Protocol 69
- SSL Record Protocol 69
- symmetric key encryption methods 69
- system security loopholes 67
- Token Devices 76
- Trojan Horse attack 67
- trusted root keys 87
- US export regulations 71
- VeriSign 72
- SENDER.C file 247
- SENDER.H file 248
- SENDER.LOG file 234
- SENDER.TST file 233
- separate machines 35
- server memory 133
- Service Pak 2 for Windows NT 144
- service provider 44
- services file 111, 126, 180, 182
- SESSION\_ID cookie 12
- SET
  - See Secure Electronic Transactions (SET)
- SFS
  - See CICS
- shipping 33, 49, 56
- shipping information 28, 32
- shipping provider 32, 35, 36
- shopper 17, 19, 21, 23, 24, 27, 29, 31, 32, 49, 50, 53, 187, 236

- shopper groups 5, 31, 50, 54, 67
- shopping cart 21, 23, 28, 30, 31, 32, 35, 38, 57
- shopping process 3, 11, 23, 25, 28, 29, 31, 32, 43
- shopping site 44, 45, 47, 69, 76, 79
- shopping trip 24, 27, 29, 31, 32
- single system 34
- Site Manager
- SMIT 81, 83, 86
- SNA Server for AIX 138
- special offers 29
- specials 36
- specific skills 43
- SQL 6, 11, 13, 43
- SSL
  - See Security
- staging server
  - See Net.Commerce Server
- store 2, 4, 11, 21, 23, 25, 27, 29, 31, 32, 34, 35, 37, 43, 49, 54, 77
- Store Manager
  - See Net.Commerce Administrator
- Structured File Server (SFS)
  - See CICS
- Sybase 50

**T**

- tax calculation 33, 59
- tax data 28, 57
- Taxware 34, 59
- team of specialists 43
- Template Designer
  - See Net.Commerce Administrator
- TESTCICS.EXE file 189
- Transaction Server for AIX
  - See CICS
- Transaction Server for NT
  - See CICS
- Transaction Server for OS/2
  - See CICS
- transparent color
  - See GIF format

**U**

- US export regulations 77
- users 53

**V**

- Vinca StandbyServer for NT 48
- VISA 72
- VTAM
  - active APPN configuration 150
  - active CDRSC major node 150
  - active SNA independent LU 6.2 configuration 149, 151
  - active SNA PU configuration 149
  - active VTAM APPN directory entry 151

VTAM (*continued*)

- adjacent CP name 158
- adjacent link station 149
- APPN functions 147
- APPN network node 157
- CDRDYN=YES option 147
- CICS application major node 151
- configuration services XID exit 147, 149
- CP-CP sessions 149
- CPNAME 147
- cross-domain resource (CDRSC) 149
- DYNADJCP=YES option 149
- dynamic definitions of adjacent CPs 149
- dynamic PU definition 147
- DYNLU=YES option 147
- IDBLK 147
- IDNUM 147, 149
- independent LUs 149
- interchange node 147
- LU62APPC log mode 151, 160
- mode table entry 152
- NETID parameter 159
- Netview NCCF 145
- NODETYPE=NN option 147
- primary LU (PLU) 152
- request units (RU) 152
- RUSIZES parameter 152, 154
- secondary LU (SLU) 152
- SNA service mode 152
- SSCPNAME parameter 159
- start options 145, 157, 159
- subarea functions 147

World Wide Web (WWW) (*continued*)

- professional Web pages (*continued*)
  - headers 21
  - home page 29, 38
  - images 21, 22, 44
  - layout 45
  - logos 21, 44
  - slogans 21
  - static 43
- static pages 63
- target audience 65
- Web browser 46, 68, 88, 114, 121, 128, 183, 190
- Web server 63, 65, 76
- Web site 47
- WWW transaction 64, 66

WWW

- See World Wide Web (WWW)

## W

- Windows NT Service Pak 104
- WINZIP 107
- World Wide Web (WWW) 1, 6, 24, 33, 45
  - browser plug-ins 21
  - commerce 1
  - dynamic pages 63, 134
  - HTTP basic security 68
  - integration 1
  - presence 1
  - professional Web pages 20
    - animation 21
    - basic layout 21
    - colors 21
    - demo mall home page 103, 130
    - dependent links 23
    - design 21, 23, 43
    - different layouts 23
    - dynamic 43
    - expired links 23
    - external links 23
    - fonts 21
    - footers 21
    - graphic designer 20
    - graphical design 45
    - graphical design skills 43





---

## ITSO Redbook Evaluation

Integrating Net.Commerce with Legacy Applications  
SG24-4933-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to [redeval@vnet.ibm.com](mailto:redeval@vnet.ibm.com)

**Please rate your overall satisfaction** with this book using the scale:  
**(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)**

**Overall Satisfaction** \_\_\_\_\_

**Please answer the following questions:**

Was this redbook published in time for your needs? Yes\_\_\_\_ No\_\_\_\_

If no, please explain:

---

---

---

---

What other redbooks would you like to see published?

---

---

---

**Comments/Suggestions:**      **( THANK YOU FOR YOUR FEEDBACK! )**

---

---

---

---

---



This soft copy for use by IBM employees only.

Printed in U.S.A.

SG24-4933-00

