



---

**TPC Benchmark™ W  
Full Disclosure Report  
DELL PowerEdge 6400/900MHz with  
PowerEdge 1550 using Volera Excelerator 2.1 and  
PowerApp.Web 120**



**First Edition**

**Submitted for Review**

**December 19, 2001**

# Table of Contents

## Preface

## Executive Summary

<b>0</b>	<b>Introduction...</b>	<b>1</b>
<b>1</b>	<b>Web Object and Logical Database Design...</b>	<b>6</b>
<b>2</b>	<b>Web Interactions and Workload...</b>	<b>8</b>
<b>3</b>	<b>Database Transaction and System Properties...</b>	<b>10</b>
<b>4</b>	<b>Scaling and Database Population...</b>	<b>17</b>
<b>5</b>	<b>Performance Metrics and Response Times...</b>	<b>18</b>
<b>6</b>	<b>SUT, RBE, and Networking ...</b>	<b>29</b>
<b>7</b>	<b>Pricing...</b>	<b>32</b>
<b>8</b>	<b>Audit Related Items...</b>	<b>33</b>

## Appendices

**Appendix A – Application Source Code**

**Appendix B – Configuration and Tuning**

**Appendix C – Database Design**

**Appendix D – HTML Code**

**Appendix E – Space Calculations**

**Appendix F – Price Quotes**

## **First Printing, December 19, 2001**

Dell believes that the information included in this document is accurate as of the publication date. The information in this document is subject to change without notice. Furthermore, Dell is not responsible for any errors contained within this document.

The pricing information given in this FDR is accurate as of the publication date, December 19, 2001 and is generally available.

Benchmark results are highly dependent upon workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result for these and other factors. Therefore, TPC Benchmark W should not be used as a substitute for a specific customer application benchmark when critical capacity planning and/or product evaluation decisions are contemplated.

All performance data contained in this report were obtained in a rigorously controlled environment. Actual performance experienced by a particular customer may vary due to differences in system layout and configuration, hardware and/or software revision levels, and background system activity. The content of this document is for informational purposes only.

Copyright 2001 Dell

All rights reserved. Permission is hereby granted to reproduce this document in whole or in part provided the copyright notice printed above is set forth in full text or on the title page of each item reproduced.

PowerApp, PowerConnect, PowerEdge and PowerVault are registered trademarks of Dell.

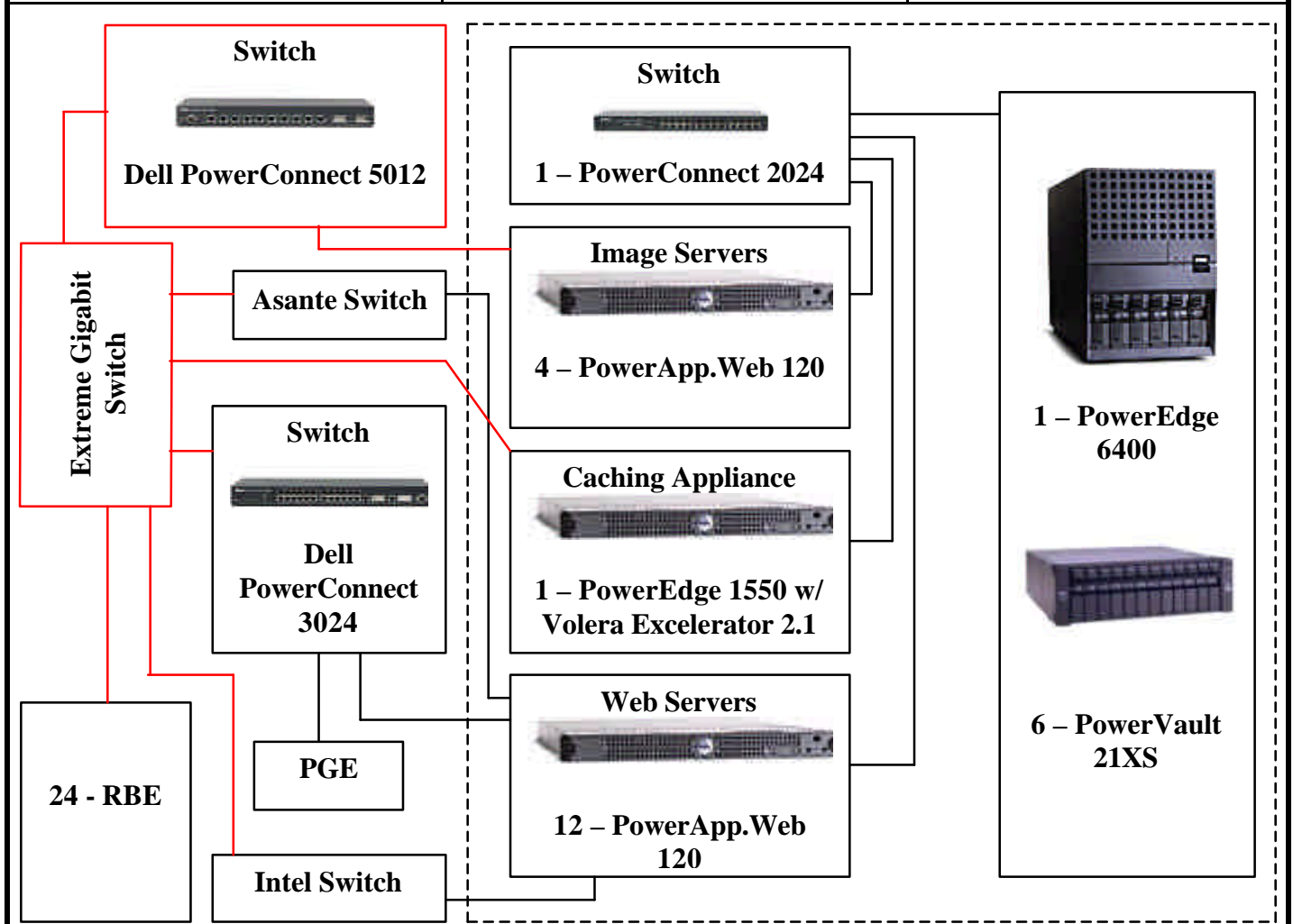
Microsoft, Windows 2000 and SQL Server are registered trademarks of Microsoft Corporation.

TPC Benchmark, TPC-W, WIPS, WIPSo and WIPSo are trademarks of the Transaction Processing Performance Council. Where appropriate, this document contains clauses copied from the Specification for TPC-W Revision 1.7 by permission of the Transaction Processing Performance Council.

Intel and Pentium are registered trademarks of Intel Corporation.


Other product names mentioned in this document may be trademarks and/or registered trademarks of their respective companies.

<b>Dell POWER</b> PowerApp   PowerEdge   PowerVault	<b>PowerEdge 6400/900MHz with PowerApp.Web 120</b>	<b>TPC-W Revision 1.7</b>
<b>Price Performance</b> <b>\$25.70 per WIPS@10,000</b>	<b>Performance</b> <b>6622.4 WIPS@10,000</b>	<b>Available</b> <b>December 19, 2001</b>
<b>Total Cost</b> \$170,039	<b>WIPSo@10,000</b> 1171.7	<b>WIPSp@10,000</b> 6636.5
<b>Scale Factor</b> 10,000 Items	<b>Number of Users</b> 50,000	<b>Number of Systems</b> 18
<b>Web Server</b> Microsoft Internet Information Server 5.0	<b>Application Server</b> ISAPI Application	<b>Database Manager</b> Microsoft SQL Server 2000 Standard Edition with SP1
<b>Load Balancer</b> Microsoft Windows 2000 DNS Server	<b>Other Software</b> Microsoft SQL Server 2000 Full Text Search	<b>Web Cache</b> PowerEdge 1550 with Volera Excelerator 2.1



Function	# of Systems	System	OS	Processors	Memory
Web Server	12	PowerApp.Web 120	Microsoft Windows Powered	2 - Pentium III 1000MHz/256kB L2	512MB
Image Server DNS	4	PowerApp.Web 120	Microsoft Windows Powered	2 - Pentium III 1000MHz/256kB L2	1GB
Web Cache	1	PowerEdge 1550	Volera Excelerator 2.1	1 - Pentium III 1.26GHz/512kB L2	512MB
Database Server	1	PowerEdge 6400	Windows 2000 Advanced Server	4 - Pentium III Xeon 900MHz/2MB L2	4GB

DELL	PowerEdge 6400/900MHz with PowerApp.Web 120		TPC-W Revision 1.7					
			Report Date 12/19/01					
Price Performance	Performance		Availability Date					
\$25.70/WIPS	6622.4 WIPS		12/19/2001					
Third Party			Extended Price	3 Yr Maintenance Price				
Description	Part No.	Brand	Price	Unit Price	Quantity			
<b>Web Server Hardware:</b>								
SYSTEM: PowerApp.Web 120 includes 1GHz/133MHz FSB	220-9998	1	1	\$1,101	12	\$13,212		
PROC: 2nd 1GHz/133MHz FSB	311-5512	1	1	\$599	12	\$7,188		
MEM: 512MB (4x128MB)	311-1225	1	1	\$408	12	\$4,896		
DISK: 1x18GB SCSI	340-2468	1	1	\$249	12	\$2,988		
NIC: Pro 100+ Adapter	430-4937	1	1	\$59	12	\$708		
NIC: Dual Integrated Pro 10/100+ Adapters	Included in SYSTEM.					\$0		
ACCESSORY: Rail Racks	310-5986	1	1	\$249	12	\$2,988		
MAINT: 3 Years 4Hour-7x24, Unisys (Initial/2Yr Ext.)	900-6350/900-6362	1	1	\$774	12		\$9,288	
MONITOR: DELL E551 15"	320-0960	1	1	\$149	1	\$149		
<b>Web Server Software:</b>								
OS: Windows for PAW Server 2K	420-0802	1	1	\$240	12	\$2,880		
ACC: Microsoft Visual C++ Professional 6.0	048-00317	2	2	\$549	1	\$549		
<b>Web Cache Appliance Hardware:</b>								
<b>Web Server Subtotal:</b>						<b>\$35,558</b>	<b>\$9,288</b>	
SYSTEM: PowerEdge 1550 includes 1.26GHz/133MHz	220-6497	1	1	\$935	1	\$935		
MEM: 512MB (4x128MB)	311-1225	1	1	\$408	1	\$408		
DISK: 1x18GB SCSI	340-2468	1	1	\$249	2	\$498		
ACCESSORY: Rail Racks	310-5986	1	1	\$249	1	\$249		
NIC: Intel Pro 1000 Server Adapter Gigabit (SX Fiber)	PWLA8490	3	3	\$435	1	\$435		
SPARES: Intel Pro 1000 Server Adapter Gigabit (SX Fiber)	Spares priced to spec.		3	3	\$435	2	\$870	
MAINT: 3 Years 4Hour-7x24, Unisys (Initial/2Yr Ext.)	900-6360/900-6362	1	1	\$774	1		\$774	
MONITOR: DELL E551 15"	320-0960	1	1	\$149	1	\$149		
<b>Web Cache Appliance Software:</b>								
OS: Volera Excelsator 2.1	06-20011	1	1	\$4,495	1	\$4,495		
MAINT: Volera Excelsator 2.1 3Yrs 24x7 Platinum Service	06-20057	1	1	\$989	3		\$2,967	
<b>Web Cache Subtotal:</b>						<b>\$8,039</b>	<b>\$3,741</b>	
<b>Image Server Hardware:</b>								
SYSTEM: PowerApp.Web 120 includes 1GHz/133MHz FSB	220-9998	1	1	\$1,101	4	\$4,404		
PROC: 2nd 1GHz/133MHz FSB	311-5512	1	1	\$599	4	\$2,396		
MEM: 1GB SDRAM,133MHZ,(4X256MB)	311-0896	1	1	\$892	4	\$3,568		
DISK: 1x18GB SCSI	340-2468	1	1	\$249	8	\$1,992		
ACCESSORY: Rail Racks	310-5986	1	1	\$249	4	\$996		
NIC: BROADCOM,1000,NIC,CAT5	430-1980	1	1	\$149	4	\$596		
MAINT: 3 Years 4Hour-7x24, Unisys (Initial/2Yr Ext.)	900-6360/900-6362	1	1	\$774	4		\$3,096	
MONITOR: DELL E551 15"	320-0960	1	1	\$149	1	\$149		
<b>Image Server Software:</b>								
OS: Windows for PAW Server 2K	420-0802	1	1	\$240	4	\$960		
<b>Img Server Subtotal:</b>						<b>\$15,061</b>	<b>\$3,096</b>	
<b>Database Server Hardware:</b>								
SYSTEM: PE6400,900MHZ/2MB,RDNT,BASE	220-1495	1	1	\$6,714	1	\$6,714		
PROC: Quad 900MHz/2MB L2 Cache	311-0939	1	1	\$16,799	1	\$16,799		
MEM: 4GB RAM, 16x256MB RGT DIMMs	311-4115	1	1	\$3,584	1	\$3,584		
BACKPLANE: 8 Bay Split 2X4 BakPln(1"only)	340-5773	1	1	\$249	1	\$249		
CTRL: Mylex ExtremeRAID 2000 Card,64MB	4	4	4	\$1,398	3	\$4,194		
SPARES: Mylex ExtremeRAID 2000 Card,64MB	Spares priced to spec.		4	4	\$1,398	2	\$2,796	
DISK: 18GB, 10k SCSI	340-8808	1	1	\$249	1	\$249		
NIC: Integrated Pro 100+ Adapter	Included in SYSTEM.					\$0		
TAPE: 20/40GB DDS4 Autoloader	340-9167	1	1	\$2,099	1	\$2,099		
MONITOR: DELL E551 15"	320-0960	1	1	\$149	1	\$149		
MAINT: 3 Years 4Hr/7x24, Unisys (Initial/2Yr Ext.)	900-6370/900-6352	1	1	\$1,783	1		\$1,783	
ACCESSORY: Rail Racks	310-0516	1	1	\$349	1	\$349		
POWER: Tower 1400 UPS 110V	310-0284	1	1	\$599	2	\$1,198		
<b>Database Server Software:</b>								
OS: Microsoft Windows 2000 Advanced Server	C10-00475	2	2	\$2,399	1	\$2,399		
APP: Microsoft SQL Server 2000 Standard Edition w/ SP1	228-00968	2	2	\$4,271	4	\$17,084		
MAINT: 3-Year Maintenance for All Microsoft Software	2	2	2	\$2,095	3		\$6,285	
<b>DB Server Subtotal:</b>						<b>\$57,863</b>	<b>\$8,068</b>	
<b>Storage Hardware:</b>								
SYSTEM: PowerVault 21x - 12 HDD Storage System	220-4099	1	1	\$1,253	6	\$7,518		
CTRL: Ultra3 SCSI, Not Clusterable	340-9556	1	1	\$409	7	\$2,863		
POWER: Redundant Power Option	310-6125	1	1	\$399	6	\$2,394		
ACCESSORY: Rack Rail Kit	310-1330	1	1	\$114	6	\$684		
RACK: PE4210,DOORS,GRND,42U,PS	220-4493	1	1	\$1,110	1	\$1,110	\$184	
DISK: 36GB, 10k SCSI	340-6762	1	1	\$499	18	\$8,982		
DISK: 18GB, 10k SCSI	340-1395	1	1	\$249	44	\$10,956		
CBL: 2 Cables,PERC,4M, PV2XXS	310-0313	1	1	\$49	7	\$343		
MAINT: 3 Years 4Hr/7x24, IGS (Initial/2Yr Ext.)	902-4310/902-1512	1	1	\$1,099	6		\$6,594	
<b>Storage Subtotal:</b>						<b>\$34,850</b>	<b>\$6,778</b>	
<b>Connectivity:</b>								
SWITCH: PowerConnect 2024	220-2099	1	1	\$173	1	\$173		
MAINT: 3 Years 4Hr/7x24, Unisys (Initial/2Yr Ext.)	950-1660/950-1662	1	1	\$65	1		\$65	
<b>Connectivity Subtotal:</b>						<b>\$173</b>	<b>\$65</b>	
<b>Total:</b>						<b>\$151,544</b>	<b>\$31,036</b>	
<b>Dell Discounts:</b>						<b>(\$12,616)</b>		
<b>3 Year Cost of Ownership</b>		<b>\$169,964</b>						
<b>WIPS @10,000 Items</b>		<b>6622.4</b>						
<b>\$ / WIPS @10,000 Items</b>		<b>\$25.70</b>						
1 Dell		2 Microsoft		3 www.lanadapters.com		4 www.compuview-ms.com		
<b>Independently Audited by:</b>		<b>Lorna Livingtree</b>		<b>Performance Metrics</b>				

	<b>PowerEdge 6400/900MHz with PowerApp.Web 120</b>	<b>TPC-W Revision 1.7</b>
		<b>Report Date 12/19/01</b>
<b>Price Performance</b>	<b>Performance</b>	<b>Availability Date</b>
<b>\$25.70 per WIPS@10,000</b>	<b>6622.4 WIPS@10,000</b>	<b>December 19, 2001</b>

### Numerical Quantities Summary

#### WIPS Interactions Summary

Interaction	Mix	Count	90% Response
Home	15.99%	1,906,590	0.21
Shopping Cart	11.61%	1,383,822	0.18
Customer Registration	3.00%	358,120	0.09
Buy Request	2.61%	310,601	0.21
Buy Confirm	1.20%	143,407	2.15
Order Inquiry	0.75%	89,739	0.12
Order Display	0.66%	79,013	0.15
Search Request	20.00%	2,383,940	0.17
Search Result	16.99%	2,025,716	0.20
New Products	5.00%	595,865	0.20
Best Sellers	4.99%	594,612	0.21
Product Detail	17.00%	2,026,528	0.12
Admin. Request	0.10%	11,829	0.12
Admin. Confirm	0.09%	10,617	1.96

#### Measurement Intervals Summary

	Shopping	Browsing	Ordering
Ramp-up Time (min)	138.0	55.0	40.0
Length of Measurement Interval (min)	30.0	30.0	30.0
Average Think Time (seconds)	7.35	7.35	7.35
Average User Session Duration (min)	14.4	14.7	14.3
Number of Users	50,000	50,000	9,000

## 0 Introduction

TPC Benchmark™ W (TPC-W) is a transactional web benchmark. The workload is performed in a controlled Internet commerce environment that simulates the activities of a business oriented transactional web server. The workload exercises a breadth of system components associated with such environments, which are characterized by:

- Multiple on-line browser sessions
- Dynamic page generation with database access and update
- Consistent web objects
- The simultaneous execution of multiple transaction types that span a breadth of complexity
- On-line transaction execution modes
- Databases consisting of many tables with a wide variety of sizes, attributes, and relationships
- Transaction integrity (ACID properties)
- Contention on data access and update

The performance metric reported by TPC-W is the number of web interactions processed per second. Multiple web interactions are used to simulate the activity of a retail store, and each interaction is subject to a response time constraint. The store size is chosen from among a set of given scale factors, which is the number of Items in inventory and varies from 1,000 items to 10,000,000 items. The performance metric for this benchmark is expressed in Web Interactions Per Second at a tested scale factor expressed by WIPS@scale factor where scale factor is the number of items in the ITEM table. For example 123WIPS@100000. All references to WIPS in this specification mean WIPS@scale factor.

TPC-W simulates three different profiles by varying the ratio of browse to buy: primarily shopping (WIPS), browsing (WIPSb), and web-based ordering (WIPSo). All references to WIPS (WIPSb, WIPSo) results must include the primary metrics, which are: the WIPS rate, the associated price per WIPS (\$/WIPS), and the availability date of the priced configuration.

### 0.1 Benchmark Sponsor

*A statement identifying the benchmark sponsor(s) and other participating companies must be provided.*

Dell Computer Corporation sponsored this TPC-W benchmark.

### 0.2 Source Code

*The source code of the application program or programs as defined in Clause 1.2.10, as well as all input for product-generated code, must be disclosed.*

Appendix A contains the application source code used.

### 0.3 Settings

*Settings must be provided for all customer-tunable parameters and options which have been changed from the defaults found in actual products, including but not limited to:*

- *Database tuning parameters.*
- *Web server tuning and logging options.*
- *Recovery/commit options.*
- *Consistency/locking options.*
- *Operating system and application configuration parameters.*
- *Compilation and linkage options and run-time optimizations used to create/install applications, OS, DBMS, web server, and/or any other commercial product.*

Appendix B contains the settings.

### 0.4 Diagrams of Measured and Priced Configurations

*Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:*

- *Number and type of processors.*
- *Physical memory actually present on the SUT.*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test.*
- *Number and type of disk units and controllers.*
- *Number of channels or bus connections to disk units, including their protocol type.*
- *Number of LAN (e.g., Ethernet) connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure.*
- *Type and the run-time execution location of software components (e.g., DBMS, web server, application server or program, transaction monitors, etc.)*
- *Number and type of cryptographic processors or cryptographic accelerators, if applicable.*

Diagrams of Dell's tested and priced TPC-W configuration are described as follows:

## 16 – Dell PowerApp.Web 120

12 Dell PowerApp.Web 120 served as web servers running Microsoft IIS 5.0, Microsoft Windows Powered, and ran the tpcw.dll and list.dll ISAPI applications. The other 4 Dell PowerApp.Web servers ran Microsoft Windows Powered and Microsoft IIS 5.0 to serve JPEG images. One of the Dell PowerApp.Web 120 image servers was configured as the primary DNS server while another was used as the secondary DNS server.

<b>Base Model Features and Specifications</b>	
Form factor	Rack
Processor type	Intel Pentium III
Processor speed	1GHz
Processor L2-cache	256kB
Hot swap HD	Yes
Disk controller	Dual-channel integrated U160 SCSI controller
Optical drive	Up to 40X CD-ROM
Network	Dual On-Board 100Mbs Ethernet
<b>Web Server Accessories and Options Selected</b>	<b>Quantity</b>
1GHz/133MHz FSB Pentium III Processor	2
256MB/133MHz FSB SDRAM	2
18GB 10k U160 SCSI HDD	1
Intel Pro 100 NIC	1
<b>Image Server Accessories and Options Selected</b>	<b>Quantity</b>
1GHz/133MHz FSB Pentium III Processor	2
256MB/133MHz FSB SDRAM	4
18GB 10k U160 SCSI HDD	2
Broadcom CAT5 NIC	1

## 1 – Dell PowerEdge 1550

The Dell PowerEdge 1550 ran Volera Excelerator 2.1 Software to provide Internet caching services.

<b>Base Model Features and Specifications</b>	
Form factor	Rack
Processor type	Intel Pentium III
Processor speed	1.26GHz
Processor L2-cache	512kB
Hot swap HD	Yes
Memory	512MB
Disk controller	On-Board Ultra3 (Ultra160) SCSI controller
Network	Dual onboard 100Mbs Ethernet
Optical drive	Up to 40X CD-ROM
<b>Accessories and Options Selected</b>	<b>Quantity</b>
1.26GHz/133MHz FSB Pentium III Processor	1
256MB/133MHz FSB SDRAM	2
18GB 10k U160 SCSI HDD	2
Intel Pro 1000 F	1

## 1 – Dell PowerEdge 6400

The Dell PowerEdge 6400 served as the database server running Microsoft Windows 2000 Advanced Server and Microsoft SQL Server 2000 Standard Edition with Service Pack 1.

<b>Base Model Features and Specifications</b>	
Form factor	Rack
Processor type	Intel Pentium III Xeon
Processor speed	900MHz
Processor L2-cache	2MB
Hot swap HD	Yes
Memory	4GB
Disk controller	On-Board Ultra3 (Ultra160) SCSI controller
Network	On-Board 100Mbs Ethernet
Optical drive	Up to 40X CD-ROM
<b>Accessories and Options Selected</b>	
<b>Quantity</b>	
900MHz/100MHz FSB Pentium III Processor	4
256MB/133MHz FSB SDRAM	16
18GB 10k U160 SCSI HDD	1
20/40GB DDS4 Autoloader	1
UPS	1
Mylex PCI eXtremeRaid 2Int/4Ext Channel Raid Controller 64MB	5 (3 are used and 2 are spared to meet the maintenance requirement.)

## The Measured Configuration Diagram

The measured configuration met the specifications for Clause 6.7.1 Uninterrupted Execution using 40-18GB HDDs for the database volume. The priced configuration includes 3 more 18GB HDDs to meet the 180-day space requirements.

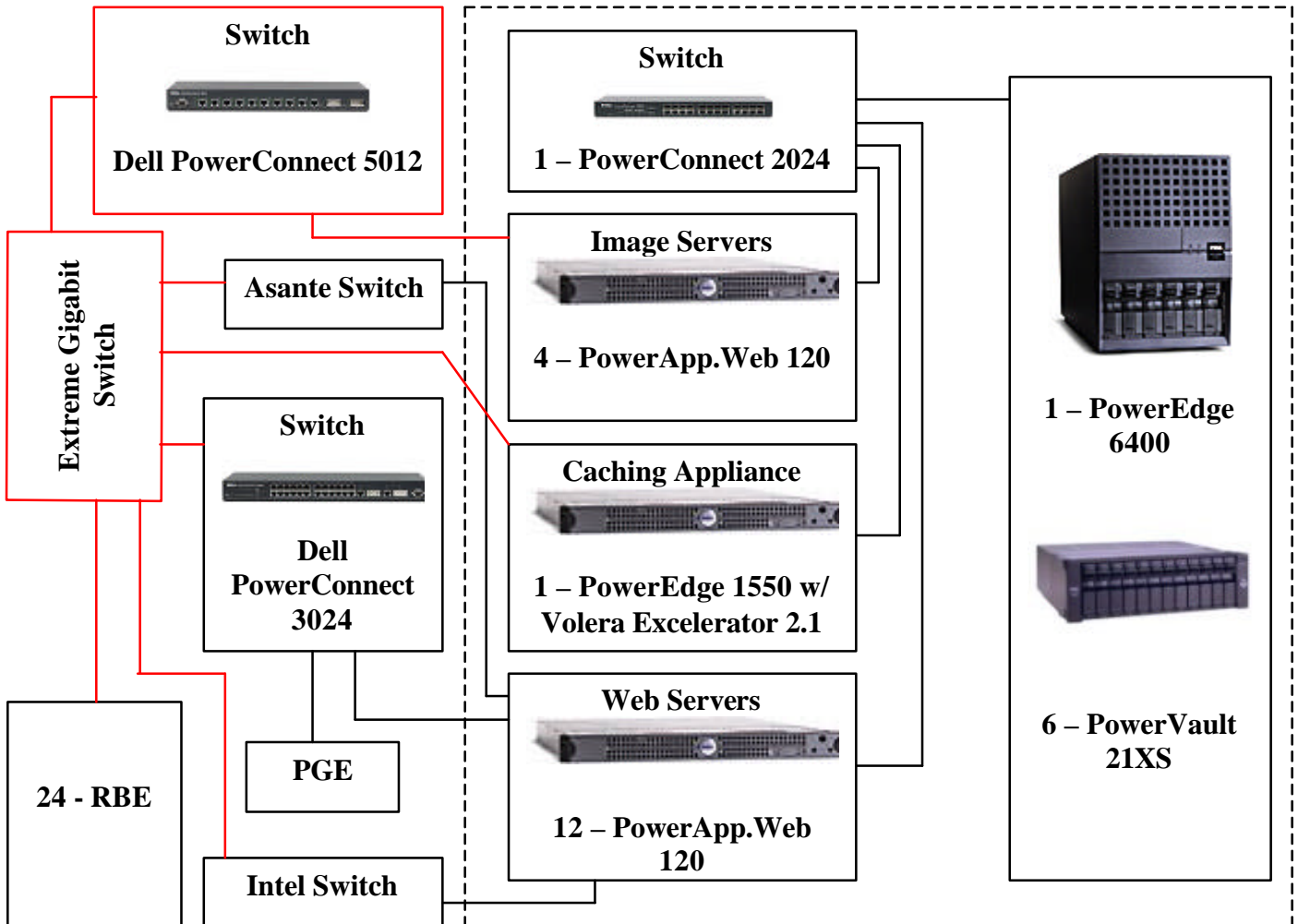


Figure 0.4.1: SUT Diagram

# 1 Web Object and Logical Database Design

## 1.1 Location of JPEG and GIF Images

*The location of the JPEG and GIF images used in the benchmark must be disclosed. This information must include on which system and in what logical structure the images are stored (database or file system).*

The JPEG images were stored on the image servers. The GIF images were located on the web servers that contain the TPC-W ISAPI code. All the JPEG and GIF images were stored on a file system.

## 1.2 Database Table Definitions

*Listings must be provided for all table definition statements and all other statements used to set up the database (8.1.2.1).*

The database table definitions are detailed in Appendix C.

## 1.3 Database Organization

*The physical organization of tables and indices, within the database, must be disclosed.*

The following figure discloses the disk configurations. Appendix C also contains the physical organization of tables and indices.

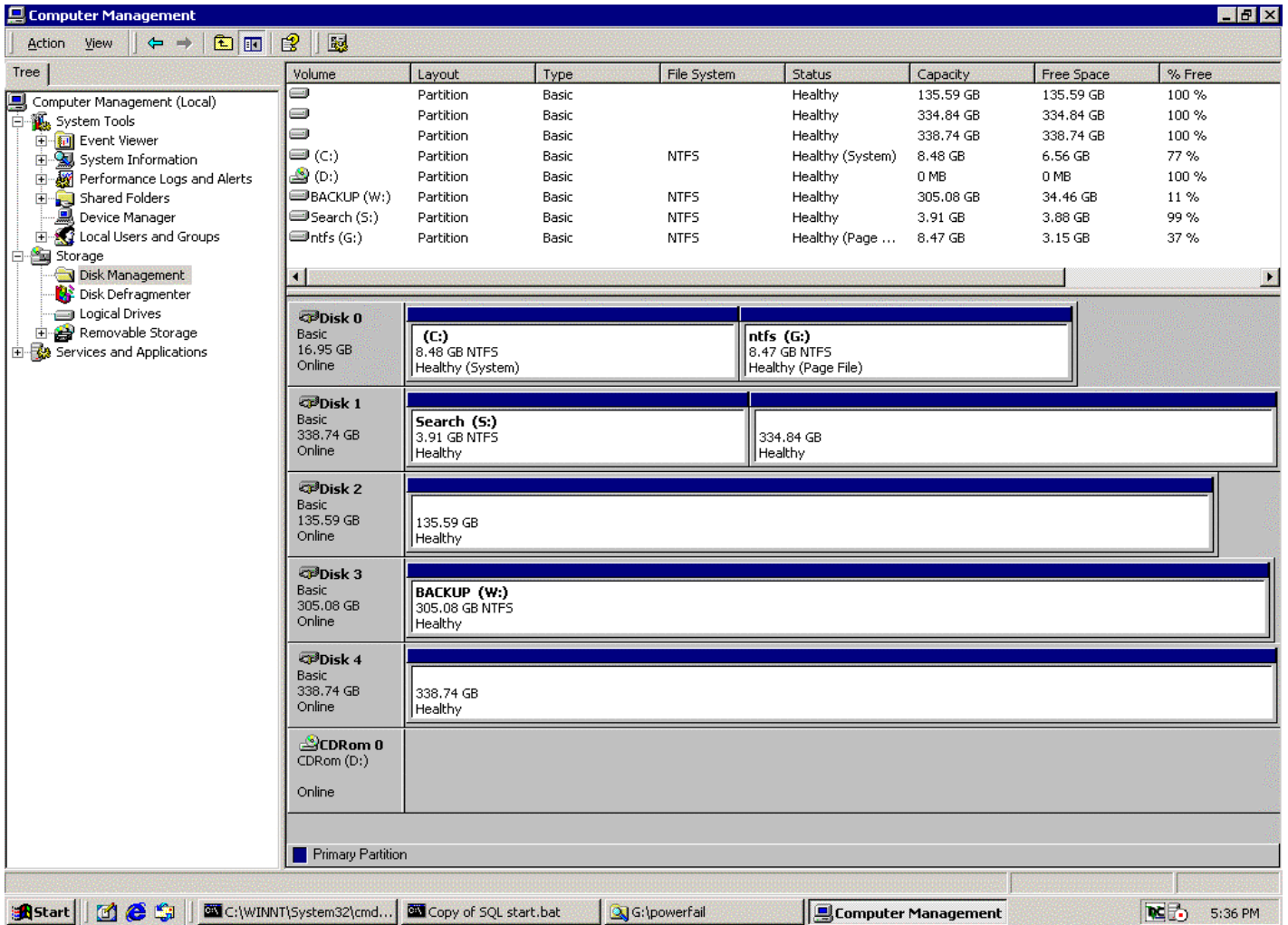


Figure 1.3.1: Disk Manager Screen Capture

## 1.4 Partitioning

Any horizontal or vertical partitioning of tables or rows in the TPC-W benchmark must be disclosed (see Clause 1.6.4 and 1.6.5). Replication of tables, if used, must be disclosed (see Clause 1.6.6).

No vertical or horizontal partitioning was used.

## 1.5 Duplicate/Additional Attributes

Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance (see Clause 1.6.7).

Additional tables SC\_LOCK, ORDER\_LOCK, CUST\_LOCK and ADDR\_LOCK were used to keep track of SC\_SHOPPING\_ID, O\_ID, C\_ID and ADDR\_ID. See Appendix C for details.

SEARCH\_ITEM and SEARCH\_AUTHOR tables were copied from the ITEM and AUTHOR tables after the initial population and used for the text searches based on I\_TITLE and A\_LNAME. See Appendix C for details.

## 2 Web Interactions and Workload

### 2.1 Random Number Generator

*The method of random number generation must be disclosed.*

The remote browser emulator (RBE) generated the random numbers using the LRAND32 function and unique seeds for each emulated browser (EB).

### 2.2 Emulated Browsers

*The number of Emulated Browsers used in the measured intervals must be disclosed.*

50,000 EBs were used in the TPC-W benchmark run.

### 2.3 User Sessions

*The percentage of User sessions started during each Measurement Interval with known Customers must be disclosed with at least 1 decimal place.*

Disclosed below are the percentages of User sessions started during each Measurement Interval with known customers for WIPS, WIPsb, and WIPSo.

WIPS	WIPsb	WIPSo
79.9%	79.9%	80.3%

### 2.4 Shopping Cart Implementation

*The implementation of the CART data must be explained in sufficient detail to allow a replication of the benchmark including the durability of the CART data over a single point of failure.*

The Cart data structure was implemented as two tables in the database. See Appendix C for table and stored procedure definitions.

### 2.5 Security Requirements

*A description of how the security requirements were met as defined in Clause 2.2.1, must be disclosed including key lengths.*

The RBE and the SUT use OpenSSL libraries to transfer SSL requests between the RBE-SUT and the SUT-PGE. Secure transactions on the web servers are internally encrypted and decrypted using Microsoft Internet Information Server. The following SSL parameters were used:

- Protocol: SSL version 3
- Cipher: RC4
- Cipher strength: 128
- Hash: MD5
- Hash Strength: 128

Key Exchange: RSA  
Key exchange strength: 1024

## **2.6 HTML Code**

*The HTML code for one representative web page for each web interaction must be included in the Full Disclosure Report.*

See Appendix D for the HTML code used for each web interaction.

## **2.7 Programming Language and API**

*A statement must be provided describing the development language(s) and the types of APIs used between commercial components to implement the interactions. This includes, but is not limited to, the interfaces to the database server, web server, commerce package or application, or any other commercial product used. For example: The commerce package calls application code through standard CGI interface. The application is written in C and interfaces to the database with ODBC calls that initiate stored procedures.*

The RBE uses standard HTTP 1.1 to communicate with the SUT. The web servers use tpcw.dll and list.dll ISAPIs written in C++. In addition, the DLLs use standard Windows Socket API to communicate with the application servers. Next, the application code invokes MS SQL stored procedures on the database from embedded C++ code. Finally, the stored procedures are written in embedded SQL code.

## 3 Database Transaction and System Properties

### 3.1 ACID Test Results

*The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met.*

ACID test results are disclosed in the following sections.

#### 3.1.1 ACID Test

*All interactions with any database maintaining the tables defined in Clause 1 must be made through a database transaction supporting full ACID properties, as defined in Clauses 3.1.2 to 3.1.4. .*

The Dell TPC-W database interactions support full ACID properties as defined in Clauses 3.1.2 to 3.1.4 and was witnessed by an independent auditor.

#### 3.1.2 Atomicity

*The system under test must guarantee that database transactions are atomic. Within a database transaction the system will either perform all individual operations on the data, or will assure that no operations leave any effects on the data.*

##### 3.1.2.1 Atomicity Tests

*The atomicity tests require that the Buy Confirm web interaction be instrumented so that the update to the database may be aborted while in progress, affecting the final outcome of the update, but without affecting the ability of the SUT to complete the web interaction.*

The following steps were performed to validate Atomicity Test 1.

- Step 1. Requested and completed a Best Sellers web interaction and chose an item from the response page.
- Step 2. Requested and completed a Product Detail web interaction for the item chosen in Step 1.
- Step 3. Requested an instrumented Buy Confirm web interaction for the item chosen in Step 1, ordered the item, and completed the web interaction without aborting its updates.
- Step 4. Requested and completed an Order detail web interaction for the order placed in Step 3.
- Step 5. The information presented in the response page from Step 4 matched the order entered in Step 3.

The following steps were performed to validate Atomicity Test 2:

- Step 1. Requested and completed a Search Result web interaction and chose an item from the response page.
- Step 2. Requested and completed a Product Detail web interaction for the item chosen in Step 1.
- Step 3. Requested an instrumented Buy Confirm web interaction for the item chosen in Step 1, ordered the item, and aborted its updates before completion.
- Step 4. Requested and completed an Order Detail web interaction for the item ordered in Step 1.

Step 5. The information presented in the response page from Step 4 made no mention of the order entered in Step 3.

### 3.1.3 Consistency

#### 3.1.3.1 Consistency Property Definition

*The system under test must guarantee that database transactions are consistent. Assuming that the database is initially in a consistent state, the system will ensure that any TPC-W database transaction takes the database from one consistent state to another.*

#### 3.1.3.2 Consistency Conditions

*A consistent state for the TPC-W database is defined to exist when:*

- 1. (I\_A\_ID) is a valid Foreign Key reference to an existing (A\_ID)*
- 2. (C\_ADDR\_ID) is a valid Foreign Key reference to an existing (ADDR\_ID)*
- 3. (O\_C\_ID) is a valid Foreign Key reference to an existing (C\_ID)*
- 4. O\_BILL\_ADDR and O\_SHIP\_ADDR are valid Foreign Key references to an existing (ADDR\_ID)*
- 5. (OL\_I\_ID) is a valid Foreign Key reference to an existing (I\_ID)*
- 6. (OL\_O\_ID) is a valid Foreign Key reference to an existing (O\_ID)*
- 7. (CX\_O\_ID) is a valid Foreign Key reference to an existing (O\_ID)*
- 8. (CX\_CO\_ID) is a valid Foreign Key reference to an existing (CO\_ID)*
- 9. CX\_XACT\_AMT = O\_TOTAL when CX\_O\_ID = O\_ID*
- 10. (ADDR\_CO\_ID) is a valid Foreign Key reference to an existing (CO\_ID)*

#### 3.1.3.3 Initial Consistency

*A TPC-W database, when populated as defined in Clause 4.7, must meet the consistency condition defined in Clause 3.1.3.2.*

The TPC-W database when initially populated met the consistency conditions stated in Clause 3.1.3.2.

#### 3.1.3.4 Consistency Integrity

*If data is replicated, as permitted under Clause 1.6.6, each copy must meet the consistency condition defined in Clause 3.1.3.2. The implementation of the web interactions must ensure that all consistency conditions defined in Clause 3.1.3.2 are maintained without relying on a limited range of input data. But the implementation of the benchmark is not required to maintain these consistency conditions under arbitrary database transactions.*

#### 3.1.3.5 Consistency Test

*The verification of the consistency between the CUSTOMER and ADDRESS tables, between the ORDERS and CUSTOMER tables, and between the ORDERS and CC\_XACTS tables is done as part of each Durability test (see Clause 3.1.5.5).*

Consistency queries were performed before and after the test run to ensure that the consistency conditions were met.

### **3.1.4 Isolation**

*Isolation can be defined in terms of phenomena that can occur during the execution of concurrent database transactions.*

#### **3.1.4.1 Isolation Tests**

*The isolation tests require that several web interactions be modified so that a query or an update to the database may be halted while in progress, without affecting the final outcome of the query or the update, and without affecting the ability of the SUT to complete the web interaction.*

##### **3.1.4.2.1 Isolation Test 1**

*To verify isolation between two TPC-W update transactions, perform the following steps:*

- Step 1. From browser A, request and complete a Best Sellers, a New Products or a Search Result web interaction and choose an item from the response page.*
- Step 2. From browser A, request and complete a Product Detail web interaction for the item chosen in the previous step.*
- Step 3. From browser B, repeat steps 1 and 2 above for the same item.*
- Step 4. From browser A, request an Admin Request web interaction from the Product Detail Page of step 2.*
- Step 5. From browser A, modify the item's image and price and confirm the item's update by requesting by initiating an instrumented Admin Confirm web interaction.*
- Step 6. Interrupt the processing of the Admin Confirm web interaction from browser A after updating the image, but before updating the price.*
- Step 7. From browser B, repeat step 4 above. Verify that the Admin Request web interaction either waits for browser A to resume, or completes. If the Admin Request web interaction completes, then repeat step 5 above, updating the item's thumbnail and price (using a different value than browser A).*
- Step 8. Verify that browser B's Admin Confirm web interaction, if requested in step 7, either waits for browser A to resume, or displays an Admin Confirm Page containing the item's image from step 2 and the price and thumbnail as updated by browser B.*
- Step 9. Resume the processing of the Admin Confirm web interaction interrupted in step 6.*
- Step 10. Verify that browser A displays an Admin Confirm Page containing the item's new image, and either the old thumbnail and the price as updated by browser A, or the new thumbnail and the price as updated by browser B.*

It was verified and witnessed by an independent auditor that browser A displayed an Admin Confirm page with the item's new image and the new thumbnail and price as updated by browser B.

##### **3.1.4.2.2 Isolation Test 2**

*To verify isolation between a TPC-W update transaction and an arbitrary transaction, perform the following steps:*

- Step 1. From a browser, request and complete a Best Sellers web interaction.*

*Step 2. From the same browser, request and complete a Product Detail web interaction for the first item found in the previous step.*

*Step 3. From the same browser, request an Admin Request web interaction from the Product Detail Page of the previous step.*

*Step 4. From the same browser, modify the item's image, thumbnail and price and confirm the item's update by requesting by initiating an instrumented Admin Confirm web interaction.*

*Step 5. Interrupt the processing of the Admin Confirm web interaction after updating the image, but before updating the price and the thumbnail.*

*Step 6. Using a database query and update utility, request an update of the description of the first item found in step 1. Commit this update as soon as allowed by the database. (The update request may hang waiting for the processing of the Admin Confirm web interaction to complete.)*

*Step 7. Resume the processing of the Admin Confirm web interaction interrupted in step 5.*

*Step 8. Verify that the browser displays an Admin Confirm Page containing the item's new image, new thumbnail and new price.*

*Step 9. Using a database query and update utility, verify that the first item found in step 1 now reflects the updates from Step 6 and from the Admin Confirm web interaction.*

It was verified that the browser displayed an Admin Confirm Page containing the item's new image, new thumbnail, and new price. Further, utilizing a database query and update, it was verified that the first item in step 1 reflected updates from step 6 and from the Admin Confirm web interaction.

### **3.1.4.2.3 Isolation Test 3**

*To verify isolation between a TPC-W read-only transaction and an arbitrary transaction, perform the following steps:*

*Step 1. From a browser, request and complete an Order Display web interaction.*

*Step 2. Using a database query and update utility, request an update of the quantity of the first line item of the order displayed in Step 1, but do not commit this update.*

*Step 3. From the same browser, request and complete another Order Display web interaction to retrieve the same order as in Step 1*

*Step 4. Verify that the browser's Order Display web interaction either waits for the update utility to commit, or displays an Order Display Page containing the same information as displayed in Step 1.*

*Step 5. Using a database query and update utility commit the pending updates to the database.*

*Step 6. If the browser was waiting in Step 4, verify that it has resumed and that it displays an Order Display Page containing the new quantity for the first line item. If the browser was not waiting in Step 4, request and complete a new Order Display web interaction to retrieve the same order as in Step 1 and verify that the Order Display Page contains the new quantity for the first line item.*

It was verified that isolation occurred between a TPC-W read-only transaction and an arbitrary transaction as the steps above produced the expected results.

## **3.1.5 Durability**

### **3.1.5.1 Durability Property Definition**

*The system under test must guarantee that database transactions are durable. The system will preserve the effects of any committed database transaction after recovery from any single point of failure.*

*For each component susceptible to one of the failure types defined in Clause 3.1.5.3 perform the following steps:*

*Step 1. Verify that Consistency Conditions 2, 3, 8 and 9, as specified in Clause 3.1.3.2, are met.*

*Step 2. Obtain the total number of rows in the ORDERS table to determine the current count of orders (count1) in the database.*

*Step 3. Start the mix of web interactions used for the Shopping Interval (see Clause 5.2.1) from a number of EBs within at least 10% of the number of EBs used for the reported WIPS metric.*

*Step 4. Run for at least 5 minutes once all EBs have started requesting web interactions; and keep a count of the number of Buy Confirm web interactions successfully completed by all EBs.*

*Step 5. Cause the failure selected from the list in Clause 3.1.5.3.*

*Step 6. Stop the RBE and collect the total number of Buy Confirm web interactions successfully completed by all EBs (RBE-count).*

*Step 7. If necessary, stop and restart the system under test using normal recovery procedures, where applicable.*

*Step 8. Repeat step 2 to determine the current count of orders (count2) in the database. Verify that (count2-count1) is greater than or equal to the number of successfully completed Buy Confirm web interactions (RBE-count). If there is an inequality, the difference must be less than or equal to the number of EBs active during this test.*

### **3.1.5.2 Durability Test 1: System Failure/Memory Failure**

The failure of the database server was incorporated into the following test:

1. An initial order count was taken. The run was started and all EBs (45,001) were running in steady state for at least 5 minutes.
2. The database server and all application servers were powered off. The system crashed and no more transactions could be completed resulting in errors begin reported by the RBEs.
3. The run was stopped.
4. The database server was powered on and the database was started normally. The database went through an automatic crash recovery process and then brought the database online.
5. A second order count was taken. It was shown that the orders on the database was greater than or equal to that reported by the RBE.
6. Consistency conditions 2, 3, 7, 8, and 9 were checked and satisfied.

### **3.1.4.3 Durability Test 2: Loss of Log, System Failure, and Memory Failure**

1. An initial count of orders was taken. The run was started and all EBs (45,001) were running in steady state for at least 5 minutes.
2. At this point a log drive was removed from the RAID-10 array. Since the array was RAID-10, processing continued without failure.
3. Database system power was removed and then restored.
4. The database was brought back up and transactions were rolled forward from the log.
5. A second orders count was taken. It was shown that the orders on the database was greater than or equal to that reported by the RBE.
6. Consistency conditions were checked and satisfied.

### 3.1.4.4 Durability Test 3: Loss of Data Drive

1. The current count of the total number of orders was determined by the count of O\_ID from the orders table.
2. A test was started with 45,001 EBs submitting requests.
3. 2 data drives from the database RAID 0 volume were removed from the storage system. The loss of a drive crashed the database. The run was stopped. The disk drive were swapped and replaced and then the database was restored from the backup. Then, a roll-forward recovery was done on the database.
4. Step 1 was repeated to obtain the current count of orders.
5. The count of orders on the database was verified to be greater than or equal to the count reported by the RBE.
6. Consistency conditions 2, 3, 7, 8, and 9 were checked and satisfied.

## 3.2 Web-Page Consistency

*Most web pages returned by the SUT to the EB reflect the content of the database by displaying plain data and web objects (GIF pictures). As database transactions update the content of the database, the web pages must display a consistent reflection of these updates, with the exception of the Search Result Page.*

### 3.2.1 Web Page Consistency Definition

*The effects of any update database transaction must be reflected with consistency in subsequent web pages returned by the SUT to the EB: the web pages must reflect either the effect of the entire update operation or none of it.*

*The Search Result Page for title and author search is the only exception to the web page consistency requirement. The Search Result Page contains the result of a query. The index used to answer this query may be based upon the initial population and is not required to be updated along with any database updates, including updates made by the Admin Confirm web interaction.*

### 3.2.2 Web Page Consistency Test

*This test requires that a series of web interactions be requested from several User Sessions while the system is operating under the following load:*

- ◆ *Load 1: The mix of web interactions used during the Shopping Interval and producing a WIPS rate greater than both 10% of the reported WIPS and 100 WIPS (or within 90% of the reported WIPS rate if it is less than 100).*
- ◆ *Load 2: A set of 30 different User Sessions, each executing the Product Detail web interaction in a continuous loop. All User Sessions must repeatedly query the same item, chosen in step 1 below, and must maintain a log of the response pages, including the date header provided by the HTTP 1.1 protocol.*

*The following list of steps must be executed in sequence:*

- Step 1. From User Session A, complete a New Products web interaction and choose an item from the response page, excluding the first ten items.*

*Step 2. Start the Load 1 above.*

*Step 3. Start the load 2 above.*

*Step 4. Wait a minimum of two minutes.*

*Step 5. From User Session A, complete an Admin Confirm web interaction for the item chosen in step 1, changing the price, the published date and the image of the item.*

*Step 6. Wait 30 seconds.*

*Step 7. From User Session B, immediately after step 6 completes, complete a New Products web interaction on the subject of the item chosen in step 1.*

*Step 8. Wait a minimum of two minutes.*

*Step 9. Terminate the load 2 and then the load 1.*

*The following conditions must be met:*

- ◆ *All Product Detail Pages returned by the load 2 meet the requirements defined in Clause 3.2.1.2.*
- ◆ *The New Products Page returned in step 7 contains the item chosen in step 1 within its first ten entries.*

These conditions were met and witnessed by the auditor.

## 4 Scaling and Database Population

### 4.1 Scaling and Database Population

The cardinality (e.g., number of rows) of each table, as initially populated (see Clause 4.3 & 4.4), must be disclosed.

Table Name	Rows
CUSTOMER	144,000,000
COUNTRY	92
ADDRESS	288,000,000
ORDERS	129,600,000
ORDER_LINE	388,832,930
AUTHOR	2,500
CC_XACTS	129,600,000
ITEM	10,000

**Table 4-1: Database Cardinality Table**

### 4.2 180-Day Space

The space required to sustain 180 days of the reported throughput as defined in Clause 4.4 must be disclosed.

Appendix E contains the 180-Day space calculations.

### 4.3 8-Hour Space

The space required for 8 hours of Web-Server access logs as defined in Clause 4.5 must be disclosed.

Appendix E contains the 8-hours of Web server logs space calculations.

### 4.4 Tables and Logs Distribution

The method for distributing table and log data across all media must be described. A detailed diagram or listing of database files indicating the disks or volumes on which they reside must be included. Simple diagrams can be used for clarification.

Controller	Drives	Logical Unit Number (LUN)	Size	Use
0	8x36GB	1	135GB	RAID 1/0 Logs
0	5x36GB	2	135GB	RAID 5 Backup
0	5x36GB	2	135GB	RAID 5 Backup
1	20x18GB	3	337GB	Customer, Country, Address, Orders, Order line, Author, CC Xacts, Item
2	20x18GB	4	337GB	

## 5 Performance Metrics and Response Times

### 5.1 TPC-W Primary Metrics

*The WIPS@scale factor and price per WIPS@scale factor must be disclosed.*

WIPS	Scale Factor
6622.4	10,000

### 5.2 Reproducibility

*A description of the method used to determine the reproducibility of the measurement results must be reported as well as the WIPS from the reproducibility run.*

To demonstrate the reproducibility of the steady state condition during the Shopping Interval, one additional non-overlapping Shopping Interval of the same duration as the Shopping Interval is measured. Its WIPS and difference percentage is disclosed below:

Interval	Duration	Start Time	End Time	WIPS	% Difference
Measurement	30	138	168	6622.4	-
Reproducibility	30	60	90	6625.5	0.05%

### 5.3 Measurement Intervals

*The duration, start time and stop time of each Measurement Interval must be disclosed. For Measurement Intervals reported from the same run, the timing precision must be sufficiently small to demonstrate each Measurement Interval is non-overlapping.*

Interval	Duration	Start Time	End Time
WIPS	30	138	168
WIPsb	30	55	85
WIPSo	30	40	70

## 5.4 Web Interaction Mix

The percentage of each web interaction executed during each measured interval must be displayed in tabular form as in the following example. Percentages must be reported to 2 decimal points.

Web Interaction	Browsing Mix (WIPsb)	Shopping Mix (WIPS)	Ordering Mix (WIPSo)
<b>Browse</b>			
Home	29.00	15.99	9.14
New Product	11.00	5.00	0.46
Best Seller	11.00	4.99	0.46
Product Detail	21.00	17.0	12.33
Search Request	12.00	20.01	14.53
Search Results	11.00	16.99	13.08
<b>Order</b>			
Shopping Cart	2.00	11.61	13.54
Customer Reg.	0.82	3.00	12.86
Buy Request	0.75	2.61	12.73
Buy Confirm	0.69	1.20	10.17
Order Inquiry	0.30	0.75	0.25
Order Display	0.25	0.66	0.22
Admin Request	0.10	0.10	0.12
Admin Confirm	0.09	0.09	0.11

## 5.5 Web Interaction Response Times

The minimum, maximum, average and 90th percentile response time must be reported for each web interaction, in tabular form with a column for each of the Measurement Intervals of WIPS, WIPSo and WIPsb.

Web Interaction	Browsing Mix (WIPsb)				Shopping Mix (WIPS)				Ordering Mix (WIPSo)			
	Avg.	90 <sup>th</sup>	Min	Max	Avg.	90 <sup>th</sup>	Min	Max	Avg.	90 <sup>th</sup>	Min	Max
Home	0.19	0.18	0.19	4.78	0.2	0.21	0.19	4.33	0.19	0.18	0.19	2.67
New Product	0.19	0.18	0.19	41.41	0.2	0.2	0.19	41.39	0.19	0.18	0.19	0.44
Best Seller	0.2	0.2	0.19	41.42	0.22	0.21	0.19	25.97	0.2	0.25	0.19	0.5
Product Detail	0.11	0.11	0.11	41.3	0.12	0.12	0.11	25.13	0.11	0.11	0.11	25.69
Search Request	0.16	0.15	0.16	1.45	0.16	0.17	0.16	3.69	0.16	0.15	0.16	0.45
Search Results	0.2	0.18	0.19	41.41	0.2	0.2	0.19	41.39	0.19	0.18	0.19	25.83
Shopping Cart	0.17	0.18	0.16	1.83	0.17	0.18	0.16	3.78	0.16	0.15	0.16	0.78
Customer Reg.	0.08	0.07	0.08	1.03	0.08	0.09	0.08	3.97	0.08	0.07	0.08	0.38
Buy Request	0.14	0.18	0.06	3.53	0.15	0.21	0.06	3.48	0.08	0.09	0.06	0.83
Buy Confirm	2.07	2.09	2.03	3.09	2.17	2.15	1.88	11.2	2.06	2.07	2.03	4.36
Order Inquiry	0.08	0.09	0.06	3.38	0.08	0.12	0.06	2	0.06	0.06	0.06	0.08
Order Display	0.08	0.12	0.05	1.26	0.1	0.15	0.05	1.56	0.09	0.15	0.05	0.86
Admin Request	0.11	0.11	0.11	0.94	0.11	0.12	0.11	1.08	0.11	0.11	0.11	0.31
Admin Confirm	0.47	0.76	0.19	4.38	1.11	1.96	0.19	14.56	0.2	0.23	0.16	0.47

## 5.6 Think Times

The minimum, maximum and average of all think times executed during each Measurement Interval must be reported.

Think Time	Browse Mix (WIPSb)	Shopping Mix (WIPS)	Ordering Mix (WIPSo)
Minimum	0	0	0
Maximum	73.5	73.61	73.5
Average	7.35	7.35	7.35

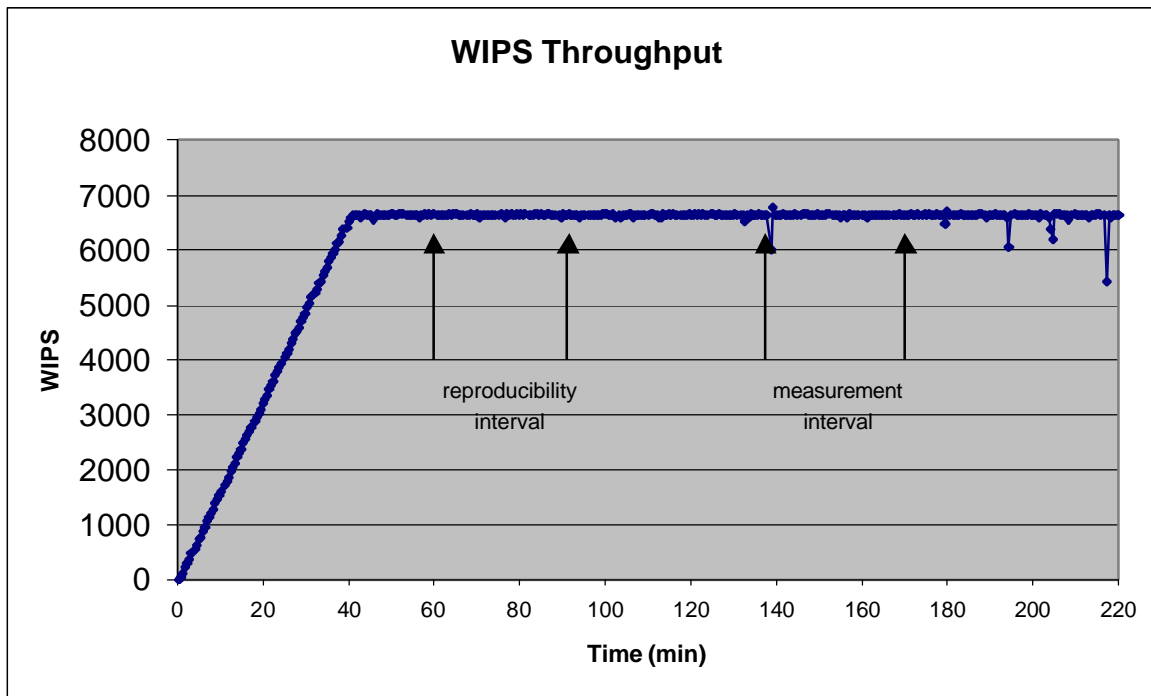
## 5.7 Checkpoint Times

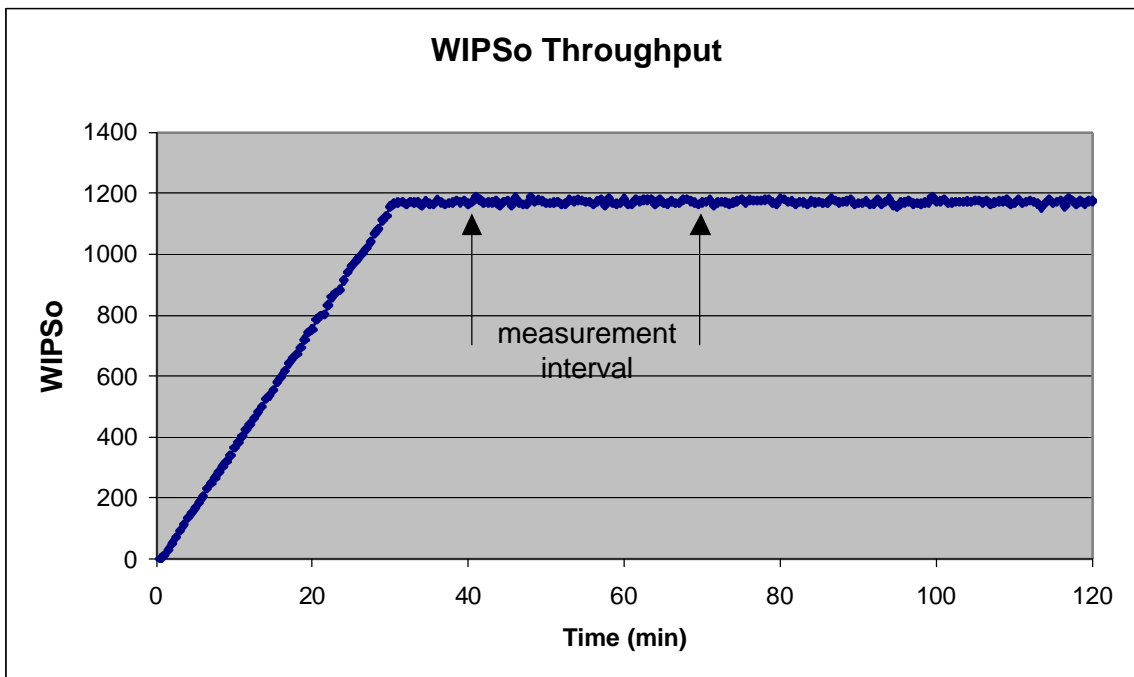
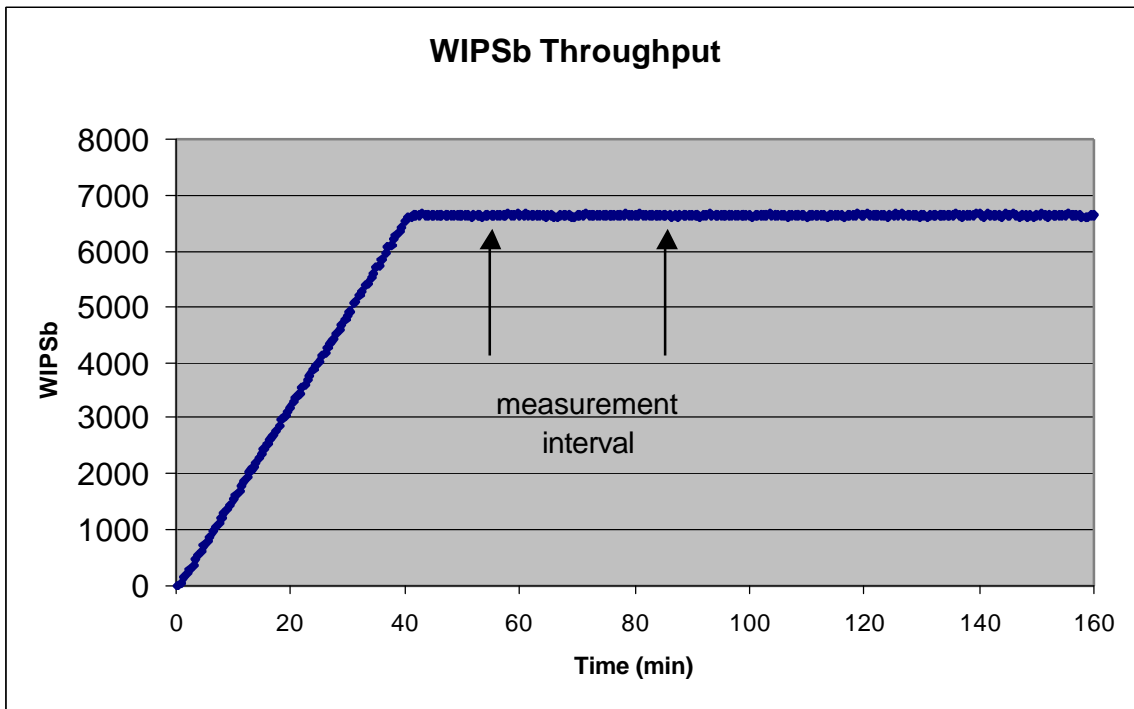
The number of checkpoints in each Measurement Interval, the time in seconds from the start of the Measurement Interval to the first checkpoint and the Checkpoint Interval must be disclosed.

Mix	# Of Check Pts	Time	Check Pt. interval
WIPSb	19	36 sec	1:09
WIPS	26	45 sec	1:07
WIPSo	29	36 sec	1:03

## 5.8 Measured Throughput

The measured throughput of each measured interval must be graphed as described in Clause 5.6.2 and the graphs must be included in the Full Disclosure Report.





## 5.10 Performance Statistics

*A set of performance statistics is required and must be reported. Clause 5.7.1 and 5.7.2 define how these monitored metrics are to be aggregated.*

Refer to section 5.13 for the performance statistics.

## 5.11 Monitoring Tools

The monitoring tools used during each Measurement Interval must be described and the operational methods invoked must be reported. For example, the monitor was invoked as a started task, and binary data was recorded every 15 seconds to a disk for post-processing.

As a monitoring tool we used Microsoft(R) Management Console 1.2 Version 5.0 (Build 2195) for each Measurement Interval. The monitor tool is started and ended manually. The monitor was configured to record the collected data to the hard disk drive in csv format every 30 seconds.

## 5.12 Sampling Technique

If a sampling technique is used to obtain performance data then the sample rate (per second) must be reported as specified in Clause 5.7.1.6

There was no sampling technique used in the collection of performance data.

## 5.13 Data Tables for Monitored Metrics

The monitored metrics as defined in Clause 5.7 must be reported in a table or tables similar to the following example... A careful definition of the data fields reported must be provided to allow the data to be understood and properly interpreted.

**Monitored Metrics (Database Server)**

Metric	WIPS	WIPsb	WIPSo
Avail. Memory (MB)	2249.983	2241.166667	2276
Swap In / Sec	.003333	0.01444555	.003333
Swap Out / Sec	0	0	0
Page In / Sec	.003333	0.042225609	.003333
Page Out / Sec	0	0	0
Avg. Device Util. %	267.6575	108.8439163	95.6288
Avg. Data Tr. Size	8319.713	7450.415951	7533.93
Avg. Service Time	0.011453	0.009769759	0.0107
Physical Reads / Sec	907.4734	328.4946774	100.6673
Total Disk I/O rate	1357.189	636.539752	492.6765
Physical Writes / Sec	449.7157	308.0450747	392.0092
Total Log. I/O rates	154.2769	148.4686404	155.5878
System I/O rate	1327.336	622.0781247	508.7304
% CPU busy	89.03	79.24020438	28.5959
Pool Memory (kB)	7545.067	3773.333333	400.5333
Logical Reads / Sec	874.0192	312.5999855	110.1598
Logical Writes / Sec	453.7423	309.9109281	398.9961
Page Size	4K	4K	4K

**Monitored Metrics (Web Servers Aggregate)**

Metric	WIPS	WIPsb	WIPSo
Avail. Memory (MB)	13.10694444	99.9375	293.9236111
Swap In / Sec	8.792827695	9.261677291	1.497448797
Swap Out / Sec	0.008127165	0.016476158	0.001018236
Page In / Sec	68.88010928	72.95450095	11.9513974
Page Out / Sec	0.127181553	0.263618528	0.01629178
Avg. Device Util. %	4.136185702	4.151877194	1.713616256
Avg. Data Tr. Size	46077.94113	47303.00822	27818.37045
Avg. Service Time	0.006424666	0.006227011	0.008569863
Physical Reads / Sec	0.260588615	0.207240744	0.000323885
Total Disk I/O rate	6.452320892	6.678147089	2.015449615
Physical Writes / Sec	24806.78875	25392.01813	4579.815197
System I/O rate	1.186064448	1.174216796	1.492072223
% CPU busy	80.3417567	69.55411664	13.39816485
Logical Reads	1.011173295	1.010947755	1.367581812
Logical Writes	0.679457594	0.672299872	0.624992554
HTTP Bytes / Sec	13362955.9	13771245.26	1760147.578
TCP/IP Conn./Sec	8.286537788	7.231225549	1.546706896
HTTP Req. / Sec	2993.668246	3224.717295	484.2275076
HTTP Errors / Sec	0	0	0
HTTP Success/Sec	2993.668246	3224.717295	484.2275076
Page Size	4K	4K	4K

#### Monitored Metrics (Image Servers Aggregate)

Metric	WIPS	WIPsb	WIPSo
Avail. Memory (MB)	321.0041667	320.1916667	539.7791667
Swap In / Sec	18.68448126	19.41187501	2.307366891
Swap Out / Sec	0.000416665	0.001805772	0
Page In / Sec	149.3912594	154.8444117	18.32754469
Page Out / Sec	0.006666635	0.028892358	0
Avg. Device Util. %	5.330785053	5.375872804	1.921685856
Avg. Data Tr. Size	31123.63965	30806.3145	25309.99506
Avg. Service Time	0.004320069	0.00420801	0.006085793
Physical Reads / Sec	0.001944741	0.151948477	0.022917295
Total Disk I/O rate	23.61797424	24.68246235	5.167645187
Physical Writes / Sec	23.6160295	24.53051387	5.144727892
System I/O rate	798.7638367	809.0268128	290.0692937
% CPU busy	80.41930013	84.89562679	7.902474073
Logical Reads	2.726795005	2.840747151	2.747646477
Logical Writes	796.4002389	806.5372966	287.6633117
HTTP Bytes / Sec	39857992.3	40190890.86	4376695.723
TCP/IP Conn. / Sec	13.47692067	13.76301409	2.320283109
HTTP Req. / Sec	6454.766148	6660.660572	787.7445862
HTTP Errors / Sec	0	0	0
HTTP Success / Sec	6454.766148	6660.660572	787.7445862
Page Size	4K	4K	4K

The following are definitions of the performance metrics used here. These definitions belong to MS Performance Monitor tool that we used.

**% CPU busy: Processor / % Processor Time (\_Total)**

% Processor Time is the percentage of time that the processor is executing a non-Idle thread. This counter was designed as a primary indicator of processor activity. It is calculated by measuring the time that the processor spends executing the thread of the Idle process in each sample interval, and subtracting that value from 100%. (Each processor has an Idle thread which consumes cycles when no other threads are ready to run). It can be viewed as the percentage of the sample interval spent doing useful work. This counter displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time the service was inactive, then, subtracting that value from 100%.

**Available Memory: Memory / Available MBytes**

Available MBytes is the amount of physical memory available to processes running on the computer, in Megabytes (Bytes / 1,048,576). It is calculated by summing space on the Zeroed, Free, and Stand by memory lists. Free memory is ready for use; Zeroed memory are pages of memory filled with zeros to prevent later processes from seeing data used by a previous process. Standby memory is memory removed from a process' working set (its physical memory) on route to disk, but is still available to be recalled. This counter displays the last observed value only; it is not an average.

**Pool Memory: SQLServer, Memory manager / Granted work space memory (KB)**

Total amount of memory granted to executing processes. This memory is used for hash, sort and create index operations.

**Page Size: Page size of the memory manager**

**Page In / Sec: Memory / Pages Input/sec**

Pages Input/sec is the number of pages read from disk to resolve hard page faults. (Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk). This counter was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

**Page Out / Sec: Memory / Pages Output/sec**

Pages Output/sec is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory, so they are likely to hold data, not code. A high rate of pages output might indicate a memory shortage. Windows NT writes more pages back to disk to free up space when physical memory is in short supply. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

#### Swap In: Memory / Page Reads/sec

Page Reads/sec is the number of times the disk was read to resolve hard page faults. (Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk). This counter was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes reads to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of read operations, without regard to the numbers of pages retrieved by each operation. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

#### Swap Out: Memory/ Page Writes/sec

Page Writes/sec is the number of times pages were written to disk to free up space in physical memory. Pages are written to disk only if they are changed while in physical memory, so they are likely to hold data, not code. This counter counts write operations, without regard to the number of pages written in each operation. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

#### Logical Reads/Sec: System / File Read Operations/sec

File Read Operations/sec is the combined rate of file system read requests to all devices on the computer, including requests to read from the file system cache. It is measured in numbers of reads. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

#### Logical Writes/Sec: System / File Write Operations/sec

File Write Operations/sec is the combined rate of the file system as it writes requests to all devices on the computer, including requests to write to data in the file system cache. It is measured in numbers of writes. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

#### Physical Reads/Sec: Physical Disk / Disk Reads/Sec

Disk Reads/sec is the rate of read operations on the disk (\_Total is for all disks).

#### Physical Writes/Sec: Physical Disk / Disk Writes/Sec

Disk Writes/sec is the rate of write operations to the disk (\_Total is for all disks).

#### System I/O rate: Process / IO Data Operations/sec (\_Total)

The rate the process is issuing read and write I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/O's.

#### Total Disk I/O rate: Physical Disk / Disk Transfers/sec (\_Total)

Counts the number of disk reads and writes per second (\_Total is for all disks).

#### Avg. device utilization: Physical Disk / % Disk Time (\_Total)

% Disk Time is the percentage of elapsed time that the selected disk drive is busy servicing read or write requests (\_Total is for all disks). Caution this number is not meaningful for a multi-disk RAID volume unless divided by the number of hard disk drives.

#### Average service time: Physical Disk / Avg. Disk sec/Transfer (\_Total)

Avg. Disk sec/Transfer is the time in seconds of the average disk transfer (\_Total is for all disks).

Average disk data transfer size: Physical Disk / Avg. Disk Bytes/Transfer (\_Total)  
 Avg. Disk Bytes/Transfer is the average number of bytes transferred to or from the disk during write or read operations (\_Total is for all disks).

Total Logging I/O rates: Physical Disk / Disk Transfers/sec (Drive G (used for log))  
 Disk Transfers/sec is the rate of read/write operations on the disk.

TCP/IP Connections / Sec: Web Service / Anonymous Users/sec  
 The rate users are making anonymous connections using the Web service.

HTTP Requests / Sec: Web Service / Total Method Requests/sec  
 The rate all HTTP requests are made.

HTTP Errors / Sec: Manually count the logs.

HTTP Bytes / Sec: Web Service / Bytes Sent/sec  
 Bytes Sent/sec is the rate that data bytes are sent by the Web service.

HTTP Success / Sec: Arithmetically calculate this metric.

#### Cache Server Monitored Metrics

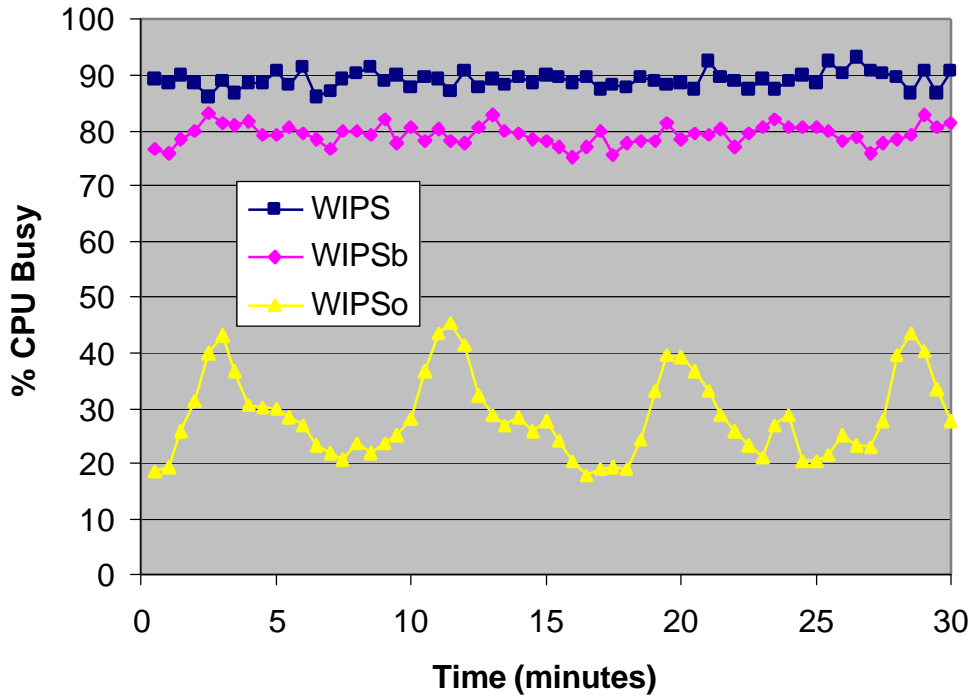
Metric	WIPS	WIPsb	WIPSo
% CPU busy	32.3833	36.5333	5.2667
HTTP KBytes/Sec	16,690	23,790	1,386

The cache server CPU utilization was monitored via SNMP, with samples every 30 seconds through the measurement interval. The HTTP KBytes/sec was calculated using the “HTTP KBytes sent” counter, subtracting the ending value from the starting value and dividing by the measurement interval time.

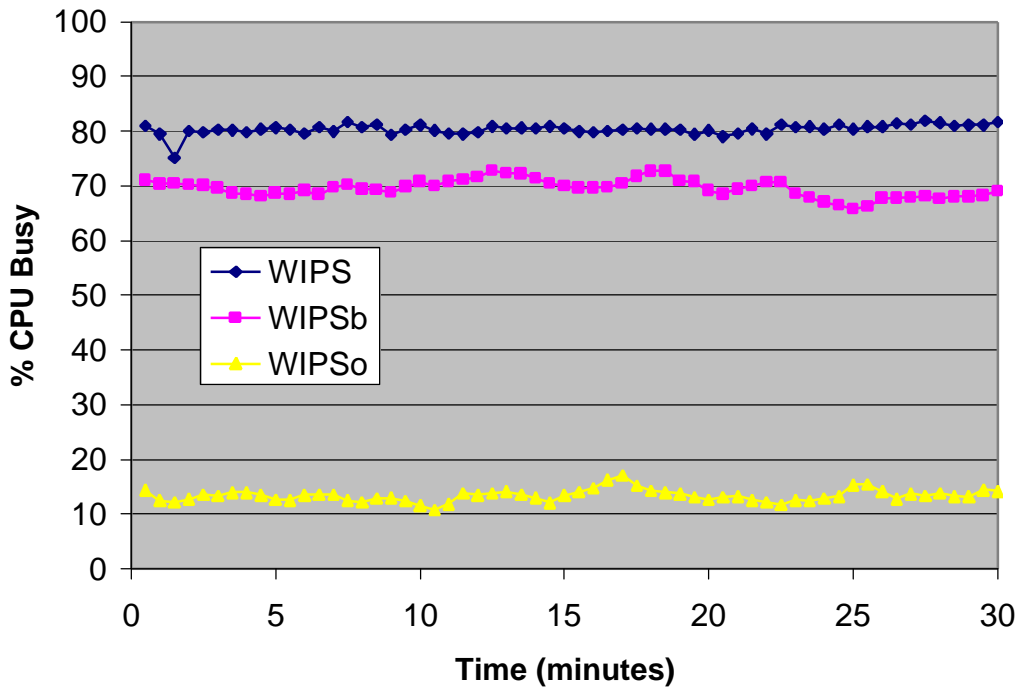
## 5.14 CPU Utilization Graphs

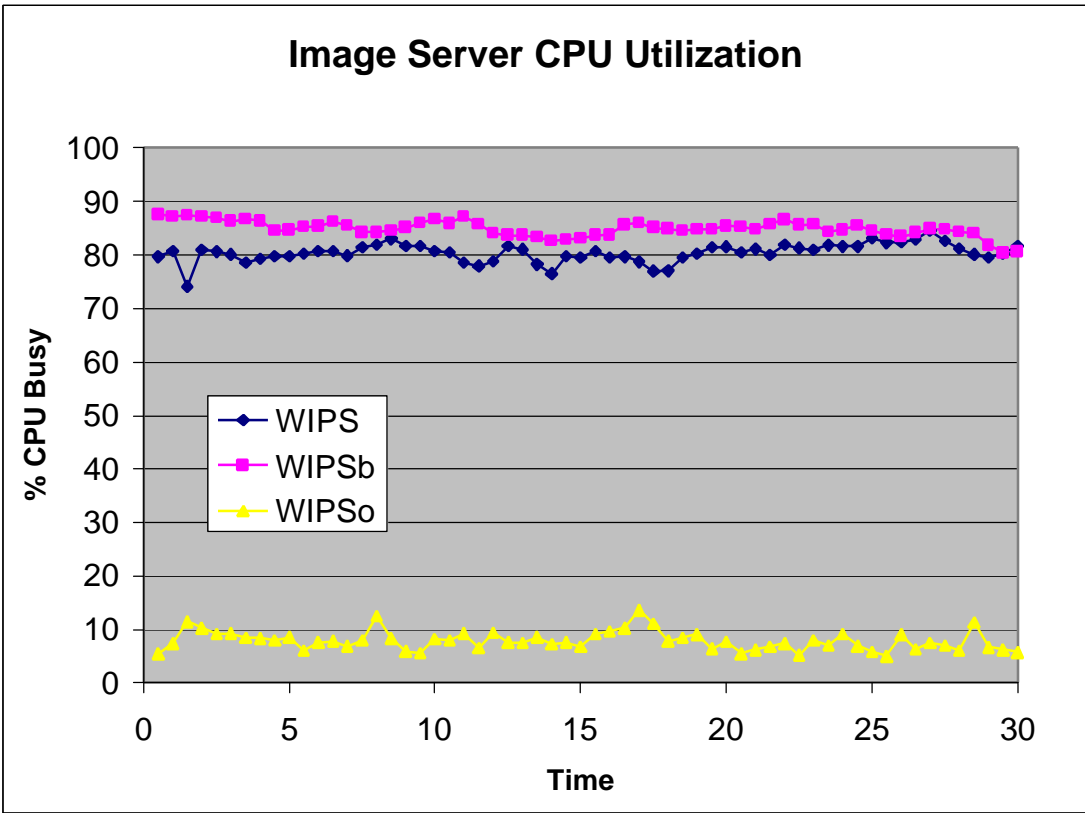
*The CPU utilization must be graphed (as defined in clause 5.7.2.1.2) and the graph must be included in the Full Disclosure Report.*

### Database Server CPU Utilization



### Web Server CPU Utilization





## 6 SUT, RBE and Network

### 6.1 Function Diagram of Measured Configuration

*A complete functional diagram of the measured configuration must be disclosed. A detailed list of all software and hardware functionality being performed on the Driver System, and its interface to the SUT must be disclosed (see Clause 6.6.3.6).*

The RBEs generated the transaction input data and transmitted it to the SUT in HTML format. The RBE then received the output from the SUT, time-stamped it, and forwarded it to the Master RBE for post-test processing. No other functionality was included on the driver RBE. Detailed diagrams of benchmarked and priced configurations are provided in **Section 0.4**.

### 6.2 Network Bandwidth

*The rated bandwidth of the network(s) component used in the measured configuration must be disclosed, along with any setting restricting that bandwidth. This includes, but is not limited to, the inter-node connections within the SUT, between the SUT and the RBE, and between the SUT and the Payment Gateway Emulation.*

The bandwidth of network components used in the measured configured was disclosed in **Section 0.4**.

### 6.3 Operator Intervention

*If the configuration requires operator intervention to meet the requirements of performance levels and uninterrupted operations, the mechanisms and the frequency of this intervention must be disclosed.*

No operator intervention is required to maintain the required performance levels for the three-hour steady state performance runs.

### 6.4 RBE Availability

*It must be disclosed if the RBE is commercially available or proprietary.*

The RBE is Dell proprietary.

## 6.5 Web Page Assembly Description

*A Web Page Assembly description is required that describes how the Web Interaction pages are assembled. When a web page is returned to the RBE, it may contain URLs to frames and/or images that must be acquired by the RBE to complete the Web Interaction page. The Web Page Assembly provides a high level description in easy to read text that describes which servers are providing which frames and/or images. Thus a quick glance of the Web Page Assembly description provides an overview of the TPC-W solution without a detailed examination of the source code provided in the FDR.*

### 1 Web Cache – PowerEdge 1550 using Volera Excelerator 2.1

- Title Search Results frame
- Title Search Results promotional list frame
- Author Search Results frame
- Author Search Results promotional list frame
- Best Sellers frame
- New Products frame
- Subject Search Results frame
- Product Detail frame

### 12 Web Servers - PowerApp.Web 120

- Button images for non-secure pages
- TPC logo images
- All other Web interaction pages/frames/images
- Resolves all stale Web Cache data

### 4 Image Servers – PowerApp.Web 120

- Item images
- Thumbnail images

## 6.6 Errors that Result in EB Session Closing and Retried

*The types of errors that result in an EB session being closed and re-initialized (as defined in 5.5.1.5) must be disclosed. Any error handling method that is not disclosed is assumed to be a retry.*

An error or failure during a Buy Confirm, Shopping Cart Refresh, or Buy Request web interaction results in an EB session being closed and re-initialized per Clause 5.5.1.5.

## 6.7 RBE Method for Generating Start Seed

*The RBE method of generating the start seeds for the EBs must be disclosed. It should be clear that the start seeds are different for each EB. Knowledge of previously used start seeds may only be used to generate a unique set of start seeds for all subsequent runs. If knowledge of previous start seeds is not used, then the period of the selection for new seeds sets must be greater than 10,000.*

*Comment: It is the intent of this clause to prevent generating similar start seeds that will result in a reduction of inserts to the database (for example, ADDRESS table).*

First a random start seed was generated. Then, this seed was added to each thread ID to ensure uniqueness. For each subsequent run, a new start seed is generated and this seed value is added to each thread ID. The period of the selection for the new seed set was 10,000,000.

## **6.8 Random Number Generators**

*The number of random number generators used for each EB must be disclosed along with their purpose(s). Examples of purposes include: choosing the next web interaction, building random text strings, deciding whether the shipping address is changed, think time generation, etc...*

One random number generator is used for each EB as each EB chooses a web interaction, builds random text strings, decides to change shipping address, and generate think times.

## **7 Pricing**

### **7.1 Hardware and Software Components**

*A detailed list of the hardware and software used in the priced system must be reported. Each item must have a vendor part number, description and release/revision level, and either general availability status or committed delivery date. If package pricing is used, contents of the package must be disclosed. Pricing source(s) and effective date(s) must also be reported.*

A detailed list of all hardware and software, including the 3-year price, is provided in the Executive Summary at the front of this report. The price quotes are included in Appendix F.

### **7.2 3-Year Cost of Configured System**

*The total 3-year price of the entire configuration must be reported, including hardware, software and maintenance charges. Separate component pricing is recommended. The basis of all discounts must be disclosed.*

A detailed list of all hardware and software, including the 3-year price, is provided in the Executive Summary at the front of this report. The price quotes are included in Appendix F.

### **7.3 Availability Dates**

*The committed delivery date for general availability (availability date) of products used in the price calculations must be reported. When the priced system includes products with different availability dates, availability date reported on the Executive Summary must be the date by which all components are committed to being available. The Full Disclosure Report must report availability dates individually for at least each of the categories for which a pricing subtotal must be provided (see Clause 7.3.1.3).*

All hardware and software used in this benchmark will be generally available on December 19, 2001.

### **7.4 Usage Pricing**

*For any usage pricing, the sponsor must disclose:*

*Usage level at which the component was priced*

*A statement of the company policy allowing such pricing*

No usage pricing is involved.

### **7.5 Country-Specific Pricing**

*Additional Clause 7 related items may be included in the Full Disclosure Report for each country-specific priced configuration. Country-specific pricing is subject to Clause 7.1.7.*

This Dell system configuration is priced for the United States of America.

## **8 Audit Related Items**

*The auditor's name, address, phone number, and a copy of the auditor's attestation letter indicating compliance must be included in the Full Disclosure Report.*

This implementation of the TPC Benchmark W was audited by Lorna Livingtree of Performance Metrics Incorporated. Further information can be obtained from:

Performance Metrics Inc.  
137 Yankton St. Suite 101  
Folsom CA 95630  
Phone: (916) 985-1131  
Fax: (916) 985-1185  
Email: Lorna@PerfMetrics.com



**PERFORMANCE METRICS INC.**  
**TPC Certified Auditors**

---

December 19, 2001

Greg Darnell  
Internet Performance  
Dell Computer Corporation  
One Dell Way  
Round Rock, TX 78682

I have verified the TPC Benchmark™ W for the following configuration:

Platform: Dell PowerEdge 6400/900Mhz  
Database Manager: Microsoft SQL Server 2000 Standard Edition  
Operating System: Microsoft Windows 2000 Advanced Server  
Web Server: Microsoft IIS 5.0

1 DB Server: (4P)				
CPU's	Memory	WIPS	WIPSo	WIPsb
4 Pentium III Xeon @ 900 Mhz	Main: 4 GB Cache: 2MB each	<b>6622.4</b>	<b>1,171.7</b>	<b>6,636.5</b>
12 Web Servers each with:				
2 Pentium III @ 1000 Mhz	Main: 512 MB Cache: 256KB	1 disk @ 18GB		
Additional Servers:				
4 Images Servers	1 Cache Server			

In my opinion, these performance results were produced in compliance with the TPC requirements for the benchmark. The following attributes of the benchmark were given special attention:

- The transactions were correctly implemented.

**PERFORMANCE METRICS INC.**  
**TPC Certified Auditors**

---

- The screens had the appropriate data and graphics.
- The distribution of graphic and web page sizes were correct.
- The database was properly sized and populated.
- The database was properly scaled with 144,000,000 customers and 10,000 items.
- The ACID properties were met.
- The durability tests were performed on the full database.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was configured on the measured system.
- Fourteen days of growth space was configured on the measured system.
- The 8-hour, 7-day, 14-day and 180-day space calculations were verified. The measured configuration has sufficient storage to satisfy these requirements.
- There were 50,000 user contexts present on the system.
- The 2-second delay and SSL handshake for the PGE were verified.
- Each emulated user started with a different random number seed.
- The steady state portion of the test was 30 minutes.
- Checkpoints were taken every 2 minutes or less and no checkpoint duration was more than 5 minutes.
- The reproducibility run exceeded the measurement performance by not more than 5%.
- The system pricing was checked for major components and maintenance.
- The appropriate portions of the FDR were review for accuracy.

Auditor Notes:

The initial population of the CUSTOMER table C\_LAST\_VISIT column was loaded with the current date. The specification requires this to be the current date plus a random number of days between 0 and 60 inclusive. This was an error in the loading program. Since this column is never processed in the benchmark, it is my opinion that this is an insignificant deviation that has no impact on the performance.

Sincerely,



Lorna Livingtree  
Auditor

## Appendix A- Application Source Code

### Tpcw.dll source code

#### Debug.c

```
//debug.c
//
//Copyright Unisys, 1999,2000

#define _WIN32_WINNT 0x0500
#define DEBUGSPINCOUNT 200
#include <windows.h>
#include <winbase.h>
#include "debug.h"
#include "tpcw.h"

#ifdef SCREEN_INFO
    long numshop;
    long numthread;
    long numbasketkill;
    long numpgereq;
    long numsslretry;
    long numwebaction[15];
    long totalwebaction[15];
    long numswitchbasket;
    CRITICAL_SECTION webaction[15];
    CRITICAL_SECTION shop;
    CRITICAL_SECTION thread;
    CRITICAL_SECTION basketkill;
    CRITICAL_SECTION pgereq;
    CRITICAL_SECTION sslretry;
    CRITICAL_SECTION switchbasket;
#endif

void Screen_Info_Init()
{
#ifdef SCREEN_INFO
    int i;
    numshop=0;
    numthread=0;
    numbasketkill=0;
    numpgereq=0;
    numsslretry=0;
    numswitchbasket=0;
    for(i=0;i<15;i++)
    {
        totalwebaction[i]=0;
        numwebaction[i]=0;
        InitializeCriticalSectionAndSpinCount(&webaction[i],
        DEBUGSPINCOUNT);
    }
#endif
}
```

```
InitializeCriticalSectionAndSpinCount(&shop ,
DEBUGSPINCOUNT);
InitializeCriticalSectionAndSpinCount(&thread ,
DEBUGSPINCOUNT);
InitializeCriticalSectionAndSpinCount(&basketkill ,
DEBUGSPINCOUNT);
InitializeCriticalSectionAndSpinCount(&pgereq ,
DEBUGSPINCOUNT);
InitializeCriticalSectionAndSpinCount(&sslretry ,
DEBUGSPINCOUNT);
InitializeCriticalSectionAndSpinCount(&switchbasket ,
DEBUGSPINCOUNT);
#endif
}
```

void Screen\_Info\_Term()

```
{
    #ifdef SCREEN_INFO
    int i;
    for(i=0;i<14;i++)
    {
        DeleteCriticalSection(&webaction[i]);
    }
    DeleteCriticalSection(&shop );
    DeleteCriticalSection(&thread );
    DeleteCriticalSection(&basketkill );
    DeleteCriticalSection(&pgereq);
    DeleteCriticalSection(&sslretry);
    DeleteCriticalSection(&switchbasket);
    #endif
}
```

#### debug.h

```
//debug.h
//
//Copyright Unisys, 1999,2000
#define SCREEN_INFO 1
void Screen_Info_Init();
void Screen_Info_Term();
```

#### display.c

```
//display.c
//
//Copyright Unisys, 1998,1999,2000
#define DBRETY
#define DBRETYSLLEEP

#include "tpcw.h"
#include "tpcw_db.h"
#include "thread.h"
```

```

#include "strings.h"
#include "display.h"
#include "encode.h"
#include "readreg.h"
#include <windows.h>
#include <sys/timeb.h>
#include <time.h>

#include <math.h>
#ifndef PGEHost
#include "pge.h"
#endif

extern long maxitem;
extern CRITICAL_SECTION g_csBasket[AR_SIZE];
extern HANDLE BasketHeap;

#ifdef SCREEN_INFO
extern long numshop;
extern long numthread;
extern long numbasketkill;
extern long numpgreq;
extern long numsslretry;
extern long numwebaction[15];
extern long totalwebaction[15];
extern long numswitchbasket;
extern CRITICAL_SECTION webaction[15];
extern CRITICAL_SECTION shop;
extern CRITICAL_SECTION thread;
extern CRITICAL_SECTION basketkill;
extern CRITICAL_SECTION pgreq;
extern CRITICAL_SECTION sslretry;
extern CRITICAL_SECTION switchbasket;
#endif

extern BOOL DayLightSaving;
extern char cachehost[20];
extern char advert1[500];
extern char advert2[50];
extern long advert1len;
extern long advert2len;
extern char homel[100];
extern char home2[500];
extern char home3[500];
extern char hometable[50000];
extern long homel1len;
extern long home2len;
extern long home3len;
extern long home4len;
extern long hometable1len;
extern char newprod1[500];
extern char newprod2[1000];
extern long newprod1len;

extern long newprod2len;
extern char bestsell1[500];
extern char bestsell2[1000];
extern long bestsell1len;
extern long bestsell2len;
extern char bookinfo1[500];
extern char bookinfo2[50];
extern long bookinfo1len;
extern long bookinfo2len;
extern char search1[500];
extern char search2[1000];
extern long search1len;
extern long search2len;
extern char searchres1[600];
extern long searchres1len;
/*
extern char result1[500];
extern char result2[500];
extern char result3[500];
extern long result1len;
extern long result2len;
extern long result3len;
*/
extern char shop1[500];
extern char shop2[500];
extern char shop3[1000];
extern long shop1len;
extern long shop2len;
extern long shop3len;
extern char orderstat1[500];
extern char orderstat2[500];
extern char orderstat3[500];
extern long orderstat1len;
extern long orderstat2len;
extern long orderstat3len;
extern char custreg1[500];
extern char custreg2[2000];
extern long custreg1len;
extern long custreg2len;
extern char neworder1[500];
extern char neworder2[500];
extern char neworder3[500];
extern char neworder4[500];
extern long neworder1len;
extern long neworder2len;
extern long neworder3len;
extern long neworder4len;
extern char buyreq1[500];
extern char buyreq2[500];
extern char buyreq3[500];
extern char buyreq4[1500];
extern char buyreq5[500];
extern long buyreq1len;

```

```

extern long buyreq2len;
extern long buyreq3len;
extern long buyreq4len;
extern long buyreq5len;
extern char show1[500];
extern char show2[500];
extern char show3[500];
extern long show1len;
extern long show2len;
extern long show3len;
extern char changeinfo1[500];
extern char changeinfo2[50];
extern long changeinfo1len;
extern long changeinfo2len;
char hostname[20];
int hostnumber;
char domainname[30];
__int64 O_ID;
__int64 C_ID;
__int64 ADDR_ID;
long Session_ID;
__int64 LimitO_ID;
__int64 LimitC_ID;
__int64 LimitADDR_ID;
long LimitSession_ID;
CRITICAL_SECTION csO_ID;
CRITICAL_SECTION csC_ID;
CRITICAL_SECTION csADDR_ID;
CRITICAL_SECTION csSession_ID;
long C_ID_ALLOC= 100;
long O_ID_ALLOC= 100;
long ADDR_ID_ALLOC= 100;
long Session_ID_ALLOC= 5000;

const static char * subject[] = {
    "ARTS", //0 = a
    "BIOGRAPHIES", //1 = b
    "BUSINESS", //2 = c
    "CHILDREN", //3 = d
    "COMPUTERS", //4 = e
    "COOKING", //5 = f
    "HEALTH", //6 = g
    "HISTORY", //7 = h
    "HOME", //8 = i
    "HUMOR", //9 = j
    "LITERATURE", //10 = k
    "NON-FICTION", //11 = l
    "PARENTING", //12 = m
    "POLITICS", //13 = n
    "REFERENCE", //14 = o
    "RELIGION", //15 = p
    "ROMANCE", //16 = q
    "SELF-HELP", //17 = r

```

```

"SCIENCE-NATURE", //18 = s
"SCIENCE-FICTION", //19 = t
"SPORTS", //20 = u
"MYSTERY", //21 = v
"TRAVEL", //22 = w
"YOUTH" //23 = x
};

void Parse(char param[],char * querystring)
{
    char buffer[1024];
    char * chptr;
    int i=0;
    buffer[0]='\0';
    chptr=querystring;
    querystring = strchr(querystring, '=')+1;
    while(*querystring!='&' && *querystring!='\0')
    {
        buffer[i]=*querystring;
        querystring++;
        i++;
    }
    buffer[i]='\0';
    if(*querystring!='\0')
    {
        querystring++;
    }
    strcpy(param,buffer);
    strcpy(chptr,querystring);
}

void Skip(char * querystring)
{
    querystring++;
}

void DecodeParse(char param[],char * querystring)
{
    char buffer[1500];
    char * chptr;
    int i=0;
    buffer[0]='\0';
    chptr=querystring;
    querystring = strchr(querystring, '=')+1;
    while(*querystring!='&'&&*querystring!='\0')
    {
        buffer[i]=*querystring;
        querystring++;
        i++;
    }
    buffer[i]='\0';
    if(*querystring!='\0')
    {

```

```

        querystring++;
    }
    URL_Decode(buffer);
    strcpy(param,buffer);
    strcpy(chptr,querystring);
}

BOOL floatprint(long num,struct FLOATPRINT * fltptr)
{
    fltptr->x=num/100;
    fltptr->y=abs(num-(fltptr->x*100));
    return TRUE;
}

void CookieDate(char cookiedate[])
{
    time_t ltime;
    struct tm * newtime;
    const static char *
dayofweek[]={"Sun","Mon","Tue","Wed","Thu","Fri","Sat"};
    const static char *
month[]={"Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct",
        "Nov","Dec"};
    time(&ltime);
    newtime= gmtime(&ltime);
    newtime->tm_year++;
    newtime->tm_isdst= DayLightSaving;
    ltime= mktime(newtime);
    newtime->tm_year= 1900+ newtime->tm_year;
    wsprintf(cookiedate,"%s,%.2d-%s-%.4d %.2d:%.2d:%.2d GMT",
        dayofweek[newtime->tm_wday],
        newtime->tm_mday, month[newtime->tm_mon], newtime-
>tm_year,
        newtime->tm_hour, newtime->tm_min, newtime->tm_sec);
}

void Listing(TPCW * tpcw)
{
    memcpy(tpcw->endptr, hometable, hometablelen);
    tpcw->endptr= tpcw->endptr+ hometablelen;
}

long Advertisement(TPCW *tpcw)
{
    int rc;
    BOOK_DETAILS_ENQ_DATA book;
    // long image[5];
    unsigned int i;
    char *chrptr;
    book.i_id= (__int64)RandomNumber(1, maxitem,&tpcw->lTLSeed);

    if(book.i_id <0 || book.i_id >(__int64)maxitem)

```

```

    {
        sprintf(tpcw->err,
            "Advertisement: BookID is not within range[0-%d]
(%I64d)",
            maxitem, book.i_id);
        ErrorMessage(tpcw);
        return 0;
    }
    rc= SQLGetAD(tpcw,&book);
    if(rc== FALSE)
    {
        strcpy(tpcw->err," Advertisement:
SQLGetAD(tpcw,&book)");
        ErrorMessage(tpcw);
        return 0;
    }
    memcpy(tpcw->endptr, advert1, advert1len);
    tpcw->endptr= tpcw->endptr+ advert1len;
    for(i= 0; i<5; i++)
    {
        strcpy(tpcw->szBuffer, book.i_also[i].i_thumbnail+ 1);
        chrptr= tpcw->szBuffer;

        chrptr = book.i_also[i].i_thumbnail;
        chrptr+=strcsn(book.i_also[i].i_thumbnail, " \");
        *chrptr = '\0';

        tpcw->len= sprintf(tpcw->endptr,
            "<TD><A HREF=\"http://%s/tpcw/tpcw.dll?z=%I64d\"
target=\"_top\">\"
            \"<IMG SRC=\"http://image.tpcw.net/tpcw/%s\"
ALT=\"Book %d\" WIDTH=\"100\" HEIGHT=\"150\"></A></TD>\",
            domainname,
            book.i_also[i].i_id,
            book.i_also[i].i_thumbnail,
            i+ 1);
        tpcw->endptr= tpcw->endptr + tpcw->len;
    }
    memcpy(tpcw->endptr, advert2, advert2len);
    tpcw->endptr= tpcw->endptr + advert2len;

    return 1;
}

BOOL Home(TPCW * tpcw, struct BASKET_LIST * Basket_List[])
{
    int rc;
    struct BASKET_LIST * p= NULL;

```

```

CUSTOMER_INFO_ENQ_DATA cust;
memcpy(tpcw->endptr, home3, home3len);
tpcw->endptr= tcw->endptr+ home3len;
__try
{
    EnterCriticalSection(&g_csBasket[tpcw->hash]);
    rc= SearchForBasket(tpcw, Basket_List,&p);
    if(rc== FALSE)
    {
        strcpy(tpcw->err,"Home: SearchForBasket()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    if(p!= NULL && p->sc_c_id== tcw->ID.C_ID && p-
>sc_c_id!= 0)
    {
        strcpy(cust.c_fname, p->sc_c_fname);
        strcpy(cust.c_lname, p->sc_c_lname);
    }
}
__finally
{
    LeaveCriticalSection(&g_csBasket[tpcw->hash]);
}
if(p== NULL&&tpcw->ID.C_ID>0)
{
    cust.c_id= tcw->ID.C_ID;
    rc= SQLGetCustName(tpcw,&cust);
    if(rc== FALSE)
    {
        strcpy(tpcw->err," Home: SQLGetCustName()");
        ErrorMessage(tpcw);
        return FALSE;
    }
}
if(tpcw->ID.C_ID>0)
{
    tcw->len= wsprintf(tpcw->endptr,
        "<H3 ALIGN=\\"CENTER\\""Welcome back <FONT
COLOR=\\"#770000\\"">%s %s</FONT></H3>",
        cust.c_fname, cust.c_lname);
    tcw->endptr= tcw->endptr+ tcw->len;
}
else
{
    memcpy(tpcw->endptr, home1, homellen);
    tcw->endptr= tcw->endptr+ homellen;
}
rc= Advertisement(tpcw);
if(rc== 0)
{
    return FALSE;
}

```

```

memcpy(tpcw->endptr, home2, home2len);
tpcw->endptr= tcw->endptr + home2len;
Listing(tpcw);
tpcw->len= wsprintf(tpcw->endptr,"</TABLE><P>"
    "<A HREF=\\"http://%s/tpcw/tpcw.dll?b&ADD_FLAG=N\\""><IMG
SRC=\\"gifs/cart.gif\\"" ALT=\\"Shopping Cart\\"" BORDER=\\"0\\"" WIDTH=\\"120\\""
HEIGHT=\\"30\\""></A>"
    "<A HREF=\\"http://%s/tpcw/tpcw.dll?h\\""><IMG
SRC=\\"gifs/search.gif\\"" ALT=\\"Search\\"" BORDER=\\"0\\"" WIDTH=\\"120\\""
HEIGHT=\\"30\\""></A>"
    "<A HREF=\\"https://%s/tpcw/tpcw.dll?f\\""><IMG
SRC=\\"gifs/status.gif\\"" ALT=\\"Order Status\\"" BORDER=\\"0\\"" WIDTH=\\"120\\""
HEIGHT=\\"30\\""></A>"
    "</P>"
    "</CENTER></BODY></HTML>",
    domainname,
    domainname,
    domainname);
tpcw->endptr= tcw->endptr+ tcw->len;
return TRUE;
}
BOOL HomeG(TPCW * tcw, struct BASKET_LIST * Basket_List[])
{
    int rc;
    struct BASKET_LIST * p= NULL;
    CUSTOMER_INFO_ENQ_DATA cust;
    memcpy(tpcw->endptr, home3, home3len);
    tcw->endptr= tcw->endptr+ home3len;
    __try
    {
        EnterCriticalSection(&g_csBasket[tpcw->hash]);
        rc= SearchForBasket(tpcw, Basket_List,&p);
        if(rc== FALSE)
        {
            strcpy(tpcw->err,"Home: SearchForBasket()");
            ErrorMessage(tpcw);
            return FALSE;
        }
        if(p!= NULL && p->sc_c_id== tcw->ID.C_ID && p-
>sc_c_id!= 0)
        {
            strcpy(cust.c_fname, p->sc_c_fname);
            strcpy(cust.c_lname, p->sc_c_lname);
        }
    }
    __finally
    {
        LeaveCriticalSection(&g_csBasket[tpcw->hash]);
    }
    if(p== NULL&&tpcw->ID.C_ID>0)
    {
        cust.c_id= tcw->ID.C_ID;
    }
}

```

```

rc= SQLGetCustName(tpcw,&cust);
if(rc== FALSE)
{
    strcpy(tpcw->err," Home: SQLGetCustName()");
    ErrorMessage(tpcw);
    return FALSE;
}
}
if(tpcw->ID.C_ID>0)
{
    tpcw->len= wsprintf(tpcw->endptr,
        "<H3 ALIGN=\"CENTER\">Welcome back <FONT
COLOR=\"#770000\">%s %s</FONT></H3>",
        cust.c_fname, cust.c_lname);
    tpcw->endptr= tpcw->endptr+ tpcw->len;
}
else
{
    memcpy(tpcw->endptr, home1, homellen);
    tpcw->endptr= tpcw->endptr+ homellen;
}
rc= Advertisement(tpcw);
if(rc== 0)
{
    return FALSE;
}
tpcw->len= wsprintf(tpcw->endptr,
    "<CENTER><IFRAME SRC=\"homebase.htm\" BORDER=0
FRAMEBORDER=0 "
    "WIDTH=750 HEIGHT=500 scrolling=no></IFRAME></CENTER>"

    "</P></CENTER>"
    "</BODY></HTML>");
tpcw->endptr= tpcw->endptr+ tpcw->len;
return TRUE;
}

```

```

BOOL PromoForListing(TPCW * tpcw)

```

```

{
    BOOK_LIST_ENQ_DATA bookdata;
    long i;
    char type[50];
    int rc;
    Parse(type, tpcw->querystring);
    Parse(bookdata.category, tpcw->querystring);
    i=(long)(bookdata.category[0]- 'a');
    strcpy(bookdata.category, subject[i]);
    if(type[0]== 'j')
    {
        strcpy(type,"New Product Page - Subject:");
    }
    else if (type[0]== 'k')

```

```

{
    strcpy(type,"Best Sellers Page - Subject:");
}
else
{
    strcpy(tpcw->err," PromoForListing: Unknown
interaction");
    ErrorMessage(tpcw);
    return FALSE;
}
tpcw->len= wsprintf(tpcw->endptr,
    "<BODY BGCOLOR=\"#FFFFFF\"><H1 ALIGN=\"CENTER\">TPC Web
Commerce Benchmark (TPC-W)</H1>"
    "<P ALIGN=\"CENTER\"><IMG SRC=\"gifs/tpclogo.gif\"
WIDTH=\"288\" HEIGHT=\"67\" ALIGN=\"BOTTOM\" BORDER=\"0\">"
    "</P><H2 ALIGN=\"CENTER\">%s %s</H2>",
    type,
    bookdata.category
    );
tpcw->endptr= tpcw->endptr+ tpcw->len;
rc= Advertisement(tpcw);
if(rc== 0)
{
    ErrorMessage(tpcw);
    return FALSE;
}
*tpcw->endptr= '\0';
return TRUE;
}

```

```

BOOL BookInfo(TPCW * tpcw)

```

```

{
    BOOK_DETAILS_ENQ_DATA bookdetail;
    int rc;
    struct _timeb ltime;
    struct tm *newtime;

    Parse(tpcw->szBuffer, tpcw->querystring);
    bookdetail.i_id= _atoi64(tpcw->szBuffer);
    memcpy(tpcw->endptr, bookinfo1, bookinfo1len);
    tpcw->endptr= tpcw->endptr+ bookinfo1len;
    rc= SQLBookDetailsProc(tpcw,&bookdetail);
    if(rc== FALSE)
    {
        strcpy(tpcw->err," BookInfo: SQLBookDetailsProc()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    memcpy(tpcw->endptr, bookinfo2, bookinfo2len);
    tpcw->endptr= tpcw->endptr+ bookinfo2len;
    _ftime(& ltime);

```

```

newtime= gmtime(&(ltime.time));
wsprintf( tpcw-> expires,"%s, %d %s %s %.4d %.2d:%.2d:%.2d GMT",
    day[ newtime-> tm_wday],
    newtime-> tm_mday,
    mon[ newtime-> tm_mon],
    newtime-> tm_year+ 1900,
    newtime-> tm_hour,
    newtime-> tm_min,
    newtime-> tm_sec);

```

```

tpcw->expiresflag= LATER;
return TRUE;
}

```

```

BOOL Search(TPCW *tpcw)

```

```

{
    int rc;
    memcpy(tpcw->endptr, search1, search1len);
    tpcw->endptr= tpcw->endptr+ search1len;
    rc= Advertisement(tpcw);
    if(rc== 0)
    {
        ErrorMessage(tpcw);
        return FALSE;
    }
    memcpy(tpcw->endptr, search2, search2len);
    tpcw->endptr= tpcw->endptr+ search2len;
    return TRUE;
}

```

```

BOOL SearchResultFrameset(TPCW * tpcw)

```

```

{
    char searchtype[SUBJ_LEN];
    char value[128];
    BOOK_DETAILS_ENQ_DATA book;

    Parse(searchtype, tpcw->querystring);
    Parse(value, tpcw->querystring);

    book.i_id= (__int64)RandomNumber(1, maxitem,&tpcw->lTLSeed);
    if(book.i_id <0 || book.i_id >(__int64)maxitem)
    {
        sprintf(tpcw->err,
            "SearchResultFrameset: BookID is not within
range[0-%d] (%I64d)",
            maxitem, book.i_id);
        ErrorMessage(tpcw);
        return 0;
    }
}

```

```

tpcw->len= wsprintf(tpcw->endptr,
    "<HTML><HEAD><TITLE>Search Result</TITLE></HEAD>"
    "<BODY BGCOLOR=#FFFFFF><H1 ALIGN=CENTER>TPC Web
Commerce Benchmark (TPC-W)</H1>"
    "<P ALIGN=CENTER><IMG SRC=gifs/tpclogo.gif\
WIDTH=288\ HEIGHT=67\ ALIGN=BOTTOM\ BORDER=0\>"
    "</P><H2 ALIGN=CENTER>Search Results Page - %s:
%s</H2>",
    searchtype, value);
tpcw->endptr= tpcw->endptr+ tpcw->len;

```

```

if (strcmp(searchtype,"subject")==0)
{
    tpcw->len= sprintf(tpcw->endptr,
        "<IFRAME SRC=http://%s/tpcw/tpcw.dll?p=%I64d\
WIDTH=800 HEIGHT=200 "
        "MARGINWIDTH=0\ MARGINHEIGHT=0\
SCROLLING=NO\ FRAMEBORDER=0\> </IFRAME>"
        "<IFRAME
SRC=http://%s/tpcw/list.dll?i=%s&value=%s\ WIDTH=800 HEIGHT=400 "
        "MARGINWIDTH=0\ MARGINHEIGHT=0\
FRAMEBORDER=0\> </IFRAME>",
        domainname, book.i_id, cachehost, searchtype,
value);
}

```

```

tpcw->endptr= tpcw->endptr+ tpcw->len;
else
{
    tpcw->len= sprintf(tpcw->endptr,
        "<IFRAME SRC=http://%s/tpcw/tpcw.dll?p=%I64d\
WIDTH=800 HEIGHT=200 "
        "MARGINWIDTH=0\ MARGINHEIGHT=0\
SCROLLING=NO\ FRAMEBORDER=0\> </IFRAME>"
        "<IFRAME
SRC=http://%s/tpcw/list.dll?i=%s&value=%s\ WIDTH=800 HEIGHT=400 "
        "MARGINWIDTH=0\ MARGINHEIGHT=0\
FRAMEBORDER=0\> </IFRAME>",
        cachehost, book.i_id, cachehost, searchtype,
value);
    tpcw->endptr= tpcw->endptr+ tpcw->len;
}
}

```

```

return TRUE;
}

```

```

BOOL PromoForSearchResult(TPCW * tpcw)

```

```

{
    BOOL rc;
    char value[128];
    BOOK_DETAILS_ENQ_DATA book;
    //
    TIMESTAMP_STRUCT btime;
}

```

```

unsigned int i;
char *chrptr;

Parse(value, tpcw->querystring);

book.i_id = _atoi64(value);

if(book.i_id <0 || book.i_id >(__int64)maxitem)
{
    sprintf(tpcw->err,
        "PromoForSearchResult: BookID is not within
range[0-%d] (%I64d)",
        maxitem, book.i_id);
    ErrorMessage(tpcw);
    return 0;
}

rc= SQLGetAD(tpcw,&book);
if(rc== FALSE)
{
    strcpy(tpcw->err, " PromoForSearchResult:
SQLGetAD(tpcw,&book)");
    ErrorMessage(tpcw);
    return 0;
}

//

memcpy(tpcw->endptr, advert1, advert1len);
tpcw->endptr= tpcw->endptr+ advert1len;
for(i= 0; i<5; i++)
{
    strcpy(tpcw->szBuffer, book.i_also[i].i_thumbnail+ 1);
    chrptr= tpcw->szBuffer;

    chrptr = book.i_also[i].i_thumbnail;
    chrptr+=strcspn(book.i_also[i].i_thumbnail, " \\");
    *chrptr = '\\0';

    tpcw->len= sprintf(tpcw->endptr,
        "<TD><A HREF=\\\"http://%s/tpcw/tpcw.dll?z=%I64d\\\"
target=\\\"_top\\\">
        <IMG SRC=\\\"http://image.tpcw.net/tpcw/%s\\\"
ALT=\\\"Book %d\\\" WIDTH=\\\"100\\\" HEIGHT=\\\"150\\\"></A></TD>",
        domainname,
        book.i_also[i].i_id,
        book.i_also[i].i_thumbnail,
        i+ 1);
    tpcw->endptr= tpcw->endptr + tpcw->len;
}

//

memcpy(tpcw->endptr, advert2, advert2len);
tpcw->endptr= tpcw->endptr + advert2len;

//

*tpcw->endptr= '\\0';
tpcw->expiresflag = NOT;

return TRUE;
}

long AddToSessionForBasketSwitch(TPCW * tpcw,
    struct BASKET_ITEM * shopitem, struct BASKET_LIST *
Basket_List[],
    struct BASKET_LIST * * ShopBasket)
{
    struct BASKET_LIST * p= NULL;
    struct BASKET_ITEM * newitem;
    int rc;
    CUSTOMER_INFO_ENQ_DATA cust;

    newitem= HeapAlloc(BasketHeap, HEAP_ZERO_MEMORY, sizeof(struct
BASKET_ITEM));
    if(newitem== NULL)
    {
        strcpy(tpcw->err, " AddToSession: newitem= HeapAlloc()");
        ErrorMessage(tpcw);
        return -1;
    }
    newitem->scl_shopping_id= tpcw->ID.SessionID;
    newitem->scl_i_id= shopitem->scl_i_id;
    newitem->scl_qty= shopitem->scl_qty;
    strcpy(newitem->scl_i_title, shopitem->scl_i_title);
    newitem->scl_i_srp= shopitem->scl_i_srp;
    newitem->scl_i_cost= shopitem->scl_i_cost;
    strcpy(newitem->scl_i_backing, shopitem->scl_i_backing);
    p= Basket_List[tpcw->hash]->next;
    while(p->next!= Basket_List[tpcw->hash] &&
        p->sc_shopping_id!= shopitem->scl_shopping_id)
    {
        p= p->next;
    }
    if(p->next== Basket_List[tpcw->hash]
        && p->sc_shopping_id!= shopitem->scl_shopping_id)
    {
        struct BASKET_LIST *newbasket;
        struct BASKET_ITEM *item;
        newbasket= HeapAlloc(BasketHeap, HEAP_ZERO_MEMORY,

```

```

        sizeof(struct BASKET_LIST));
if(newbasket== NULL)
{
    strcpy(tpcw->err," AddToSession: newbasket=
HeapAlloc()");
    ErrorMessage(tpcw);
    return -3;
}
item= HeapAlloc(BasketHeap, HEAP_ZERO_MEMORY,
    sizeof(struct BASKET_ITEM));
if(item== NULL)
{
    strcpy(tpcw->err," AddToSession: item=
HeapAlloc()");
    ErrorMessage(tpcw);
    return -4;
}
if(tpcw->ID.C_ID== 0)
{
    cust.c_fname[0]= ' ';
    cust.c_lname[0]= ' ';
    cust.c_fname[1]= '\0';
    cust.c_lname[1]= '\0';
}
else
{
    cust.c_id= tpcw->ID.C_ID;
    rc= SQLGetCustName(tpcw,&cust);
    if(rc== FALSE)
    {
        strcpy(tpcw->err," AddToSession:
SQLGetCustName()");
        ErrorMessage(tpcw);
        return -2;
    }
}
item->scl_i_id=-1L;
strcpy(item->scl_i_title," item header");
item->prev= newitem;
item->next= newitem;
newitem->prev= item;
newitem->next= item;
newbasket->Item= item;
newbasket->sc_shopping_id= shopitem->scl_shopping_id;
newbasket->sc_c_id= tpcw->ID.C_ID;
newbasket->sc_c_sub_total= shopitem->scl_i_cost;
strcpy(newbasket->sc_c_fname, cust.c_fname);
strcpy(newbasket->sc_c_lname, cust.c_lname);
newbasket->sc_c_ship_cost= 1;
if(tpcw->ID.C_ID>0)
{
    newbasket->sc_c_discount= cust.c_discount;

```

```

else
{
    newbasket->sc_c_discount= 0;
}
p= Basket_List[tpcw->hash];
newbasket->next= p->next;
p->next= newbasket;
newbasket->prev= p;
newbasket->next->prev= newbasket;
*ShopBasket= newbasket;
newbasket->valid=0;
#ifdef SCREEN_INFO
    EnterCriticalSection(&shop);
    numshop++;
    LeaveCriticalSection(&shop);
#endif
return 1;
}
else
{
    struct BASKET_ITEM *q= p->Item;
    p->sc_c_sub_total= 0;
    while(q->next!= p->Item &&q->scl_i_id!= newitem-
>scl_i_id)
    {
        q= q->next;
        p->sc_c_sub_total= p->sc_c_sub_total+(q-
>scl_i_cost*q->scl_qty);
    }
    if(q->scl_i_id!= newitem->scl_i_id)
    {
        q= p->Item;
        p->sc_c_sub_total= p->sc_c_sub_total+ newitem-
>scl_i_cost;
        newitem->next= q->next;
        q->next->prev= newitem;
        q->next= newitem;
        newitem->prev= q;
        *ShopBasket= p;
        return 2;
    }
else if(q->scl_i_id== newitem->scl_i_id)
{
    q->scl_qty++;
    shopitem->scl_qty= q->scl_qty;
    p->sc_c_sub_total= p->sc_c_sub_total+ newitem-
>scl_i_cost;
    while(q->next!= p->Item)
    {
        q= q->next;
        p->sc_c_sub_total= p->sc_c_sub_total+ q-

```

```

rc= HeapFree(BasketHeap, 0, newitem);
if(rc== 0)
{
    strcpy(tpcw->err, " AddToSession:
HeapFree()");
    ErrMessage(tpcw);
    return -5;
}
*ShopBasket= p;
return 3;
}
else
{
    strcpy(tpcw->err, " AddToSession: Should never
reach this point");
    ErrMessage(tpcw);
    return -6;
}
}
return 0;
}
long AddToSessionForRecovery(TPCW * tpcw,
    struct BASKET_ITEM * shopitem, struct BASKET_LIST *
Basket_List[],
    struct BASKET_LIST * * ShopBasket)
{
    struct BASKET_LIST * p= NULL;
    struct BASKET_ITEM * newitem;
    int rc;
    CUSTOMER_INFO_ENQ_DATA cust;

    newitem= HeapAlloc(BasketHeap, HEAP_ZERO_MEMORY, sizeof(struct
BASKET_ITEM));
    if(newitem== NULL)
    {
        strcpy(tpcw->err, " AddToSession: newitem= HeapAlloc()");
        ErrMessage(tpcw);
        return -1;
    }
    newitem->scl_shopping_id= tpcw->ID.SessionID;
    newitem->scl_i_id= shopitem->scl_i_id;
    newitem->scl_qty= shopitem->scl_qty;
    strcpy(newitem->scl_i_title, shopitem->scl_i_title);
    newitem->scl_i_srp= shopitem->scl_i_srp;
    newitem->scl_i_cost= shopitem->scl_i_cost;
    strcpy(newitem->scl_i_backing, shopitem->scl_i_backing);
    p= Basket_List[tpcw->hash]->next;
    while(p->next!= Basket_List[tpcw->hash] &&
        (p->sc_shopping_id!= shopitem->scl_shopping_id))
    {
        p= p->next;
    }
    if(p->next== Basket_List[tpcw->hash]

```

```

    && (p->sc_shopping_id!= shopitem->scl_shopping_id))
    {
        struct BASKET_LIST *newbasket;
        struct BASKET_ITEM *item;

        newbasket= HeapAlloc(BasketHeap, HEAP_ZERO_MEMORY,
            sizeof(struct BASKET_LIST));
        if(newbasket== NULL)
        {
            strcpy(tpcw->err, " AddToSession: newbasket=
HeapAlloc()");
            ErrMessage(tpcw);
            return -3;
        }
        item= HeapAlloc(BasketHeap, HEAP_ZERO_MEMORY,
            sizeof(struct BASKET_ITEM));
        if(item== NULL)
        {
            strcpy(tpcw->err, " AddToSession: item=
HeapAlloc()");
            ErrMessage(tpcw);
            return -4;
        }
        if(tpcw->ID.C_ID== 0)
        {
            cust.c_fname[0]= ' ';
            cust.c_lname[0]= ' ';
            cust.c_fname[1]= '\0';
            cust.c_lname[1]= '\0';
        }
        else
        {
            cust.c_id= tpcw->ID.C_ID;
            rc= SQLGetCustName(tpcw,&cust);
            if(rc== FALSE)
            {
                strcpy(tpcw->err, " AddToSession:
SQLGetCustName()");
                ErrMessage(tpcw);
                return -2;
            }
        }
        item->scl_i_id=-1L;
        strcpy(item->scl_i_title, " item header");
        item->prev= newitem;
        item->next= newitem;
        newitem->prev= item;
        newitem->next= item;
        newbasket->Item= item;
        newbasket->sc_shopping_id= shopitem->scl_shopping_id;
        newbasket->sc_c_id= tpcw->ID.C_ID;
        newbasket->sc_c_sub_total= shopitem->scl_i_cost;
        strcpy(newbasket->sc_c_fname, cust.c_fname);
    }
}

```

```

strcpy(newbasket->sc_c_lname, cust.c_lname);
newbasket->sc_c_ship_cost= 1;
if(tpcw->ID.C_ID>0)
{
    newbasket->sc_c_discount= cust.c_discount;
}
else
{
    newbasket->sc_c_discount= 0;
}
p= Basket_List[tpcw->hash];
newbasket->next= p->next;
p->next= newbasket;
newbasket->prev= p;
newbasket->next->prev= newbasket;
*ShopBasket= newbasket;
#ifdef SCREEN_INFO
    EnterCriticalSection(&shop);
    numshop++;
    LeaveCriticalSection(&shop);
#endif
return 1;
}
else
{
    struct BASKET_ITEM *q= p->Item;

    p->sc_c_sub_total= 0;
    while(q->next!= p->Item &&q->scl_i_id!= newitem-
>scl_i_id)
    {
        q= q->next;
        p->sc_c_sub_total= p->sc_c_sub_total+(q-
>scl_i_cost*q->scl_qty);
    }
    if(q->scl_i_id!= newitem->scl_i_id)
    {
        q= p->Item;
        p->sc_c_sub_total= p->sc_c_sub_total+ newitem-
>scl_i_cost;

        newitem->next= q->next;
        q->next->prev= newitem;
        q->next= newitem;
        newitem->prev= q;
        *ShopBasket= p;
        return 2;
    }
    else if(q->scl_i_id== newitem->scl_i_id)
    {
        q->scl_qty++;
        shopitem->scl_qty= q->scl_qty;
        p->sc_c_sub_total= p->sc_c_sub_total+ newitem-
>scl_i_cost;

        while(q->next!= p->Item)
        {
            q= q->next;
            p->sc_c_sub_total= p->sc_c_sub_total+ q-
>scl_i_cost;
        }
        rc= HeapFree(BasketHeap, 0, newitem);
        if(rc== 0)
        {
            strcpy(tpcw->err, " AddToSession:
                ErrMessage(tpcw);
                return -5;
        }
        *ShopBasket= p;
        return 3;
    }
    else
    {
        strcpy(tpcw->err, " AddToSession: Should never
                reach this point");
        ErrMessage(tpcw);
        return -6;
    }
}
return 0;
}

long AddToSession(TPCW * tpcw, struct BASKET_ITEM * shopitem,
    struct BASKET_LIST * Basket_List[],
    struct BASKET_LIST * * ShopBasket)
{
    struct BASKET_LIST *p= NULL;
    struct BASKET_ITEM *newitem;
    int rc;
    CUSTOMER_INFO_ENQ_DATA cust;

    newitem= HeapAlloc(BasketHeap, HEAP_ZERO_MEMORY, sizeof(struct
    BASKET_ITEM));
    if(newitem== NULL)
    {
        strcpy(tpcw->err, "AddToSession: newitem= HeapAlloc()");
        ErrMessage(tpcw);
        return -1;
    }
    newitem->scl_shopping_id= tpcw->ID.SessionID;
    newitem->scl_i_id= shopitem->scl_i_id;
    newitem->scl_qty= shopitem->scl_qty;
    strcpy(newitem->scl_i_title, shopitem->scl_i_title);
    newitem->scl_i_srp= shopitem->scl_i_srp;
    newitem->scl_i_cost= shopitem->scl_i_cost;
    strcpy(newitem->scl_i_backing, shopitem->scl_i_backing);
    rc= SearchForBasket(tpcw, Basket_List, &p);
}

```

```

if(rc== FALSE)
{
    strcpy(tpcw->err,"AddToSession: SearchForBasket()");
    ErrorMessage(tpcw);
    return -7;
}
if(p== NULL)
{
    struct BASKET_LIST *newbasket;
    struct BASKET_ITEM *item;
    newbasket= HeapAlloc(BasketHeap, HEAP_ZERO_MEMORY,
        sizeof(struct BASKET_LIST));
    if(newbasket== NULL)
    {
        strcpy(tpcw->err," AddToSession: newbasket=
HeapAlloc()");
        ErrorMessage(tpcw);
        return -3;
    }
    item= HeapAlloc(BasketHeap, HEAP_ZERO_MEMORY,
        sizeof(struct BASKET_ITEM));
    if(item== NULL)
    {
        strcpy(tpcw->err," AddToSession: item=
HeapAlloc()");
        ErrorMessage(tpcw);
        return -4;
    }
    if(tpcw->ID.C_ID== 0)
    {
        cust.c_fname[0]= ' ';
        cust.c_lname[0]= ' ';
        cust.c_fname[1]= '\0';
        cust.c_lname[1]= '\0';
    }
    else
    {
        cust.c_id= tpcw->ID.C_ID;
        rc= SQLGetCustName(tpcw,&cust);

        if(rc== FALSE)
        {
            strcpy(tpcw->err," AddToSession:
SQLGetCustName()");
            ErrorMessage(tpcw);
            return -2;
        }
    }
    item->scl_i_id=-1L;
    strcpy(item->scl_i_title," item header");
    item->prev= newitem;
    item->next= newitem;
    newitem->prev= item;
    newitem->next= item;
    newbasket->Item= item;
    newbasket->sc_shopping_id= shopitem->scl_shopping_id;
    newbasket->sc_c_id= tpcw->ID.C_ID;
    newbasket->sc_c_sub_total= shopitem->scl_i_cost;
    strcpy(newbasket->sc_c_fname, cust.c_fname);
    strcpy(newbasket->sc_c_lname, cust.c_lname);
    newbasket->sc_c_ship_cost= 1;
    if(tpcw->ID.C_ID>0)
    {
        newbasket->sc_c_discount= cust.c_discount;
    }
    else
    {
        newbasket->sc_c_discount= 0;
    }
    p= Basket_List[tpcw->hash];
    newbasket->next= p->next;
    p->next= newbasket;
    newbasket->prev= p;
    newbasket->next->prev= newbasket;
    newbasket->valid=0;
    *ShopBasket= newbasket;
#ifdef SCREEN_INFO
    EnterCriticalSection(&shop);
    numshop++;
    LeaveCriticalSection(&shop);
#endif
    return 1;
}
else
{
    struct BASKET_ITEM *q= p->Item;
    p->valid=0;
    p->sc_c_sub_total= 0;
    while(q->next!= p->Item &&q->scl_i_id!= newitem-
>scl_i_id)
    {
        q= q->next;
        p->sc_c_sub_total= p->sc_c_sub_total+(q-
>scl_i_cost*q->scl_qty);
    }
    if(q->scl_i_id!= newitem->scl_i_id)
    {
        q= p->Item;
        p->sc_c_sub_total= p->sc_c_sub_total+ newitem-
>scl_i_cost;
        newitem->next= q->next;
        q->next->prev= newitem;
        q->next= newitem;
        newitem->prev= q;
        *ShopBasket= p;
        return 2;
    }
}

```

```

    }
else if(q->scl_i_id== newitem->scl_i_id)
{
    q->scl_qty++;
    shopitem->scl_qty= q->scl_qty;
    p->sc_c_sub_total=p->sc_c_sub_total+newitem-
>scl_i_cost;
    while(q->next!=p->Item)
    {
        q=q->next;
        p->sc_c_sub_total=p->sc_c_sub_total+q-
>scl_i_cost;
    }
rc=HeapFree(BasketHeap,0,newitem);
if(rc==0)
{
    strcpy(tpcw->err,"AddToSession:
HeapFree()");
    ErrorMessage(tpcw);
    return -5;
}
*ShopBasket=p;
return 3;
}
else
{
    strcpy(tpcw->err,"AddToSession: Should never
reach this point");
    ErrorMessage(tpcw);
    return -6;
}
}
return 0;
}

BOOL RemoveBasket(TPCW * tpcw,
                  struct BASKET_ITEM * shopitem,
                  struct BASKET_LIST * * ShopBasket)
{
    struct BASKET_LIST * p;
    struct BASKET_LIST * t;
    int rc;
    p=*ShopBasket;
    if(p->sc_shopping_id==shopitem->scl_shopping_id)
    {
        p->prev->next=p->next;
        p->next->prev=p->prev;
        t=p->prev;
        rc=HeapFree(BasketHeap,0,p);
        if(rc==0)
            ErrorLog("HeapFree Error","RemoveBasket()");
            return FALSE;
        }
    }
else
    {
        return FALSE;
    }
}
else
    {
        return FALSE;
    }
}
return TRUE;
}

BOOL RemoveItem(TPCW * tpcw, struct BASKET_ITEM * shopitem,
                struct BASKET_LIST * * ShopBasket)
{
    struct BASKET_LIST * p;
    struct BASKET_ITEM * q;
    struct BASKET_ITEM * d;
    int rc;
    p=*ShopBasket;
    if(p->sc_shopping_id==shopitem->scl_shopping_id)
    {
        q=p->Item->next;
        while(q->scl_shopping_id!=shopitem-
>scl_shopping_id&&q!=p->Item)
        {
            q=q->next;
        }
        if(q->scl_shopping_id==shopitem->scl_shopping_id)
        {
            q->prev->next=q->next;
            q->next->prev=q->prev;
            d=q->prev;
            rc= HeapFree(BasketHeap, 0, q);
            if(rc== 0)
            {
                ErrorLog(" HeapFree Error", "
RemoveBasket()");
                return FALSE;
            }
        }
    }
else
    {
        return FALSE;
    }
}
else
    {
        return FALSE;
    }
}
return TRUE;
}

```

```

BOOL RemoveUpdate(TPCW * tpcw, struct BASKET_ITEM * shopitem,
                  struct BASKET_LIST * * ShopBasket)
{
    struct BASKET_LIST *p;
    struct BASKET_ITEM *q;
    p=*ShopBasket;
    if(p->scl_shopping_id== shopitem->scl_shopping_id)
    {
        q= p->Item->next;
        while(q->scl_shopping_id!= shopitem->
>scl_shopping_id&&q!= p->Item)
        {
            q= q->next;
        }
        if(q->scl_shopping_id== shopitem->scl_shopping_id)
        {
            q->scl_qty--;
        }
        else
        {
            return FALSE;
        }
    }
    else
    {
        return FALSE;
    }
    return TRUE;
}

LONG AddToBasket(TPCW * tpcw, struct BASKET_LIST * Basket_List[],
                BOOK_DETAILS_ENQ_DATA book)
{
    struct BASKET_ITEM shopitem;
    long increment;
    int rc;
    struct BASKET_LIST *pShopBasket= NULL;
    strcpy(shopitem.scl_i_backing, book.i_backing);
    shopitem.scl_i_cost= book.i_cost;
    shopitem.scl_i_id= book.i_id;
    shopitem.scl_i_srp= book.i_srp;
    strcpy(shopitem.scl_i_title, book.i_title);
    shopitem.scl_qty= 1;
    shopitem.scl_shopping_id= tpcw->ID.SessionID;
    MakeAlphaString(20,100,100, shopitem.scl_i_comments,&tpcw-
>lTLSeed);
    shopitem.scl_qty= 1;
    __try
    {
        EnterCriticalSection(&g_csBasket[tpcw->hash]);
        increment= AddToSession(tpcw,&shopitem,
Basket_List,&pShopBasket);
        pShopBasket->valid=0;
    }
}

```

```

if(increment<0)
{
    return FALSE;
}
switch(increment)
{
case 1:
    rc= SQLCreateBasket(tpcw, pShopBasket);
    if(rc== FALSE)
    {
        rc= RemoveBasket(tpcw,&shopitem,
Basket_List);
        wsprintf(tpcw->err, " AddToBasket:
SQLCreateBasket() Remove: %d", rc);
        ErrorMessage(tpcw);
        return FALSE;
    }
case 2:
    rc= SQLAddToBasket(tpcw,&shopitem, pShopBasket);
    if(rc== FALSE)
    {
        rc= RemoveItem(tpcw,&shopitem,
Basket_List);
        wsprintf(tpcw->err, " AddToBasket:
SQLAddToBasket() Remove: %d", rc);
        ErrorMessage(tpcw);
        pShopBasket->valid=1;
        return FALSE;
    }
    break;
case 3:
    __try
    {
        rc= SQLBeginTran(tpcw);
        if(rc== FALSE)
        {
            strcpy(tpcw->err, " AddToBasket:
SQLBeginTran()");
            ErrorMessage(tpcw);
            return FALSE;
        }
        rc= SQLUpdateItem(tpcw,&shopitem,
pShopBasket);
        if(rc== FALSE)
        {
            rc= RemoveUpdate(tpcw,&shopitem,
Basket_List);
            wsprintf(tpcw->err, " AddToBasket:
SQLUpdateItem() Remove: %d", rc);
            ErrorMessage(tpcw);
            return FALSE;
        }
    }
}
}

```

```

        __finally
        {
            rc= SQLCommitTran(tpcw);
            if(rc== FALSE)
            {
                strcpy(tpcw->err, " AddToBasket:
SQLCommitTran()");
                ErrMessage(tpcw);
                return FALSE;
            }
        }
        break;
    default:
        strcpy(tpcw->err, " AddToBasket: no match on
switch(increment)");
        ErrMessage(tpcw);
        pShopBasket->valid=1;
    }
    __finally
    {
        LeaveCriticalSection(&g_csBasket[tpcw->hash]);
    }
    return TRUE;
}

```

```

BOOL UpdateBasketCosts(struct BASKET_LIST * Basket, TPCW * tpcw)
{
    int rc;
    rc= SQLUpdateBasketCosts(tpcw, Basket);
    if(rc== FALSE)
    {
        strcpy(tpcw->err, " UpdateBasketCosts:
SQLUpdateBasketCosts()");
        ErrMessage(tpcw);
        return FALSE;
    }
    return TRUE;
}

```

```

BOOL SearchForBasket(TPCW * tpcw,
    struct BASKET_LIST * Basket_List[],
    struct BASKET_LIST * * ShopBasket)
{
    struct BASKET_LIST * p;
    int rc;
    *ShopBasket= NULL;
    p= Basket_List[tpcw->hash]->next;

    while ( p->next!= Basket_List[tpcw->hash]
        && (p->sc_shopping_id!= tpcw->ID.SessionID))
    {
        p= p->next;
    }
}

```

```

}
if((p->sc_shopping_id == tpcw->ID.SessionID) && p->valid
    && (tpcw->hostnum == hostnumber))
{
    *ShopBasket= p;
    return TRUE;
}
else if ((p->sc_shopping_id == tpcw->ID.SessionID)
    && (!p->valid || (tpcw->hostnum != hostnumber)))
{
    struct BASKET_LIST * t;
    struct BASKET_ITEM * q;
    struct BASKET_ITEM * d;

    if (p->Item != NULL)
    {
        if( p->Item->next!= NULL)
        {
            q= p->Item->next;
            while( q!= p->Item)
            {
                q->prev->next= q->next;
                q->next->prev= q->prev;
                d= q->prev;
                rc= HeapFree( BasketHeap, 0, q);
                if( rc== 0)
                {
                    ErrorLog(" HeapFree
Error", " DeleteBasket");
                }
                q= d;
                q= q->next;
            }
        }
        rc= HeapFree( BasketHeap, 0, p->Item);
        if( rc== 0)
        {
            ErrorLog(" HeapFree Error", "
DeleteBasket");
        }
    }
    p->prev->next= p->next;
    p->next->prev= p->prev;
    t= p->prev;
    rc= HeapFree( BasketHeap, 0, p);
    if( rc== 0)
    {
        ErrorLog(" HeapFree Error", " DeleteBasket");
    }
    p= t;
}
//

```

```

rc= SQLSwitchBasketItem(tpcw, Basket_List);
if(rc== FALSE)
{
    strcpy(tpcw->err, " SearchForBasket:
SQLSwitchBasketItem()");
    ErrorMessage(tpcw);
    return FALSE;
}
while(p->next!= Basket_List[tpcw->hash]
    && p->sc_shopping_id!= tpcw->ID.SessionID)
{
    p= p->next;
}
if(p->sc_shopping_id== tpcw->ID.SessionID)
{
    *ShopBasket= p;
#ifdef SCREEN_INFO
    EnterCriticalSection(&switchbasket);
    numswitchbasket++;
    LeaveCriticalSection(&switchbasket);
#endif
    return TRUE;
}
}
else if(tpcw->ID.SessionID <(LimitSession_ID-Session_ID_ALLOC)
    || tpcw->ID.SessionID >LimitSession_ID)
{
    rc= SQLSwitchBasketItem(tpcw, Basket_List);
    if(rc== FALSE)
    {
        strcpy(tpcw->err, " SearchForBasket:
SQLSwitchBasketItem()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    while(p->next!= Basket_List[tpcw->hash]
        && p->sc_shopping_id!= tpcw->ID.SessionID)
    {
        p= p->next;
    }
    if(p->sc_shopping_id== tpcw->ID.SessionID)
    {
        *ShopBasket= p;
#ifdef SCREEN_INFO
        EnterCriticalSection(&switchbasket);
        numswitchbasket++;
        LeaveCriticalSection(&switchbasket);
        #endif
        return TRUE;
    }
}

```

```

}
return TRUE;
}
}
BOOL ShoppingBasket(TPCW * tpcw, struct BASKET_LIST * Basket_List[])
{
    BOOK_DETAILS_ENQ_DATA book;
    struct BASKET_LIST * p=NULL;
    struct BASKET_ITEM * q;
    int rc;
    char Sessiondate[30];
    char * ptr;
    int i;
    Parse(tpcw->szBuffer, tpcw->querystring);
    if(strcmp(tpcw->szBuffer, "Y")== 0)
    {
        Parse(tpcw->szBuffer, tpcw->querystring);
        book.i_id= _atoi64(tpcw->szBuffer);
        ptr= tpcw->endptr;
        rc= SQLBookDetailsProc(tpcw, &book);
        if(rc== FALSE)
        {
            strcpy(tpcw->err, "ShoppingBasket:
SQLBookDetailsProc()");
            ErrorMessage(tpcw);
            return FALSE;
        }
        tpcw->endptr= ptr;
        rc= AddToBasket(tpcw, Basket_List, book);
        if(rc== FALSE)
        {
            return FALSE;
        }
    }
    CookieDate(Sessiondate);
    sprintf(tpcw->szHeader, "Set-Cookie: Session_ID=%d; expires=%s;
path=\\r\\nSet-Cookie: Web_ID=%d; expires=%s; path=\\r\\n%s",
        tpcw->ID.SessionID, Sessiondate, hostnumber,
Sessiondate, HEADER);
    memcpy(tpcw->endptr, shop1, shop1len);
    tpcw->endptr= tpcw->endptr+ shop1len;
    rc= Advertisement(tpcw);
    if(rc== 0)
    {
        ErrorMessage(tpcw);
        return FALSE;
    }
    memcpy(tpcw->endptr, shop2, shop2len);
}

```

```

tpcw->endptr= tpcw->endptr+ shop2len;

__try
{
    EnterCriticalSection(&g_csBasket[tpcw->hash]);
    rc= SearchForBasket(tpcw, Basket_List,&p);
    if(rc== FALSE)
    {
        strcpy(tpcw->err, " ShoppingBasket:
SearchForBasket()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    if(p== NULL|| p->sc_shopping_id!= tpcw->ID.SessionID||
p->Item->next== p->Item)
    {
        LeaveCriticalSection(&g_csBasket[tpcw->hash]);
        book.i_id= (__int64)RandomNumber(1,
maxitem,&tpcw->lTLSeed);

        if(book.i_id <0 || book.i_id > (__int64)maxitem)
        {
            sprintf(tpcw->err,"Advertisement: BookID
is not within range[0-%d] (%I64d)", maxitem, book.i_id);
            ErrorMessage(tpcw);
            return 0;
        }
        rc= SQLGetRelated1(tpcw,&book);
        if(rc== FALSE)
        {
            strcpy(tpcw->err, "ShoppingBasket:
SQLBookDetailsProc()");
            ErrorMessage(tpcw);
            return FALSE;
        }
        rc= AddToBasket(tpcw, Basket_List, book);
        if(rc== FALSE)
        {
            return FALSE;
        }
        EnterCriticalSection(&g_csBasket[tpcw->hash]);
        rc= SearchForBasket(tpcw, Basket_List,&p);
        if(rc== FALSE|| p== NULL)
        {
            strcpy(tpcw->err, "ShoppingBasket:
SearchForBasket()");
            ErrorMessage(tpcw);
            return FALSE;
        }
    }
}

```

```

//=====
=====
//=====
if(p->sc_shopping_id== tpcw->ID.SessionID)
{
    q= p->Item->next;
    i= 1;
    while(q!= p->Item)
    {
        floatprint(q->scl_i_srp,&tpcw->flt1);
        floatprint(q->scl_i_cost,&tpcw->flt2);
        tpcw->len= sprintf(tpcw->endptr,
            "<TR><TD VALIGN=\"TOP\">"
            "<INPUT TYPE=\"HIDDEN\"
NAME=\"ITEM%d\" VALUE=\"%I64d\">"
            "<INPUT NAME=\"QTY%d\"
VALUE=\"%d\" SIZE=\"3\"></TD>"
            "<TD VALIGN=\"TOP\"><I>%s</I>-
Backing: %s<BR>"
            "SRP.$%d.%2d, <FONT
COLOR=\"#AA0000\"><B>Your Price: $%d.%2d</B></FONT></TD></TR>",
            i,
            q->scl_i_id,
            i,
            q->scl_qty,
            q->scl_i_title,
            q->scl_i_backing,
            tpcw->flt1.x, tpcw->flt1.y,
            tpcw->flt2.x, tpcw->flt2.y);
        tpcw->endptr= tpcw->endptr+ tpcw->len;
        q= q->next;
    }
    floatprint(p->sc_c_sub_total,&tpcw->flt1);
    tpcw->len= wsprintf(tpcw->endptr,
        "</TABLE><B><I>Subtotal price: $"
        "%d.%2d </B><BR>%s</I></P>",
        tpcw->flt1.x, tpcw->flt1.y,
        ctime(&p->sc_date));
    tpcw->endptr= tpcw->endptr+ tpcw->len;
}
}
__finally
{
    LeaveCriticalSection(&g_csBasket[tpcw->hash]);
}
memcpy(tpcw->endptr, shop3, shop3len);
tpcw->endptr= tpcw->endptr+ shop3len;
return TRUE;
}

```

```

BOOL Adjustment(TPCW *tpcw, struct BASKET_LIST *Basket_List[])

```





```

tpcw->len= wsprintf(tpcw->endptr, "<HTML><HEAD><TITLE>TPC-W
Order Inquiry Page</TITLE></HEAD>"
"<BODY BGCOLOR=#FFFFFF">"
"<H1 ALIGN=#\"CENTER\">TPC Web
Commerce Benchmark (TPC-W)</H1>"
"<H2 ALIGN=#\"CENTER\">Order
Inquiry Page</H2>"
"<FORM
ACTION=#\"https://s/tpcw/tpcw.dll?g\" METHOD=#\"POST\">"
"<TABLE ALIGN=#\"CENTER\">"
"<TR><TD><H4>Username:</TD><TD><INPUT NAME=#\"USERNAME\"
SIZE=#\"10\"></TD></TR>"
"<TR><TD><H4>Password:</TD><TD><INPUT TYPE=#\"PASSWORD\"
NAME=#\"password\" SIZE=#\"10\"></TD></TR>"
"</TABLE>"
"<P ALIGN=#\"CENTER\"><BR>"
"<INPUT TYPE=#\"IMAGE\"
NAME=#\"Submit\" SRC=#\"gifs/ord_sta.gif\" WIDTH=#\"120\"
HEIGHT=#\"30\">"
"<A
HREF=#\"http://s/tpcw/tpcw.dll?h\"><IMG SRC=#\"gifs/search.gif\"
ALT=#\"Search\" BORDER=#\"0\" WIDTH=#\"120\" HEIGHT=#\"30\"></A>"
"<A
HREF=#\"http://s/tpcw/tpcw.dll?a\"><IMG SRC=#\"gifs/home.gif\"
ALT=#\"Home\" BORDER=#\"0\" WIDTH=#\"120\" HEIGHT=#\"30\"></A>"
"</FORM></BODY></HTML>" ,
domainname,
domainname,
domainname,
domainname);
*/
//After code changed
//=====
//=====
CUSTOMER_INFO_ENQ_DATA cust;
BOOL rc;
if(tpcw->ID.C_ID== 0)
{
    cust.c_uname[0]= '\0';
}
else
{
    cust.c_id= tpcw->ID.C_ID;
    rc= SQLGetCustUname(tpcw,&cust);
    if(rc== FALSE)
    {
        strcpy(tpcw->err, " OrderStatus:
SQLGetCustUname()");
        ErrorMessage(tpcw);
        return FALSE;
    }
}
tpcw->len= wsprintf(tpcw->endptr, "<HTML><HEAD><TITLE>TPC-W
Order Inquiry Page</TITLE></HEAD>"

```

```

"<BODY BGCOLOR=#\"FFFFFF\">"
"<H1 ALIGN=#\"CENTER\">TPC Web Commerce Benchmark (TPC-
W)</H1>"
"<H2 ALIGN=#\"CENTER\">Order Inquiry Page</H2>"
"<FORM ACTION=#\"https://s/tpcw/tpcw.dll?\"
METHOD=#\"GET\">"
"<INPUT TYPE=#\"HIDDEN\" NAME=#\"g\" VALUE=#\"\">"
"<TABLE ALIGN=#\"CENTER\">"
"<TR><TD><H4>Username:</TD><TD><INPUT NAME=#\"USERNAME\"
SIZE=#\"10\" value=#\"%s\"></TD></TR>"
"<TR><TD><H4>Password:</TD><TD><INPUT TYPE=#\"PASSWORD\"
NAME=#\"password\" SIZE=#\"10\"></TD></TR>"
"</TABLE>"
"<P ALIGN=#\"CENTER\"><BR>"
"<INPUT TYPE=#\"IMAGE\" NAME=#\"Submit\"
SRC=#\"gifs/status.gif\" WIDTH=#\"120\" HEIGHT=#\"30\">"
"<A HREF=#\"http://s/tpcw/tpcw.dll?h\"><IMG
SRC=#\"gifs/search.gif\" ALT=#\"Search\" BORDER=#\"0\" WIDTH=#\"120\"
HEIGHT=#\"30\"></A>"
"<A HREF=#\"http://s/tpcw/tpcw.dll?a\"><IMG
SRC=#\"gifs/home.gif\" ALT=#\"Home\" BORDER=#\"0\" WIDTH=#\"120\"
HEIGHT=#\"30\"></A>"
"</FORM></BODY></HTML>" ,
domainname,
cust.c_uname,
domainname,
domainname);
//=====
//=====
tpcw->endptr= tpcw->endptr+ tpcw->len;
return TRUE;
}
BOOL OrderStatusResult(TPCW * tpcw)
{
    CUSTOMER_INFO_ENQ_DATA cust;
    ADDRESS_ENQ_DATA shipaddr, billaddr;
    COUNTRY_ENQ_DATA shipcountry, billcountry;
    ORDER_ENQ_DATA order;
    int rc;

    DecodeParse(cust.c_uname, tpcw->querystring);
    DecodeParse(cust.c_passwd, tpcw->querystring);
    cust.c_id= 0;
    rc= SQLUnamePwdCheck(tpcw,&cust);
    if(rc== FALSE)
    {
        strcpy(tpcw->err, " OrderStatusResult:
SQLUnamePwdCheck()");
        ErrorMessage(tpcw);
        return FALSE;
    }
}

```

```

memcpy(tpcw->endptr, orderstat1, orderstatllen);
tpcw->endptr= tpcw->endptr+ orderstatllen;
if(cust.c_id!= 0)
{
    order.o_id= 0;
    rc=
SQLGetCustOrderInfo(tpcw,&cust,&billaddr,&billcountry,&order);
    if(rc== FALSE)
    {
        strcpy(tpcw->err, " OrderStatusResult:
SQLGetCustOrderInfo()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    if(order.o_id== 0)
    {
        tpcw->len= sprintf(tpcw->endptr,
            "<H2 ALIGN=\"CENTER\">No order
found</H2>"
                "<BR><CENTER>"
                "<A
HREF=\"http://s/tpcw/tpcw.dll?h\"><IMG SRC=\"gifs/search.gif\"
ALT=\"Search\" BORDER=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>"
                "<A
HREF=\"http://s/tpcw/tpcw.dll?a\"><IMG SRC=\"gifs/home.gif\"
ALT=\"Home\" BORDER=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>"
                "</CENTER></BODY></HTML>",
                domainname,
                domainname);
        tpcw->endptr= tpcw->endptr+ tpcw->len;
        return TRUE;
    }
    if(order.o_bill_addr_id== order.o_ship_addr_id)
    {
        strcpy(shipaddr.a_street1, billaddr.a_street1);
        strcpy(shipaddr.a_street2, billaddr.a_street2);
        strcpy(shipaddr.a_city, billaddr.a_city);
        strcpy(shipaddr.a_state, billaddr.a_state);
        strcpy(shipaddr.a_zip, billaddr.a_zip);
        strcpy(shipcountry.co_name,
billcountry.co_name);
    }
    else
    {
        shipaddr.a_id= order.o_ship_addr_id;
        rc= SQLGetAddr(tpcw,&shipaddr,&shipcountry);
        if(rc== FALSE)
        {
            strcpy(tpcw->err, " OrderStatusResult:
SQLGetAddr()");
            ErrorMessage(tpcw);
            return FALSE;
        }
    }
}

```

```

}
floatprint(order.o_subtotal,&tpcw->flt1);
floatprint(order.o_tax,&tpcw->flt2);
floatprint(order.o_total,&tpcw->flt3);
tpcw->len= sprintf(tpcw->endptr, " Order ID: %I64d<BR>"
    "Authorization ID: %s<BR>" //
gnd 02-01-01
    "Order Placed on %d/%d/%d<BR>"
    "Shipping Type: %s<BR>"
    "Ship Date: %d/%d/%d<BR>"
    "Order Subtotal: $%d.%2d<BR>"
    "Order Tax: $%d.%2d<BR>"
    "Order Total: $%d.%2d<BR>"
    "</P>",
    order.o_id,
    order.o_cc_auth, // gnd 02-
    order.o_date.month, order.o_date.day,
    order.o_date.year,
    order.o_ship_type,
    order.o_ship_date.month, order.o_ship_date.day,
    order.o_ship_date.year,
    tpcw->flt1.x, tpcw->flt1.y,
    tpcw->flt2.x, tpcw->flt2.y,
    tpcw->flt3.x, tpcw->flt3.y);
tpcw->endptr= tpcw->endptr+ tpcw->len;
memcpy(tpcw->endptr, orderstat2, orderstat2len);
tpcw->endptr= tpcw->endptr+ orderstat2len;
tpcw->len= sprintf(tpcw->endptr,
    "<TR><TD COLSPAN=\"2\"><H4>%s %s</TD></TR>"
    "<TR><TD WIDTH=\"50%%\"><ADDRESS>%s<BR>"
    "%s<BR>"
    "%s<BR>"
    "%s %s<BR>"
    "%s<BR><BR>"
    "Email: %s<BR>"
    "Phone: %s</ADDRESS><BR>"
    "<P>"
    "Credit Card Type: %s<BR>"
    "Order Status: %s"
    "</P></TD>"
    "<TD WIDTH=\"50%%\"
VALIGN=\"TOP\"><ADDRESS>%s<BR>"
    "%s<BR>"
    "%s<BR>"
    "%s %s<BR>"
    "%s</ADDRESS>"
    "</TD></TR></TABLE>",
    cust.c_fname,
    cust.c_lname,
    billaddr.a_street1,
    billaddr.a_street2,
    billaddr.a_city,

```

```

        billaddr.a_state,
        billaddr.a_zip,
        billcountry.co_name,
        cust.c_email,
        cust.c_phone,
        order.o_cc_type,
        order.o_status,
        shipaddr.a_street1,
        shipaddr.a_street2,
        shipaddr.a_city,
        shipaddr.a_state,
        shipaddr.a_zip,
        shipcountry.co_name);
    tpcw->endptr = tpcw->endptr + tpcw->len;
    memcpy(tpcw->endptr, orderstat3, orderstat3len);
    tpcw->endptr = tpcw->endptr + orderstat3len;
    rc = SQLGetOrderLine(tpcw, order.o_id);
    if(rc == FALSE)
    {
        strcpy(tpcw->err, " OrderStatusResult:
SQLGetOrderLine()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    tpcw->len = sprintf(tpcw->endptr,
        "</TABLE>"
        "<BR>"
        "<A HREF=\"http://%s/tpcw/tpcw.dll?h\"><IMG
SRC=\"gifs/search.gif\" ALT=\"Search\" BORDER=\"0\" WIDTH=\"120\"
HEIGHT=\"30\"></A>"
        "<A HREF=\"http://%s/tpcw/tpcw.dll?a\"><IMG
SRC=\"gifs/home.gif\" ALT=\"Home\" BORDER=\"0\" WIDTH=\"120\"
HEIGHT=\"30\"></A>"
        "</CENTER></BODY></HTML>",
        domainname,
        domainname);
    tpcw->endptr = tpcw->endptr + tpcw->len;
}
else
{
    tpcw->len = sprintf(tpcw->endptr, "<H2
ALIGN=\"CENTER\">Record does not match</H2>"
        "<BR><CENTER>"
        "<A HREF=\"http://%s/tpcw/tpcw.dll?h\"><IMG
SRC=\"gifs/search.gif\" ALT=\"Search\" BORDER=\"0\" WIDTH=\"120\"
HEIGHT=\"30\"></A>"
        "<A HREF=\"http://%s/tpcw/tpcw.dll?a\"><IMG
SRC=\"gifs/home.gif\" ALT=\"Home\" BORDER=\"0\" WIDTH=\"120\"
HEIGHT=\"30\"></A>"
        "</CENTER></BODY></HTML>",
        domainname,
        domainname);
    tpcw->endptr = tpcw->endptr + tpcw->len;
}

```

```

    }
    return TRUE;
}
BOOL CustomerReg(TPCW *tpcw)
{
    memcpy(tpcw->endptr, custreg1, custreg1len);
    tpcw->endptr = tpcw->endptr + custreg1len;
    tpcw->len = sprintf(tpcw->endptr,
        "<FORM ACTION=\"https://%s/tpcw/tpcw.dll?\"
METHOD=\"GET\">"
        "<INPUT TYPE=\"HIDDEN\" NAME=\"d\" VALUE=\"\">",
        domainname);
    tpcw->endptr = tpcw->endptr + tpcw->len;
    memcpy(tpcw->endptr, custreg2, custreg2len);
    tpcw->endptr = tpcw->endptr + custreg2len;
    tpcw->len = sprintf(tpcw->endptr,
        "<A HREF=\"http://%s/tpcw/tpcw.dll?h\"><IMG
SRC=\"gifs/search.gif\" ALT=\"Search Item\" BORDER=\"0\"
VSPACE=\"0\" HSPACE=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>"
        "<A HREF=\"http://%s/tpcw/tpcw.dll?a\"><IMG
SRC=\"gifs/home.gif\" ALT=\"Home Page\" BORDER=\"0\" VSPACE=\"0\"
HSPACE=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>"
        "</P></FORM></BODY></HTML>",
        domainname,
        domainname);
    tpcw->endptr = tpcw->endptr + tpcw->len;
    return TRUE;
}
BOOL NewOrderLine(TPCW *tpcw, ORDER_ENQ_DATA order, ADDRESS_ENQ_DATA
    shipaddress, COUNTRY_ENQ_DATA shipcountry, struct BASKET_LIST
    *Basket_List[])
{
    struct BASKET_ITEM *d;
    struct BASKET_LIST *p = NULL;
    CUSTOMER_INFO_ENQ_DATA cust;
    struct BASKET_ITEM *q;
    char auth_id[15];
    long rc;

    if(shipaddress.a_street1[0] != '\0')
    {
        __try
        {
            EnterCriticalSection(&csADDR_ID);
            shipaddress.a_id = ADDR_ID;
            ADDR_ID++;
            if(ADDR_ID > LimitADDR_ID)
            {
                rc = ReadMaxADDR_ID(tpcw, ADDR_ID_ALLOC);
            }
        }
    }
}

```

```

        if(rc== 0)
        {
            strcpy(tpcw->err,"NewOrderLine:
rc= ReadMaxADDR_ID(tpcw, O_ID_ALLOC)== 0");
            ErrorMessage(tpcw);
            return FALSE;
        }
        else
        {
            shipaddress.a_id= ADDR_ID;
            ADDR_ID++;
        }
    }
    }
    __finally
    {
        LeaveCriticalSection(&csADDR_ID);
    }
    rc= SQLAddNewAddress(tpcw,&shipaddress,&shipcountry);
    if(rc== FALSE)
    {
        strcpy(tpcw->err," NewOrderLine:
SQLAddNewAddress()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    else
    {
        rc= SQLGetCountryID(tpcw,&shipcountry);
        if(rc== FALSE)
        {
            strcpy(tpcw->err," NewOrderLine:
SQLGetCountryID()");
            ErrorMessage(tpcw);
            return FALSE;
        }
    }
    cust.c_id= tpcw->ID.C_ID;
    rc= SQLGetCustAddrID(tpcw,&cust);
    if(rc== FALSE)
    {
        strcpy(tpcw->err," NewOrderLine: SQLGetCustAddrID()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    order.o_c_id= cust.c_id;
    order.o_bill_addr_id= cust.c_addr_id;
    if(shipaddress.a_street1[0]!='\0')
    {
        order.o_ship_addr_id= shipaddress.a_id;
    }
    else

```

```

    {
        order.o_ship_addr_id= cust.c_addr_id;
    }
    }
    __try
    {
        EnterCriticalSection(&g_csBasket[tpcw->hash]);
        rc= SearchForBasket(tpcw, Basket_List,&p);
        if(rc== FALSE)
        {
            strcpy(tpcw->err," Adjustment:
SearchForBasket()");
            ErrorMessage(tpcw);
        }
        if(p== NULL|| p->Item== p->Item->next)
        {
            tpcw->len= wsprintf(tpcw->endptr,
                "HTML<HEAD><TITLE>TPC-W Shopping Cart
Page</TITLE></HEAD>"
                "<BODY BGCOLOR=\#FFFFFF\>"
                "<H2 ALIGN=\\"CENTER\>"
                "Thank for shopping at TPC-W"
                "<H2 ALIGN=\\"CENTER\>"
                "<A
                HREF=\\"http://%s/tpcw/tpcw.dll?h\><IMG SRC=\\"gifs/cart.gif\\"
                ALT=\\"Shopping Cart\\" BORDER=\\"0\\" VSPACE=\\"0\\" HSPACE=\\"0\\"
                WIDTH=\\"120\\" HEIGHT=\\"30\></A>"
                "<A
                HREF=\\"http://%s/tpcw/tpcw.dll?a\><IMG SRC=\\"gifs/home.gif\\"
                ALT=\\"Home\\" BORDER=\\"0\\" VSPACE=\\"0\\" HSPACE=\\"0\\" WIDTH=\\"120\\"
                HEIGHT=\\"30\></A></CENTER></BODY></HTML>",
                domainname,
                domainname);
            tpcw->endptr= tpcw->endptr+ tpcw->len;
            return TRUE;
        }
        cust.c_discount= p->sc_c_discount;
        order.o_subtotal= p->sc_c_sub_total;
        order.o_tax= p->sc_c_tax;
        order.o_total= p->sc_c_total;
    }
    }
    __finally
    {
        LeaveCriticalSection(&g_csBasket[tpcw->hash]);
    }
    order.o_id= 0;
    }
    }
    __try
    {
        EnterCriticalSection(&csO_ID);
        order.o_id= O_ID;
        O_ID++;
        if(O_ID >LimitO_ID)

```

```

    {
        rc= ReadMaxO_ID(tpcw, O_ID_ALLOC);
        if(rc== 0)
        {
            strcpy(tpcw->err, " NewOrderLine: rc=
ReadMaxO_ID(tpcw, O_ID_ALLOC)= 0");
            ErrorMessage(tpcw);
            return FALSE;
        }
        else
        {
            order.o_id= O_ID;
            O_ID++;
        }
    }
}
__finally
{
    LeaveCriticalSection(&csO_ID);
}

#ifdef SCREEN_INFO
EnterCriticalSection(&pgereq);
numpgereq++;
LeaveCriticalSection(&pgereq);
#endif

rc= PGERequest(&tpcw->pge, order.o_id, order.o_cc_num,
auth_id);

#ifdef SCREEN_INFO
EnterCriticalSection(&pgereq);
numpgereq--;
LeaveCriticalSection(&pgereq);
#endif

if(rc!= 1)
{
    wsprintf(tpcw->err, "NewOrderLine: PGERequest() returned
%d", rc);
    ErrorMessage(tpcw);
    return FALSE;
}
p->valid = 0;
rc= SQLAddNewOrder(tpcw, &order, shipcountry.co_id,
cust.c_discount, auth_id, p);
if(rc== FALSE)
{
    sprintf(tpcw->err, " NewOrderLine: SQLAddNewOrder() Order
ID: %I64d Session ID: %d", order.o_id, tcw->ID.SessionID);
    ErrorMessage(tpcw);
    return FALSE;
}

```

```

memcpy(tpcw->endptr, neworder1, neworder1len);
tpcw->endptr= tcw->endptr+ neworder1len;
memcpy(tpcw->endptr, neworder2, neworder2len);
tpcw->endptr= tcw->endptr+ neworder2len;
__try
{
    EnterCriticalSection(&g_csBasket[tpcw->hash]);
    q= p->Item->next;
    while(q!= p->Item)
    {
        floatprint(q->scl_i_srp, &tpcw->flt1);
        floatprint(q->scl_i_cost, &tpcw->flt2);
        tcw->len= wsprintf(tpcw->endptr,
            "<TR><TD VALIGN=\"TOP\">%d</TD>"
            "<TD VALIGN=\"TOP\"><I>Title: %s</I>-
Backing: %s<BR>"
            "SRP.$%d.%2d, <FONT
COLOR=\"#AA0000\"><B>Your Price: $%d.%2d</B></FONT></TD></TR>",
            q->scl_qty,
            q->scl_i_title,
            q->scl_i_backing,
            tcw->flt1.x, tcw->flt1.y,
            tcw->flt2.x, tcw->flt2.y);
        tcw->endptr= tcw->endptr+ tcw->len;
        q->prev->next= q->next;
        q->next->prev= q->prev;
        d= q->prev;
        rc= HeapFree(BasketHeap, 0, q);
        if(rc== 0)
        {
            strcpy(tpcw->err, " NewOrderLine:
HeapFree(BasketHeap, 0, q)");
            ErrorMessage(tpcw);
            return FALSE;
        }
        q= d;
        q= q->next;
    }
}
memcpy(tpcw->endptr, neworder3, neworder3len);
tpcw->endptr= tcw->endptr+ neworder3len;

}
__finally
{
    LeaveCriticalSection(&g_csBasket[tpcw->hash]);
}
memcpy(tpcw->endptr, neworder4, neworder4len);
tpcw->endptr= tcw->endptr+ neworder4len;
floatprint(order.o_subtotal, &tpcw->flt1);
floatprint(order.o_tax, &tpcw->flt2);
floatprint(order.o_total, &tpcw->flt3);
floatprint(p->sc_c_ship_cost, &tpcw->flt4);
tpcw->len= sprintf(tpcw->endptr,

```

```

        "<TR><TD><H4>Subtotal with
discount(%d%%):</TD><TD><H4>$.2d</TD></TR>"
        "<TR><TD><H4>Tax
(8.25%%):</TD><TD><H4>$.2d</TD></TR>"
        "<TR><TD><H4>Shipping:</TD><TD><H4>$.2d</TD></TR>"
        "<TR><TD><H4>Total:</TD><TD><H4>$.2d</TD></TR>"
        "</TABLE>"
        "<P><BR></P>"
        "<H2>Confirmation Number: %I64d</H2>"
        "<H1>Thank you for shopping at TPC-W</H1>"
        "<P></P><CENTER>"
        "<A HREF=\"http://%s/tpcw/tpcw.dll?h\"><IMG
SRC=\"gifs/search.gif\" ALT=\"Search Item\" BORDER=\"0\" VSPACE=\"0\"
HSPACE=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>"
        "<A HREF=\"http://%s/tpcw/tpcw.dll?a\"><IMG
SRC=\"gifs/home.gif\" ALT=\"Home Page\" BORDER=\"0\" VSPACE=\"0\"
HSPACE=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>"

        "</CENTER></BLOCKQUOTE></BLOCKQUOTE></BLOCKQUOTE></BLOCKQUOTE><
/BODY></HTML>" ,
        cust.c_discount,
        tpcw->flt1.x, tpcw->flt1.y,
        tpcw->flt2.x, tpcw->flt2.y,
        tpcw->flt4.x, tpcw->flt4.y,
        tpcw->flt3.x, tpcw->flt3.y,
        order.o_id,
        domainname,
        domainname);
        tpcw->endptr= tpcw->endptr+ tpcw->len;
        p->sc_c_ship_cost= 0;
        p->sc_c_sub_total= 0;
        p->sc_c_tax= 0;
        p->sc_c_total= 0;
        p->valid = 1;
        return TRUE;
}

BOOL BuyRequest(TPCW *tpcw, CUSTOMER_INFO_ENQ_DATA cust,
                ADDRESS_ENQ_DATA addr, COUNTRY_ENQ_DATA
country,
                struct BASKET_LIST *Basket_List[])
{
    char C_IDdate[30];
    long rc;
    struct BASKET_LIST *p;
    struct BASKET_ITEM *q;
    double cost;
    double discount;
    CookieDate(C_IDdate);
    sprintf(tpcw->szHeader,"Set-Cookie: C_ID=%I64d; expires=%s;
path=\\r\\n%s",
            cust.c_id, C_IDdate, HEADER);

```

```

        _try
        {
            EnterCriticalSection(&g_csBasket[tpcw->hash]);
            rc= SearchForBasket(tpcw, Basket_List,&p);
            if(rc== FALSE)
            {
                strcpy(tpcw->err, " BuyRequest:
SearchForBasket()");
                ErrorMessage(tpcw);
                return FALSE;
            }
            if(p!= NULL && p->sc_shopping_id==tpcw->ID.SessionID)
            {
                p->valid = 0;
                if(cust.c_id!= tpcw->ID.C_ID)
                {
                    p->sc_c_id= cust.c_id;
                    tpcw->ID.C_ID= cust.c_id;
                    strcpy(p->sc_c_fname, cust.c_fname);
                    strcpy(p->sc_c_lname, cust.c_lname);
                }

                memcpy(tpcw->endptr, buyreq5, buyreq5len);
                tpcw->endptr= tpcw->endptr+ buyreq5len;
                tpcw->len= sprintf(tpcw->endptr,
                    "<TABLE BORDER=\"0\" WIDTH=\"90%\"
CELLSPACING=\"0\" CELLPADDING=\"0\">"
                    "<TR ALIGN=\"LEFT\" VALIGN=\"TOP\">"
                    "<TD VALIGN=\"TOP\" WIDTH=\"45%\">"
                    "<H2>Billing Information:</H2>"
                    "<TABLE WIDTH=\"100%\" BORDER=\"0\"
CELLSPACING=\"0\" CELLPADDING=\"0\">"
                    "<TR><TD>Firstname:</TD><TD>%s</TD></TR>"
                    "<TR><TD>Lastname:</TD><TD>%s</TD></TR>"
                    "<TR><TD>Addr_street_1:</TD><TD>%s</TD></TR>"
                    "<TR><TD>Addr_street_2:</TD><TD>%s</TD></TR>"
                    "<TR><TD>City:</TD><TD>%s</TD></TR>"
                    "<TR><TD>State:</TD><TD>%s</TD></TR>"
                    "<TR><TD>Zip:</TD><TD>%s</TD></TR>"
                    "<TR><TD>Country:</TD><TD>%s</TD></TR>"
                    "<TR><TD>Email:</TD><TD>%s</TD></TR>"
                    "<TR><TD>Phone:</TD><TD>%s</TD></TR>"
                    "<TR><TD>Username:</TD><TD>%s</TD></TR>"
                    "<TR><TD>User
ID:</TD><TD>%I64d</TD></TR>"
                    "</TABLE></TD>",
                    cust.c_fname,
                    cust.c_lname,
                    addr.a_street1,
                    addr.a_street2,

```

```

        addr.a_city,
        addr.a_state,
        addr.a_zip,
        country.co_name,
        cust.c_email,
        cust.c_phone,
        cust.c_uname,
        cust.c_id
    );

    tpcw->endpstr= tpcw->endpstr+ tpcw->len;
    tpcw->len= sprintf(tpcw->endpstr,
        "<TD ALIGN=\"TOP\" WIDTH=\"45%\">"
        "<H2>Shipping Information:</H2>"
        "<TABLE WIDTH=\"100%\" BORDER=\"0\"
CELLSPACING=\"0\" CELLPADDING=\"0\">"
        "<TR>"
        "<TD WIDTH=\"50%\">Addr_street_1:</TD>"
        "<TD WIDTH=\"50%\"><INPUT
NAME=\"SHIPADD1\" SIZE=\"40\" VALUE=\"%s\"></TD>"
        "</TR>"
        "<TR>"
        "<TD>Addr_street_2:</TD>"
        "<TD><INPUT NAME=\"SHIPADD2\" SIZE=\"40\"
VALUE=\"%s\"></TD>"
        "</TR>"
        "<TR>"
        "<TD>City:</TD>"
        "<TD><INPUT NAME=\"SHIPCITY\" SIZE=\"30\"
VALUE=\"%s\"></TD>"
        "</TR>"
        "<TR>"
        "<TD>State:</TD>"
        "<TD><INPUT NAME=\"SHIPSTATE\"
SIZE=\"20\" VALUE=\"%s\"></TD>"
        "</TR>"
        "<TR>"
        "<TD>Zip:</TD>"
        "<TD><INPUT NAME=\"SHIPZIP\" SIZE=\"10\"
VALUE=\"%s\"></TD>"
        "</TR>"
        "<TR>"
        "<TD>Country:</TD>"
        "<TD><INPUT NAME=\"SHIPCOUNTRY\"
SIZE=\"40\" VALUE=\"%s\"></TD>"
        "</TR>"
        "</TABLE>"
        "</TD></TR></TABLE>",
        addr.a_street1,
        addr.a_street2,
        addr.a_city,
        addr.a_state,

```

```

        addr.a_zip,
        country.co_name
    );

    tpcw->endpstr= tpcw->endpstr+ tpcw->len;
    memcpy(tpcw->endpstr, buyreq1, buyreq1len);
    tpcw->endpstr= tpcw->endpstr+ buyreq1len;
    p->sc_c_discount= cust.c_discount;
    rc= UpdateBasketCosts(p, tpcw);
    if(rc== FALSE)
    {
        return FALSE;
    }
    memcpy(tpcw->endpstr, buyreq2, buyreq2len);
    tpcw->endpstr= tpcw->endpstr+ buyreq2len;
    q= p->Item->next;
    p->sc_c_sub_total= 0;
    p->sc_c_ship_cost= 0;
    discount=(100- p->sc_c_discount)/100.0;
    while(q!= p->Item)
    {
        floatprint(q->scl_i_srp,&tpcw->flt1);
        floatprint(q->scl_i_cost,&tpcw->flt2);
        tpcw->len= sprintf(tpcw->endpstr,
            "<TR><TD VALIGN=\"TOP\">%d</TD>"
            "<TD VALIGN=\"TOP\">Title:
<I>%s</I>- Backing: %s<BR>"
            "SRP.$d.%2d, <FONT
COLOR=\"#AA0000\"><B>Your Price: $%d.%2d</B></FONT></TR>",
            q->scl_qty,
            q->scl_i_title,
            q->scl_i_backing,
            tpcw->flt1.x, tpcw->flt1.y,
            tpcw->flt2.x, tpcw->flt2.y);
        tpcw->endpstr= tpcw->endpstr+ tpcw->len;
        cost=((double)(q->scl_i_cost*q-
>scl_qty)*discount)+ 0.5;
        p->sc_c_sub_total= p-
>sc_c_sub_total+(long)(cost);
        p->sc_c_ship_cost= p->sc_c_ship_cost+(q-
>scl_qty*100);
        q= q->next;
    }
    memcpy(tpcw->endpstr, buyreq3, buyreq3len);
    tpcw->endpstr= tpcw->endpstr+ buyreq3len;
    p->sc_c_tax=(long)(p->sc_c_sub_total*0.0825+
0.5);
    p->sc_c_ship_cost= p->sc_c_ship_cost+ 300;
    p->sc_c_total= p->sc_c_sub_total+ p->sc_c_tax+
    p->sc_c_ship_cost;
    floatprint(p->sc_c_sub_total,&tpcw->flt1);
    floatprint(p->sc_c_tax,&tpcw->flt2);

```

```

floatprint(p->sc_c_total,&tpcw->flt3);
floatprint(p->sc_c_ship_cost,&tpcw->flt4);
tpcw->len= sprintf(tpcw->endptr,
    "<BR>"
    "<TABLE BORDER=\"0\">"
    "<TR><TD><B>Subtotal with
discount (%d%%):</B></TD>"
    "<TD
ALIGN=\"RIGHT\"><B>$%d.%.2d</B></TD></TR>"
    "<TR><TD><B>Tax</B></TD>"
    "<TD
ALIGN=\"RIGHT\"><B>$%d.%.2d</B></TD></TR>"
    "<TR><TD><B>Shipping & Handling</B></TD>"
    "<TD
ALIGN=\"RIGHT\"><B>$%d.%.2d</B></TD></TR>"
    "<TR><TD><B>Total</B></TD>"
    "<TD
ALIGN=\"RIGHT\"><B>$%d.%.2d</B></TD></TR>"
    "</TABLE>",
    p->sc_c_discount,
    tpcw->flt1.x, tpcw->flt1.y,
    tpcw->flt2.x, tpcw->flt2.y,
    tpcw->flt4.x, tpcw->flt4.y,
    tpcw->flt3.x, tpcw->flt3.y
);
tpcw->endptr= tpcw->endptr+ tpcw->len;
memcpy(tpcw->endptr, buyreq4, buyreq4len);
tpcw->endptr= tpcw->endptr+ buyreq4len;
tpcw->len= sprintf(tpcw->endptr,
    "<A
HREF=\"http://%s/tpcw/tpcw.dll?b&ADD_FLAG=N\"><IMG
SRC=\"gifs/cart.gif\" ALT=\"Shopping Cart\" BORDER=\"0\" VSPACE=\"0\"
HSPACE=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>"
    "<A
HREF=\"http://%s/tpcw/tpcw.dll?a\"><IMG SRC=\"gifs/home.gif\"
ALT=\"Home\" BORDER=\"0\" VSPACE=\"0\" HSPACE=\"0\" WIDTH=\"120\"
HEIGHT=\"30\"></A>"
    "</P></BLOCKQUOTE></BLOCKQUOTE></BLOCKQUOTE></BODY></HTML>",
    domainname,
    domainname);
tpcw->endptr= tpcw->endptr+ tpcw->len;
p->valid = 1;
}
else
{
    sprintf(tpcw->err,"BuyRequest: Can't find
Session ID: %d Shopping ID: %d",
        tpcw->ID.SessionID, p->sc_shopping_id);
    ErrorMessage(tpcw);
    return FALSE;
}
}
}

```

```

__finally
{
    LeaveCriticalSection(&g_csBasket[tpcw->hash]);
}

return TRUE;
}

BOOL ExistingCustomer(TPCW *tpcw, struct BASKET_LIST *Basket_List[])
{
    COUNTRY_ENQ_DATA country;
    ADDRESS_ENQ_DATA addr;
    CUSTOMER_INFO_ENQ_DATA cust;
    int rc;
    DecodeParse(cust.c_uname, tpcw->querystring);
    DecodeParse(cust.c_passwd, tpcw->querystring);
    cust.c_id= 0;

    rc= SQLUnamePwdCheck(tpcw,&cust);

    if(rc== FALSE)
    {
        strcpy(tpcw->err, " ExistingCustomer:
SQLUnamePwdCheck()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    if(cust.c_id!= 0)
    {
        rc= SQLGetCustInfo(tpcw,&cust,&addr,&country);

        if(rc== FALSE)
        {
            strcpy(tpcw->err, " ExistingCustomer:
SQLGetCustInfo()");
            ErrorMessage(tpcw);
            return FALSE;
        }
        BuyRequest(tpcw, cust, addr, country, Basket_List);
    }
    else
    {
        strcpy(tpcw->err, " ExistingCustomer: Record does not
match!!!");
        ErrorMessage(tpcw);
    }
    return TRUE;
}

```

```

BOOL BuyConfirm(TPCW *tpcw, struct BASKET_LIST *Basket_List[])
{
    ORDER_ENQ_DATA order;
    ADDRESS_ENQ_DATA shipaddress;
    COUNTRY_ENQ_DATA shipcountry;
    int rc;
    char month[5];
    char day[5];
    char year[5];

    //
    DecodeParse(shipaddress.a_street1, tpcw->querystring);
    DecodeParse(shipaddress.a_street2, tpcw->querystring);
    DecodeParse(shipaddress.a_city, tpcw->querystring);
    DecodeParse(shipaddress.a_state, tpcw->querystring);
    DecodeParse(shipaddress.a_zip, tpcw->querystring);
    DecodeParse(shipcountry.co_name, tpcw->querystring);
    if(shipaddress.a_street1[0] != '\0')
    {
        shipcountry.co_id= 0;
        rc= SQLCountryCheck(tpcw,&shipcountry);
        if(rc== FALSE)
        {
            strcpy(tpcw->err, " BuyConfirm:
SQLCountryCheck()");
            ErrorMessage(tpcw);
            return FALSE;
        }
        if(shipcountry.co_id== 0)
        {
            strcpy(tpcw->err, " BuyConfirm: Invalid country name!");
            ErrorMessage(tpcw);
            return FALSE;
        }
    }
    Parse(order.o_cc_type, tpcw->querystring);
    DecodeParse(order.o_cc_name, tpcw->querystring);
    Parse(order.o_cc_num, tpcw->querystring);
    DecodeParse(tpcw->szBuffer, tpcw->querystring);
    month[0]= tpcw->szBuffer[0];
    month[1]= tpcw->szBuffer[1];
    month[2]= '\0';
    order.o_cc_expiration_date.month=(SQLUSMALLINT) atol(month);
    if(order.o_cc_expiration_date.month >12)
    {
        wsprintf(tpcw->err, " BuyConfirm: MONTH TOO LARGE: %d",
            order.o_cc_expiration_date.month);
        ErrorMessage(tpcw);
        return FALSE;
    }
    day[0]= tpcw->szBuffer[3];
    day[1]= tpcw->szBuffer[4];
    day[2]= '\0';

```

```

order.o_cc_expiration_date.day=(SQLUSMALLINT) atol(day);
if(order.o_cc_expiration_date.day >31)
{
    wsprintf(tpcw->err, " BuyConfirm: DAY TOO LARGE: %d",
        order.o_cc_expiration_date.day);
    ErrorMessage(tpcw);
    return FALSE;
}
year[0]= tpcw->szBuffer[6];
year[1]= tpcw->szBuffer[7];
year[2]= tpcw->szBuffer[8];
year[3]= tpcw->szBuffer[9];
year[4]= '\0';
order.o_cc_expiration_date.year=(SQLUSMALLINT) atol(year);
if(order.o_cc_expiration_date.year >3000)
{
    wsprintf(tpcw->err, " BuyConfirm: YEAR TOO LARGE: %d",
        order.o_cc_expiration_date.year);
    ErrorMessage(tpcw);
    return FALSE;
}
order.o_cc_expiration_date.hour= 0;
order.o_cc_expiration_date.minute= 0;
order.o_cc_expiration_date.second= 0;
order.o_cc_expiration_date.fraction= 0;
Parse(order.o_ship_type, tpcw->querystring);
NewOrderLine(tpcw, order, shipaddress, shipcountry,
Basket_List);
return TRUE;
}

BOOL NewCustomer(TPCW *tpcw, struct BASKET_LIST *Basket_List[])
{
    int rc;
    COUNTRY_ENQ_DATA country;
    ADDRESS_ENQ_DATA addr;
    CUSTOMER_INFO_ENQ_DATA cust;
    char year[5];
    char month[5];
    char day[5];
    struct BASKET_LIST *p= NULL;

    DecodeParse(cust.c_uname, tpcw->querystring);
    DecodeParse(cust.c_passwd, tpcw->querystring);
    Parse(tpcw->szBuffer, tpcw->querystring);
    URL_Decode(tpcw->szBuffer);
    month[0]= tpcw->szBuffer[0];
    month[1]= tpcw->szBuffer[1];

```

```

month[2]= '\0';
cust.c_birthdate.month=(SQLUSMALLINT) atol(month);
if(cust.c_birthdate.month >12)
{
    wsprintf(tpcw->szContent, " MONTH TOO LARGE: %d",
cust.c_birthdate.month);
    ErrorMessage(tpcw);
    return FALSE;
}
day[0]= tpcw->szBuffer[3];
day[1]= tpcw->szBuffer[4];
day[2]= '\0';
cust.c_birthdate.day=(SQLUSMALLINT) atol(day);
if(cust.c_birthdate.day >31)
{
    wsprintf(tpcw->szContent, " DAY TOO LARGE: %d",
cust.c_birthdate.day);
    ErrorMessage(tpcw);
    return FALSE;
}
year[0]= tpcw->szBuffer[6];
year[1]= tpcw->szBuffer[7];
year[2]= tpcw->szBuffer[8];
year[3]= tpcw->szBuffer[9];
year[4]= '\0';
cust.c_birthdate.year=(SQLUSMALLINT) atol(year);
if(cust.c_birthdate.year >3000)
{
    wsprintf(tpcw->szContent, " YEAR TOO LARGE: %d",
cust.c_birthdate.year);
    ErrorMessage(tpcw);
    return FALSE;
}
cust.c_birthdate.hour= 0;
cust.c_birthdate.minute= 0;
cust.c_birthdate.second= 0;
cust.c_birthdate.fraction= 0;
DecodeParse(cust.c_fname, tpcw->querystring);
DecodeParse(cust.c_lname, tpcw->querystring);
DecodeParse(addr.a_street1, tpcw->querystring);
DecodeParse(addr.a_street2, tpcw->querystring);
DecodeParse(addr.a_city, tpcw->querystring);
DecodeParse(addr.a_state, tpcw->querystring);
DecodeParse(addr.a_zip, tpcw->querystring);
DecodeParse(country.co_name, tpcw->querystring);
country.co_id= 0;
rc= SQLCountryCheck(tpcw,&country);
if(rc== FALSE)
{
    strcpy(tpcw->err, " NewCustomer: SQLCountryCheck()");
    ErrorMessage(tpcw);
    return FALSE;
}

```

```

if(country.co_id== 0)
{
    strcpy(tpcw->err, " NewCustomer: Invalid country name!");
    ErrorMessage(tpcw);
    return FALSE;
}
Parse(cust.c_phone, tpcw->querystring);
DecodeParse(cust.c_email, tpcw->querystring);
DecodeParse(cust.c_data, tpcw->querystring);
/*=====
=====
-- Code changed on 11/13/00 to meet clause 2.6.3.2 of 1.2 spec
-- This line of code added on 11/13/00 to meet clause 2.6.3.2
of 1.2
spec
-- It was not used during the measurement intervals of the 11/6
thru
11/10 benchmark but tests were run 11/08/00 to
demonstrate to the auditor that it had no significant impact on
performance.
-- See auditor's attestation letter for details.
*/
//Before code changed
//=====
//cust.c_discount= RandomNumber(0,50,&tpcw->lTLSeed);
//=====
//After code changed
//=====
=====
cust.c_discount= RandomNumber(0,50,&tpcw->lTLSeed);
//=====
=====
if(cust.c_discount<0 || cust.c_discount>50)
{
    wsprintf(tpcw->err,
        "NewCustomer: cust.c_discount<0 ||
cust.c_discount>50 (%d)",
        cust.c_discount);
    ErrorMessage(tpcw);
    return FALSE;
}
__try
{
    EnterCriticalSection(&g_csBasket[tpcw->hash]);
    rc= SearchForBasket(tpcw, Basket_List,&p);
    if(rc== FALSE)
    {

```

```

SearchForBasket());
    strcpy(tpcw->err, " NewCustomer:
    ErrorMessage(tpcw);
    return FALSE;
}
if(p== NULL)
{
    strcpy(tpcw->err, " NewCustomer: Cannot find
customer");
    ErrorMessage(tpcw);
    return FALSE;
}
p->sc_c_discount= cust.c_discount;
}
__finally
{
    LeaveCriticalSection(&g_csBasket[tpcw->hash]);
}
cust.c_id= 0;
__try
{
    EnterCriticalSection(&csC_ID);
    cust.c_id= C_ID;
    C_ID++;
    if(C_ID >LimitC_ID)
    {
        rc= ReadMaxC_ID(tpcw, C_ID_ALLOC);
        if(rc== 0)
        {
            strcpy(tpcw->err, " NewCustomer: rc=
ReadMaxC_ID(tpcw, C_ID_ALLOC)== 0");
            ErrorMessage(tpcw);
            return FALSE;
        }
        else
        {
            cust.c_id= C_ID;
            C_ID++;
        }
    }
}
__finally
{
    LeaveCriticalSection(&csC_ID);
}
DigSyl(cust.c_id, cust.c_uname);
strcpy(cust.c_passwd, cust.c_uname);
_strlwr(cust.c_passwd);
__try
{
    EnterCriticalSection(&csADDR_ID);

```

```

    addr.a_id= ADDR_ID;
    ADDR_ID++;
    if(ADDR_ID >LimitADDR_ID)
    {
        rc= ReadMaxADDR_ID(tpcw, ADDR_ID_ALLOC);
        if(rc== 0)
        {
            strcpy(tpcw->err, " NewOrderLine: rc=
ReadMaxADDR_ID(tpcw, O_ID_ALLOC)== 0");
            ErrorMessage(tpcw);
            return FALSE;
        }
        else
        {
            addr.a_id= ADDR_ID;
            ADDR_ID++;
        }
    }
}
__finally
{
    LeaveCriticalSection(&csADDR_ID);
}
rc= SQLAddNewCustomer(tpcw, &country, &addr, &cust);
if(rc== FALSE)
{
    strcpy(tpcw->err, "NewCustomer: SQLAddNewCustomer()");
    ErrorMessage(tpcw);
    return FALSE;
}
if(cust.c_id >9999999999L || cust.c_id <= 0)
{
    sprintf(tpcw->err, "NewCustomer: SQLAddNewCustomer:
%I64d", cust.c_id);
    ErrorMessage(tpcw);
    return FALSE;
}

BuyRequest(tpcw, cust, addr, country, Basket_List);
return TRUE;
}
BOOL UserType(TPCW *tpcw, struct BASKET_LIST *Basket_List[])
{
    BOOL ret;
    Parse(tpcw->szBuffer, tcw->querystring);
    if(strcmp(tpcw->szBuffer, "N")== 0)
    {
        ret= NewCustomer(tpcw, Basket_List);
    }
    else

```

```

    {
        ret= ExistingCustomer(tpcw, Basket_List);
    }
    return ret;
}

BOOL ShowBook(TPCW *tpcw)
{
    BOOK_DETAILS_ENQ_DATA book;
    int rc;
    char *chrptr;
    Parse(tpcw->szBuffer, tcpw->querystring);
    book.i_id= _atoi64(tpcw->szBuffer);
    rc= SQLBookDetailsProc(tpcw,&book);
    if(rc== FALSE)
    {
        strcpy(tpcw->err, " ShowBook: SQLBookDetailsProc()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    tcpw->endptr= tcpw->szContent;

    chrptr = book.i_image;
    chrptr+=strcsn(book.i_image, " \");
    *chrptr = '\0';
    chrptr = book.i_thumbnail;
    chrptr+=strcsn(book.i_thumbnail, " \");
    *chrptr = '\0';

/*
*/

    memcpy(tpcw->endptr, show1, show1len);
    tcpw->endptr= tcpw->endptr+ show1len;
    floatprint(book.i_srp,&tcpw->flt1);
    floatprint(book.i_cost,&tcpw->flt2);
    tcpw->len= sprintf(tcpw->endptr,
        "<H2>Title: %s</H2>"
        "<P ALIGN=\<LEFT\>Author %s %s<BR></P>"
        "<IMG SRC=\<http://image.tpcw.net/tpcw/%s\> BORDER=\<0\>"
ALIGN=\<RIGHT\> WIDTH=\<200\> HEIGHT=\<200\>"
        "<IMG SRC=\<http://image.tpcw.net/tpcw/%s\> BORDER=\<0\>"
ALIGN=\<RIGHT\> WIDTH=\<100\> HEIGHT=\<150\>"
        "<P><BR><BR></P>"
        "<FORM ACTION=\<http://%s/tpcw/tpcw.dll?\>"
METHOD=\<GET\>"
        "<INPUT TYPE=\<HIDDEN\> NAME=\<n\> VALUE=\<\>"
        "<INPUT TYPE=\<HIDDEN\> NAME=\<I_ID\> VALUE=\<%I64d\>"
        "<TABLE BORDER=\<0\>"
        "<TR><TD><B>Suggested
Retail:</B></TD><TD><B>$$$d.%d</B></TD></TR>"

```

```

        "<TR><TD><B>Our Current Price:</B></TD><TD><FONT
COLOR=\<#dd0000\><B>$$$d.%d</B></FONT></TD></TR>" ,
        book.i_title,
        book.a_fname,
        book.a_lname,
        book.i_image,
        book.i_thumbnail,
        domainname,
        book.i_id,
        tcpw->flt1.x, tcpw->flt1.y,
        tcpw->flt2.x, tcpw->flt2.y);
    tcpw->endptr= tcpw->endptr + tcpw->len;
    memcpy(tpcw->endptr, show2, show2len);
    tcpw->endptr= tcpw->endptr + show2len;
    memcpy(tpcw->endptr, show3, show3len);
    tcpw->endptr= tcpw->endptr + show3len;
    return TRUE;
}

BOOL ChangeInfo(TPCW *tpcw)
{
    BOOK_DETAILS_ENQ_DATA book;
    int rc;
    book.i_id= _atoi64(tpcw->szBuffer);
    Parse(tpcw->szBuffer, tcpw->querystring);
    book.i_cost=(long)(0.5+atof(tpcw->szBuffer)*100);
    DecodeParse(book.i_image, tcpw->querystring);
    DecodeParse(book.i_thumbnail, tcpw->querystring);

    rc= SQLItemChange(tpcw,&book);

    if(rc== FALSE)
    {
        strcpy(tpcw->err,"ChangeInfo: SQLItemChange()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    memcpy(tpcw->endptr, changeinfo1, changeinfo1len);
    tcpw->endptr= tcpw->endptr+ changeinfo1len;

    rc= SQLProductUpdated(tpcw,&book);

    if(rc== FALSE)
    {
        strcpy(tpcw->err,"ChangeInfo: SQLProductUpdated()");
        ErrorMessage(tpcw);
        return FALSE;
    }
    memcpy(tpcw->endptr, changeinfo2, changeinfo2len);
    tcpw->endptr= tcpw->endptr+ changeinfo2len;
    return TRUE;
}

```

```

#ifdef SCREEN_INFO
BOOL ShowStat(TPCW *tpcw)
{
    int i;
    long total= 0;
    long sumtotal = 0;
    tpcw->len= sprintf(tpcw->endptr,
        "<table border= 0><tr valign= top><td>"
        "<table border= 1>"
        "<tr><td>number of shopping basket<td>%d"
        "<tr><td>outstanding pge request<td>%d"
        "<tr><td>number thread local storage<td>%d"
        "<tr><td>number of basket killed<td>%d"
        "<tr><td>number of ssl retry to pge<td>%d"
        "<tr><td>number of switch basket<td>%d"
        "</table>",
        numshop,
        numpgereq,
        numthread,
        numbasketkill,
        numsslretry,
        numswitchbasket
    );
    tpcw->endptr= tpcw->endptr+ tpcw->len;
    for(i= 0; i<15; i++)
    {
        total= numwebaction[i]+ total;
        sumtotal = totalwebaction[i]+sumtotal;
    }
    tpcw->len= sprintf(tpcw->endptr, "<td width= 50><td><table
border= 1>"
    "<tr><td>home<td>%d<td>%d"
    "<tr><td>shopping basket<td>%d<td>%d"
    "<tr><td>adjustment<td>%d<td>%d"
    "<tr><td>cust registration<td>%d<td>%d"
    "<tr><td>buy request<td>%d<td>%d"
    "<tr><td>buy confirm<td>%d<td>%d"
    "<tr><td>order status<td>%d<td>%d"
    "<tr><td>order result<td>%d<td>%d"
    "<tr><td>search request<td>%d<td>%d"
    "<tr><td>search result<td>%d<td>%d"
    "<tr><td>best & new<td>%d<td>%d"
    "<tr><td>product detail<td>%d<td>%d"
    "<tr><td>admin request<td>%d<td>%d"
    "<tr><td>admin confirm<td>%d<td>%d"
    "<tr><td>total<td>%d<td>%d"
    "</table></table>",
    numwebaction[0], totalwebaction[0 ],
    numwebaction[1], totalwebaction[ 1],
    numwebaction[14], totalwebaction[14 ],
    numwebaction[2], totalwebaction[2 ],
    numwebaction[3], totalwebaction[3 ],

```

```

        numwebaction[4], totalwebaction[4 ],
        numwebaction[5], totalwebaction[5 ],
        numwebaction[6], totalwebaction[6 ],
        numwebaction[7], totalwebaction[7 ],
        numwebaction[8], totalwebaction[8 ],
        numwebaction[9], totalwebaction[ 9],
        numwebaction[11], totalwebaction[11 ],
        numwebaction[12], totalwebaction[12 ],
        numwebaction[13], totalwebaction[13 ],
        total, sumtotal
    );
    tpcw->endptr= tpcw->endptr+ tpcw->len;
    return TRUE;
}
#endif

```

#### display.h

```

//display.h
//
//Copyright Unisys, 1998,1999,2000
#include "debug.h"

static char *
day[]={ "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };

static char *
mon[]={ "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };

BOOL floatprint(long num, struct FLOATPRINT *fltpr);

BOOL Home(TPCW *tpcw, struct BASKET_LIST *Basket_List[]);
BOOL HomeG(TPCW *tpcw, struct BASKET_LIST *Basket_List[]);
BOOL BookInfo(TPCW *tpcw);
BOOL PromoForListing(TPCW *tpcw);
BOOL FrameListing(TPCW *tpcw);
BOOL NewProductList(TPCW *tpcw);
BOOL BestSellerList(TPCW *tpcw);
BOOL Adjustment(TPCW *tpcw, struct BASKET_LIST *Basket_List[]);
BOOL CustomerReg(TPCW *tpcw);
BOOL ShoppingBasket(TPCW *tpcw, struct BASKET_LIST *Basket_List[]);
BOOL UserType(TPCW *tpcw, struct BASKET_LIST *Basket_List[]);
BOOL BuyConfirm(TPCW *tpcw, struct BASKET_LIST *Basket_List[]);
BOOL OrderStatus(TPCW *tpcw);
BOOL OrderStatusResult(TPCW *tpcw);
BOOL Search(TPCW *tpcw);
BOOL SearchResultFrameset(TPCW *tpcw);
BOOL PromoForSearchResult(TPCW *tpcw);

void Parse(char param[], char *querystring);
void Skip(char *querystring);

```

```

void DecodeParse(char param[], char *querystring);
BOOL ShowBook(TPCW *tpcw);
BOOL ChangeInfo(TPCW *tpcw);
long AddToSession(TPCW *tpcw, struct BASKET_ITEM *shopitem, struct
    BASKET_LIST *Basket_List[], struct BASKET_LIST **ShopBasket);
long AddToSessionForRecovery(TPCW *tpcw, struct BASKET_ITEM *shopitem,
    struct
        BASKET_LIST *Basket_List[], struct BASKET_LIST **ShopBasket);
long AddToSessionForBasketSwitch(TPCW *tpcw, struct BASKET_ITEM *
    shopitem, struct BASKET_LIST *Basket_List[], struct BASKET_LIST
**
    ShopBasket);
BOOL SearchForBasket(TPCW *tpcw, struct BASKET_LIST *
    Basket_List[], struct BASKET_LIST **ShopBasket);
void CookieDate(char cookiedate[]);

#ifdef SCREEN_INFO
BOOL ShowStat(TPCW *tpcw);
#endif

```

#### encode.c

```

// encode.c
//
// Copyright Unisys, 1999,2000
#include <windows.h>
#include "encode.h"
const char sp_char[27]={'!', '@', '#', '$', '%', '^', '&', '*', '(',
    ')', '_', '-', '=', '+', '{', '}', '[', ']', '|', ':', ';', ',', '.',
    '?', '/', '~', ' '};
const char sp_hex[27][3]=
{
    {'%', '2', '1'}, //!
    {'%', '4', '0'}, //@
    {'%', '2', '3'}, //#
    {'%', '2', '4'}, //$
    {'%', '2', '5'}, //%
    {'%', '5', 'E'}, //^
    {'%', '2', '6'}, //&
    {'%', '2', 'A'}, //*
    {'%', '2', '8'}, //(
    {'%', '2', '9'}, //(
    {'%', '5', 'F'}, //_
    {'%', '2', 'D'}, // -
    {'%', '3', 'D'}, // =
    {'%', '2', 'B'}, // +
    {'%', '7', 'B'}, // {
    {'%', '7', 'D'}, // }
    {'%', '5', 'B'}, // [
    {'%', '5', 'D'}, // ]
    {'%', '7', 'C'}, // |
    {'%', '3', 'A'}, // :
    {'%', '3', 'B'}, // ;

```

```

{'%', '2', 'C'}, //,
{'%', '2', 'E'}, //.
{'%', '3', 'F'}, //?
{'%', '2', 'F'}, ///
{'%', '7', 'E'}, //~
{'%', '2', '0'}
};

int URL_Decode( char *target)
{
    char temp[ 1500];
    char c1, c2;
    int k;
    int j;
    strcpy( temp, target);
    for( j= 0, k= 0; temp[ j]!= '\0'; j++, k++)
    {
        switch( temp[ j])
        {
            case '+':
                target[k]= ' ';
                break;
            case '%':
                c1= tolower( temp[++ j]);
                c2= tolower( temp[++ j]);
                if( isdigit( c1))
                {
                    c1-= '0';
                }
                else
                {
                    c1= c1- 'a'+ 10;
                }
                if( isdigit( c2))
                {
                    c2-= '0';
                }
                else
                {
                    c2= c2- 'a'+ 10;
                }
                target[ k]= c1* 16+ c2;
                break;
            default:
                target[ k]= temp[ j];
                break;
        }
    }
    target[ k]= '\0';
    return (k);
}

```

#### encode.h

```
// encode.h
//
// Copyright Unisys, 1999,2000

int URL_Decode(char * target);
```

#### htmltext.c

```
// htmltext.c
#include <stdio.h>
#include "htmltext.h"
#include "tpcw.h"
```

```
extern char domainname[ 20];
char advert1[ 500];
char advert2[ 100];
long advert1len;
long advert2len;
char home1[ 500];
char home2[ 500];
char home3[ 500];
char hometable[ 51000];
long home1len;
long home2len;
long home3len;
long hometablelen;
```

```
char newprod1[ 1000];
char newprod2[ 1000];
long newprod1len;
long newprod2len;
char bestsell1[ 1000];
char bestsel2[ 1000];
long bestsell1len;
long bestsel2len;
char bookinfo1[ 1000];
char bookinfo2[ 100];
long bookinfo1len;
long bookinfo2len;
char search1[ 1000];
char search2[ 2000];
long search1len;
long search2len;
char searchres1[600];
long searchres1len;
char shop1[ 1000];
char shop2[ 1000];
char shop3[ 2000];
long shop1len;
long shop2len;
long shop3len;
char orderstat1[ 1000];
```

```
char orderstat2[ 1000];
char orderstat3[ 1000];
long orderstat1len;
long orderstat2len;
long orderstat3len;
char custreg1[ 1000];
char custreg2[ 3000];
long custreg1len;
long custreg2len;
char neworder1[ 1000];
char neworder2[ 1000];
char neworder3[ 1000];
char neworder4[ 1000];
long neworder1len;
long neworder2len;
long neworder3len;
long neworder4len;
char buyreq1[ 1000];
char buyreq2[ 1000];
char buyreq3[ 1000];
char buyreq4[ 2500];
char buyreq5[ 1000];
long buyreq1len;
long buyreq2len;
long buyreq3len;
long buyreq4len;
long buyreq5len;
char show1[ 1000];
char show2[ 1000];
char show3[ 1000];
long show1len;
long show2len;
long show3len;
char changeinfo1[ 1000];
char changeinfo2[ 100];
long changeinfo1len;
long changeinfo2len;
char produpdate1[ 1000];
long produpdate1len;
char cachehost[ 20];
```

```
/*
advertisement
*/
void AdLen()
{
    advert1len= sprintf( advert1,"<TABLE BORDER=\"0\" WIDTH=\"660\"
ALIGN=\"CENTER\">"
    "<TR ALIGN=\"CENTER\" VALIGN=\"TOP\">"
    "<TD COLSPAN=\"5\"><B><FONT SIZE=\"+ 1\" color=\"#ff0000\">"
Click on one of our latest books to find out
more!</FONT></B></TD></TR>"
    "<TR ALIGN=\"CENTER\" VALIGN=\"TOP\">");
```

```

    advert2len= sprintf( advert2,"</TABLE>");
}
/*
home
*/
void HomeLen()
{
    homellen= sprintf( home1,"<H3 ALIGN=\"CENTER\"> Welcome <FONT
COLOR=#770000></H3>");
    home2len= sprintf( home2,"<CENTER><TABLE BORDER=\"0\"
WIDTH=700\" ALIGN=\"CENTER\" CELLPADDING=6\" CELLSPACING=0\"
BGCOLOR=#C0C0C0\">
    <TR ALIGN=\"CENTER\" VALIGN=\"TOP\"
BGCOLOR=#FFFFFF\">
    <TD COLSPAN=2\" VALIGN=\"MIDDLE\" WIDTH=300\"><IMG
SRC=\"gifs/new.gif\" WIDTH=300\" ALT=\"New Product\"></TD>
    <TD BGCOLOR=#FFFFFF\" WIDTH=100\"></TD>
    <TD COLSPAN=2\" WIDTH=300\"><IMG
SRC=\"gifs/best.gif\" WIDTH=300\" ALT=\"Best Seller\"></TD>
    </TR>");
    home3len= sprintf( home3,"<HTML><HEAD><TITLE>TPC-W Home
Page</TITLE><BODY BGCOLOR=#FFFFFF\"><H1 ALIGN=\"CENTER\"> TPC Web
Commerce Benchmark (TPC-W)</H1>
    <P ALIGN=\"CENTER\"><IMG SRC=\"gifs/tpclogo.gif\"
WIDTH=288\" HEIGHT=67\" ALIGN=\"BOTTOM\"
BORDER=0\" naturalsizeflag=3\"></CENTER>
    <H2 ALIGN=\"CENTER\"> Home Page</H2>");
    hometablelen= sprintf( hometable,
    "<TR><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_arts.htm\">ARTS</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_parenting.htm\">PARENTING</A></TD><TD
    WIDTH=50\" BGCOLOR=#FFFFFF\"></TD><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_arts.htm\">ARTS</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_parenting.htm\">PARENTING</A></TD></TR>
    <TR><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_biographies.htm\">BIOGRAPHIES</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_politics.htm\">POLITICS</A></TD><TD
    WIDTH=50\" BGCOLOR=#FFFFFF\"></TD><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_biographies.htm\">BIOGRAPHIES</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_politics.htm\">POLITICS</A></TD></TR>

```

```

    <TR><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_business.htm\">BUSINESS</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_reference.htm\">REFERENCE</A></TD><TD
    WIDTH=50\" BGCOLOR=#FFFFFF\"></TD><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_business.htm\">BUSINESS</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_reference.htm\">REFERENCE</A></TD></TR>
    <TR><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_children.htm\">CHILDREN</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_religion.htm\">RELIGION</A></TD><TD
    WIDTH=50\" BGCOLOR=#FFFFFF\"></TD><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_children.htm\">CHILDREN</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_religion.htm\">RELIGION</A></TD></TR>
    <TR><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_computers.htm\">COMPUTERS</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_romance.htm\">ROMANCE</A></TD><TD
    WIDTH=50\" BGCOLOR=#FFFFFF\"></TD><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_computers.htm\">COMPUTERS</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_romance.htm\">ROMANCE</A></TD></TR>
    <TR><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_cooking.htm\">COOKING</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_self-help.htm\">SELF-
    HELP</A></TD><TD
    WIDTH=50\" BGCOLOR=#FFFFFF\"></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_cooking.htm\">COOKING</A></TD><TD><P
    ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/best_self-help.htm\">SELF-
    HELP</A></TD></TR>
    <TR><TD><P ALIGN=\"CENTER\">
    <A
    HREF=\"http://%s/tpcw/new_health.htm\">HEALTH</A></TD><TD><P
    ALIGN=\"CENTER\">

```

```

        <A HREF=\"http://%s/tpcw/new_science-
nature.htm\">SCIENCE-NATURE</A></TD><TD
WIDTH=\"50\"BGCOLOR=\"#FFFFFF\"></TD><TD><P ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/best_health.htm\">HEALTH</A></TD><TD><P
ALIGN=\"CENTER\">\"
        <A HREF=\"http://%s/tpcw/best_science-
nature.htm\">SCIENCE-NATURE</A></TD></TR>\"
        <TR><TD><P ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/new_history.htm\">HISTORY</A></TD><TD><P
ALIGN=\"CENTER\">\"
        <A HREF=\"http://%s/tpcw/new_science-
fiction.htm\">SCIENCE-FICTION</A></TD><TD
WIDTH=\"50\"BGCOLOR=\"#FFFFFF\"></TD><TD><P ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/best_history.htm\">HISTORY</A></TD><TD><P
ALIGN=\"CENTER\">\"
        <A HREF=\"http://%s/tpcw/best_science-
fiction.htm\">SCIENCE-FICTION</A></TD></TR>\"
        <TR><TD><P ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/new_home.htm\">HOME</A></TD><TD><P
ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/new_sports.htm\">SPORTS</A></TD><TD
WIDTH=\"50\"BGCOLOR=\"#FFFFFF\"></TD><TD><P ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/best_home.htm\">HOME</A></TD><TD><P
ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/best_sports.htm\">SPORTS</A></TD></TR>\"
        <TR><TD><P ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/new_humor.htm\">HUMOR</A></TD><TD><P
ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/new_mystery.htm\">MYSTERY</A></TD><TD
WIDTH=\"50\"BGCOLOR=\"#FFFFFF\"></TD><TD><P ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/best_humor.htm\">HUMOR</A></TD><TD><P
ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/best_mystery.htm\">MYSTERY</A></TD></TR>\"
        <TR><TD><P ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/new_literature.htm\">LITERATURE</A></TD><TD><P
ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/new_travel.htm\">TRAVEL</A></TD><TD
WIDTH=\"50\"BGCOLOR=\"#FFFFFF\"></TD><TD><P ALIGN=\"CENTER\">\"

```

```

        <A
HREF=\"http://%s/tpcw/best_literature.htm\">LITERATURE</A></TD><TD><P
ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/best_travel.htm\">TRAVEL</A></TD></TR>\"
        <TR><TD><P ALIGN=\"CENTER\">\"
        <A HREF=\"http://%s/tpcw/new_non-fiction.htm\">NON-
FICTION</A></TD><TD><P ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/new_youth.htm\">YOUTH</A></TD><TD
WIDTH=\"50\"BGCOLOR=\"#FFFFFF\"></TD><TD><P ALIGN=\"CENTER\">\"
        <A HREF=\"http://%s/tpcw/best_non-fiction.htm\">NON-
FICTION</A></TD><TD><P ALIGN=\"CENTER\">\"
        <A
HREF=\"http://%s/tpcw/best_youth.htm\">YOUTH</A></TD></TR>\",
        domainname, domainname, domainname, domainname,
        domainname,
        domainname, domainname, domainname, domainname,
        domainname,
        domainname, domainname, domainname, domainname,
        domainname, domainname, domainname, domainname, domainname,
        domainname, domainname, domainname, domainname, domainname,
        domainname, domainname, domainname, domainname, domainname,
        domainname, domainname, domainname, domainname, domainname,
        domainname, domainname, domainname, domainname, domainname,
        domainname);
    }
}
/*
new product
*/
void NewProdLen()
{
    newprodllen= sprintf( newprod1,\"<TABLE BORDER=\"1\"
CELLPADDING=\"1\" CELLSPACING=\"1\">\"
        <TR><TD><FONT SIZE=\"+ 1\"> Author</FONT></TD>\"
        <TD><FONT SIZE=\"+ 1\"> Title</FONT></TD></TR>\";
    newprod2len= sprintf( newprod2,\"</TABLE><P ALIGN=\"CENTER\">\"
        <A HREF=\"http://%s/tpcw/tpcw.dll?b&ADD_FLAG=N
\"target=\"_top\"><IMG SRC=\"gifs/cart.gif\" ALT=\"Shopping Cart\"
BORDER=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>\"
        <A HREF=\"http://%s/tpcw/tpcw.dll?h \"target=\"_top\"><IMG
SRC=\"gifs/search.gif\" ALT=\"Search\" BORDER=\"0\" WIDTH=\"120\"
HEIGHT=\"30\"></A>\"
        <A HREF=\"http://%s/tpcw/tpcw.dll?a \"target=\"_top\"><IMG
SRC=\"gifs/home.gif\" ALT=\"Home\" BORDER=\"0\" WIDTH=\"120\"
HEIGHT=\"30\"></A>\"
        </BODY></HTML>\", domainname, domainname, domainname);
}
}
/*
best seller

```

```

*/
void BestSelLen()
{
    bestselllen= sprintf( bestsell, "<TABLE BORDER=\\"1\\"
CELLPADDING=\\"1\\"CELLSPACING=\\"1\\">"
    "<TR><TD><FONT SIZE=\\"+ 1\\"> Author</FONT></TD>"
    "<TD><FONT SIZE=\\"+ 1\\"> Title</FONT></TD></TR>");
    bestsel2len= sprintf( bestsel2, "</TABLE><P ALIGN=\\"CENTER\\">"
    "<A HREF=\\"http://%s/tpcw/tpcw.dll?b&ADD_FLAG=N"
    "\\target=\\"_top\\"><IMG SRC=\\"gifs/cart.gif\\" ALT=\\"Shopping Cart\\"
    BORDER=\\"0\\" WIDTH=\\"120\\" HEIGHT=\\"30\\"></A>"
    "<A HREF=\\"http://%s/tpcw/tpcw.dll?h\\"target=\\"_top\\"><IMG
    SRC=\\"gifs/search.gif\\" ALT=\\"Search\\" BORDER=\\"0\\" WIDTH=\\"120\\"
    HEIGHT=\\"30\\"></A>"
    "<A HREF=\\"http://%s/tpcw/tpcw.dll?a\\"target=\\"_top\\"><IMG
    SRC=\\"gifs/home.gif\\" ALT=\\"Home\\" BORDER=\\"0\\" WIDTH=\\"120\\"
    HEIGHT=\\"30\\"></A>"
    "</BODY></HTML>",
    domainname, domainname, domainname);
}

/*
best seller
*/
void SearchResLen()
{
    searchresllen= sprintf( searchres1, "</TABLE><P
ALIGN=\\"CENTER\\">"
    "<A HREF=\\"tpcw.dll?b&ADD_FLAG=N\\"target=\\"_top\\"><IMG
    SRC=\\"gifs/cart.gif\\" ALT=\\"Shopping Cart\\" BORDER=\\"0\\" WIDTH=\\"120\\"
    HEIGHT=\\"30\\"></A>"
    "<A HREF=\\"tpcw.dll?h\\"target=\\"_top\\"><IMG
    SRC=\\"gifs/search.gif\\" ALT=\\"Search\\" BORDER=\\"0\\" WIDTH=\\"120\\"
    HEIGHT=\\"30\\"></A>"
    "<A HREF=\\"tpcw.dll?a\\"target=\\"_top\\"><IMG
    SRC=\\"gifs/home.gif\\" ALT=\\"Home\\" BORDER=\\"0\\" WIDTH=\\"120\\"
    HEIGHT=\\"30\\"></A>"
    "</BODY></HTML>");
}

/*
bookinfo
*/
void BookInfoLen()
{
    bookinfoflen= sprintf( bookinfo1, "<HTML><HEAD><TITLE>TPC-W
Product Detail Page</TITLE></HEAD><BODY BGCOLOR=\\"#FFFFFF\\">"
    "<H1 ALIGN=\\"CENTER\\"> TPC Web Commerce Benchmark (TPC-W)</H1>"
    "<CENTER><IMG
    SRC=\\"http://tpcwww.tpcw.net/tpcw/gifs/tpclogo.gif\\" ALIGN=\\"BOTTOM\\"
    BORDER=\\"0\\" WIDTH=\\"288\\" HEIGHT=\\"67\\"></CENTER>"

```

```

"<H2 ALIGN=\\"CENTER\\"> Product Detail Page</H2>");
bookinfo2len= sprintf( bookinfo2, "</BODY></HTML>");
}

/*
search
*/
void SearchLen()
{
    searchllen= sprintf( search1, "<HTML><HEAD><TITLE>Search Request
Page</TITLE></HEAD>"
    "<BODY><CENTER><H1 ALIGN=\\"CENTER\\"> TPC W Commerce Benchmark
(TPC-W)</H1>"
    "<H2 ALIGN=\\"CENTER\\"><IMG SRC=\\"gifs/tpclogo.gif\\"
ALIGN=\\"BOTTOM\\" BORDER=\\"0\\" WIDTH=\\"288\\" HEIGHT=\\"67\\"></H2>"
    "<H2 ALIGN=\\"CENTER\\"> Search Request Page</H2>");
    search2len= sprintf( search2, "<FORM
ACTION=\\"/tpcw/tpcw.dll?i\\"METHOD=\\"GET\\">"
    "<TABLE ALIGN=\\"CENTER\\">"
    "<TR>"
    "<TD ALIGN=\\"RIGHT\\"><H3> Search by:</h3></TD>"
    "<TD WIDTH=\\"100\\"></TD></TR>"
    "<TR><TD ALIGN=\\"RIGHT\\">"
    "<select NAME=\\"i\\" SIZE=\\"1\\">"
    "<option VALUE=\\"author\\" selected=\\"selected\\">
Author</option>"
    "<option VALUE=\\"title\\"> Title</option>"
    "<option VALUE=\\"subject\\"> Subject</option></select></TD>"
    "<TD><INPUT NAME=\\"SEARCHSTRING\\" SIZE=\\"30\\"></TD></TR>"
    "</TABLE>"
    "<P ALIGN=\\"CENTER\\"><INPUT TYPE=\\"IMAGE\\" NAME=\\"Submit\\"
SRC=\\"gifs/submit.gif\\" WIDTH=\\"120\\" HEIGHT=\\"30\\">"
    "<A HREF=\\"http://%s/tpcw/tpcw.dll?a\\"><IMG
SRC=\\"gifs/home.gif\\" ALT=\\"Home\\" BORDER=\\"0\\" WIDTH=\\"120\\"
HEIGHT=\\"30\\"></A>"
    "<A HREF=\\"http://%s/tpcw/tpcw.dll?b&ADD_FLAG=N\\"><IMG
SRC=\\"gifs/cart.gif\\" ALT=\\"Shopping Cart\\" BORDER=\\"0\\" WIDTH=\\"120\\"
HEIGHT=\\"30\\"></A>"
    "</P></FORM></BODY></HTML>",
    domainname, domainname);
}

/*
shopping basket
*/
void ShopLen()
{
    shopllen= sprintf( shop1, "<HTML><HEAD><TITLE>TPC W Shopping
Cart</TITLE></HEAD>"
    "<BODY BGCOLOR=\\"#FFFFFF\\">"
    "<H1 ALIGN=\\"CENTER\\">TPC Web Commerce Benchmark (TPC-W)</H1>"
    "<CENTER><IMG SRC=\\"gifs/tpclogo.gif\\" ALIGN=\\"BOTTOM\\"
BORDER=\\"0\\" WIDTH=\\"288\\" HEIGHT=\\"67\\"></CENTER>"
    "<H2 ALIGN=\\"CENTER\\"> Shopping Cart Page</H2>");
}

```

```

shop2len= sprintf( shop2,"<FORM
ACTION="http://%s/tpcw/tpcw.dll?"METHOD="GET">"
"<INPUT TYPE="HIDDEN" NAME="o" VALUE="">"
"<CENTER><P><TABLE BORDER="0">"
"<TR><TD><B> Qty</B></TD><TD><B> Product</B></TD></TR>",
domainname);
shop3len= sprintf( shop3,"<P><BR>" \

"<A HREF="http://%s/tpcw/custreg.htm"><IMG
SRC="gifs/checkout.gif" ALT="Check Out" BORDER="0" WIDTH="120"
HEIGHT="30"></A>"

"<A HREF="http://%s/tpcw/tpcw.dll?a"><IMG
SRC="gifs/home.gif" ALT="Home" BORDER="0" WIDTH="120"
HEIGHT="30"></A>"
"<P> If you have changed the quantities and/or taken anything
out<BR> of your "
"shopping cart, click here to refresh your shopping cart:</P>"
"<P><INPUT TYPE="IMAGE" NAME="Refresh"
SRC="gifs/refresh.gif" WIDTH="120" HEIGHT="30">"
"</CENTER></FORM></BODY></HTML>",
domainname, domainname);
}

/*
order status
*/
void OrderStatusLen()
{
    orderstat1len= sprintf( orderstat1,"<HTML><HEAD><TITLE>TPC-W
Order Display Page</TITLE></HEAD><BODY BGCOLOR="#FFFFFF">"
"<H1 ALIGN="CENTER"> TPC Web Commerce Benchmark (TPC-W)</H1>"
"<H2 ALIGN="CENTER"> Order Display Page</H2>"
"<BLOCKQUOTE><BLOCKQUOTE><BLOCKQUOTE>"
"<HR><P>");
    orderstat2len= sprintf( orderstat2,"<TABLE BORDER="0"
WIDTH="80%">"
"<TR><TD><B> Bill To:</B></TD><TD><B> Ship To:</B></TD></TR>");
    orderstat3len= sprintf(
orderstat3,"</BLOCKQUOTE></BLOCKQUOTE></BLOCKQUOTE></BLOCKQUOTE>"
"<CENTER>"
"<TABLE BORDER="1" CELLPADDING="5" CELLSPACING="0">"
"<TR><TD><H4> Item #</TD><TD><H4> Title</TD><TD><H4>
Cost</TD><TD><H4> Qty</TD><TD><H4> Discount</TD><TD><H4>
Comment</TD></TR>");
}

/*
customer reg
*/
void CustReg()
{

```

```

    custreg1len= sprintf( custreg1,"<HTML><HEAD><TITLE>Customer
Registration</TITLE></HEAD><BODY BGCOLOR="#FFFFFF">"
"<H1 ALIGN="CENTER"> TPC Web Commerce Benchmark (TPC-W)</H1>"
"<H1 ALIGN="CENTER"><IMG SRC="gifs/tpclogo.gif"
ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67"></H1>"
"<H2 ALIGN="CENTER"> Customer Registration Page</H2>");
    custreg2len= sprintf( custreg2,"<BLOCKQUOTE><BLOCKQUOTE><HR>"
"<TABLE BORDER="0">"
"<TR><TD><INPUT TYPE="RADIO" NAME="RETURNFLAG" VALUE="Y">I
am an existing customer</TD></TR>"
"<TR><TD><INPUT TYPE="RADIO"
NAME="RETURNFLAG" VALUE="N" CHECKED="CHECKED">I am a first time
customer</TD></TR>"
"</TABLE>"
"<HR><P><B>If you're an existing customer, enter your Username
and Password:</B>"
"<BR><BR></P>"
"<TABLE>"
"<TR ALIGN="LEFT"><TD>User ID: <INPUT
NAME="USERNAME" SIZE="23"></TD></TR>"
"<TR ALIGN="LEFT"><TD> Password: <INPUT TYPE="PASSWORD"
NAME="password" SIZE="14"></TD></TR>"
"</TABLE>"
"<HR><P><B>If you're a first time customer, enter the detail
below:</B>"
"<BR><BR></P>"
"<TABLE>"
"<TR><TD> Enter your birth date (mm/dd/yyyy):</TD><TD><INPUT
NAME="BIRTHDATE" SIZE="10" VALUE="01/01/1998"></TD></TR>"
"<TR><TD> Enter your First Name:</TD><TD><INPUT
NAME="FIRSTNAME" SIZE="15" VALUE="firstname"></TD></TR>"
"<TR><TD> Enter your Last Name:</TD><TD><INPUT
NAME="LASTNAME" SIZE="15" VALUE="lastname"></TD></TR>"
"<TR><TD> Enter your Address 1:</TD><TD><INPUT
NAME="ADDRESS1" SIZE="40" VALUE="street1"></TD></TR>"
"<TR><TD> Enter your Address 2:</TD><TD><INPUT
NAME="ADDRESS2" SIZE="40" VALUE="street2"></TD></TR>"
"<TR><TD> Enter your City, State, Zip:</TD><TD><INPUT
NAME="CITY" SIZE="30" VALUE="city"> <INPUT
NAME="STATE" SIZE="20" VALUE="state"> <INPUT
NAME="ZIP" SIZE="10" VALUE="zip"></TD></TR>"
"<TR><TD> Enter your Country:</TD><TD><INPUT
NAME="COUNTRY" SIZE="50" VALUE="United States"></TD></TR>"
"<TR><TD> Enter your Phone:</TD><TD><INPUT
NAME="PHONE" SIZE="16" VALUE="123-456-7890"></TD></TR>"
"<TR><TD> Enter your E- mail:</TD><TD><INPUT
NAME="EMAIL" SIZE="50" VALUE="email"></TD></TR>"
"</TABLE>"
"<HR>"
"<TABLE>"
"<TR><TD COLSPAN="2"> Other Customer Data: <TEXTAREA
NAME="DATA" ROWS="4" COLS="65"> information</TEXTAREA></TD></TR>"
"</TABLE>"

```

```

" <HR></BLOCKQUOTE></BLOCKQUOTE>"
" <CENTER>"
" <INPUT TYPE=\\"IMAGE\\" SRC=\\"gifs/submit.gif\\"NAME=\\"Submit\\"
BORDER=\\"0\\"VSPACE=\\"0\\" WIDTH=\\"120\\" HEIGHT=\\"30\\">");
}

/*
new order line
*/
void NewOrderLen()
{
    neworder1len= sprintf( neworder1, "<HTML><HEAD><TITLE> Order
Confirmation</TITLE></HEAD>"
" <BODY BGCOLOR=\\"#FFFFFF\\">"
" <H1 ALIGN=\\"CENTER\\">TPC Web Commerce Benchmark (TPC-W)</H1>"
" <H2 ALIGN=\\"CENTER\\">Buy Confirm Page</H2>"
" <BLOCKQUOTE><BLOCKQUOTE><BLOCKQUOTE>"
" <H2 ALIGN=\\"LEFT\\"> Order Information:</H2>");
    neworder2len= sprintf( neworder2, "<TABLE BORDER=\\"1\\"
CELLSPACING=\\"0\\" CELLPADDING=\\"0\\"><TR><TD><B> Qty</B></TD><TD><B>
Product</B></TD></TR>");
    neworder3len= sprintf( neworder3, "</TABLE>");
    neworder4len= sprintf( neworder4, "<H2 ALIGN=\\"LEFT\\"> Your
Order has been processed.</H2>"
" <TABLE BORDER=\\"1\\" CELLPADDING=\\"5\\"CELLSPACING=\\"0\\">");
}

/*
buy request
*/
void BuyReqLen()
{
    buyreq1len= sprintf( buyreq1, "<HR><H2>Order
Information:</H2>");
    buyreq2len= sprintf( buyreq2, "<TABLE BORDER=\\"1\\"
CELLSPACING=\\"0\\" CELLPADDING=\\"0\\"><TR><TD><B> Qty</B></TD><TD><B>
Product</B></TD></TR>");
    buyreq3len= sprintf( buyreq3, "</TABLE>");
    buyreq4len= sprintf( buyreq4, "<HR><TABLE BORDER=\\"1\\"
CELLPADDING=\\"5\\"CELLSPACING=\\"0\\"><TR><TD> Credit Card Type</TD>"
" <TD><INPUT TYPE=\\"RADIO\\" NAME=\\"CCTYPE\\" VALUE=\\"Visa\\"
checked> VISA"
" <INPUT TYPE=\\"RADIO\\"NAME=\\"CCTYPE\\" VALUE=\\"MASTERCARD\\">
MASTERCARD"
" <INPUT TYPE=\\"RADIO\\"NAME=\\"CCTYPE\\" VALUE=\\"DISCOVER\\">
DISCOVER"
" <INPUT TYPE=\\"RADIO\\"NAME=\\"CCTYPE\\" VALUE=\\"AMEX\\"> AMERICAN
EXPRESS"
" <INPUT TYPE=\\"RADIO\\"NAME=\\"CCTYPE\\" VALUE=\\"DINERS\\"> DINERS"
" </TD></TR><TR><TD> Name on Credit Card</TD><TD>"
" <INPUT NAME=\\"CCNAME\\"SIZE=\\"30\\" VALUE=\\"John
Doe\\"></TD></TR>"
" <TR><TD> Credit Credit Number</TD><TD>"

```

```

" <INPUT NAME=\\"CCNUM\\"SIZE=\\"16\\"
VALUE=\\"1234567890123456\\"></TD></TR>"
" <TR><TD> Credit Credit Expiration Date</TD><TD>"
" <INPUT NAME=\\"CCDATE\\"SIZE=\\"15\\"
VALUE=\\"02/02/1999\\"></TD></TR>"
" <TR><TD> Shipping Method</TD><TD>"
" <INPUT TYPE=\\"RADIO\\" NAME=\\"SHIPMETHOD\\" VALUE=\\"AIR\\"
checked>AIR"
" <INPUT TYPE=\\"RADIO\\" NAME=\\"SHIPMETHOD\\" VALUE=\\"UPS\\">UPS"
" <INPUT TYPE=\\"RADIO\\" NAME=\\"SHIPMETHOD\\"
VALUE=\\"FEDEX\\">FEDEX"
" <INPUT TYPE=\\"RADIO\\" NAME=\\"SHIPMETHOD\\" VALUE=\\"SHIP\\">SHIP"
" <INPUT TYPE=\\"RADIO\\" NAME=\\"SHIPMETHOD\\"
VALUE=\\"COURIER\\">COURIER"
" <INPUT TYPE=\\"RADIO\\" NAME=\\"SHIPMETHOD\\" VALUE=\\"MAIL\\">MAIL"
" </TD></TR></TABLE><P ALIGN=\\"CENTER\\">"
" <INPUT TYPE=\\"IMAGE\\"NAME=\\"Submit\\"
SRC=\\"gifs/processor.gif\\" WIDTH=\\"120\\" HEIGHT=\\"30\\">");
    buyreq5len= sprintf( buyreq5, "<HTML><HEAD><TITLE> TPC-W Buy
Request</TITLE></HEAD><BODY BGCOLOR=\\"#FFFFFF\\">"
" <H1 ALIGN=\\"CENTER\\"> TPC Web Commerce Benchmark (TPC-W)</H1>"
" <H2 ALIGN=\\"CENTER\\"> Buy Request Page</H2>"
" <HR><FORM ACTION=\\"https://%s/tpcw/tpcw.dll?\"
METHOD=\\"GET\\">"
" <INPUT TYPE=\\"HIDDEN\\" NAME=\\"e\\" VALUE=\\"\">",
    domainname);
}

/*
show book
*/
void ShowLen()
{
    show1len= sprintf( show1, "<HTML><HEAD><TITLE>TPC-W Product
Update Page</TITLE></HEAD><BODY BGCOLOR=\\"#FFFFFF\\">"
" <H1 ALIGN=\\"CENTER\\">TPC Web Commerce Benchmark (TPC-
W)</H1>"
" <H2 ALIGN=\\"CENTER\\"><IMG SRC=\\"gifs/tpclogo.gif\\"
ALIGN=\\"BOTTOM\\" BORDER=\\"0\\" WIDTH=\\"288\\" HEIGHT=\\"67\\"></H2>"
" <H2 ALIGN=\\"CENTER\\"> Admin Request Page</H2>");
    show2len= sprintf( show2, "<TR><TD><B> Enter New
Price</B></TD><TD ALIGN=\\"RIGHT\\"> $<INPUT TYPE=\\"text\\"
NAME=\\"NEWPRICE\\"></TD></TR>"
" <TR><TD><B> Enter New Picture</B></TD><TD
ALIGN=\\"RIGHT\\"><INPUT TYPE=\\"text\\" NAME=\\"NEWIMAGE\\"></TD></TR>"
" <TR><TD><B> Enter New Thumbnail</B></TD><TD
ALIGN=\\"RIGHT\\"><INPUT TYPE=\\"text\\" NAME=\\"NEWTUMB\\"></TD></TR>"
" </TABLE>"
" <P ALIGN=\\"CENTER\\"><INPUT
TYPE=\\"IMAGE\\"NAME=\\"Submit\\" SRC=\\"gifs/changes.gif\\" BORDER=\\"0\\"
HEIGHT=\\"30\\" WIDTH=\\"130\\">");
    show3len= sprintf( show3, "<A
HREF=\\"http://%s/tpcw/tpcw.dll?h\\"><IMG SRC=\\"gifs/search.gif\\"
ALT=\\"Search\\" BORDER=\\"0\\" WIDTH=\\"120\\" HEIGHT=\\"30\\"></A>"

```

```

        "<A HREF=\"http://%s/tpcw/tpcw.dll?a\"><IMG
SRC=\"gifs/home.gif\" ALT=\"Home\" BORDER=\"0\" VSPACE=\"0\" HSPACE=\"0\"
WIDTH=\"120\" HEIGHT=\"30\"></A>"
        "</P></FORM></BODY></HTML>", domainname, domainname);
}
/*
change info
*/
void ChangeInfoLen()
{
    changeinfollen= sprintf( changeinfo1,"<HTML><HEAD><TITLE>TPC-W
Admin Response Page</TITLE></HEAD><BODY BGCOLOR=\"#FFFFFF\">"
        "<H1 ALIGN=\"CENTER\">TPC Web Commerce Benchmark (TPC-W)</H1>"
        "<H2 ALIGN=\"CENTER\"><IMG SRC=\"gifs/tpclogo.gif\"
ALIGN=\"BOTTOM\" BORDER=\"0\" WIDTH=\"288\" HEIGHT=\"67\"></H2>"
        "<H2 ALIGN=\"CENTER\"> Admin Confirm Page</H2>"
        "<H2> Product Updated</H2>");
    changeinfo2len= sprintf( changeinfo2,"</BODY></HTML>");
}

/*
product update
*/
void ProdUpdate()
{
    produpdatellen= sprintf( produpdatel,"</FONT></DT></DL><P
ALIGN=\"CENTER\">"
        "<A HREF=\"http://%s/tpcw/tpcw.dll?h\"><IMG
SRC=\"gifs/search.gif\" ALT=\"Search\" BORDER=\"0\" WIDTH=\"120\"
HEIGHT=\"30\"></A>"
        "<A HREF=\"http://%s/tpcw/tpcw.dll?a\"><IMG
SRC=\"gifs/home.gif\" ALT=\"Home\" BORDER=\"0\" WIDTH=\"120\"
HEIGHT=\"30\"></A>", domainname, domainname);
}

void FillHTML()
{
    AdLen();
    HomeLen();
    NewProdLen();
    BestSelLen();
    BookInfoLen();
    SearchLen();
    SearchResLen();
    ShopLen();
    OrderStatusLen();
    CustReg();
    NewOrderLen();
    BuyReqLen();
    ShowLen();
    ChangeInfoLen();
    ProdUpdate();
}

```

## htmltext.h

```

// htmltext.h

void FillHTML();
// pagesize. h
//
// Copyright Unisys, 2000
#define HOMESIZE 7172
#define SHOPSIZE 2056
#define CUSTREGSIZE 2769
#define BUYREQSIZE 3690
#define BUYCONFSIZE 1530
#define ORDERINQ 953
#define ORDERDISP 1715
#define SEARCHREQ 2004
#define SEARCHRES 1891
#define NEWPROD 8095
#define BESTPROD 6640
#define PRODDT 1726
#define ADMINREQ 1551
#define ADMINCONF 1349

```

## pge.c

```

// pge. c
//
// Copyright Unisys, 1999,2000

#define _WIN32_WINNT 0x0500
#define PGESPINCOUNT 200

#include <windows.h>
#include "pge.h"
#include "tpcw.h"
#include "strings.h"
#include "readreg.h"

CRITICAL_SECTION csPGE;
extern char hostname[ 20];
extern int hostnumber;

BOOL PGEThreadDetach( PGE * pge)
{
    if( pge != NULL)
    {
        DisconnectFromServer( pge->ssSecureWebConn.sSocket,
            &g_SecurityFunc, &ClientCreds, &pge->hContext,
FALSE);
    }
    return TRUE;
}

```

```

        ErrorLog("Seek is less than 0","PGEThreadAttach()");
        return FALSE;
    }
    pge->TransactionCount = 0;
    return TRUE;
}

int PGEProcess( PGE * pge)
{
    int i;
    HRESULT hRet;
    HtmlBuffer * phb;
    SOCKET_STATE * pss;
    char * authptr= NULL;
    char * msgptr;
    char * reqptr;
    long msglen;
    long reqlen;
    time_t endtime;

    pge->request[0]= '\0';
    pge->embedmsg[0]= '\0';
    pge->padstring[0]= '\0';
    pge->auth[0]= '\0';
    pge->pgemsg[0]= '\0';
    msgptr= pge->pgemsg;
    reqptr= pge->request;
    MakeAlphaString( 5952,5952,5952, pge->padstring,&pge->lTLSeed);

    msglen= sprintf( pge-
>embedmsg,"<NUM>%s</NUM><OID>%.10I64d</OID>", pge->cc_num, pge->o_id);
    memcpy( msgptr, pge->embedmsg, msglen);
    msgptr= msgptr + msglen;
    memcpy( msgptr, pge->padstring, 5952);
    msglen= msglen + 5952;
    reqlen= sprintf( reqptr,"POST /pge.dll? HTTP/1.0\r\n"
        "Connection: Keep-Alive\r\n"
        "Content-Length: %d\r\n\r\n",
        msglen);
    reqptr= reqptr + reqlen;
    memcpy( reqptr, pge->pgemsg, msglen);
    reqptr= reqptr + msglen;
    phb=&pge->SendBuffer;
    pss=&pge->ssSecureWebConn;
    memcpy( pss->SendMsg, pge->request, 6080);
    phb->pbBuffer = pss->SendMsg;
    phb->cbBuffer = 6080;
    phb->cbOffset = 0;
    phb=&pge->RecvBuffer;
    phb->pbBuffer = pss->RecvMsg;
    phb->cbBuffer = pss->RecvMsgSz;
    phb->cbOffset = 0;

BOOL PGEThreadAttach(PGE * pge)
{
    SOCKET_STATE *pss;
    char StringBuf[128];
    char StringBuf2[128];
    LONG lMaxInteractions = 100000000;

    pge->PGEThreadID = GetCurrentThreadId();
    pss=&pge->ssSecureWebConn;
    pss->uPort = 443;
    strcpy( pss->szIPAddr, pgehost);
    pss->ConnMade = FALSE;
    pss->bRenegotiateConnection = TRUE;
    pge->ServerNameGenerator = 0;
    wprintf(StringBuf,"%lX", pge->PGEThreadID);
    if( strlen(StringBuf) > sizeof( pge->MyIDString) )
    {
        ErrorLog("strlen(StringBuf) > sizeof(pge->MyIDString)","
PGEThreadAttach");
        return FALSE;
    }
    else
    {
        strcpy(pge->MyIDString, StringBuf);
    }
    wprintf(StringBuf,"%s.%s", pge->MyIDString,
        _ltoa(pge->ServerNameGenerator, StringBuf2,10));
    if(strlen(StringBuf) > sizeof(pge->szServerName))
    {
        ErrorLog("strlen(StringBuf) > sizeof(pge-
>szServerName)", "PGEThreadAttach");
        return FALSE;
    }
    else
    {
        strcpy(pge->szServerName, StringBuf);
    }
    strcpy(pss->SendBuffer, "");
    pss->SendMsg = pss->SendBuffer;
    pss->SendMsgSz = strlen(pss->SendBuffer);
    pss->RecvMsg = pss->RecvBuffer;
    pss->RecvMsgSz = sizeof(pss->RecvBuffer);

    if( SetThreadPriority(GetCurrentThread(),
THREAD_PRIORITY_HIGHEST)== 0)
    {
        ErrorLog("SetThreadPriority Error", "PGEThreadAttach()");
        return FALSE;
    }
    pge->lTLSeed = InitRnd();
    if( pge->lTLSeed < 0)
    {

```

```

time(&endtime);
endtime= endtime - 200;

if(( pge->TransactionCount == 100 || pge->starttime < endtime)
    &&pge->starttime!= 0)
{
    if( pss->ConnMade)
    {
        i= DisconnectFromServer( pss-
>sSocket, &g_SecurityFunc, &hClientCreds, &pge->hContext, FALSE);
        if( i== 0)
        {
            pss->ConnMade= FALSE;
        }
        else
        {
            pss->ConnMade= FALSE;
        }
    }
    pss->bRenegotiateConnection= TRUE;
    pge->TransactionCount= 0;
    time(&pge->starttime);
}
if(( hRet= SecureSendThenRecvHtml(&pge->SendBuffer,
    &pge->RecvBuffer,
    &pge->RecvHeader,
    pss,
    pge,
    &g_SecurityFunc,
    &hClientCreds,
    &pge->hContext,
    &pge->Sizes)))
{
    return hRet;
}
else
{
    pge->TransactionCount++;
}
if( pge->RecvBuffer.cbBuffer < 6000)
{
    return -4;
}
authptr= strstr( pge->RecvBuffer.pbBuffer, "PGE ERROR");
if( authptr!= NULL)
{
    return -2;
}
authptr= strstr( pge->RecvBuffer.pbBuffer, "<CODE>")+ 6;
if( authptr== NULL)
{
    return -3;
}

```

```

}
i= 0;
while(* authptr!= '<')
{
    pge->auth[ i]=* authptr;
    authptr++;
    i++;
}
return 1;
}

int PGERequest( PGE * pge, __int64 o_id, char cc_num[], char auth_id[])
{
    int rc;

    pge->o_id= o_id;
    strcpy( pge->cc_num, cc_num);
    rc= PGEProcess( pge);
    memcpy( auth_id, pge->auth, 15);
    return rc;
}

BOOL PGEInit()
{
    if(! ReadPGEHost())
    {
        return FALSE;
    }
    return TRUE;
}

BOOL CleanPGE()
{
    return TRUE;
}

pge.h

// pge.h
//
// Copyright Unisys, 1999,2000

#define SECURITY_WIN32
#define PGEHost "pge" //
#define MAX_CONN_OUT 1024
#define O_CC_NUM_LEN 17
#define MAX_SECURE_RETRIES 10
#define MAX_SSL_HANDSHAKE_RETRIES 10
#define MAX_OPEN_SOCKET_RETRIES 5

```

```

#include <windows.h>
#include <httpext.h>
#include <sql.h>
#include <time.h>
#include "basetsd.h"
#include "sspi.h"

extern HMODULE g_hSecurity;
extern SecurityFunctionTable g_SecurityFunc;
extern CredHandle hClientCreds;

//
// HtmlBuffer
// A buffer to receive/ contain HTML code.
typedef struct {
    PBYTE pbBuffer; // Pointer to the buffer.
    LONG cbBuffer; // Size of buffer in (contiguous) bytes.
    // (This is that max available storage in the buffer.
    LONG cbOffset; // An offset into the buffer.
    // This is a utility var that is used for various
    // purposes depending on context.
    // (E.g., Returned by a receive data function to
    // use it to point to the next byte past the last
    // byte received into the buffer.)
    int i;
} HtmlBuffer;

//
// HtmlHeader
// Contains fields for certain components that can occur in the
// header of an HTML Response.
typedef struct {
    INT RespStatusCode; // The Response Status Code
    CHAR RespStatusMsg[ 16]; // First 15 chars of RespStatusMsg
subfield
    LONG ContentLength; // Content-Length header field.
    CHAR SetCookie[ 64]; // Set-Cookie field
    CHAR Location[ 128]; // Location field
} HtmlHeader;

#define MAX_BUFF_SZ 8000
typedef struct
{
    UINT uSocketNo;
    SOCKET sSocket;
    BOOL bRenegotiateConnection;
    // If TRUE then indicates that it is desired that a new full
    // handshake be performed between the caller and the server.
    BOOL ConnMade;
    // Flag that is set TRUE when a connection is established via
this socket.
    ULONG ulSendBuffSz;
    ULONG ulRecvBuffSz;

```

```

    CHAR szIPAddr[ 20];
    USHORT uPort;
    CHAR SendBuffer[ MAX_BUFF_SZ];
    CHAR RecvBuffer[ MAX_BUFF_SZ];
    ULONG SendMsgSz;
    PBYTE SendMsg;
    ULONG RecvMsgSz;
    PBYTE RecvMsg;
} SOCKET_STATE;

typedef struct
{
    DWORD PGThreadID;
    // SOCKET Socket;
    int TransactionCount;
    __int64 o_id;
    char cc_num[ O_CC_NUM_LEN];
    long lTLSeed;
    char request[ 8000];
    char embedmsg[ 60];
    char padstring[ 6000];
    char pgemsg[ 7000];
    char auth[ 16];
    time_t starttime;
    SOCKET_STATE ssSecureWebConn;
    CtxtHandle hContext;
    SecPkgContext_StreamSizes Sizes;
    LONG ServerNameGenerator;
    CHAR MyIDString[ 16];
    CHAR szServerName[ 32];
    HtmlBuffer SendBuffer;
    HtmlBuffer RecvBuffer;
    HtmlHeader RecvHeader;
} PGE;

char pgehost[ 20];
BOOL PGEInit();
BOOL CleanPGE();
BOOL PGERequest( PGE * pge, __int64 o_id, char cc_num[], char
auth_id[]);
BOOL PGThreadAttach( PGE * pge);
BOOL PGThreadDetach( PGE * pge);
HRESULT SecureSendThenRecvHtml(
    HtmlBuffer * pSendBuffer,
    HtmlBuffer * pRecvBuffer,
    HtmlHeader * pRecvHeader,
    SOCKET_STATE * pss,
    PGE * pts,
    SecurityFunctionTable * g_pSecurityFunc,
    CredHandle * phClientCreds,
    CtxtHandle * phContext,
    SecPkgContext_StreamSizes * pSizes
);

```

```

INT
DisconnectFromServer(
    SOCKET Socket,
    SecurityFunctionTable *g_pSecurityFunc,
    PCredHandle phCreds,
    CtxtHandle * phContext,
    BOOL fVerbose
);
#define TPCW_E_OK ((HRESULT)0x00000000L)
#define TPCW_E_SECURE_SEND_THEN_RECV_HTML_BAD_HTTP_PAGE_STATUS
((HRESULT)0xE0010031L)
#define TPCW_E_SECURE_SEND_THEN_RECV_HTML_BAD_HTTP_IMAGE_STATUS
((HRESULT)0xE0010032L)
// ParseHtmlHeader codes
#define TPCW_E_PARSE_HTML_HEADER__FAILED_TO_CONVERT_STATUS_TO_INT
((HRESULT)0xE0010201L)
#define
TPCW_E_PARSE_HTML_HEADER__FAILED_TO_CONVERT_CONTENTLENGTH_TO_INT
((HRESULT)0xE0010202L)
#define TPCW_E_PARSE_HTML_HEADER__HEADER_DOESNOT_BEGIN_WITH_HTTP_SLASH
((HRESULT)0xE0010203L)
#define TPCW_E_PARSE_HTML_HEADER__FIRST_BLANK_CHAR_NOT_FOUND
((HRESULT)0xE0010204L)
#define
TPCW_E_PARSE_HTML_HEADER__DIDNOT_FIND_BLANK_CHAR_AFTER_STATUSCODE
((HRESULT)0xE0010205L)
#define
TPCW_E_PARSE_HTML_HEADER__DIDNOT_FIND_CRLF_TO_TERMINATE_STATUSCODE_LINE
((HRESULT)0xE0010206L)
#define
TPCW_E_PARSE_HTML_HEADER__NOT_ENOUGH_ROOM_IN_HEADER_LOCATION_FIELD
((HRESULT)0xE0010207L)
// SecureSendThenRecvHtml codes
#define TPCW_E_SECURE_SEND_THEN_RECV_HTML__SOCKIOINIT_FAILED
((HRESULT)0xE0010301L)
#define
TPCW_E_SECURE_SEND_THEN_RECV_HTML__MAX_SSL_HANDSHAKE_RETRIES_FAILED
((HRESULT)0xE0010302L)
#define TPCW_E_SECURE_SEND_THEN_RECV_HTML__MAX_SECURE_RETRIES_FAILED
((HRESULT)0xE0010303L)
#define
TPCW_E_SECURE_SEND_THEN_RECV_HTML__FATAL_ERR_FROM_SECURE_SEND_THEN_RECV
((HRESULT)0xE0010304L)
#define TPCW_E_SECURE_SEND_THEN_RECV_HTML__HTTP_STATUS_NOT_200
((HRESULT)0xE0010305L)
// SecureSendThenRecv codes
#define TPCW_E_SECURE_SEND_THEN_RECV__PARSE_HTML_HEADER_FAILED
((HRESULT)0xE0010501L)
#define TPCW_E_SECURE_SEND_THEN_RECV__CONTENT_LENGTH_EQ_MINUS_1
((HRESULT)0xE0010502L)
#define TPCW_E_SECURE_SEND_THEN_RECV__NOT_ENOUGH_ROOM_IN_RECVBUFFER
((HRESULT)0xE0010503L)

```

```

#define TPCW_E_SECURE_SEND_THEN_RECV_BUFFER_ALLOC_FAILED
((HRESULT)0xE0010504L)
// PerformClientHandShake codes
#define TPCW_E_PERFORMCLIENTHANDSHAKE_ERROR_SENDING_DATA_TO_SERVER_1
((HRESULT)0xE0010601L)
// ClientHandShakeLoop codes
#define
TPCW_E_CLIENTHANDSHAKELOOP_SERVER_REQUESTED_CLIENT_AUTHORIZATION
((HRESULT)0xE0010701L)
#define TPCW_E_CLIENTHANDSHAKELOOP_ERROR_SENDING_DATA_TO_SERVER_2
((HRESULT)0xE0010702L)
#define TPCW_E_CLIENTHANDSHAKELOOP_OUT_OF_MEMORY_2
((HRESULT)0xE0010703L)
#define TPCW_E_CLIENTHANDSHAKELOOP_BUFFER_ALLOC_FAILED
((HRESULT)0xE0010704L)

```

#### readreg.c

```

// readreg.c
//
// Copyright Unisys, 1999,2000
#include "tpcw.h"
#include "tpcw_db.h" //
#include "thread.h"
#ifndef PGEHost
#include "pge.h"
#endif
extern char hostname[ 20];
extern int hostnumber;
extern long maxitem;
extern char cachehost[ 20];
extern __int64 O_ID;
extern __int64 C_ID;
extern __int64 ADDR_ID;
extern long Session_ID;
extern __int64 LimitO_ID;
extern __int64 LimitC_ID;
extern __int64 LimitADDR_ID;
extern long LimitSession_ID;
extern char domainname[ 30];

BOOL ReadMaxItem( TPCW * tpcw)
{
    int rc;
    rc= SQLGetMaxItem( tpcw);
    maxitem= tpcw->hash;

    if( rc< 0)
    {
        return FALSE;
    }
    return TRUE;
}

```

```

long ReadMaxC_ID( TPCW * tpcw, long range)
{
    CUSTOMER_INFO_ENQ_DATA cust;
    tpcw->retcode= SQLGetMaxC_ID( tpcw,&cust, range);
    if( tpcw->retcode== FALSE)
    {
        return 0;
    }
    if( cust.c_id< 0)
    {
        return 0;
    }
    C_ID= cust.c_id;
    LimitC_ID= C_ID+ (__int64)range;
//
    return 1;
}

long ReadMaxO_ID( TPCW * tpcw, long range)
{
    ORDER_ENQ_DATA order;
    tpcw->retcode= SQLGetMaxO_ID( tpcw,&order, range);
    if( tpcw->retcode== FALSE)
    {
        return 0;
    }
    if( order.o_id< 0)
    {
        return 0;
    }
    O_ID= order.o_id;
    LimitO_ID= O_ID+ (__int64)range;
//
    return 1;
}

long ReadMaxADDR_ID( TPCW * tpcw, long range)
{
    ADDRESS_ENQ_DATA addr;
    tpcw->retcode= SQLGetMaxADDR_ID( tpcw,&addr, range);
    if( tpcw->retcode== FALSE)
    {
        return 0;
    }
    if( addr.a_id< 0)
    {
        return 0;
    }
    ADDR_ID= addr.a_id;
    LimitADDR_ID= ADDR_ID+ (__int64)range;
//
}

}

return 1;

long ReadMaxSession_ID( TPCW * tpcw, long range)
{
    tpcw->retcode= SQLGetMaxSessionID( tpcw, range);
    if( tpcw->retcode== FALSE)
    {
        return 0;
    }
    if( tpcw->hash < 0)
    {
        return FALSE;
    }
    Session_ID= tpcw->hash;
    LimitSession_ID= Session_ID+ range;
    return Session_ID;
}

BOOL ReadHostName()
{
    int i;
    i= gethostname( hostname, sizeof( hostname));
    if( i!= 0)
    {
        ErrorLog(" Fail on gethostname", " ReadHostName()");
        return FALSE;
    }
    i = sscanf(hostname, "tpcwww%d", &hostnumber);
    if( i == 0)
        i = sscanf(hostname, "TPCWWW%d", &hostnumber);
    if( i== 0)
    {
        ErrorLog(" Fail on gethostnumber", " ReadHostName()");
        return FALSE;
    }
    return TRUE;
}

BOOL ReadCacheHost()
{
    HKEY hkTPCW;
    DWORD dwType= REG_SZ;
    DWORD dwSize;
    cachehost[ 0]= '\0';
    if( RegOpenKeyEx( HKEY_LOCAL_MACHINE, "SOFTWARE\\DELL\\TPCW", 0
,KEY_READ, &hkTPCW)== ERROR_SUCCESS)
    {
        dwSize= sizeof( cachehost);
}
}

```

```

        if( RegQueryValueEx( hkTPCW, "CacheHost", NULL, &dwType, (
unsigned      char *) cachehost, &dwSize) != ERROR_SUCCESS)
        {
            ErrorLog("CacheHost key value does not exist in
registry", "ReadCacheHost()");
            return FALSE;
        }
    }
    else
    {
        ErrorLog(" Cannot open register key", " ReadCacheHost()");
        return FALSE;
    }
    if( RegCloseKey( hkTPCW) != ERROR_SUCCESS)
    {
        return FALSE;
    };
    if( strlen( cachehost) > 0)
    {
        return TRUE;
    }
    else
    {
        ErrorLog(" CacheHost length is less than zero", "
ReadCacheHost()");
        return FALSE;
    }
}

BOOL ReadDomainName()
{
    HKEY hkTPCW;
    DWORD dwType= REG_SZ;
    DWORD dwSize;
    domainname[ 0]= '\0';

    if( RegOpenKeyEx( HKEY_LOCAL_MACHINE, "SOFTWARE\\DELL\\TPCW", 0
, KEY_READ, &hkTPCW) == ERROR_SUCCESS)
    {
        dwSize= sizeof( domainname);
        if( RegQueryValueEx( hkTPCW, "DomainName", NULL, &dwType, (
unsigned char *) domainname, &dwSize) != ERROR_SUCCESS)
        {
            ErrorLog(" DomainName key value does not exist
in register", " ReadDomainName()");
            return FALSE;
        }
    }
    else
    {

```

```

        ErrorLog(" Cannot open register key", "
ReadDomainName()");
        return FALSE;
    }
    if( RegCloseKey( hkTPCW) != ERROR_SUCCESS)
    {
        return FALSE;
    };
    if( strlen( domainname) > 0)
    {
        return TRUE;
    }
    else
    {
        ErrorLog("DomainName length is less than zero", "
ReadDomainName()");
        return FALSE;
    }
}

BOOL ReadPGEHost()
{
    HKEY hkTPCW;
    DWORD dwType= REG_SZ;
    DWORD dwSize;
    pgehost[ 0]= '\0';

    //
    if( RegOpenKeyEx( HKEY_LOCAL_MACHINE, "SOFTWARE\\Dell\\TPCW", 0
, KEY_READ, &hkTPCW) == ERROR_SUCCESS)
    {
        dwSize= sizeof( pgehost);
        if( RegQueryValueEx( hkTPCW, "PGEHost",
NULL, &dwType, (unsigned
char *) pgehost, &dwSize) != ERROR_SUCCESS)
        {
            ErrorLog(" PGEHost key value does not exist in
registry", " ReadPGEHost()");
            return FALSE;
        }
    }
    else
    {
        ErrorLog("Cannot open registry key", " ReadPGEHost()");
        return FALSE;
    }
    if( RegCloseKey( hkTPCW) != ERROR_SUCCESS)
    {
        return FALSE;
    };
    if( strlen( pgehost) > 0)
    {

```

```

        return TRUE;
    }
    else
    {
        ErrorLog(" PGEHost length is less than zero", "
ReadPGEHost()");
        return FALSE;
    }
}

```

#### readreg.h

```

// readreg. h
//
// Copyright Unisys, 1999,2000

```

```

BOOL ReadHostName();
BOOL ReadPGEHost();
BOOL ReadDomainName();
BOOL ReadCacheHost();
BOOL ReadMaxItem(TPCW * tpcw);
long ReadMaxO_ID(TPCW * tpcw, long range);
long ReadMaxC_ID(TPCW * tpcw, long range);
long ReadMaxADDR_ID(TPCW * tpcw, long range);
long ReadMaxSession_ID(TPCW * tpcw, long range);

```

#### ssl.c

```

// ssl.c
// Copyright Unisys, 1999,2000
#include "ssl.h"
#include "debug.h"

extern long numsslretry;
extern CRITICAL_SECTION sslretry;

```

```

HMODULE g_hSecurity = NULL;
SecurityFunctionTable g_SecurityFunc;
HCERTSTORE hMyCertStore = NULL;
SCHANNEL_CRED SchannelCred;
CredHandle hClientCreds;
DWORD dwProtocol= 0;
LPSTR pszUserName = NULL;
ALG_ID aiKeyExch = 0;
BOOL fVerbose = FALSE;
BOOL fVerboseSSL = FALSE;

```

```

//=====
//
// Function name: SockioInit
//
//=====

```

```

BOOL SockioInit( SOCKET_STATE * pss)
{
    static True = TRUE; // useful when you need to point to the
vaue TRUE
    SOCKADDR_IN sinRemoteAddr;
    IN_ADDR INADDR_IPAddress;
    INT iErr;

    pss->ulRecvBuffSz = sizeof( pss->RecvBuffer);
    pss->ulSendBuffSz = sizeof( pss->SendBuffer);
    pss->sSocket = socket( AF_INET, SOCK_STREAM, 0);
    if (pss->sSocket == INVALID_SOCKET)
    {
        return( TRUE);
    };
    // Set the receive buffer size...
    iErr = setsockopt( pss->sSocket, SOL_SOCKET, SO_RCVBUF,
(char *) &pss->ulRecvBuffSz,
sizeof( pss->ulRecvBuffSz));
    if (iErr == SOCKET_ERROR)
    {
        closesocket( pss->sSocket);
        return( TRUE);
    };
    // Set the send buffer size
    iErr = setsockopt( pss->sSocket, SOL_SOCKET, SO_SNDBUF,
(char *) &pss->ulSendBuffSz,
sizeof( pss->ulSendBuffSz));
    if (iErr == SOCKET_ERROR)
    {
        closesocket( pss->sSocket);
        return( TRUE);
    };
    // Disable Nagling for our sends.
    iErr = setsockopt( pss->sSocket, IPPROTO_TCP, TCP_NODELAY,
(const char *)& True,
sizeof( True));
    if (iErr == SOCKET_ERROR)
    {
        closesocket( pss->sSocket);
        return( TRUE);
    };
    ZeroMemory(& sinRemoteAddr, sizeof( sinRemoteAddr));
    INADDR_IPAddress.s_addr = inet_addr( pss->szIPAddr);
    if (INADDR_IPAddress.s_addr == -1)
    {
        closesocket( pss->sSocket);
        return( TRUE);
    };
    sinRemoteAddr.sin_family = AF_INET;
    sinRemoteAddr.sin_port = htons( pss->uPort);
    sinRemoteAddr.sin_addr = INADDR_IPAddress;
    iErr = connect( pss->sSocket,( PSOCKADDR) &sinRemoteAddr,

```

```

        sizeof (sinRemoteAddr));
        if (iErr == SOCKET_ERROR)
        {
            closesocket( pss->sSocket);
            return( TRUE);
        };
        return( FALSE);
    }; // SockioInit

//
// OpenNewSocketWithRetries
//
// Tries to establish a socket connection, retrying to do this if
necessary.
// Inputs:
// pss : pointer to SOCKET_STATE structure to initialize
// iMaxRetries : Max number of retries to attempt. Must be greater than
or equal to 0 else fatal error
//
// Outputs:
// Function returns # of retries attempted that failed.
// If this returned value is < iMaxRetries, then the connection was
established
// If this returned value == iMaxRetries, then the connection was not
established
//
INT OpenNewSocketWithRetries (SOCKET_STATE * pss, INT iMaxRetries)
{
    INT iRetryFailureCount;
    if( iMaxRetries< 0) exit( 555);
    if (! SockioInit(pss)) // SockioInit succeeded here - we needed
no retries
        return(- 1); //( remember: if returned value < iMaxRetries this
indicates success)
    else if( iMaxRetries== 0) // We aren't allowed any retries
    {
        return( 0); // indicates failure
    }
    else
    { // do retries of SockioInit()
        for (iRetryFailureCount= 0;
            iRetryFailureCount< iMaxRetries;
iRetryFailureCount++)
        {
            if (SockioInit( pss))
            {
                // SockioInit failed
                continue;
            }
            else
            break; // SockioInit succeeded
        }
    }
}

```

```

    }
    return( iRetryFailureCount);
}

// end OpenNewSocketWithRetries
//
// ParseHtmlHeader(* pBuffer, *pHtmlHeader);
//
// ParseHtmlHeader expects that the HTML header starts at the 1st byte
of
// *pBuffer.(It checks for this and if not true returns error.)
//
// It requires that *pBuffer be a null- delimited string.
//
// If the RespStatusValue (e.g.200 for an "OK"response)
// is 200 then ParseHtmlHeader expects a Content- Length: header field.
// If it is not found, then pHtmlHeader->ContentLength is returned as -
1.
// If ParseHtmlHeader fails for any other reason, ParseHtmlHeader
returns an error code,
// Else it returns TPCW_E_OK.
//
HRESULT ParseHtmlHeader(
    char * pBuffer, // In : The HTML
Header string (/ 0 delimited) to parse.
    HtmlHeader * pHtmlHeader) // In : Pointsto HtmlHeader to
recv the HTML header
// :
components found.
{
    char * pbCbeg, *pbCend, * pbC1, * pbC2; // cursors into the
char buffer called Buffer.
    INT RespStatusValue;
    INT I;
    LONG L;

    // Set all default return values in Header struct:
    pHtmlHeader->RespStatusCode=-1;
    pHtmlHeader->RespStatusMsg[ 0]= '\0';
    pHtmlHeader->ContentLength=-1;
    pHtmlHeader->SetCookie[0]= '\0';

    pbCbeg = pbC1 = pBuffer; // Point to beginning of buffer
    pbCend = pbCbeg + strlen( pBuffer); // Point to byte after last
of Header.
    // Check that Header begins with "HTTP/"
    if(!( strcmp( pbC1,"HTTP/"))> 0)
        return(
TPCW_E_PARSE_HTML_HEADER__HEADER_DOESNOT_BEGIN_WITH_HTTP_SLASH);
    pbC1 += strlen("HTTP/");- 1;
    // Get the response status value as a token
    if(( pbC2= strstr( pbC1, " "))= NULL)

```

```

        return(
TPCW_E_PARSE_HTML_HEADER__FIRST_BLANK_CHAR_NOT_FOUND);
    pbC1 = pbC2+ strlen(" ");
    // Convert the numeric status value to an int
    if(( RespStatusValue= atoi( pbC2))== 0)
        return(
TPCW_E_PARSE_HTML_HEADER__FAILED_TO_CONVERT_STATUS_TO_INT);
    pHtmlHeader->RespStatusCode = RespStatusValue;
    // Copy first so- many characters of the Response Status Msg,
    which follows the numeric status code.
    if(( pbC2= strstr( pbC1," ")== NULL) // Assumption: There's
only one " "preceding the numeric Response Status Code
        return(
TPCW_E_PARSE_HTML_HEADER__DIDNOT_FIND_BLANK_CHAR_AFTER_STATUSCODE);
    if(( pbC1= strstr( pbC2,"\r\n"))== NULL)
        return(
TPCW_E_PARSE_HTML_HEADER__DIDNOT_FIND_CRLF_TO_TERMINATE_STATUSCODE_LINE
);
    pbC2++; // Point at beginning of Response Status Msg.
    I = sizeof( pHtmlHeader->RespStatusMsg);
    if(( pbC1- pbC2)<( I- 1))
    {
        *( pbC1)= '\0'; // Delimit the Msg
        strncpy( pHtmlHeader->RespStatusMsg, pbC2, I- 1);
        // strncpy will null pad tail of RespStatusMsg
        *( pbC1)= '\r'; // Restore '\r' in the Msg
    }
    else
    {
        memcpy( pHtmlHeader->RespStatusMsg, pbC2, I- 1);
        pHtmlHeader->RespStatusMsg[ I- 1]= '\0'; // Delimit the
string in RespStatusMsg
    }
    pbCbeg = pbC1+ strlen("\r\n"); // Reposition pbCbeg to beg. of
additional Response Header lines
    //
    // Process the additional Response Header fields
    //
    // CONSTRAINT THAT MUST BE MET:
    // Going into the foll.while loop, pbC1 must point to beg. of
"\r\n"that ends the first
    // Header Response line (the one containing the numeric
Response Status Code).
    while (pbCbeg < pbCend)
    {
        // Reposition pbCbeg to beg.of additional Response Header lines
        // Note: pbCbeg will gradually move thru the rest of the
Header.It means "begin your next search here"
        pbCbeg = pbC1+ strlen("\r\n");
        // Look for next "\r\n"that delimits next response header line.
        if(( pbC1= strstr( pbCbeg,"\r\n"))== NULL) // This ends our
search for additional header response lines of interest.
            return( TPCW_E_OK);

```

```

        // pbCbeg now points to the beginning of the next Header
response line
        // and pbC1 now points just past this Header response line,
        // So ...
        // now try to match beg.of just found line with a field we're
interested in.
        //
        // Look for Content- Length: field
        if(_strnicmp("Content-Length:", pbCbeg, strlen("Content-
Length:"))== 0)
        {
            pbC2 = pbCbeg + strlen("Content-Length:");
            if(( L= atol( pbC2))== 0L)
            {
                // do what here? return error AND return w pHtmlHeader-
>ContentLength = 1?
                return(
TPCW_E_PARSE_HTML_HEADER__FAILED_TO_CONVERT_CONTENTLENGTH_TO_INT);
            }
            pHtmlHeader->ContentLength = L;
            continue;
        }
        // Look for Set-Cookie: field
        if(_strnicmp("Set-Cookie:", pbCbeg, strlen("Set-Cookie:"))== 0)
        {
            pbC2 = pbCbeg + strlen("Set-Cookie:");
            I = sizeof( pHtmlHeader->SetCookie);
            if(( pbC1- pbC2)<( I- 1))
            {
                *( pbC1)= '\0'; // Delimit the Msg
                strncpy( pHtmlHeader->SetCookie, pbC2, I-1); // strncpy will
null pad tail of RespStatusMsg
                *( pbC1)= '\r'; // Restore '\r' in the Msg
            }
            else
            {
                memcpy( pHtmlHeader->SetCookie, pbC2, I-1);
                pHtmlHeader->SetCookie[ I- 1]= '\0';
                // Delimit the string in RespStatusMsg
            }
            continue;
        }
        switch (RespStatusValue)
        {
            case 302:
                // Look for the Location: field
                if(_strnicmp("Location:", pbCbeg, strlen("Location:"))== 0)
                {
                    pbC2 = pbCbeg + strlen("Location:");
                    I = sizeof( pHtmlHeader->Location);
                    if(( pbC1- pbC2)>( I- 1))
                        return(
TPCW_E_PARSE_HTML_HEADER__NOT_ENOUGH_ROOM_IN_HEADER_LOCATION_FIELD);

```

```

        if(( pbC1- pbC2)<( I- 1))
        {
            *( pbC1)= '\0'; // Delimit the Msg
            strncpy( pHtmlHeader->Location, pbC2, I- 1); // strncpy will
null pad tail of RespStatusMsg
            *( pbC1)= '\r'; // Restore '\r' in the Msg
        }
        else
        {
            memcpy( pHtmlHeader->Location, pbC2, I- 1);
            pHtmlHeader->Location[ I-1]= '\0'; // Delimit the string in
RespStatusMsg
        }
        continue;
    }
    break;
default:
    break;
} // end while (pbCbeg < pbCend)
return( TPCW_E_OK);
} // end ParseHtmlHeader

/*****
*****/
SECURITY_STATUS CreateCredentials(
    SecurityFunctionTable *g_pSecurityFunc, // in
    HCERTSTORE * phMyCertStore, // in
    LPSTR pszUserName, // in
    SCHANNEL_CRED * pSchannelCred, // in
    DWORD dwProtocol, // in
    ALG_ID aiKeyExch, // in
    PCredHandle phCreds) // out
{
    TimeStamp tsExpiry;
    SECURITY_STATUS Status;
    DWORD cSupportedAlgs = 0;
    ALG_ID rgbSupportedAlgs[ 16];
    PCCERT_CONTEXT pCertContext = NULL;
    //
    // Build Schannel credential structure. Currently, this sample
only
    // specifies the protocol to be used (and optionally the
certificate,
    // of course). Real applications may wish to specify other
parameters
    // as well.
    //
    ZeroMemory( pSchannelCred, sizeof(*( pSchannelCred)));
    pSchannelCred->dwVersion = SCHANNEL_CRED_VERSION;
    if( pCertContext)

```

```

    {
        pSchannelCred->cCreds = 1;
        pSchannelCred->paCred = &pCertContext;
    }
    pSchannelCred->grbitEnabledProtocols = dwProtocol;
    if( aiKeyExch)
    {
        rgbSupportedAlgs[ cSupportedAlgs++] = aiKeyExch;
    }
    if( cSupportedAlgs)
    {
        pSchannelCred->cSupportedAlgs = cSupportedAlgs;
        pSchannelCred->palgSupportedAlgs = rgbSupportedAlgs;
    }
    pSchannelCred->dwFlags |= SCH_CRED_NO_DEFAULT_CREDS;
    pSchannelCred->dwFlags |= SCH_CRED_MANUAL_CRED_VALIDATION;
    //
    // Create an SSPI credential.
    //
    Status = (* g_pSecurityFunc).AcquireCredentialsHandleA(
        NULL, // Name of principal
        UNISP_NAME_A, // Name of package
        SECPKG_CRED_OUTBOUND, // Flags indicating use
        NULL, // Pointer to logon ID
        pSchannelCred, // Package specific data
        NULL, // Pointer to GetKey() func
        NULL, // Value to pass to GetKey()
        phCreds, // (out) Cred Handle
        &tsExpiry); // (out) Lifetime (optional)
    if( Status != SEC_E_OK)
    {
        return Status;
    }
    //
    // Free the certificate context. Schannel has already made its
own copy.
    //
    if( pCertContext)
    {
        CertFreeCertificateContext( pCertContext);
    }
    return SEC_E_OK;
}

BOOL LoadSecurityLibrary( void)
{
    PSecurityFunctionTable pSecurityFunc;
    INIT_SECURITY_INTERFACE pInitSecurityInterface;
    g_hSecurity = LoadLibrary("security.dll");
    if( g_hSecurity == NULL)
    {
        return FALSE;
    }
}

```

```

    pInitSecurityInterface = (INIT_SECURITY_INTERFACE)
GetProcAddress(
    g_hSecurity,
    "InitSecurityInterfaceA");
if( pInitSecurityInterface == NULL)
{
return FALSE;
}
pSecurityFunc = pInitSecurityInterface();
if( pSecurityFunc == NULL)
{
return FALSE;
}
CopyMemory(& g_SecurityFunc, pSecurityFunc,
sizeof( g_SecurityFunc));
return TRUE;
}

/*****
SECURITY_STATUS
ClientHandshakeLoop(
    SOCKET Socket, // in
    SecurityFunctionTable *g_pSecurityFunc, // in
    HCERTSTORE * phMyCertStore, // in
    SCHANNEL_CRED * pSchannelCred, // in
    PCredHandle phCreds, // in
    CtxtHandle * phContext, // in, out
    BOOL fDoInitialRead, // in
    SecBuffer * pExtraData, // out
    BOOL fVerbose) // in
{
SecBufferDesc InBuffer;
SecBuffer InBuffers[ 2];
SecBufferDesc OutBuffer;
SecBuffer OutBuffers[ 1];
DWORD dwSSPIFlags;
DWORD dwSSPIOutFlags;
TimeStamp tsExpiry;
SECURITY_STATUS scRet;
DWORD cbData;
PUCHAR IoBuffer;
DWORD cbIoBuffer;
BOOL fDoRead;
dwSSPIFlags = ISC_REQ_SEQUENCE_DETECT |
ISC_REQ_REPLAY_DETECT |
ISC_REQ_CONFIDENTIALITY |
ISC_RET_EXTENDED_ERROR |
ISC_REQ_ALLOCATE_MEMORY |
ISC_REQ_STREAM;
//
// Allocate data buffer.

```

```

//
IoBuffer = LocalAlloc( LMEM_FIXED, IO_BUFFER_SIZE);
if( IoBuffer == NULL)
{
return TPCW_E_CLIENHANDSHAKELOOP_BUFFER_ALLOC_FAILED;
}
cbIoBuffer = 0;
fDoRead = fDoInitialRead;
//
// Loop until the handshake is finished or an error occurs.
//
scRet = SEC_I_CONTINUE_NEEDED;
while( scRet == SEC_I_CONTINUE_NEEDED ||
scRet == SEC_E_INCOMPLETE_MESSAGE ||
scRet == SEC_I_INCOMPLETE_CREDENTIALS)
{
//
// Read data from server.
//
if( 0 == cbIoBuffer || scRet == SEC_E_INCOMPLETE_MESSAGE)
{
if( fDoRead)
{
cbData = recv( Socket,
IoBuffer + cbIoBuffer,
IO_BUFFER_SIZE - cbIoBuffer,
0);
if( cbData == SOCKET_ERROR)
{
scRet = SEC_E_INTERNAL_ERROR;
break;
}
else if( cbData == 0)
{
scRet = SEC_E_INTERNAL_ERROR;
break;
}
if( fVerbose) printf("%d bytes of handshake data received\n", cbData);
cbIoBuffer += cbData;
}
else
{
fDoRead = TRUE;
}
}
//
// Set up the input buffers.Buffer 0 is used to pass in data
// received from the server.Schannel will consume some or all
// of this.Leftover data (if any) will be placed in buffer 1 and
// given a buffer type of SECBUFFER_EXTRA.
//
InBuffers[ 0].pvBuffer = IoBuffer;
InBuffers[ 0].cbBuffer = cbIoBuffer;

```

```

InBuffers[ 0].BufferType = SECBUFFER_TOKEN;
InBuffers[ 1].pvBuffer = NULL;
InBuffers[ 1].cbBuffer = 0;
InBuffers[ 1].BufferType = SECBUFFER_EMPTY;
InBuffer.cbBuffers = 2;
InBuffer.pBuffers = InBuffers;
InBuffer.ulVersion = SECBUFFER_VERSION;
//
// Set up the output buffers. These are initialized to NULL
// so as to make it less likely we'll attempt to free random
// garbage later.
//
OutBuffers[ 0].pvBuffer = NULL;
OutBuffers[ 0].BufferType = SECBUFFER_TOKEN;
OutBuffers[ 0].cbBuffer = 0;
OutBuffer.cbBuffers = 1;
OutBuffer.pBuffers = OutBuffers;
OutBuffer.ulVersion = SECBUFFER_VERSION;
//
// Call InitializeSecurityContext.
//
scRet =
(* g_pSecurityFunc).InitializeSecurityContextA( phCreds,
phContext,
NULL,
dwSSPIFlags,
0,
SECURITY_NATIVE_DREP,
&InBuffer,
0,
NULL,
&OutBuffer,
&dwSSPIOutFlags,
&tsExpiry);
//
// If InitializeSecurityContext was successful (or if the error was
// one of the special extended ones), send the contents of the output
// buffer to the server.
//
if( scRet == SEC_E_OK ||
scRet == SEC_I_CONTINUE_NEEDED ||
FAILED( scRet) && (dwSSPIOutFlags &
ISC_RET_EXTENDED_ERROR))
{
if( OutBuffers[ 0].cbBuffer != 0 && OutBuffers[ 0].pvBuffer
!= NULL)
{
cbData = send( Socket,
OutBuffers[ 0].pvBuffer,
OutBuffers[ 0].cbBuffer,
0);
if( cbData == SOCKET_ERROR || cbData == 0)
{

```

```

(* g_pSecurityFunc).FreeContextBuffer( OutBuffers[ 0].pvBuffer);
(* g_pSecurityFunc).DeleteSecurityContext( phContext);
LocalFree( IoBuffer);
return
TPCW_E_CLIENHANDSHAKELOOP_ERROR_SENDING_DATA_TO_SERVER_2;
}
if( fVerbose) printf("%d bytes of handshake data sent\n", cbData);
// Free output buffer.
(* g_pSecurityFunc).FreeContextBuffer( OutBuffers[ 0].pvBuffer);
OutBuffers[ 0].pvBuffer = NULL;
}
//
// If InitializeSecurityContext returned SEC_E_INCOMPLETE_MESSAGE,
// then we need to read more data from the server and try again.
//
if( scRet == SEC_E_INCOMPLETE_MESSAGE)
{
continue;
}
//
// If InitializeSecurityContext returned SEC_E_OK, then the
// handshake completed successfully.
//
if( scRet == SEC_E_OK)
{
//
// If the "extra"buffer contains data, this is encrypted application
// protocol layer stuff. It needs to be saved. The application layer
// will later decrypt it with DecryptMessage.
//
if( fVerbose) printf("Handshake was successful\n");
if( InBuffers[ 1].BufferType == SECBUFFER_EXTRA)
{
pExtraData->pvBuffer = LocalAlloc( LMEM_FIXED,
InBuffers[ 1].cbBuffer);
if( pExtraData->pvBuffer == NULL)
{
return
TPCW_E_CLIENHANDSHAKELOOP_OUT_OF_MEMORY_2;
}
MoveMemory( pExtraData->pvBuffer,
IoBuffer + (cbIoBuffer -
InBuffers[ 1].cbBuffer),
InBuffers[ 1].cbBuffer);
pExtraData->cbBuffer = InBuffers[ 1].cbBuffer;
pExtraData->BufferType = SECBUFFER_TOKEN;
if( fVerbose) printf("%d bytes of app data was bundled with handshake
data\n",
pExtraData->cbBuffer);
}
else
{

```

```

pExtraData->pvBuffer = NULL;
pExtraData->cbBuffer = 0;
pExtraData->BufferType = SECBUFFER_EMPTY;
}
//
// Bail out to quit
//
break;
}
//
// Check for fatal error.
//
if( FAILED( scRet))
{
break;
}
//
// If InitializeSecurityContext returned SEC_I_INCOMPLETE_CREDENTIALS,
// then the server just requested client authentication.
//
if( scRet == SEC_I_INCOMPLETE_CREDENTIALS)
{
scRet =
TPCW_E_CLIENHANDSHAKELOOP_SERVER_REQUESTED_CLIENT_AUTHORIZATION;
break;
}
//
// Copy any leftover data from the "extra"buffer, and go around
// again.
//
if ( InBuffers[ 1].BufferType == SECBUFFER_EXTRA )
{
MoveMemory( IoBuffer,
IoBuffer + (cbIoBuffer -
InBuffers[ 1].cbBuffer),
InBuffers[ 1].cbBuffer);
cbIoBuffer = InBuffers[ 1].cbBuffer;
}
else
{
cbIoBuffer = 0;
} // end while
// Delete the security context in the case of a fatal error.
if( FAILED( scRet))
{
(* g_pSecurityFunc).DeleteSecurityContext( phContext);
}
LocalFree( IoBuffer);
return scRet;
} // ClientHandshakeLoop

```

```

/*****
SECURITY_STATUS
PerformClientHandshake(
    SOCKET Socket, // in
    SecurityFunctionTable *g_pSecurityFunc, // in
    HCERTSTORE * phMyCertStore, // in
    SCHANNEL_CRED * pSchannelCred, // in
    PCredHandle phCreds, // in
    LPSTR pszServerName, // in
    CtxtHandle * phContext, // out
    SecBuffer * pExtraData, // out
    BOOL fVerbose) // in
{
    SecBufferDesc OutBuffer;
    SecBuffer OutBuffers[ 1];
    DWORD dwSSPIFlags;
    DWORD dwSSPIOutFlags;
    TimeStamp tsExpiry;
    SECURITY_STATUS scRet;
    DWORD cbData;
    dwSSPIFlags = ISC_REQ_SEQUENCE_DETECT |
        ISC_REQ_REPLAY_DETECT |
        ISC_REQ_CONFIDENTIALITY |
        ISC_RET_EXTENDED_ERROR |
        ISC_REQ_ALLOCATE_MEMORY |
        ISC_REQ_STREAM;

    //
    // Initiate a ClientHello message and generate a token.
    //
    OutBuffers[ 0].pvBuffer = NULL;
    OutBuffers[ 0].BufferType = SECBUFFER_TOKEN;
    OutBuffers[ 0].cbBuffer = 0;
    OutBuffer.cBuffers = 1;
    OutBuffer.pBuffers = OutBuffers;
    OutBuffer.ulVersion = SECBUFFER_VERSION;
    scRet = (* g_pSecurityFunc).InitializeSecurityContextA(
        phCreds,
        NULL,
        pszServerName,
        dwSSPIFlags,
        0,
        SECURITY_NATIVE_DREP,
        NULL,
        0,
        phContext,
        &OutBuffer,
        &dwSSPIOutFlags,
        &tsExpiry);
    if( scRet != SEC_I_CONTINUE_NEEDED)
    {
        return scRet;
    }
}

```

```

// Send response to server if there is one.
if( OutBuffers[ 0].cbBuffer != 0 && OutBuffers[ 0].pvBuffer != NULL)
{
    cbData = send( Socket,
    OutBuffers[ 0].pvBuffer,
    OutBuffers[ 0].cbBuffer,
    0);
    if( cbData == SOCKET_ERROR || cbData == 0)
    {
        (* g_pSecurityFunc).FreeContextBuffer( OutBuffers[ 0].pvBuffer);
        (* g_pSecurityFunc).DeleteSecurityContext( phContext);
        return
        TPCW_E_PERFORMCLIENTHANDSHAKE_ERROR_SENDING_DATA_TO_SERVER_1;
    }
    if( fVerbose) printf("%d bytes of handshake data sent\n",
    cbData);
    // Free output buffer.
    (* g_pSecurityFunc).FreeContextBuffer( OutBuffers[ 0].pvBuffer);
    OutBuffers[ 0].pvBuffer = NULL;
}
return ClientHandshakeLoop( Socket, g_pSecurityFunc,
    phMyCertStore, pSchannelCred, phCreds, phContext, TRUE, pExtraData,
    fVerbose);
}
/*****
*****/
INT
DisconnectFromServer(
    SOCKET Socket,
    SecurityFunctionTable *g_pSecurityFunc,
    PCredHandle phCreds,
    CtxtHandle * phContext,
    BOOL fVerbose
)
{
    DWORD dwType;
    PBYTE pbMessage;
    DWORD cbMessage;
    DWORD cbData;
    SecBufferDesc OutBuffer;
    SecBuffer OutBuffers[ 1];
    DWORD dwSSPIFlags;
    DWORD dwSSPIOutFlags;
   TimeStamp tsExpiry;
    DWORD Status;
    //
    // Notify schannel that we are about to close the connection.
    //
    dwType = SCHANNEL_SHUTDOWN;
    OutBuffers[ 0].pvBuffer = &dwType;
    OutBuffers[ 0].BufferType = SECBUFFER_TOKEN;
    OutBuffers[ 0].cbBuffer = sizeof( dwType);
    OutBuffer.cBuffers = 1;

```

```

OutBuffer.pBuffers = OutBuffers;
OutBuffer.ulVersion = SECBUFFER_VERSION;
Status = (* g_pSecurityFunc).ApplyControlToken( phContext,
    &OutBuffer);
if( FAILED( Status))
{
    goto cleanup;
}
//
// Build an SSL close notify message.
//
dwSSPIFlags = ISC_REQ_SEQUENCE_DETECT |
    ISC_REQ_REPLAY_DETECT |
    ISC_REQ_CONFIDENTIALITY |
    ISC_RET_EXTENDED_ERROR |
    ISC_REQ_ALLOCATE_MEMORY |
    ISC_REQ_STREAM;
OutBuffers[ 0].pvBuffer = NULL;
OutBuffers[ 0].BufferType = SECBUFFER_TOKEN;
OutBuffers[ 0].cbBuffer = 0;
OutBuffer.cBuffers = 1;
OutBuffer.pBuffers = OutBuffers;
OutBuffer.ulVersion = SECBUFFER_VERSION;
Status = (* g_pSecurityFunc).InitializeSecurityContextA(
    phCreds,
    phContext,
    NULL,
    dwSSPIFlags,
    0,
    SECURITY_NATIVE_DREP,
    NULL,
    0,
    phContext,
    &OutBuffer,
    &dwSSPIOutFlags,
    &tsExpiry);
if( FAILED( Status))
{
    goto cleanup;
}
pbMessage = OutBuffers[ 0].pvBuffer;
cbMessage = OutBuffers[ 0].cbBuffer;
//
// Send the close notify message to the server.
//
if( pbMessage != NULL && cbMessage != 0)
{
    cbData = send( Socket, pbMessage, cbMessage, 0);
    if( cbData == SOCKET_ERROR || cbData == 0)
    {
        Status = WSAGetLastError();
        (* g_pSecurityFunc).FreeContextBuffer( pbMessage);
        goto cleanup;
    }
}

```

```

}
// Free output buffer.
(* g_pSecurityFunc).FreeContextBuffer( pbMessage);
}
cleanup:
// Free the security context.
(* g_pSecurityFunc).DeleteSecurityContext( phContext);
// Close the socket.
closesocket( Socket);
return Status;
}
//
// SecureConnect
//
BOOL
SecureConnect(
SecurityFunctionTable *g_pSecurityFunc, // In
HtmlBuffer * pRecvBuffer, // In/ Out: Pointer to receive Server's
Response.
PGE * pts,
// In : Points to PGE struct for the calling thread.
SOCKET_STATE * pss, // In : Points to SOCKET_STATE struct for this Web
Connection.
CtxtHandle * phContext, // Out
BOOL bRenegotiate // Out
)
{
SECURITY_STATUS scRet;
SecBuffer ExtraData;
PCCERT_CONTEXT pRemoteCertContext = NULL;
LONG L;
char StringBuf[ 10]; // Work buf for building strings.
//
// Perform handshake
//
// Prep for handshake
if( bRenegotiate) // Establish a new SSL session using a full SSL
handshake
{
// by generating a new szServerName so that Schannel will not reuse a
past SSL session ID
L=++( pts->ServerNameGenerator);
wsprintf( pts->szServerName,"%s.%s", pts->MyIDString,_ltoa( L,
StringBuf, 10));
}
// else do a handshake that might reuse an established SSL session
// Do the handshake
if(( scRet= PerformClientHandshake( pss->sSocket,
&g_SecurityFunc,
&hMyCertStore,
&SchannelCred,
&hClientCreds,
(LPSTR)& pts->szServerName,
phContext,
&ExtraData,
fVerboseSSL)))
{
return( FALSE);
}
if( ExtraData.pvBuffer || ExtraData.cbBuffer != 0)
{
// If PerformClientHandshake() returns extra data, then it has
LocalAlloc() 'd a buffer to hold that
// data and to which ExtraData.pvBuffer now points. So, after we do
something with the extra data
// we need to LocalFree() that buffer.
// (But right now, we're not doing anything with the extra data.)
if( LocalFree( ExtraData.pvBuffer)!= NULL)
{
exit( 888);
}
}
//
// Authenticate server's credentials.
//
// Get server's certificate.
scRet = g_SecurityFunc.QueryContextAttributes( phContext,
SECPKG_ATTR_REMOTE_CERT_CONTEXT,
(PVOID)& pRemoteCertContext);
if( scRet != SEC_E_OK)
{
return( FALSE);
}
#if 0 // swml01399
// Display server certificate chain.
if( fVerboseSSL)
DisplayCertChain( pRemoteCertContext, FALSE);
#endif // swml01399end
#if 0 // swml01399
//
// Display connection info.
//
if( fVerboseSSL)
DisplayConnectionInfo(& g_SecurityFunc, phContext);
#endif // swml01399end
return( TRUE);
} // end SecureConnect

//
// SecureSendThenRecv
//
SECURITY_STATUS
SecureSendThenRecv(
HtmlBuffer * pSendBuffer, // In : Pointer to request to send to Server.

```

```

HtmlBuffer * pRecvBuffer, // In/ Out: Pointer to receive Server's
Response.
HtmlHeader * pRecvHeader, // In : Pointer to an HtmlHeader into which
// : HTML header info of the Server's response
// : is copied from pRecvBuffer.
SOCKET_STATE * pss, // In : Points to SOCKET_STATE struct for this Web
Connection.
SecurityFunctionTable * g_pSecurityFunc, // In :
HCERTSTORE * phMyCertStore, // In :
SCHANNEL_CRED * pSchannelCred, // In :
PCredHandle phClientCreds, // In :
CtxtHandle * phContext, // In :
SecPkgContext_StreamSizes * pSizes, // In : Points to a Stream Sizes
struct
PGE * pts, // In : Points to thread state of caller
BOOL fVerbose // In :
)
{
SECURITY_STATUS scRet;
SecBufferDesc Message;
SecBuffer Buffers[ 4];
PBYTE pbIoBuffer;
DWORD cbIoBuffer;
DWORD cbIoBufferLength;
PBYTE pbMessage;
DWORD cbMessage;
DWORD cbData;
INT i;
LONG j;
char * pbC; // Work pointer into a buffer
BOOL fEndOfHeaderFound;
LONG HeaderPlusContentLength;
SecBuffer * pDataBuffer;
SecBuffer * pExtraBuffer;
SecBuffer ExtraBuffer;
fEndOfHeaderFound = FALSE;
//
// Read stream encryption properties.
//
scRet = (* g_pSecurityFunc).QueryContextAttributes( phContext,
SECPKG_ATTR_STREAM_SIZES,
pSizes);
if( scRet != SEC_E_OK)
{
return( scRet);
}
//
// Allocate a working buffer. The plaintext sent to EncryptMessage
// should never be more than 'Sizes.cbMaximumMessage', so a buffer
// size of this plus the header and trailer sizes should be safe
// enough.
//

```

```

cbIoBufferLength = pSizes->cbHeader + pSizes->cbMaximumMessage +
pSizes->cbTrailer;
pbIoBuffer = LocalAlloc( LMEM_FIXED, cbIoBufferLength);
if( pbIoBuffer == NULL)
{
return TPCW_E_SECURE_SEND_THEN_RECV_BUFFER_ALLOC_FAILED;
}
// Copy the HTML string to send that's been built in *pSendBuffer into
the recv buffer.
// ( SecureSendThenRecv only use one buffer for send and recv.
// Here we assume the pbIoBuffer will be big enough to hold the send
string.)
// Build the HTTP request offset into the data buffer by "header size"
// bytes. This enables Schannel to perform the encryption in place,
// which is a significant performance win.
strcpy( pbIoBuffer+ pSizes->cbHeader, pSendBuffer->pbBuffer);
pbMessage = pbIoBuffer + pSizes->cbHeader;
cbMessage = strlen( pbMessage);
//
// Encrypt the HTTP request.
//
Buffers[ 0].pvBuffer = pbIoBuffer;
Buffers[ 0].cbBuffer = pSizes->cbHeader;
Buffers[ 0].BufferType = SECBUFFER_STREAM_HEADER;
Buffers[ 1].pvBuffer = pbMessage;
Buffers[ 1].cbBuffer = cbMessage;
Buffers[ 1].BufferType = SECBUFFER_DATA;
Buffers[ 2].pvBuffer = pbMessage + cbMessage;
Buffers[ 2].cbBuffer = pSizes->cbTrailer;
Buffers[ 2].BufferType = SECBUFFER_STREAM_TRAILER;
Buffers[ 3].BufferType = SECBUFFER_EMPTY;
Message.ulVersion = SECBUFFER_VERSION;
Message.cBuffers = 4;
Message.pBuffers = Buffers;

scRet = (* g_pSecurityFunc).EncryptMessage( phContext, 0,
&Message, 0);
if( FAILED( scRet))
{
goto ExitSecureSendThenRecv;
}
//
// Send the encrypted data to the server.
//
cbData = send( pss->sSocket,
pbIoBuffer,
Buffers[ 0].cbBuffer +
Buffers[ 1].cbBuffer + Buffers[ 2].cbBuffer,
0);
if( cbData == SOCKET_ERROR || cbData == 0)
{
(* g_pSecurityFunc).DeleteSecurityContext( phContext);
scRet = SEC_E_INTERNAL_ERROR;
}

```

```

goto ExitSecureSendThenRecv;
}
//
// Read data from server until done.
//
cbIoBuffer = 0;
while( TRUE)
{
//
// Read some data.
//
if( 0 == cbIoBuffer || scRet == SEC_E_INCOMPLETE_MESSAGE)
{
cbData = recv( pss->sSocket,
pbIoBuffer +
cbIoBuffer,
cbIoBufferLength - cbIoBuffer,
0);
if( cbData == SOCKET_ERROR)
{
scRet = SEC_E_INTERNAL_ERROR;
goto ExitSecureSendThenRecv;
}
else if( cbData == 0)
{
// Server disconnected.
if( cbIoBuffer)
{
scRet = SEC_E_INTERNAL_ERROR;
goto ExitSecureSendThenRecv;
}
else
{
scRet = -1; // Send back something non- zero for now
goto ExitSecureSendThenRecv;
}
}
else
{
cbIoBuffer += cbData;
}
//
// Attempt to decrypt the received data.
//
Buffers[ 0].pvBuffer = pbIoBuffer;
Buffers[ 0].cbBuffer = cbIoBuffer;
Buffers[ 0].BufferType = SECBUFFER_DATA;
Buffers[ 1].BufferType = SECBUFFER_EMPTY;
Buffers[ 2].BufferType = SECBUFFER_EMPTY;
Buffers[ 3].BufferType = SECBUFFER_EMPTY;
Message.ulVersion = SECBUFFER_VERSION;
Message.cBuffers = 4;

```

```

Message.pBuffers = Buffers;
scRet = (* g_pSecurityFunc).DecryptMessage( phContext,
&Message, 0, NULL);
if( scRet == SEC_E_INCOMPLETE_MESSAGE)
{
// The input buffer contains only a fragment of an
// encrypted record. Loop around and read some more
// data.
continue;
}
// Server signalled end of session
if( scRet == SEC_I_CONTEXT_EXPIRED)
{
goto ExitSecureSendThenRecv;
}
if( scRet != SEC_E_OK &&
scRet != SEC_I_RENEGOTIATE &&
scRet != SEC_I_CONTEXT_EXPIRED)
{
goto ExitSecureSendThenRecv;
}
// Locate data and (optional) extra buffers.
pDataBuffer = NULL;
pExtraBuffer = NULL;
for( i = 1; i < 4; i++)
{
// In the range Buffer[ 1] to Buffer[ 3], this loop will :
// 1) Find the first buffer with BufferType == SECBUFFER_DATA and point
to it
// via pDataBuffer.
// 2) Find the first buffer with BufferType == SECBUFFER_EXTRA and
point to it
// via pExtraBuffer.
if( pDataBuffer == NULL &&
Buffers[ i].BufferType == SECBUFFER_DATA)
{
pDataBuffer = &Buffers[ i];
}
if( pExtraBuffer == NULL &&
Buffers[ i].BufferType == SECBUFFER_EXTRA)
{
pExtraBuffer = &Buffers[ i];
}
}
// Display or otherwise process the decrypted data.
if( pDataBuffer)
{
if( pRecvBuffer->cbOffset+( LONG) pDataBuffer->cbBuffer > pRecvBuffer-
>cbBuffer - 1) // check for room in RecvBuffer
// The - 1 the last byte of the buffer for
// storing a \0 string delimiter.
{

```

```

pRecvBuffer->cbOffset=- 1;
scRet= TPCW_E_SECURE_SEND_THEN_RECV__NOT_ENOUGH_ROOM_IN_RECVBUFFER;
goto ExitSecureSendThenRecv;
}
MoveMemory( pRecvBuffer->pbBuffer+ pRecvBuffer->cbOffset, pDataBuffer-
>pvBuffer, pDataBuffer->cbBuffer);
pRecvBuffer->cbOffset += pDataBuffer->cbBuffer;
// Now look for end of Header in the received data.
if(! fEndOfHeaderFound && (pbC= strstr(( const
char *) pRecvBuffer->pbBuffer, "\r\n\r\n"))!= NULL)
{
fEndOfHeaderFound = TRUE;
//
// End of Header found - now parse Header in hopes of finding Content-
Length field
//
// Delimit end of header by overwriting 2nd '\n' with '\ 0'.
// ParseHtmlHeader uses strstr to search for components in the incoming
HtmlBuffer.
*( pbC+ strlen("\r\n\r\n")- 1) = '\0';
j= strlen( pRecvBuffer->pbBuffer) + 1;
// Only using j here because I want to reuse below for debugging.

if(( scRet=( SECURITY_STATUS) ParseHtmlHeader( pRecvBuffer->pbBuffer,
pRecvHeader))
{
scRet= TPCW_E_SECURE_SEND_THEN_RECV__PARSE_HTML_HEADER_FAILED;
goto ExitSecureSendThenRecv;
}
if( pRecvHeader->ContentLength== - 1)
{
scRet= TPCW_E_SECURE_SEND_THEN_RECV__CONTENT_LENGTH_EQ_MINUS_1;
goto ExitSecureSendThenRecv;
}
HeaderPlusContentLength =
strlen( pRecvBuffer->pbBuffer) + 1 + pRecvHeader->ContentLength;
strcpy( pbC, "\1\1\1"); // Change the "\r\n\r\n" sequence to "\ 1\ 1\ 1\
0"
// so it doesn't come back as a residual if the caller
// re- uses the pRecvBuffer on on a subsequent call (which
// can trigger a premature and improper strstr find of end of header).
*( pbC+ strlen("\1\1\1") = '\1';
// Finally, replace the \0 with a \1 so that subsequent string searches
// of the receive buffer don't see a premature string delimiter.
} // end if(! fEndOfHeaderFound
} // end if( pDataBuffer)
// Move any "extra" data to the input buffer.
if( pExtraBuffer)
{
// swm- 020999
// Move unprocessed (not decrypted) data to front of pbIoBuffer.
// (and we'll loop to recv() the rest of the data to be decrypted.)

```

```

// Presumably, when there is data in pExtraBuffer, then the
DecryptMessage
// call also returns scRet == SEC_E_INCOMPLETE_MESSAGE, otherwise we
won't
// call recv() again at the top of the loop.
// Note that SEC_E_INCOMPLETE_MESSAGE is tested for above, and if true,
// a continue is executed.
// swm- end
MoveMemory( pbIoBuffer, pExtraBuffer->pvBuffer,
pExtraBuffer->cbBuffer);
cbIoBuffer = pExtraBuffer->cbBuffer;
}
else
{
cbIoBuffer = 0;
} // end if( pExtraBuffer)
if( scRet == SEC_I_RENEGOTIATE)
{
// The server wants to perform another handshake
// sequence.
scRet = ClientHandshakeLoop( pss->sSocket,
g_pSecurityFunc,
phMyCertStore,
pSchannelCred,
phClientCreds,
phContext,
FALSE,
&ExtraBuffer,
fVerbose);

if( scRet != SEC_E_OK)
{
goto ExitSecureSendThenRecv;
}
// Move any "extra" data to the input buffer.
if( ExtraBuffer.pvBuffer)
{
MoveMemory( pbIoBuffer,
ExtraBuffer.pvBuffer, ExtraBuffer.cbBuffer);
cbIoBuffer = ExtraBuffer.cbBuffer;
}
} // end if( scRet == SEC_I_RENEGOTIATE)
if( fEndOfHeaderFound)
{
if( pRecvBuffer->cbOffset >
HeaderPlusContentLength)
{
scRet= 3;
goto ExitSecureSendThenRecv;
}
if( pRecvBuffer->cbOffset ==
HeaderPlusContentLength) break; // We've recd all the data we're
expecting

```

```

}
} // end while( TRUE)
*( pRecvBuffer->pbBuffer + pRecvBuffer->cbOffset) = '\0'; // Delimit
the received cleartext msg.
if( LocalFree( pbIoBuffer)!= NULL)
{
exit( 666);
}
return( SEC_E_OK);
ExitSecureSendThenRecv:
if( LocalFree( pbIoBuffer)!= NULL)
{
exit( 777);
}
return( scRet);
} // end SecureSendThenRecv

//
//*****
//*****
// SecureSendThenRecvHtml
//
//*****
//*****
HRESULT
SecureSendThenRecvHtml(
    HtmlBuffer * pSendBuffer, // In : Buffer to send to the Server
    HtmlBuffer * pRecvBuffer, // In/ Out: Pointer to receive
    Server's Response.
    HtmlHeader * pRecvHeader, // In : Pointer to an HtmlHeader into
which
    // : HTML header info of the Server's response
    // : is copied from pRecvBuffer.
    SOCKET_STATE * pss, // In : Points to SOCKET_STATE struct for
secure Web Connection being used
    PGE * pts, // In :
    SecurityFunctionTable * g_pSecurityFunc, // In :
    CredHandle * phClientCreds, // In : Points to Security
Credentials
    CtxtHandle * phContext, // In : Points to Context handle
    SecPkgContext_StreamSizes * pSizes // In : Points to a Stream
Sizes struct
)
// Global inputs :
// g_SecurityFunc,
// hMyCertStore,
// SchannelCred
{
    SECURITY_STATUS scRet;
    INT SecXferRetryCount;
    INT SecConnRetryCount;

```

```

INT i;
SecXferRetryCount = 0;
while( SecXferRetryCount< MAX_SECURE_RETRIES)
{
    //
    // See if we need to establish a secure connection first
    //
    if(! pss->ConnMade // ConnMade flag indicates that a
secure connection is open
        || pss->bRenegotiateConnection //
bRenegotiateConnection flag indicates it is desired that a
        // new full handshake be performed
between the caller and the server.
    )
    {
        for( SecConnRetryCount= 0; SecConnRetryCount<
MAX_SSL_HANDSHAKE_RETRIES; SecConnRetryCount++)
        {
            i= OpenNewSocketWithRetries( pss,
MAX_OPEN_SOCKET_RETRIES);
            if( i< MAX_OPEN_SOCKET_RETRIES)
            { // Opened socket successfully
                pss->ConnMade= TRUE;
            }
            else
            { // Open of new socket failed
                pss->ConnMade= FALSE;
                scRet=
TPCW_E_SECURE_SEND_THEN_RECV_HTML__SOCKIOINIT_FAILED;
                goto SSTRHfail; // this will then
try to DeleteSecurityContext( phContext) even if one doesn't exist
            }
            // Try to make the secure connection.
            if( pss->bRenegotiateConnection)
            {
                if(! SecureConnect(
g_pSecurityFunc, pRecvBuffer, pts, pss, phContext, TRUE))
                {
                    closesocket( pss-
>sSocket); // Might already be closed - but make sure
                    pss->sSocket = 0;
                    pss->ConnMade= FALSE;
                }
                else
                {
                    pss->ConnMade= TRUE; //
Note: this might already have been TRUE.
                    pss-
>bRenegotiateConnection= FALSE; // Clear the request to renegotiate
                }
            }
            else // !pss->bRenegotiateConnection
            {

```

```

        if(! SecureConnect(
g_pSecurityFunc, pRecvBuffer, pts, pss, phContext, FALSE))
        {
            closesocket( pss->sSocket); // Might already be closed - but make sure
            pss->sSocket = 0;
            pss->ConnMade= FALSE;
        }
        else
        {
            pss->ConnMade= TRUE;
        }
    }
    if( pss->ConnMade) break;
} // for( SecConnRetryCount= 0,...
if( SecConnRetryCount==
MAX_SSL_HANDSHAKE_RETRIES)
{
    scRet=
TPCW_E_SECURE_SEND_THEN_RECV_HTML__MAX_SSL_HANDSHAKE_RETRIES_FAILED;
    goto SSTRHfail;
} // end if(! pss->ConnMade) || pss->bRenegotiateConnection
//
// Try to securely send the HTML request and securely
receive the HTML response
//
scRet= SecureSendThenRecv( pSendBuffer,
    pRecvBuffer,
    pRecvHeader,
    pss,
    &g_SecurityFunc,
    &hMyCertStore,
    &SchannelCred,
    &hClientCreds,
    phContext,
    pSizes,
    pts,
    fVerboseSSL);
if( scRet== SEC_E_OK)
{
    break;
}
else
{
    switch( scRet)
    {
    case
TPCW_E_SECURE_SEND_THEN_RECV__PARSE_HTML_HEADER_FAILED:
    case
TPCW_E_SECURE_SEND_THEN_RECV__CONTENT_LENGTH_EQ_MINUS_1:

```

```

// If get these errors, we bail without any further
retries
    scRet=
TPCW_E_SECURE_SEND_THEN_RECV_HTML__FATAL_ERR_FROM_SECURE_SEND_THEN_RECV
;
    goto SSTRHfail;
    break;
    default:
    // Check for bad HTTP status codes
    if( pRecvHeader->RespStatusCode != 200
    )
    {
    // If get these errors, we bail without any further
retries
    scRet=
TPCW_E_SECURE_SEND_THEN_RECV_HTML__HTTP_STATUS_NOT_200;
    goto SSTRHfail;
    }
    else
    {
    DisconnectFromServer( pss->sSocket, &g_SecurityFunc, &
hClientCreds, &pts->hContext, FALSE);
    pss->ConnMade= FALSE;
    pss->bRenegotiateConnection= TRUE; // establish a new
SSL session using a full handshake
    SecXferRetryCount++;
    #ifdef SCREEN_INFO
    EnterCriticalSection(&sslretry);
    numsslretry++;
    LeaveCriticalSection(&sslretry);
    #endif
    } // if( pRecvHeader->RespStatusCode !=200 )
    } // switch( scRet)
    } // if( scRet== SEC_E_OK)
} // while( SecXferRetryCount< MAX_SECURE_RETRIES)
if( SecXferRetryCount== MAX_SECURE_RETRIES)
return(
TPCW_E_SECURE_SEND_THEN_RECV_HTML__MAX_SECURE_RETRIES_FAILED);
else
return( TPCW_E_OK);
SSTRHfail:
(* g_pSecurityFunc).DeleteSecurityContext( phContext);
// phContext is established by SecureConnect()
return( scRet);
} // SecureSendThenRecvHtml

```

#### ssl.h

```

// ssl.h
//
// Copyright Unisys, 1999,2000
#include <windows.h>
#include "stdio.h"

```

```

#include "stdlib.h"
#include "windows.h"
#include "winsock.h"
#include "wincrypt.h"
#include "wintrust.h"
#include "schannel.h"
#define SECURITY_WIN32
#include "security.h"
#include "sspi.h"
#ifdef PGEHost
#include "pge.h"
#endif
#define SEC_I_CONTEXT_EXPIRED ((HRESULT)0x00090317L)
#define IO_BUFFER_SIZE 0x10000
BOOL SockioInit( SOCKET_STATE * pss);

HRESULT ParseHtmlHeader(
    char * pBuffer, // In : The HTML Header string (// 0delimited) to
    parse.
    HtmlHeader * pHtmlHeader // In : Points to HtmlHeader to recv the HTML
    header
    // : components found.
);

SECURITY_STATUS
CreateCredentials(
    SecurityFunctionTable * g_pSecurityFunc, // in
    HCERTSTORE * phMyCertStore, // in
    LPSTR pszUserName, // in
    SCHANNEL_CRED * pSchannelCred, // in
    DWORD dwProtocol, // in
    ALG_ID aiKeyExch,
    // in
    PCredHandle phCreds
);

BOOL LoadSecurityLibrary( void);

SECURITY_STATUS ClientHandshakeLoop(
    SOCKET Socket, // in
    SecurityFunctionTable *g_pSecurityFunc, // in
    HCERTSTORE * phMyCertStore, // in
    SCHANNEL_CRED * pSchannelCred, // in
    PCredHandle phCreds, // in
    CtxtHandle * phContext, // in, out
    BOOL fDoInitialRead, // in
    SecBuffer * pExtraData, // out
    BOOL fVerbose); // in

SECURITY_STATUS PerformClientHandshake(
    SOCKET Socket, // in
    SecurityFunctionTable *g_pSecurityFunc, // in
    HCERTSTORE * phMyCertStore, // in
    SCHANNEL_CRED * pSchannelCred, // in
    PCredHandle phCreds, // in
    CtxtHandle * phContext, // in
    SecPkgContext_StreamSizes * pSizes, // In : Points to a Stream
    Sizes struct
);

INT DisconnectFromServer(
    SOCKET Socket,
    SecurityFunctionTable *g_pSecurityFunc,
    PCredHandle phCreds,
    CtxtHandle * phContext,
    BOOL fVerbose
);

BOOL
SecureConnect(
    SecurityFunctionTable *g_pSecurityFunc, // In
    HtmlBuffer * pRecvBuffer, // In/ Out: Pointer to receive
    Server's Response.
    PGE * pts,
    // In : Points to PGE struct for the callingthread.
    SOCKET_STATE * pss, // In : Points to SOCKET_STATE struct for
    this Web Connection.
    CtxtHandle * phContext, // Out
    BOOL bRenegotiate
    // Out
);

SECURITY_STATUS
SecureSendThenRecv(
    HtmlBuffer * pSendBuffer, // In :Pointer to request to send to
    Server.
    HtmlBuffer * pRecvBuffer, // In/ Out:Pointer to receive
    Server's Response.
    HtmlHeader * pRecvHeader, // In :Pointer to an HtmlHeader into
    which
    // : HTML
    header info of theServer's response
    // : is
    copied from pRecvBuffer.
    SOCKET_STATE * pss, // In : Points to
    SOCKET_STATE struct for this WebConnection.
    SecurityFunctionTable * g_pSecurityFunc, // In :
    HCERTSTORE * phMyCertStore, // In :
    SCHANNEL_CRED * pSchannelCred, // In :
    PCredHandle phClientCreds, // In :
    CtxtHandle * phContext, // In :
    SecPkgContext_StreamSizes * pSizes, // In : Points to a Stream
    Sizes struct
);

```

```

    PGE * pts, // In :
Points to thread state of caller
    BOOL fVerbose // In :
);
HRESULT
SecureSendThenRecvHtml(
    HtmlBuffer * pSendBuffer, // In : Buffer to send to theServer
    HtmlBuffer * pRecvBuffer, // In/ Out: Pointer to receive
Server's Response.
    HtmlHeader * pRecvHeader, // In : Pointer to an HtmlHeader into
which
// : HTML header info of the Server's
response
// : is copied from pRecvBuffer.
    SOCKET_STATE * pss, // In : Points to SOCKET_STATE struct for
secure Web Connection being used
    PGE * pts, // In : Points to PGE struct of caller
    SecurityFunctionTable * g_pSecurityFunc, // In :
    CredHandle * phClientCreds, // In : Points to Security
Credentials
    CtxtHandle * phContext, // In : Points to Context handle
    SecPkgContext_StreamSizes * pSizes // In : Points to a Stream
Sizes struct
);

```

#### strings.c

```

// strings.c
//
// Copyright Unisys, 1999,2000
#include <windows.h>
#include <time.h>
#include "strings.h"

BOOL DigSyl( __int64 lNum, char *pszOut)
{
    char *pszI, *pszO;
    char szDecRep[ 10];
    long lLenDecRep, I;
    if( lNum< 1)
    {
        return( FALSE);
    }
    lLenDecRep= strlen(_i64toa( lNum, szDecRep, 10));
    pszO= pszOut;
    for( I= 0, pszI= szDecRep; I< lLenDecRep; I++, pszI++)
    {
        switch(* pszI)
        {
            case '0':
                *pszO= 'B'; pszO++;* pszO= 'A'; pszO++;
                break;
            case '1':

```

```

*pszO= 'O'; pszO++;* pszO= 'G'; pszO++;
break;
case '2':
*pszO= 'A'; pszO++;* pszO= 'L'; pszO++;
break;
case '3':
*pszO= 'R'; pszO++;* pszO= 'I'; pszO++;
break;
case '4':
*pszO= 'R'; pszO++;* pszO= 'E'; pszO++;
break;
case '5':
*pszO= 'S'; pszO++;* pszO= 'E'; pszO++;
break;
case '6':
*pszO= 'A'; pszO++;* pszO= 'T'; pszO++;
break;
case '7':
*pszO= 'U'; pszO++;* pszO= 'L'; pszO++;
break;
case '8':
*pszO= 'I'; pszO++;* pszO= 'N'; pszO++;
break;
case '9':
*pszO= 'N'; pszO++;* pszO= 'G'; pszO++;
break;
default:
return( FALSE);

```

```

    }
}
*pszO= '\0';
return( TRUE);
}

int MakeAlphaString( int x, int y, int z, char *str, long * lTLSeed)
{
    int len;
    int i;
    static char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
};
    static int chArrayMax = 61;
    len=( int) RandomNumber( x, y, lTLSeed);
    for( i= 0; i< len; i++)
    {
        str[ i] = chArray[( int) RandomNumber( 0,
            chArrayMax, lTLSeed)];
    }
    if ( len < z )
    {
        memset( str+ len, ' ', z - len);
    }
}

```

```

    str[ len] = 0;
    return len;
}

long irand( long * Seed)
{
    register long s; /* copy of seed */
    register long test; /* test flag */
    register long hi; /* tmp value for speed */
    register long lo; /* tmp value for speed */
    s = *Seed;
    hi = s / Q;
    lo = s % Q;
    test = A * lo - R * hi;
    if ( test > 0 )
    {
        *Seed = test;
    }
    else
    {
        *Seed = test + M;
    }
    return( s );
}

long RandomNumber( long lower, long upper, long * Seed)
{
    long rand_num;
    if ( upper == lower ) /* pgd 08- 13- 96 perf enhancement */
        return lower;
    upper++;
    if ( upper <= lower )
    {
        rand_num = upper;
    }
    else
    {
        rand_num = lower + irand( Seed) % (upper - lower); /*
        pgd 08- 13- 96 perf enhancement */
    }
    return rand_num;
}

long InitRnd()
{
    long seed= time( NULL);
    if (seed < 0 )
    {
        seed = abs( seed);
    }
}

```

```

        return seed;
};

```

#### strings.h

```

// strings. h
//
// Copyright Unisys, 1999,2000
// Defines
#define A 16807
#define M 2147483647
#define Q 127773 /* M div A */
#define R 2836 /* M mod A */
#define Thread __declspec( thread)

BOOL DigSyl(__int64 lNum, char *pszOut);
int MakeAlphaString( int x, int y, int z, char *str, long * lTLSeed);
long irand(long * Seed);
long RandomNumber(long lower, long upper, long * Seed);
long InitRnd();

```

#### thread.c

```

// thread.c
//
// Copyright Unisys, 1998,1999,2000
#define _WIN32_WINNT 0x0500
#define SessionSPINCOUNT 200
#define BasketSPINCOUNT 200
#define MAX_IDSPINCOUNT 200
#include <math.h>
#include "tpcw.h"
#include "tpcw_db.h"
#include "thread.h"
#include "strings.h"
#include "display.h"
#include "readreg.h"
#include "pagesize.h"
#include "htmltext.h"

extern char hostname[ 20];
extern int hostnumber;
extern HANDLE BasketHeap;
extern CRITICAL_SECTION csO_ID;
extern CRITICAL_SECTION csC_ID;
extern CRITICAL_SECTION csADDR_ID;
extern CRITICAL_SECTION csSession_ID;
extern long O_ID_ALLOC;
extern long C_ID_ALLOC;
extern long ADDR_ID_ALLOC;
extern long Session_ID_ALLOC;
extern long Session_ID;

```

```

extern long LimitSession_ID;

#ifdef SCREEN_INFO
extern long numshop;
extern long numthread;
extern long numbasketkill;
extern long numpgreq;
extern long numsslretry;
extern long numwebaction[ 15];
extern long totalwebaction[15];
extern CRITICAL_SECTION webaction[ 15];
extern CRITICAL_SECTION shop;
extern CRITICAL_SECTION thread;
extern CRITICAL_SECTION basketkill;
extern CRITICAL_SECTION pgreq;
extern CRITICAL_SECTION sslretry;
#endif

long num_to_and;
long maxitem;
BOOL DayLightSaving;
CRITICAL_SECTION g_csBasket[ AR_SIZE];
struct BASKET_LIST * Basket_List[ AR_SIZE];

#define WAITSLEEP 300000
#define EXPIRE 7200

```

```

VOID WINAPI DeleteBasket (LPDWORD lpParams)
{
    struct BASKET_LIST * p;
    struct BASKET_LIST * t;
    struct BASKET_ITEM * q;
    struct BASKET_ITEM * d;
    time_t currenttime;
    int rc;
    int i;
    BOOL reconnect;
    TPCW clean;
    HANDLE ThreadID;
    ThreadID= GetCurrentThread();
    if( SetThreadPriority( ThreadID, THREAD_PRIORITY_HIGHEST)== 0)
    {
        ErrorLog(" SetThreadPriority Error"," ThreadAttach()");
    }
    strcpy( clean.host, hostname);
    clean.host_len= strlen( clean.host);
    clean.querystring[ 0]= '\0';
    clean.retcode= SQLInit(& clean);
    if( clean.retcode== SQL_ERROR|| clean.retcode==
SQL_INVALID_HANDLE)
    {
        reconnect= TRUE;
    }
}

```

```

    }
    reconnect= FALSE;
    while( TRUE)
    {
        for( i= 0; i< AR_SIZE; i++)
        {
            if( reconnect== TRUE)
            {
                rc= SQLInit(& clean);
                if( rc== FALSE)
                {
                    ErrorLog(" Reconnect
UnSuccessful"," DeleteBasket");
                    goto next_run;
                }
            }
            else
            {
                reconnect= FALSE;
                ErrorLog(" Reconnect
Successful"," DeleteBasket");
            }
        }
        __try
        {
            EnterCriticalSection(& g_csBasket[ i]);
            p= Basket_List[ i]->next;
            while( p!= Basket_List[ i]&& p!= NULL)
            {
                time(& currenttime);
                currenttime= currenttime- EXPIRE;
                if( currenttime > p->sc_date &&
p->Item!= NULL)
                {
                    if( p->Item->next!= NULL)
                    {
                        clean.ID.SessionID= p->sc_shopping_id;
                        q= p->Item->next;
                        while( q!= p-
>Item)
                        {
                            q->prev=
                            q->next=
                            q->next=
                            d= q->prev;
                            rc=
                            if( rc== 0)
                            {
                                ErrorLog(" HeapFree Error"," DeleteBasket");
                            }
                        }
                    }
                }
            }
        }
        __except( EXCEPTION_EXECUTE_HANDLER)
        {
            reconnect= TRUE;
        }
    }
}

```



```

        if( rc== 0)
        {
            ErrorLog(" HeapFree Error"," DeallocBasket");
            return FALSE;
        }
    }
    return TRUE;
}

BOOL Recovery( TPCW * rec)
{
    int rc;
    rec->lTlSeed= InitRnd();
    if( rec->lTlSeed< 0)
    {
        return FALSE;
    }
    rc= SQLGetBasketItem( rec, Basket_List);
    if( rc== FALSE)
    {
        return FALSE;
    }
    return TRUE;
}

BOOL DayLightSavingCheck()
{
    time_t now;
    struct tm now2;
    time(& now);
    now2=* localtime(& now);
    DayLightSaving= now2.tm_isdst;
    return TRUE;
}

BOOL Initialize()
{
    BOOL rc= TRUE;
    int ret;

    int i;
    DWORD dwThreadId;
    TPCW init;
    Screen_Info_Init();
    DayLightSavingCheck();
    rc= ReadHostName();
    if( rc== FALSE)
    {
        ErrorLog(" ReadHostName Error"," Initialize()");
        return FALSE;
    }
}

```

```

    }
    rc= ReadCacheHost();
    if( rc== FALSE)
    {
        ErrorLog(" ReadCacheHost Error"," Initialize()");
        return FALSE;
    }
    rc= ReadDomainName();
    if( rc== FALSE)
    {
        ErrorLog(" ReadDomainName Error"," Initialize()");
        return FALSE;
    }
    FillHTML();
    strcpy( init.host, hostname);

    init.host_len= strlen( init.host);
    init.retcode= SQLInit(& init);
    if( init.retcode== FALSE)
    {
        return FALSE;
    }
    rc= ReadMaxItem(& init);
    if( rc== FALSE)
    {
        ErrorLog(" ReadMaxItem Error"," Initialize()");
        return FALSE;
    }
    rc= ReadMaxO_ID(& init, O_ID_ALLOC);
    if( rc== 0)
    {
        ErrorLog(" ReadMaxO_ID Error"," Initialize()");
        return FALSE;
    }
    rc= ReadMaxC_ID(& init, C_ID_ALLOC);
    if( rc== 0)
    {
        ErrorLog(" ReadMaxC_ID Error"," Initialize()");
        return FALSE;
    }
    rc= ReadMaxADDR_ID(& init, ADDR_ID_ALLOC);
    if( rc== 0)
    {
        ErrorLog(" ReadMaxADDR_ID Error"," Initialize()");
        return FALSE;
    }
    rc= ReadMaxSession_ID(& init, Session_ID_ALLOC);
    if( rc== 0)
    {
        ErrorLog(" ReadMaxSession_ID Error"," Initialize()");
        return FALSE;
    }
    InitializeCriticalSectionAndSpinCount( &csSession_ID ,

```

```

        SessionSPINCOUNT);
InitializeCriticalSectionAndSpinCount( &csO_ID ,
MAX_IDSPINCOUNT);
InitializeCriticalSectionAndSpinCount( &csC_ID ,
MAX_IDSPINCOUNT);
InitializeCriticalSectionAndSpinCount( &csADDR_ID ,
MAX_IDSPINCOUNT);
num_to_and=( long) pow( 2, DIG_TO_MATCH)- 1;
for( i= 0; i< AR_SIZE; i++)
{
    rc= InitializeCriticalSectionAndSpinCount(
        &g_csBasket[ i] , BasketSPINCOUNT);
    if( rc== 0)
    {
        ErrorLog(" InitializeCriticalSectionAndSpinCount
Error", " Initialize()");
        return FALSE;
    }
    Basket_List[ i]= HeapAlloc( BasketHeap,
HEAP_ZERO_MEMORY,
sizeof( struct BASKET_LIST));
    if( Basket_List[ i]== NULL)
    {
        ErrorLog(" HeapAlloc BasketHeap return NULL", "
Initialize()");
        return FALSE;
    }
    Basket_List[ i]->sc_shopping_id=- 1;
    Basket_List[ i]->sc_c_id=- 1L;
    Basket_List[ i]->Item= NULL;
    strcpy( Basket_List[ i]->sc_c_fname, " array header");
    Basket_List[ i]->next= Basket_List[ i];
    Basket_List[ i]->prev= Basket_List[ i];
}
rc= Recovery(& init);
if( rc== FALSE)
{
    ErrorLog(" Recovery Error", " Initialize()");
    return FALSE;
}
SQLFreeHandle( SQL_HANDLE_STMT, init.hstmt);
SQLDisconnect( init.hdbc);
SQLFreeHandle( SQL_HANDLE_DBC, init.hdbc);
ret=_beginthreadex( NULL, 0, ( LPTHREAD_START_ROUTINE)
DeleteBasket, NULL, 0, & dwThreadId );
if( ret== - 1)
{
    ErrorLog(" Cannot start DeleteBasket thread", "
Initialize()");
    return FALSE;
}
#endif
#ifdef PERFMON_ENABLE
ret=_beginthreadex( NULL, 0, ( LPTHREAD_START_ROUTINE)

```

```

        LoopSetShareMem, NULL, 0, & dwThreadId );
if( ret== - 1)
{
    ErrorLog(" Cannot start LoopSetShareMem
thread", " Initialize()");
    return FALSE;
}
#endif
return TRUE;
}

```

```

void Terminate()
{
int i;
Screen_Info_Term();
DeleteCriticalSection( &csSession_ID );
DeleteCriticalSection( &csO_ID );
DeleteCriticalSection( &csC_ID );
DeleteCriticalSection( &csADDR_ID );
for( i= 0; i< AR_SIZE; i++)
{
DeleteCriticalSection( &g_csBasket[ i] );
}
}

```

```

BOOL ParseCookie( char param[], char * querystring)
{
    int i= 0;
    char buffer[ 128];
    buffer[ 0]= '\0';
    querystring = strchr( querystring, '=')+ 1;

    while(*querystring!= ',' && *querystring!= ';' &&
*querystring!= '\0')
    {
        buffer[ i]=* querystring;
        querystring++;
        i++;
    }
    buffer[ i]= '\0';
    strcpy( param, buffer);
    return TRUE;
}

```

```

BOOL CookieCheck( TPCW * tpcw)
{
    BOOL GotSession;
    DWORD dwSize;
    char * cptrl;
    char * cptr2;
    char * cptr3;
}

```

```

char c_id[ 10];
char s_id[ 10];
char w_id[ 10];
int rc;

dwSize= LEN_BUFFER;
GotSession= tpcw->pcb->GetServerVariable(tpcw->pcb->ConnID,"HTTP_COOKIE", tpcw->szBuffer, &dwSize);
if( GotSession== FALSE)
{
    tpcw->ID.C_ID= 0;
    __try
    {
        EnterCriticalSection(&csSession_ID);
        tpcw->ID.SessionID= Session_ID;
        Session_ID++;
        if( Session_ID == LimitSession_ID)
        {
            rc= ReadMaxSession_ID( tpcw,
Session_ID_ALLOC);
            if( rc== 0)
            {
                strcpy( tpcw->err,"CookieCheck:
rc= ReadMaxSession_ID( tpcw, Session_ID_ALLOC)== 0");
                ErrorMessage( tpcw);
                return FALSE;
            }
            else
            {
                tpcw->ID.SessionID= Session_ID;
                Session_ID++;
            }
        }
    }
    __finally
    {
        LeaveCriticalSection(& csSession_ID);
    }
}
else
{
    cptr1= strstr( tpcw->szBuffer,"C_ID");
    cptr2= strstr( tpcw->szBuffer,"Session_ID");
    cptr3= strstr( tpcw->szBuffer,"Web_ID");
    if( cptr1!= NULL)
    {
        ParseCookie( c_id, cptr1);
        tpcw->ID.C_ID= _atoi64( c_id); //
    }
    else
    {
        tpcw->ID.C_ID= 0;
    }
}

```

```

if( cptr2!= NULL)
{
    ParseCookie( s_id, cptr2);
    tpcw->ID.SessionID= atol( s_id);
}
else
{
    __try
    {
        EnterCriticalSection(&csSession_ID);
        tpcw->ID.SessionID= Session_ID;
        Session_ID++;
        if( Session_ID == LimitSession_ID)
        {
            rc= ReadMaxSession_ID( tpcw,
Session_ID_ALLOC);
            if( rc== 0)
            {
                strcpy( tpcw->err,"
CookieCheck: rc= ReadMaxSession_ID( tpcw, Session_ID_ALLOC)== 0");
                ErrorMessage( tpcw);
                return FALSE;
            }
            else
            {
                tpcw->ID.SessionID=
Session_ID;
                Session_ID++;
            }
        }
    }
    __finally
    {
        LeaveCriticalSection(&csSession_ID);
    }
}
if( cptr3!= NULL)
{
    ParseCookie( w_id, cptr3);
    tpcw->hostnum = atoi( w_id); //
}
else
{
    tpcw->hostnum = 0;
}
}
return TRUE;
}

```

```

void AddSpace( TPCW * tpcw, char interaction)
{
    long diff= 0;
    tpcw->len= tpcw->endptr- tpcw->szContent;
    switch( interaction)
    {
        case 'a':
        case 0:
            diff= HOMESIZE- tpcw->len;
            break;
        case 'b':
        case 'o':
            diff= SHOPSIZE- tpcw->len;
            break;
        case 'c':
            diff= CUSTREGSIZE- tpcw->len;
            break;
        case 'd':
            diff= BUYREQSIZE- tpcw->len;
            break;
        case 'e':
            diff= BUYCONFSIZE- tpcw->len;
            break;
        case 'f':
            diff= ORDERINQ- tpcw->len;
            break;
        case 'g':
            diff= ORDERDISP- tpcw->len;
            break;
        case 'h':
            diff= SEARCHREQ- tpcw->len;
            break;
        case 'i':
            diff= SEARCHRES- tpcw->len;
            break;
        case 'l':
        case 'z':
            diff= PRODDT- tpcw->len;
            break;
        case 'm':
            diff= ADMINREQ- tpcw->len;
            break;
        case 'n':
            diff= ADMINCONF- tpcw->len;
            break;
        default:
            break;
    }
    if( diff > 0)
    {
        FillMemory( tpcw->endptr, diff, ' ');
        tpcw->endptr= tpcw->endptr+ diff;
    }
}

```

```

}
*tpcw->endptr= '\0';

BOOL Cookie( TPCW * tpcw)
{
    BOOL ret= FALSE;
    ret= CookieCheck( tpcw);
    if( ret== FALSE)
    {
        return FALSE;
    }
    if( tpcw->ID.C_ID > 999999999L || tpcw->ID.C_ID < 0)
    {
        sprintf( tpcw->err, " Process: Problem with C_ID: %I64d",
        tpcw->ID.C_ID);
        ErrorMessage( tpcw);
        return FALSE;
    }
    if( tpcw->ID.SessionID > 999999999 || tpcw->ID.SessionID < 1)
    {
        wsprintf( tpcw->err, " Process: Problem with SessionID:
%d", tpcw->ID.SessionID);
        ErrorMessage( tpcw);
        return FALSE;
    }
    tpcw->hash= tpcw->ID.SessionID & num_to_and;
    if( tpcw->hash < 0 || tpcw->hash >= AR_SIZE)
    {
        wsprintf( tpcw->err, " Process: Problem with hash: %d",
        tpcw->hash);
        ErrorMessage( tpcw);
        return FALSE;
    }
    return TRUE;
}

void EnterInteraction( char interaction)
{
    #ifdef SCREEN_INFO
    long num;
    num= interaction - '0' - 49;
    EnterCriticalSection(& webaction[num]);
    numwebaction[ num]++;
    totalwebaction[num]++;
    LeaveCriticalSection(& webaction[num]);
    #endif
}

void LeaveInteraction( char interaction)
{
    #ifdef SCREEN_INFO
    long num;

```

```

num= interaction- '0'- 49;
EnterCriticalSection(& webaction[ num]);
numwebaction[ num]--;
LeaveCriticalSection(& webaction[ num]);
#endif
}

BOOL process( TPCW * tpcw)
{
    char * cptr1;
    char interaction;
    BOOL ret= FALSE;

    tpcw-> expiresflag= NOW;
    memcpy( tpcw->szHeader, HEADER, HEADERLEN);
    tpcw->endptr= tpcw->szContent;
    tpcw->szContent[ 0]= '\0';
    strcpy( tpcw->querystring, tpcw->pecb->lpszQueryString);
    interaction= tpcw->querystring[0];

    tpcw->ID.C_ID= 0;
    tpcw->ID.SessionID= 0;

    /*=====
=====
-- Code changed on 11/ 10/ 00 to meet clause 2.8.3.1 of 1.2
spec
-- This portion of codes added on 11/ 10/ 00 to meet clause
2.8.3.1 of
1.2 spec
-- It was not used during the measurement intervals of the 11/
6 thru
11/ 10 benchmark but tests were run 11/ 08/ 00 to
demonstrate to the auditor that it had no significant impact on
performance.
-- See auditor's attestation letter for details.
Before code changed
=====
=====
if( interaction== 'a' || interaction== 0 || // home
interaction== 'o' || interaction== 'b' || // shopping cart
interaction== 'e' || // buy confirm
interaction== 'd' // buy request
)
=====
=====
*/
// After code changed

```

```

//=====
=====
//=====
if( interaction== 'a' || interaction== 0 ||
    interaction== 'o' || interaction== 'b' ||
    interaction== 'e' ||
    interaction== 'd' ||
    interaction== 'f'
    || interaction == 'v'
)
//=====
//=====
{
    ret= Cookie( tpcw);
    if( ret== FALSE)
    {
        sprintf(tpcw->szContent, " Missing cookie on
query %s", tpcw->querystring);
        ErrorMessage(tpcw);
        return FALSE;
    }
}

if( interaction== 'a' || interaction== 0)
{
    EnterInteraction( 'a');
    ret= Home( tpcw, Basket_List);
    LeaveInteraction( 'a');
}
else if( interaction== 'v') //
{
    EnterInteraction( 'a');
    ret= HomeG( tpcw, Basket_List);
    LeaveInteraction( 'a');
}

else if (interaction== 'r') //
{
    EnterInteraction( 'j'); //
    ret= PromoForListing( tpcw);
    LeaveInteraction( 'j');
}
else if (interaction== 'l' || interaction== 'z') //
{
    EnterInteraction( 'l');
    ret= BookInfo( tpcw);
    LeaveInteraction( 'l');
}
else if (interaction== 'o') //
{
    EnterInteraction( interaction);
}

```

```

        if (tpcw->pecb->cbAvailable > 0) //
        {
            memcpy( tpcw->querystring, ( LPCTSTR) tpcw->pecb-
>lpbData, tpcw->pecb->cbAvailable);
            tpcw->querystring[ tpcw->pecb->cbAvailable]=
'\0';
        }
        strcpy( tpcw->querystring, strstr( tpcw-
>querystring, "&")+ 1);

        ret= Adjustment( tpcw, Basket_List);
        LeaveInteraction( interaction);
    }
    else if (interaction== 'c') //
    {
        EnterInteraction( interaction);
        ret= CustomerReg( tpcw);
        LeaveInteraction( interaction);
    }
    else if (interaction== 'b') //
    {
        EnterInteraction( interaction);
        strcpy( tpcw->querystring, strstr( tpcw-
>querystring, "&")+ 1);
        ret= ShoppingBasket( tpcw, Basket_List);
        LeaveInteraction( interaction);
    }
    else if (interaction== 'd') //
    {
        EnterInteraction( interaction);

        if (tpcw->pecb->cbAvailable > 0) //
        {
            memcpy( tpcw->querystring, ( LPCTSTR) tpcw->pecb-
>lpbData, tpcw->pecb->cbAvailable);
            tpcw->querystring[ tpcw->pecb->cbAvailable]=
'\0';
        }
        strcpy( tpcw->querystring, strstr( tpcw-
>querystring, "&")+ 1);

        ret= UserType( tpcw, Basket_List);
        LeaveInteraction( interaction);
    }
    else if (interaction== 'e') //
    {
        EnterInteraction( interaction);

        if (tpcw->pecb->cbAvailable > 0) //
        {
            memcpy( tpcw->querystring, ( LPCTSTR) tpcw->pecb-
>lpbData, tpcw->pecb->cbAvailable);

```

```

        tpcw->querystring[ tpcw->pecb->cbAvailable]=
'\0';
    }
    strcpy( tpcw->querystring, strstr( tpcw-
>querystring, "&")+ 1);

        ret= BuyConfirm( tpcw, Basket_List);
        LeaveInteraction( interaction);
    }
    else if (interaction== 'f') //
    {
        EnterInteraction( interaction);
        ret= OrderStatus( tpcw);
        LeaveInteraction( interaction);
    }
    else if (interaction== 'g') //
    {
        EnterInteraction( interaction);
        if (tpcw->pecb->cbAvailable > 0) //
        {
            memcpy( tpcw->querystring, ( LPCTSTR) tpcw->pecb-
>lpbData, tpcw->pecb->cbAvailable);
            tpcw->querystring[ tpcw->pecb->cbAvailable]=
'\0';
        }
        strcpy( tpcw->querystring, strstr( tpcw-
>querystring, "&")+ 1);
        ret= OrderStatusResult( tpcw);
        LeaveInteraction( interaction);
    }
    else if (interaction== 'h')
    {
        EnterInteraction( interaction);
        ret= Search( tpcw);
        LeaveInteraction( interaction);
    }
    else if( interaction== 'i') //
    {
        EnterInteraction( interaction);
        ret= SearchResultFrameset( tpcw);
        LeaveInteraction( interaction);
    }
    else if( interaction== 'p') //
    {
        ret= PromoForSearchResult( tpcw);
    }
    else if( interaction== 'm') //
    {
        EnterInteraction( interaction);
        strcpy( tpcw->querystring, strstr( tpcw-
>querystring, "&")+ 1);
        ret= ShowBook( tpcw);
        LeaveInteraction( interaction);

```

```

    }
else if( interaction== 'n')
{
    EnterInteraction( interaction);
    cptrl= strstr( tpcw->querystring,"I_ID");
    Parse( tpcw->szBuffer, cptrl);
    if (tpcw->pecb->cbAvailable > 0)
    {
        memcpy( tpcw->querystring,( LPCTSTR) tpcw->pecb-
>lpbData, tpcw->pecb->cbAvailable);
        tpcw->querystring[ tpcw->pecb->cbAvailable]=
'\0';
    }
    strcpy( tpcw->querystring, strstr( tpcw-
>querystring,"&")+ 1);
    ret= ChangeInfo( tpcw);
    LeaveInteraction( interaction);
}
#ifdef SCREEN_INFO
else if( interaction== 'w')
{
    ret= ShowStat( tpcw);
    ret= FALSE;
}
#endif

if( interaction != 'r' && interaction != 'i'
    && interaction != 'p'

    && interaction != 'v')
{
    AddSpace( tpcw, interaction);
}
return TRUE;
}

```

#### thread.h

```

// thread.h
//
// Copyright Unisys, 1998,1999,2000
#define DIG_TO_MATCH 14
#define AR_SIZE 16384

struct BASKET_ITEM
{
    long scl_shopping_id;
    __int64 scl_i_id;
    long scl_qty;
    char scl_i_title [TITLE_LEN+ 1];
    long scl_i_srp;
    long scl_i_cost;
    char scl_i_backing [BACKING_LEN+ 1];

```

```

char scl_i_comments[ COMMENT_LEN+ 1];
struct BASKET_ITEM * next;
struct BASKET_ITEM * prev;
};

```

```

struct BASKET_LIST
{
    long valid;
    long sc_shopping_id; //
    __int64 sc_c_id; //
    time_t sc_date; //
    char sc_c_lname [FNAME_LEN+ 1];
    char sc_c_fname [LNAME_LEN+ 1];
    long sc_c_discount;
    long sc_c_sub_total;
    long sc_c_tax;
    long sc_c_ship_cost;
    long sc_c_total;
    struct BASKET_ITEM * Item;
    struct BASKET_LIST * prev;
    struct BASKET_LIST * next;
};

```

```

BOOL Initialize();
BOOL process( TPCW * tpcw);
void Terminate();
BOOL DeallocBasket();

```

#### Tpcw.c

```

// tpcw.c
//
// Copyright Unisys, 1998,1999,2000

```

```

#define WIN32_LEAN_AND_MEAN
#define _WIN32_WINNT 0x0500
#define CSDLSPINCOUNT 200
#include "tpcw.h"
#include "tpcw_db.h"
#include "thread.h"
#include "strings.h"
#include "ssl.h"
#include "debug.h"
#include "display.h"
#include <sys/timeb.h>
#include <time.h>

```

```

extern HCERTSTORE hMyCertStore;
extern SCHANNEL_CRED SchannelCred;
extern dwProtocol;
extern LPSTR pszUserName;

```

```

extern ALG_ID aiKeyExch;
#ifdef SCREEN_INFO
extern long numshop;
extern long numthread;
extern long numbasketkill;
extern long numpgreq;
extern long numsslretry;
extern long numwebaction[ 15];
extern long totalwebaction[15];
extern CRITICAL_SECTION webaction[ 15];
extern CRITICAL_SECTION shop;
extern CRITICAL_SECTION thread;
extern CRITICAL_SECTION basketkill;
extern CRITICAL_SECTION pgreq;
extern CRITICAL_SECTION sslretry;
#endif
extern char hostname[ 20];
extern int hostnumber;
HANDLE BasketHeap;
HANDLE TPCWHeap;
DWORD TlsIndex;
SQLHENV henv;
FILE * flog;
CRITICAL_SECTION csFile;
BOOL SendToClient( TPCW * tpcw);

void ErrorLog( char ErrMsg[], char func[])
{
#ifdef FILE_LOG
char dbuffer[ 9];
char tbuffer[ 9];
EnterCriticalSection(&csFile);
_strdate( dbuffer );
_strtime( tbuffer );
fprintf( flog,"%s %s: ThreadID: %x %s - %s\n",
        dbuffer, tbuffer, GetCurrentThreadId(), func,
ErrMsg);
LeaveCriticalSection(& csFile);
fflush( flog);
#endif
}

BOOL HeapInit()
{
    TPCWHeap= HeapCreate( HEAP_GENERATE_EXCEPTIONS, sizeof( TPCW)*
2048, sizeof( TPCW)* 2048);
    if( TPCWHeap== NULL)
    {
        ErrorLog(" TPCWHeap is NULL"," HeapInit()");
        return FALSE;
    }
    BasketHeap= HeapCreate( HEAP_GENERATE_EXCEPTIONS,

```

```

        (sizeof( struct BASKET_LIST)+ sizeof( struct
BASKET_ITEM)* 3)* AR_SIZE* 10,
        (sizeof( struct BASKET_LIST)+ sizeof( struct
BASKET_ITEM)* 3)* AR_SIZE* 10);
    if( BasketHeap== NULL)
    {
        ErrorLog("BasketHeap is NULL"," HeapInit()");
        return FALSE;
    }
    return TRUE;
}

BOOL APIENTRY DllMain( HINSTANCE hModule, DWORD
        ul_reason_for_call, LPVOID lpReserved)
{
    TPCW * tpcw= NULL;
    RETCODE retcode;
    switch (ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
            if(( flog= fopen("c:\\tpcw.log","w"))== NULL)
            {
                DWORD err;
                err= GetLastError();
                return FALSE;
            }
            InitializeCriticalSectionAndSpinCount(& csFile,
CDDLSPINCOUNT);
            retcode= SQLAllocHandle( SQL_HANDLE_ENV, NULL,&
henv);
            if( retcode== SQL_ERROR|| retcode==
SQL_INVALID_HANDLE)
            {
                ErrorLog(" SQLAllocHandle return
SQL_ERROR or SQL_INVALID_HANDLE"," SQLAllocHandle()");
                DeleteCriticalSection(& csFile);
                fclose( flog);
                return FALSE;
            }
            retcode = SQLSetEnvAttr( henv,
SQL_ATTR_ODBC_VERSION,
(SQLPOINTER) SQL_OV_ODBC3, SQL_IS_INTEGER);
            if( retcode== SQL_ERROR|| retcode==
SQL_INVALID_HANDLE)
            {
                ErrorLog(" SQLSetEnvAttr return SQL_ERROR
or SQL_INVALID_HANDLE"," SQLSetEnvAttr()");
                DeleteCriticalSection(& csFile);
                fclose( flog);
                return FALSE;
            }
            if(! HeapInit())
            {

```



```

BOOL ThreadAttach( TPCW * * tpcw)
{
    BOOL rc;
    HANDLE ThreadID;
    PGE * pge;
    *tpcw= HeapAlloc( TPCWHeap, HEAP_ZERO_MEMORY, sizeof( TPCW));
    if(* tpcw== NULL)
    {
        ErrorLog(" HeapAlloc is NULL"," ThreadAttach()");
        return FALSE;
    }
    else
    {
        (* tpcw)->querystring[ 0]= '\0';
        rc= TlsSetValue( TlsIndex,* tpcw);
        if( rc== FALSE)
        {
            ErrorLog(" Cannot set TLS value","
ThreadAttach()");
            return FALSE;
        }
        rc= SQLInit(* tpcw);
        if( rc== FALSE)
        {
            rc= HeapFree( TPCWHeap, 0, *tpcw);
            *tpcw= NULL;
            TlsSetValue( TlsIndex, *tpcw);

            if( rc== 0)
            {
                ErrorLog(" HeapFree Error","
ThreadAttach");
                return FALSE;
            }

            ErrorLog(" Cannot connect to database","
ThreadAttach()");
            return FALSE;
        }
        (* tpcw)->lTlsSeed= InitRnd();
        if((* tpcw)->lTlsSeed< 0)
        {
            ErrorLog(" Seek is less than 0","
ThreadAttach()");
            return FALSE;
        }
        ThreadID= GetCurrentThread();
        if( SetThreadPriority( ThreadID,
THREAD_PRIORITY_HIGHEST)== 0)
    {
        ErrorLog(" SetThreadPriority Error","
ThreadAttach()");
        return FALSE;
    }
    strcpy((* tpcw)->host, hostname);
    (* tpcw)->host_len= strlen((* tpcw)->host);
    (* tpcw)->reconnect= FALSE;
    pge=&(* tpcw)->pge;
    rc= PGEThreadAttach( pge);
    if( rc== FALSE)
    {
        ErrorLog(" PGEThreadAttach Error","
ThreadAttach()");
        return FALSE;
    }
    #ifdef SCREEN_INFO
        EnterCriticalSection(& thread);
        numthread++;
        LeaveCriticalSection(& thread);
    #endif
    }
    return TRUE;
}

BOOL WINAPI GetExtensionVersion (OUT HSE_VERSION_INFO * pVer)
{
    if(! Initialize())
    {
        DeleteCriticalSection(& csFile);
        fclose( flog);
        HeapDestroy( TPCWHeap);
        HeapDestroy( BasketHeap);
        return FALSE;
    }
    pVer->dwExtensionVersion = MAKELONG(HSE_VERSION_MINOR,
HSE_VERSION_MAJOR );
    lstrcpy(pVer->lpszExtensionDesc,"TPCW ISAPI, v 1.0",
HSE_MAX_EXT_DLL_NAME_LEN );
    return TRUE;
}

#define TLSERR1 "<TITLE>TPC-W ERROR</TITLE><H1> TPCW ERROR: <BR> Cannot
allocate TLS<BR></H1>"
#define TLSERR2 "<TITLE>TPC-W ERROR</TITLE><H1> TPCW ERROR: <BR> Error
in process()<BR></H1>"
#define TLSERR3 "<TITLE>TPC-W ERROR</TITLE><H1> TPCW ERROR: <BR> Error
in SendToClient<BR></H1>"
#define TLSERRDEF "<TITLE>TPC-W ERROR</TITLE><H1> TPCW ERROR: <BR>
UH?<BR></H1>"

```

```

        {
            return HSE_STATUS_ERROR;
        }
        goto exit;
    }
    goto exit;
}
#endif

while( tpcw== NULL && i< TLSRETRY)
{
    tpcw= TlsGetValue( TlsIndex);
    if( tpcw== NULL)
    {
        rc= ThreadAttach(& tpcw);
        if( rc== FALSE)
        {
            Sleep( WAITDELAY);
        }
    }
    i++;
}
if( tpcw== NULL)
{
    ErrorLog(" TPCW= NULL Error","HttpExtensionProc()");
    TLSERROR( pecb, 1);
    goto exit2;
}
if( rc== FALSE)
{
    ErrorLog(" ThreadAttach Error","HttpExtensionProc()");
    TLSERROR( pecb, 1);
    goto exit2;
}

tpcw->pecb= pecb;
rc= process( tpcw);

rc= SendToClient( tpcw);
if( rc== FALSE)
{
    ErrorLog( tpcw->szContent,"HttpExtensionProc");
    rc= TLSERROR( pecb, 3);
    if( rc== FALSE)
    {
        return HSE_STATUS_ERROR;
    }
    goto exit;
}
}
exit:
}

BOOL TLSERROR( EXTENSION_CONTROL_BLOCK * pecb, long num)
{
    TPCW tpcw;
    BOOL rc;
    tpcw.pecb= pecb;
    tpcw.expiresflag= QUICK;

    strcpy( tpcw.szHeader, HEADER);
    tpcw.endptr= tpcw.szContent;
    switch( num)
    {
        case 1:
            strcpy( tpcw.szContent, TLSERR1); break;
        case 2:
            strcpy( tpcw.szContent, TLSERR2); break;
        case 3:
            strcpy( tpcw.szContent, TLSERR3); break;
        default:
            strcpy( tpcw.szContent, TLSERRDEF); break;
    }
    tpcw.endptr= tpcw.endptr+ strlen( tpcw.szContent);
    rc= SendToClient(& tpcw);
    return rc;
}

#define TLSRETRY 5
#define WAITDELAY 1000

DWORD WINAPI HttpExtensionProc( EXTENSION_CONTROL_BLOCK * pecb)
{
    TPCW * tpcw;
    BOOL rc;
    int i= 0;
    rc= TRUE;
    tpcw= NULL;

#ifdef SCREEN_INFO
    if( pecb->lpszQueryString[ 0]== 'w')
    {
        TPCW stat;
        memcpy( stat.szHeader, HEADER, HEADERLEN);
        stat.endptr= stat.szContent;
        stat.szContent[ 0]= '\0';
        stat.pecb= pecb;
        ShowStat(& stat);
        rc= SendToClient(& stat);
        if( rc== FALSE)
        {
            ErrorLog( tpcw->szContent,"HttpExtensionProc");
            rc= TLSERROR( pecb, 3);
            if( rc== FALSE)
            exit:
        }
    }
}

```

```

    if( tpcw->reconnect== TRUE)
    {
        ThreadDetach( tpcw);
    }
    exit2:
    return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}

void ErrorMessage( TPCW * tpcw)
{
    tpcw->endptr= tpcw->szContent;
    tpcw->len= sprintf( tpcw->szContent,"<TITLE>TPC-W
ERROR</TITLE>"
        "<H1>TPCW ERROR:<BR>"
        "%s<BR>"
        "%s</H1>",
        tpcw->err, tpcw->odbcerrmsg);
    tpcw->endptr= tpcw->endptr+ tpcw->len;
    ErrorLog( tpcw->szContent,"TPCW");
}

BOOL SendToClient( TPCW * tpcw)
{
    HSE_SEND_HEADER_EX_INFO HeaderExInfo;
    BOOL ret;
    DWORD err;
    unsigned long len;
    len= tpcw->endptr- tpcw->szContent;
    tpcw->endptr= tpcw->szBuffer;

    tpcw->len= sprintf( tpcw->szBuffer, tpcw->szHeader, len);

    //
    if( tpcw->expiresflag== LATER)
    {
        tpcw->len= sprintf( tpcw->szBuffer, HEADEREXPIRE, len,
tpcw->expires);
    }
    else if( tpcw->expiresflag== NOT)
    {
        tpcw->len= sprintf( tpcw->szBuffer, HEADERNOTEXPIRE,
len);
    }
    else if( tpcw->expiresflag== QUICK)
    {
        //
        struct _timeb ltime;
        struct tm *newtime;
        _ftime(& ltime);
        newtime= gmtime(&(ltime.time));
        sprintf( tpcw-> expires,"%s, %d %s %.4d %.2d:%.2d:%.2d
GMT",
            day[ newtime-> tm_wday],

```

```

        newtime-> tm_mday,
        mon[ newtime-> tm_mon],
        newtime-> tm_year+ 1900,
        newtime-> tm_hour,
        newtime-> tm_min,
        newtime-> tm_sec);

        tpcw->len= sprintf( tpcw->szBuffer, HEADEREXPIRE5, len,
tpcw->expires);
    }

    tpcw->endptr= tpcw->endptr+ tpcw->len;
    strcpy( tpcw->endptr, tpcw->szContent);
    tpcw->len= tpcw->len+ len;
    HeaderExInfo.pszHeader = tpcw->szBuffer;
    HeaderExInfo.cchHeader = tpcw->len;
    HeaderExInfo.pszStatus = "200 OK";
    HeaderExInfo.cchStatus = 6;
    HeaderExInfo.fKeepConn = TRUE;
    ret= tpcw->pcb->ServerSupportFunction(
        tpcw->pcb->ConnID,
        HSE_REQ_SEND_RESPONSE_HEADER_EX,
        &HeaderExInfo,
        NULL, NULL);
    if( ret== FALSE)
    {
        ErrorLog(tpcw->szContent,"in SendToClient");
        err= 0;
        err= GetLastError();
        sprintf( tpcw->szContent,"SendToClient Error: %d", err);
    }
    return ret;
}

tpcw.h

//tpcw.h
//
//Copyright Unisys, 1998,1999,2000

#define SCREEN_INFO 1 //

#define FILE_LOG 1
#define LEN_BUFFER 20000
#define TLS_NULL 0xFFFFFFFF
#define HEADER "Content-Length: %lu\r\nContent-Type:
text/html\r\nCache-Control: private\r\n\r\n"

#define HEADERNOTEXPIRE "Content-Length: %lu\r\nContent-Type:
text/html\r\nCache-Control: public\r\n\r\n"

```

```

#define HEADERLEN 73

#define HEADEREXPIRENOW "Content-Length: %lu\r\nContent-Type:
text/html\r\nCache-Control: private\r\n\r\n"
#define HEADEREXPIRE "Content-Length: %lu\r\nContent-Type:
text/html\r\nLast-Modified: %s\r\nCache-control: max-age=30\r\n\r\n"
#define HEADEREXPIRE5 "Content-Length: %lu\r\nContent-Type:
text/html\r\nLast-Modified: %s\r\nCache-control: max-age=5\r\n\r\n"
enum EXP {NOW, LATER, NOT, QUICK};

#include <windows.h>
#include <winsock.h>
#include <stdio.h>
#include <process.h>
#include <malloc.h>
#include <sql.h>
#include <sqlext.h>
#include <odbcss.h>
#include <time.h>
#include <httpext.h>

#ifdef PGEHost
#include "pge.h"
#endif

struct C_Session_ID
{
    __int64 C_ID;
    long SessionID;
};

struct FLOATPRINT
{
    long x;
    long y;
};

typedef struct
{
    EXTENSION_CONTROL_BLOCK * pecb;
    long lTLSeed;
    long hash;
    struct C_Session_ID ID;
    char querystring[700];
    char expires[64];
    enum EXP expiresflag; //
    long len;
    char szContent[LEN_BUFFER];
    char * endptr;
    char szHeader[256];
    char szBuffer[LEN_BUFFER];
    char host[20];

```

```

    int errcode;
    int hostnum;
    int host_len;
    SQLHDBC hdbc;
    SQLHSTMT hstmt;
    BOOL reconnect;
    RETCODE retcode;
    SQLINTEGER cbValue;
    int deadlock;
    SQLCHAR SqlState[6];
    SQLCHAR Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLINTEGER NativeError;
    SQLSMALLINT MsgLen;
    char odbcerrmsg[512];
    char err[128];
    struct FLOATPRINT flt1;
    struct FLOATPRINT flt2;
    struct FLOATPRINT flt3;
    struct FLOATPRINT flt4;
    struct FLOATPRINT flt5;
    PGE pge;
} TPCW;

BOOL ThreadDetach( TPCW * tpcw);
BOOL ThreadAttach( TPCW * * tpcw);
BOOL SendDebug( TPCW * tpcw, char msg[]);
void Message( char msg[]);
void ErrorMessage( TPCW * tpcw);
void ErrorLog( char ErrMsg[], char func[]);
LIBRARY tpcw2

DESCRIPTION 'TPC-W ISAPI'

EXPORTS
    HttpExtensionProc
    GetExtensionVersion

Tpcw_db.c

// tpcw_db.c
//
// Copyright Unisys, 1998,1999,2000
#define RECORD_LEN 8192
#define BUFFER_LEN 1024
#define DEADLOCK_RETRY 5
#define DEADLOCK_SLEEP 500
#define MAX_CONN_OUT 1024
#define RETRY_LOGGING
#include <stdio.h>
#include "tpcw.h"
#include "tpcw_db.h"
#include "thread.h"
#include "display.h"

```

```

#include "encode.h"
#ifdef PGEHOST
#include "pge.h"
#endif
#include <sys/timeb.h>
#include <time.h>

extern char hostname[ 20];
extern int hostnumber;
extern long num_to_and;
extern long maxitem;
extern BOOL DayLightSaving;
extern HANDLE BasketHeap;
extern SQLHENV henv;
extern char produpdate1[ 500];
extern long produpdate1len;
extern char domainname[ 20];

void ODBCErrorHandleSTMT( TPCW * tpcw)
{
    tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1,
                                tpcw->SqlState,&tpcw->NativeError,
                                tpcw->Msg,
                                sizeof( tpcw->Msg),&tpcw->MsgLen);
    if( tpcw->retcode== SQL_NO_DATA)
    {
        wsprintf( tpcw->odbcerrmsg," No Error code returned");
    }
    else
    {
        sprintf( tpcw->odbcerrmsg,
                " SQL State (STMT): %s<br> NativeError: %d<br>
                "Deadlock retry: %d<br> S_ID: %d<br> C_ID:
                %I64d<br> Querystring: %s<br>errcode:%d \0",
                tpcw->SqlState, tpcw->NativeError, tpcw->Msg,
                tpcw->deadlock,
                tpcw->ID.SessionID, tpcw->ID.C_ID, tpcw->querystring,
                tpcw->errcode);
        ErrorLog( tpcw->odbcerrmsg," SQL");
        tpcw->reconnect= TRUE;
    }
}

void ODBCErrorHandleDBC( TPCW * tpcw)
{
    tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_DBC, tpcw->hdbc, 1,
                                tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw->MsgLen);
    if( tpcw->retcode== SQL_NO_DATA)

```

```

{
    wsprintf( tpcw->odbcerrmsg," No Error code returned");
}
else
{
    sprintf( tpcw->odbcerrmsg," SQL State (DBC): %s<br>
NativeError: %d<br> Msg: %s<br>
"Deadlock retry: %d<br> S_ID: %d<br> C_ID:%I64d<br>
Querystring: %s\0",
            tpcw->SqlState, tpcw->NativeError, tpcw->Msg, tpcw->
            >deadlock,
            tpcw->ID.SessionID, tpcw->ID.C_ID, tpcw->querystring);
}
ErrorLog( tpcw->odbcerrmsg," SQL");
}

BOOL SQLInit( TPCW * tpcw)
{
    struct _timeb timeBuffer1;
    struct _timeb timeBuffer2;

    char string[ 128];
    SQLCHAR szOutConn[ MAX_CONN_OUT];
    SQLSMALLINT cbOutConn;
    tpcw->retcode = SQLAllocHandle( SQL_HANDLE_DBC, henv, &tpcw->
    >hdbc);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
    SQL_INVALID_HANDLE)
    {
        ODBCErrorHandleDBC( tpcw);
        return FALSE;
    }

    strcpy( string,"DSN=tpcwDSN; UID=sa; PWD=; APP=");
    strcat( string, tpcw->host);

    _ftime(&timeBuffer1);
    tpcw->retcode = SQLDriverConnect(
        tpcw->hdbc,
        NULL,
        (SQLTCHAR*) string,
        sizeof( string),
        szOutConn,
        MAX_CONN_OUT,
        &cbOutConn,
        SQL_DRIVER_NOPROMPT);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
    SQL_INVALID_HANDLE)
    {

```

```

        ODBCErrorHandleDBC( tpcw);
        ErrorLog("SQLDriverConnect failed", "SQLInit");
        SQLFreeHandle( SQL_HANDLE_DBC, tpcw->hdbc);
        _ftime(&timeBuffer2);
        wsprintf( tpcw->odbcerrmsg,"Elapsed SQLDriverConnect
time is %ds %dms",
                timeBuffer2.time - timeBuffer1.time,
                timeBuffer2.millitm - timeBuffer1.millitm);
        ErrorLog(tpcw->odbcerrmsg,"SQLInit");

        return FALSE;
    }
    tpcw->retcode = SQLAllocHandle( SQL_HANDLE_STMT, tpcw->hdbc,
        &tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE)
    {
        ODBCErrorHandleDBC( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLExecDirect( tpcw->hstmt,"set nocount on",
        SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE)
    {
        ODBCErrorHandleDBC( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLExecDirect( tpcw->hstmt,"set XACT_ABORT ON",
SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE)
    {
        ODBCErrorHandleDBC( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

BOOL NoData( TPCW * tpcw)
{
    long len;
    len= wsprintf( tpcw->odbcerrmsg,"No Data Returned");
    if( len== 0)
    {
        return FALSE;
    }
    return TRUE;
}

```

```

BOOL SQLCountryCheck( TPCW * tpcw, COUNTRY_ENQ_DATA * country)
{
    int co_len;
    co_len= strlen( country->co_name);
    country->co_id= 0;
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
            co_len, 0, country->co_name, 0, &co_len);
        if( tpcw->retcode== SQL_ERROR
            || tpcw->retcode== SQL_INVALID_HANDLE
            || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
CountryCheck(?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);
            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                wsprintf( tpcw->odbcerrmsg,"Deadlock
retry: %d Parameters: %s",
                        tpcw->deadlock,
                        country->co_name);
                ErrorLog( tpcw-
>odbcerrmsg,"SQLCountryCheck");
                #endif
                Sleep( DEADLOCK_SLEEP);
            }
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
    else
    {
        break;
    }
}

```

```

    }
}while( tpcw->deadlock<DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_LONG,& country->co_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
    NoData( tpcw);
    return FALSE;
}
if( country->co_id== 0)
{
    return FALSE;
}
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLGetAD( TPCW * tpcw, BOOK_DETAILS_ENQ_DATA * book)
{
    int i= 0;
    __int64 i_id;
    char thumbnail[ IMAGE_LEN];
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
            &book->i_id, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
}

```

```

        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
GetAD(?)", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);
            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                sprintf( tpcw->odbcerrmsg," Deadlock
retry: %d Parameters: %I64d",
                    tpcw->deadlock,
                    book->i_id);
                ErrorLog( tpcw->odbcerrmsg," SQLGetAD");
                #endif
                Sleep( DEADLOCK_SLEEP);
            }
            else
            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                return FALSE;
            }
        }
        else
        {
            break;
        }
    }
}while( tpcw->deadlock<DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,&
i_id,
    0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,
SQL_C_CHAR, thumbnail, IMAGE_LEN,
&tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}

```

```

}
while (TRUE)
{
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {
        break;
    }
    book->i_also[ i].i_id= i_id;
    strcpy( book->i_also[ i].i_thumbnail, thumbnail);
    i++;
}
/*
tpcw->retcode= SQLMoreResults( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
    SQL_C_TYPE_TIMESTAMP, btime, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
*/
if( i < 5)
{
    NoData( tpcw);
    return FALSE;
}

```

```

SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLGetBasketItem( TPCW * tpcw, struct BASKET_LIST * Basket_List[])
{
    BOOL rc;
    struct BASKET_ITEM item;
    struct BASKET_LIST ShopBasket;
    struct BASKET_LIST * pShopBasket= NULL;
    TIMESTAMP_STRUCT btime;
    TPCW * db;
    BOOL ret= TRUE;
    double srp;
    double cost;
    double subtotal;
    double tax;
    double ship_cost;
    double total;
    double discount;
    struct tm newtime;
    db= malloc( sizeof( TPCW));
    if( db== NULL)
    {
        return FALSE;
    }
    strcpy( db->host, hostname);
    db->host_len= strlen( db->host);
    rc= SQLInit( db);
    if( db== FALSE)
    {
        free( db);
        return FALSE;
    }

    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
            tpcw->host_len, 0, tpcw->host, 0, &tpcw-
>host_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            ret= FALSE;
            goto Done;
        }
    }
}

```

```

        tpcw->retcode = SQLExecDirect( tpcw->hstmt, "{ call
GetItem(?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);
            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                    wsprintf( tpcw->odbcerrmsg,"
Deadlock retry: %d Parameters: %s",
                            tpcw->deadlock,
                            tpcw->host);
                    ErrorLog( tpcw->odbcerrmsg,"
SQLGetBasketItem Step 1");
                #endif
                Sleep( DEADLOCK_SLEEP);
            }
            else
            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                ret= FALSE;
                goto Done;
            }
        }
        else
        {
            break;
        }
    }
}while( tpcw->deadlock<DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_LONG,& item.scl_shopping_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,
SQL_C_UBIGINT,& item.scl_i_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);

```

```

SQLCloseCursor( tpcw->hstmt);
ret= FALSE;
goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,
SQL_C_LONG,& item.scl_qty, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 4,
SQL_C_DOUBLE,& cost, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 5, SQL_C_DOUBLE,& srp,
0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 6,
SQL_C_CHAR, item.scl_i_title, TITLE_LEN, &tpcw-
>cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 7,
SQL_C_CHAR, item.scl_i_backing, BACKING_LEN, &tpcw-
>cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);

```

```

        ret= FALSE;
        goto Done;
    }
    while (TRUE)
    {
        tpcw->retcode = SQLFetch( tpcw->hstmt);
        if (tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            ret= FALSE;
            goto Done;
        }
        if( tpcw->retcode== SQL_NO_DATA)
        {
            break;
        }
        item.scl_i_cost=( long)(0.5+ cost* 100);
        item.scl_i_srp=( long)(0.5+ srp* 100);
        pShopBasket= NULL;
        db->ID.C_ID= 0;
        db->hash= item.scl_shopping_id & num_to_and;

        db->ID.SessionID = item.scl_shopping_id;

        AddToSessionForRecovery( db,& item, Basket_List,&
pShopBasket);
    }
    SQLCloseCursor( tpcw->hstmt);

    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
            tpcw->host_len, 0, tpcw->host, 0, &tpcw-
>host_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            ret= FALSE;
            goto Done;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
GetBasket(?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {

```

```

        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);
        if( strcmp( tpcw->SqlState,"40001")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            #ifdef RETRY_LOGGING
            wsprintf( tpcw->odbcerrmsg,"Deadlock
retry: %d Parameters: %s",
            tpcw->deadlock,
            tpcw->host);
            ErrorLog( tpcw-
>odbcerrmsg,"SQLGetBasketItem Step 2");
            #endif
            Sleep( DEADLOCK_SLEEP);
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            ret= FALSE;
            goto Done;
        }
    }
    else
    {
        break;
    }
}while( tpcw->deadlock<DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_LONG,& ShopBasket.sc_shopping_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,
SQL_C_UBIGINT,& ShopBasket.sc_c_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,
SQL_C_TYPE_TIMESTAMP,& btime, 0, &tpcw->cbValue);

```

```

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 4,
SQL_C_DOUBLE,& subtotal, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 5, SQL_C_DOUBLE,& tax,
0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 6,
SQL_C_DOUBLE,& ship_cost, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 7,
SQL_C_DOUBLE,& total, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 8,
SQL_C_CHAR, ShopBasket.sc_c_fname, FNAME_LEN, &tpcw-
>cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 9,
SQL_C_CHAR, ShopBasket.sc_c_lname, LNAME_LEN, &tpcw-
>cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 10,
SQL_C_DOUBLE,& discount, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
}
while (TRUE)
{
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
}
if( tpcw->retcode== SQL_NO_DATA)
{
    break;
}
ShopBasket.sc_c_sub_total=( long)( subtotal* 100);
ShopBasket.sc_c_tax=( long)( tax* 100);
ShopBasket.sc_c_ship_cost=( long)( ship_cost* 100);
ShopBasket.sc_c_total=( long)( total* 100);
ShopBasket.sc_c_discount=( long)( discount* 100);
db->hash= ShopBasket.sc_shopping_id& num_to_and;
pShopBasket= Basket_List[ db->hash]->next;

while( pShopBasket->next!= Basket_List[ db->hash] &&
pShopBasket->sc_shopping_id!=
ShopBasket.sc_shopping_id)
{

```

```

        pShopBasket= pShopBasket->next;
    }

    if( pShopBasket->sc_shopping_id==
ShopBasket.sc_shopping_id)
    {
        strcpy( pShopBasket->sc_c_fname,
ShopBasket.sc_c_fname);
        strcpy( pShopBasket->sc_c_lname,
ShopBasket.sc_c_lname);
        pShopBasket->sc_c_discount=
ShopBasket.sc_c_discount;
        pShopBasket->sc_c_id= ShopBasket.sc_c_id;
        pShopBasket->sc_c_ship_cost=
ShopBasket.sc_c_ship_cost;
        pShopBasket->sc_c_sub_total=
ShopBasket.sc_c_sub_total;
        pShopBasket->sc_c_tax= ShopBasket.sc_c_tax;
        pShopBasket->sc_c_total= ShopBasket.sc_c_total;
    }
    else
    {
        struct BASKET_LIST * newbasket;
        struct BASKET_ITEM * item;

        newbasket= HeapAlloc( BasketHeap,
HEAP_ZERO_MEMORY,
        sizeof( struct BASKET_LIST));
        if( newbasket== NULL)
        {
            strcpy( tpcw->err, " AddToSession:
newbasket= HeapAlloc(");
            ErrorMessage( tpcw);
            goto Done;
        }
        item= HeapAlloc( BasketHeap, HEAP_ZERO_MEMORY,
sizeof( struct BASKET_ITEM));
        if( item== NULL)
        {
            strcpy( tpcw->err, " AddToSession: item=
HeapAlloc(");
            ErrorMessage( tpcw);
            goto Done;
        }
        item->scl_i_id=- 1L;
        strcpy( item->scl_i_title, " item header");
        item->next= item;
        item->prev= item;
        newbasket->Item= item;
        strcpy( newbasket->sc_c_fname,
ShopBasket.sc_c_fname);

```

```

        strcpy( newbasket->sc_c_lname,
ShopBasket.sc_c_lname);
        newbasket->sc_c_discount=
ShopBasket.sc_c_discount;
        newbasket->sc_c_id= ShopBasket.sc_c_id;
        newbasket->sc_c_ship_cost=
ShopBasket.sc_c_ship_cost;
        newbasket->sc_c_sub_total=
ShopBasket.sc_c_sub_total;
        newbasket->sc_c_tax= ShopBasket.sc_c_tax;
        newbasket->sc_c_total= ShopBasket.sc_c_total;
        newbasket->sc_shopping_id=
ShopBasket.sc_shopping_id;
        pShopBasket= Basket_List[ db->hash]->next;
        newbasket->next= pShopBasket->next;
        pShopBasket->next= newbasket;
        newbasket->prev= pShopBasket;
        newbasket->next->prev= newbasket;
        pShopBasket= newbasket;
    }
    newtime.tm_year= btime.year- 1900;
    newtime.tm_mon= btime.month- 1; //0- 11
    newtime.tm_mday= btime.day;
    newtime.tm_hour= btime.hour;
    newtime.tm_min= btime.minute;
    newtime.tm_sec= btime.second;
    newtime.tm_isdst= DayLightSaving;
    pShopBasket->sc_date= mktime(& newtime);
    pShopBasket->valid=1;
}
SQLCloseCursor( tpcw->hstmt);
Done:
SQLFreeHandle( SQL_HANDLE_STMT, db->hstmt);
SQLDisconnect( db->hdbc);
SQLFreeHandle( SQL_HANDLE_DBC, db->hdbc);
free( db);
SQLCloseCursor( tpcw->hstmt);
return ret;
}

BOOL SQLGetCountryID( TPCW * tpcw, COUNTRY_ENQ_DATA * country)
{
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
            &tpcw->ID.C_ID, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);

```

```

SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call GetCountryID(?)}",
SQL_NTS);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);
if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
sprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: %I64d",
tpcw->deadlock,
tpcw->ID.C_ID);
ErrorLog( tpcw->odbcerrmsg," SQLGetCountryID");
#endif
Sleep( DEADLOCK_SLEEP);
}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_LONG,& country->co_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)

```

```

{
NoData( tpcw);
return FALSE;
}
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLGetAddr( TPCW * tpcw, ADDRESS_ENQ_DATA * addr, COUNTRY_ENQ_DATA
* country)
{
tpcw->deadlock= 0;
do
{
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
&addr->a_id, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call GetAddr(?)}",
SQL_NTS);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);
if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
sprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: %I64d",
tpcw->deadlock,
addr->a_id);
ErrorLog( tpcw->odbcerrmsg," SQLGetAddr");
#endif
Sleep( DEADLOCK_SLEEP);
}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else

```

```

{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_CHAR, addr-
>a_street1, A_STREET_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2, SQL_C_CHAR, addr-
>a_street2, A_STREET_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3, SQL_C_CHAR, addr->a_city,
A_CITY_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 4, SQL_C_CHAR, addr->a_state,
A_STATE_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 5, SQL_C_CHAR, addr->a_zip,
A_ZIP_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 6,
SQL_C_CHAR, country->co_name, CO_NAME_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
NoData( tpcw);
return FALSE;
}
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLItemChange( TPCW * tpcw, BOOK_DETAILS_ENQ_DATA * book)
{
    long i_size, t_size;
    __int64 update[ 5];
    __int64 bookid;
    int i= 0, j, k;
    __int64 updatetemp;
    double cost;

    i_size= strlen( book->i_image);
    t_size= strlen( book->i_thumbnail);
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
            0,
            &book->i_id, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        cost=( double)( book->i_cost/100.0);
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
            SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_DOUBLE, 0, 0,
            &cost, 0, 0);
    }
}

```

```

        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 3,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, i_size,
            0, book->i_image, 0, &i_size);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 4,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, t_size,
            0, book->i_thumbnail, 0,
            &t_size);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }

        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
ItemChange(?,?,?,?)", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);
            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                sprintf( tpcw->odbcerrmsg," Deadlock
retry: %d Parameters: %I64d %.2f %s %s",
                    tpcw->deadlock,
                    book->i_id,
                    cost,
                    book->i_image,
                    book->i_thumbnail);
                ErrorLog( tpcw->odbcerrmsg,"
SQLItemChange Step 1");
            }
            #endif
            Sleep( DEADLOCK_SLEEP);
        }

        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
    else
    {
        break;
    }
}
while( tpcw->deadlock<DEADLOCK_RETRY);

SQLCloseCursor( tpcw->hstmt);

tpcw->deadlock= 0;

do
{
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
        SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
        0,
        &book->i_id, 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
GetNewRelated(?)", SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);
        if( strcmp( tpcw->SqlState,"40001")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            #ifdef RETRY_LOGGING
            sprintf( tpcw->odbcerrmsg," Deadlock
retry: %d Parameters: %I64d",
                tpcw->deadlock,
                book->i_id);
            ErrorLog( tpcw->odbcerrmsg,"
SQLItemChange Step 2");
        }
        #endif
    }
}

```

```

        Sleep( DEADLOCK_SLEEP);
    }
    else
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
}
else
{
    break;
}
}
while( tpcw->deadlock<DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
    SQL_C_UBIGINT,& updatetemp, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
i= 0;
while (TRUE)
{
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);
        if( strcmp( tpcw->SqlState,"24000")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            break;
        }
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {
        break;
    }
    update[ i]= updatetemp;
    i++;
}
if( i<5)
{

```

```

switch (i)
{
    case 0:
        bookid= book->i_id;
        for( j= 0; j<5; j++)
        {
            if( bookid+ (__int64)7 >
                (__int64)maxitem)
            {
                bookid= bookid+ (__int64)7
            }
            else
            {
                bookid= bookid+
                (__int64)7;
            }
            update[ j]= bookid;
        }
        break;
    default:
        bookid= update[ i- 1];
        for( j= i; j<5; j++)
        {
            if( bookid+ (__int64)1 >
                (__int64)maxitem)
            {
                bookid= bookid+
                (__int64)1- (__int64)maxitem;
            }
            else
            {
                bookid= bookid+
                (__int64)1;
            }
            for( k= 0; k<j; k++)
            {
                if( bookid== update[ k])
                {
                    bookid= bookid+
                    (__int64)1;
                }
            }
            update[ j]= bookid;
        }
        break;
    }
}
SQLCloseCursor( tpcw->hstmt);

tpcw->deadlock= 0;
do

```

```

{
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
        SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
        &book->i_id, 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
        SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
        &update[ 0], 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 3,
        SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
        &update[ 1], 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 4,
        SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
        &update[ 2], 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 5,
        SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
        &update[ 3], 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 6,
        SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
        &update[ 4], 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
UpdateRelated(?,?,?,?,?)", SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);
        if( strcmp( tpcw->SqlState,"40001")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            #ifdef RETRY_LOGGING
            sprintf( tpcw->odbcerrmsg," Deadlock
retry: %d Parameters: %I64d %I64d %I64d %I64d %I64d %d",
                tpcw->deadlock,
                book->i_id,
                update[ 0],
                update[ 1],
                update[ 2],
                update[ 3],
                update[ 4]);
            ErrorLog( tpcw->odbcerrmsg,"
SQLItemChange Step 3");
            #endif
            Sleep( DEADLOCK_SLEEP);
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
    else
    {
        break;
    }
}
}

```

```

while( tpcw->deadlock<DEADLOCK_RETRY);

SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLAddNewAddress (TPCW * tpcw, ADDRESS_ENQ_DATA *
    shipaddress, COUNTRY_ENQ_DATA * shipcountry)
{
    int st1_len;
    int st2_len;
    int cty_len;
    int st_len;
    int zip_len;
    st1_len= strlen( shipaddress->a_street1);
    st2_len= strlen( shipaddress->a_street2);
    cty_len= strlen( shipaddress->a_city);
    st_len= strlen( shipaddress->a_state);
    zip_len= strlen( shipaddress->a_zip);
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
            0,
            &shipaddress->a_id, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
            SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
            SQL_PARAM_INPUT, SQL_C_LONG, SQL_SMALLINT, 0, 0,
            &shipcountry->co_id, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
            SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 3,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, st1_len,
            0, shipaddress->a_street1, 0,
            &st1_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
            SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);

```

```

        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 4,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, st2_len,
        0, shipaddress->a_street2, 0,
        &st2_len);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
        SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 5,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, cty_len,
        0, shipaddress->a_city, 0,
        &cty_len);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
        SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 6,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, st_len,
        0, shipaddress->a_state, 0,
        &st_len);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
        SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 7,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, zip_len,
        0, shipaddress->a_zip, 0,
        &zip_len);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
        SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
        AddNewAddress(?,?,?,?,?,?,?)", SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
        SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {

```

```

        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);
        if( strcmp( tpcw->SqlState,"40001")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            #ifdef RETRY_LOGGING
            sprintf( tpcw->odbcerrmsg," Deadlock
retry: %d Parameters: %I64d %d %s %s %s %s %s",
tpcw->deadlock,
shipaddress->a_id,
shipcountry->co_id,
shipaddress->a_street1,
shipaddress->a_street2,
shipaddress->a_city,
shipaddress->a_state,
shipaddress->a_zip);
            ErrorLog( tpcw->odbcerrmsg,"
SQLAddNewAddress");
            #endif
            Sleep( DEADLOCK_SLEEP);
        }
        else if( strcmp( tpcw->SqlState,"23000")== 0)
        {
            break;
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
    else
    {
        break;
    }
}while( tpcw->deadlock<DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_UBIGINT, &shipaddress->a_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {
        NoData( tpcw);
        return FALSE;
    }
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

BOOL SQLGetOrderLine( TPCW * tpcw, __int64 o_id)
{
    BOOK_DETAILS_ENQ_DATA item;
    NEW_ORDER_LINE_DATA orderline;
    double cost;
    double discounts;
    tpcw->szBuffer[ 0]= '\0';
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
&o_id, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call GetOrderLine(?)}",
SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);
            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                sprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: %I64d",
tpcw->deadlock,
o_id);
                ErrorLog( tpcw->odbcerrmsg," SQLGetOrderLine");
                #endif
                Sleep( DEADLOCK_SLEEP);
            }
        }
    }
}

```

```

}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_UBIGINT,& item.i_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,
SQL_C_CHAR, item.i_title, TITLE_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,
SQL_C_DOUBLE,& cost, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 4,
SQL_C_LONG,& orderline.qty, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 5,
SQL_C_DOUBLE,&discounts, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 6,
SQL_C_CHAR, orderline.comments, C_DATA_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 7,
SQL_C_CHAR, item.i_publisher, PUB_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
while (TRUE)
{
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
break;
}
item.i_cost=( long)( cost* 100);
floatprint( item.i_cost,&tpcw->flt1);
tpcw->len= sprintf( tpcw->endptr,"<TR><TD><H4>%I64d</TD>"
" <TD VALIGN=\" TOP\"><H4> Title:%s<BR> Publisher:
%s</H4></TD>"
" <TD><H4>$.%d.%.2d</H4></TD><TD><H4>%d</H4></TD><TD><H4>%d%%</H4>
</TD>"
" <TD><H4>%s</H4></TD></TR>",
item.i_id,
item.i_title,
item.i_publisher,
tpcw->flt1.x,
tpcw->flt1.y,
orderline.qty,
(long)((discounts+0.005)*100),

```

```

        orderline.comments);
    tpcw->endptr= tpcw->endptr+ tpcw->len;
}
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLGetCustOrderInfo( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA *
    cust, ADDRESS_ENQ_DATA * addr, COUNTRY_ENQ_DATA * country,
    ORDER_ENQ_DATA * order)
{
    double subtotal;
    double tax;
    double total;
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
            &cust->c_id, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
GetCustOrderInfo(?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1,
tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);
            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                sprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters:
%I64d",
                    tpcw->deadlock,
                    cust->c_id);
                ErrorLog( tpcw->odbcerrmsg," SQLGetCustOrderInfo");
            }
            #endif
            Sleep( DEADLOCK_SLEEP);
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);

```

```

return FALSE;
}
}
else
{
    break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_CHAR, cust-
>c_fname, FNAME_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 2, SQL_C_CHAR, cust-
>c_lname, LNAME_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 3, SQL_C_CHAR, cust-
>c_email, C_EMAIL_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 4, SQL_C_CHAR, cust-
>c_phone, C_PHONE_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 5, SQL_C_UBIGINT,&
order->o_id, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
}

```

```

    tpcw->retcode = SQLBindCol( tpcw->hstmt, 6,
    SQL_C_TYPE_TIMESTAMP,& order->o_date, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 7,
    SQL_C_DOUBLE,& subtotal, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 8, SQL_C_DOUBLE,& tax,
    0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 9,
    SQL_C_DOUBLE,& total, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 10, SQL_C_CHAR, order-
>o_ship_type, O_SHIP_TYPE_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 11,
    SQL_C_TYPE_TIMESTAMP,& order->o_ship_date, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 12,
    SQL_C_UBIGINT,& order->o_bill_addr_id, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 13,
    SQL_C_UBIGINT,& order->o_ship_addr_id, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 14, SQL_C_CHAR, order-
>o_cc_type, O_CC_TYPE_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 15, SQL_C_CHAR, order-
>o_status, O_STATUS_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 16, SQL_C_CHAR, addr-
>a_street1, A_STREET_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 17, SQL_C_CHAR, addr-
>a_street2, A_STREET_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
    }
}

```

```

return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 18, SQL_C_CHAR, addr-
>a_city, A_CITY_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 19, SQL_C_CHAR, addr-
>a_state, A_STATE_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 20, SQL_C_CHAR, addr-
>a_zip, A_ZIP_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 21,
SQL_C_CHAR, country->co_name, CO_NAME_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}

tpcw->retcode = SQLBindCol( tpcw->hstmt, 22, SQL_C_CHAR, order-
>o_cc_auth, O_CC_AUTH_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}

```

```

tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
order->o_subtotal=( long)( subtotal* 100);
order->o_tax=( long)( tax* 100);
order->o_total=( long)( total* 100);
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLGetCustInfo( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA *
cust, ADDRESS_ENQ_DATA * addr, COUNTRY_ENQ_DATA * country)
{
double discount;
tpcw->deadlock= 0;
do
{
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
&cust->c_id, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
&tpcw->ID.SessionID, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 3,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw->host_len,
0, tpcw->host, 0, &tpcw->host_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
}

```

```

    tpcw->retcode = SQLExecDirect( tpcw->hstmt, "{ call
GetCustInfo(?,?,?)}", SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1,
tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);
        if( strcmp( tpcw->SqlState,"40001")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            #ifdef RETRY_LOGGING
            sprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters:
%i64d %d %s",
                tpcw->deadlock,
                cust->c_id,
                tpcw->ID.SessionID,
                tpcw->host);
            ErrorLog( tpcw->odbcerrmsg," SQLGetCustInfo");
            #endif
            Sleep( DEADLOCK_SLEEP);
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
    else
    {
        break;
    }
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_CHAR, cust-
>c_uname, C_UNAME_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 2, SQL_C_CHAR, cust-
>c_passwd, C_UNAME_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
}

```

```

    tpcw->retcode = SQLBindCol( tpcw->hstmt, 3, SQL_C_CHAR, cust-
>c_fname, FNAME_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 4, SQL_C_CHAR, cust-
>c_lname, LNAME_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 5, SQL_C_CHAR, cust-
>c_phone, C_PHONE_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 6, SQL_C_CHAR, cust-
>c_email, C_EMAIL_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 7,
SQL_C_TYPE_TIMESTAMP,& cust->c_birthdate, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 8, SQL_C_CHAR, cust-
>c_data, C_DATA_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
}

```

```

    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 9, SQL_C_CHAR, addr-
>a_street1, A_STREET_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 10, SQL_C_CHAR, addr-
>a_street2, A_STREET_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 11, SQL_C_CHAR, addr-
>a_city, A_CITY_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 12, SQL_C_CHAR, addr-
>a_state, A_STATE_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 13, SQL_C_CHAR, addr-
>a_zip, A_ZIP_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 14,
SQL_C_CHAR, country->co_name, CO_NAME_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);

```

```

        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 15,
SQL_C_DOUBLE,& discount, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {
        NoData( tpcw);
        return FALSE;
    }
    cust->c_discount=( long)( discount* 100);
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

```

```

BOOL SQLGetCustAddrID( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA * cust)
{
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
&cust->c_id, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call GetCustAddrID(?)}",
SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

```

```

if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
sprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: %I64d",
        tpcw->deadlock,
        cust->c_id);
ErrorLog( tpcw->odbcerrmsg," SQLGetCustAddrID");
#endif
Sleep( DEADLOCK_SLEEP);
}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,& cust-
>c_addr_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
NoData( tpcw);
return FALSE;
}
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

```

```

BOOL SQLGetMaxC_ID( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA * cust, long
range)

```

```

{
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
            &range, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{call
GetMaxC_ID (?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);
            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                wsprintf( tpcw->odbcerrmsg," Deadlock
retry: %d Parameters: %d",
                    tpcw->deadlock,
                    range);
                ErrorLog( tpcw->odbcerrmsg,"
SQLGetMaxC_ID");
            }
            #endif
            Sleep( DEADLOCK_SLEEP);
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
    else
    {
        break;
    }
}
}while( tpcw->deadlock<DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,&
cust->c_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {
        NoData( tpcw);
        return FALSE;
    }
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

BOOL SQLAddressCheck( TPCW * tpcw, COUNTRY_ENQ_DATA *
country, ADDRESS_ENQ_DATA * addr)
{
int st1_len;
int st2_len;
int city_len;
int state_len;
int zip_len;
st1_len= strlen( addr->a_street1);
st2_len= strlen( addr->a_street2);
city_len= strlen( addr->a_city);
state_len= strlen( addr->a_state);
zip_len= strlen( addr->a_zip);
addr->a_id= 0;
tpcw->deadlock= 0;
do
{
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
&country->co_id, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, st1_len,
0, addr->a_street1, 0,

```

```

&st1_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 3,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, st2_len,
0, addr->a_street2, 0,
&st2_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 4,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, city_len,
0, addr->a_city, 0, &city_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 5,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, state_len,
0, addr->a_state, 0,
&state_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 6,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, zip_len,
0, addr->a_zip, 0, &zip_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
AddressCheck(?,?,?,?,?)}", SQL_NTS);

```

```

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);
if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
wsprintf( tpcw->odbcerrmsg, " Deadlock retry: %d Parameters: %d %s %s %s
%s %s",
tpcw->deadlock,
country->co_id,
addr->a_street1,
addr->a_street2,
addr->a_city,
addr->a_state,
addr->a_zip);
ErrorLog( tpcw->odbcerrmsg, " SQLAddressCheck");
#endif
Sleep( DEADLOCK_SLEEP);
}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,& addr->a_id,
0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}

```

```

if( tpcw->retcode== SQL_NO_DATA)
{
NoData( tpcw);
return FALSE;
}
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLGetCustName( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA * cust)
{
double discount;
tpcw->deadlock= 0;
do
{
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
&cust->c_id, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call GetCustName(?)}",
SQL_NTS);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);
if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
sprintf( tpcw->odbcerrmsg, " Deadlock retry: %d Parameters: %I64d",
tpcw->deadlock,
cust->c_id);
ErrorLog( tpcw->odbcerrmsg, " SQLGetCustName");
#endif
Sleep( DEADLOCK_SLEEP);
}
}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
}

```

```

else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_CHAR, cust->c_fname,
FNAME_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2, SQL_C_CHAR, cust->c_lname,
LNAME_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,
SQL_C_DOUBLE,& discount, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
cust->c_discount=( long)( discount* 100);
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
NoData( tpcw);
return FALSE;
}
}
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLUpdateBasketCosts( TPCW * tpcw, struct BASKET_LIST *
ShopBasket)

```

```

{
__int64 id;
double cost, srp;
struct BASKET_ITEM * q;
TIMESTAMP_STRUCT btime;
struct tm newtime;
tpcw->deadlock= 0;
do
{
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
&ShopBasket->sc_shopping_id, 0,
0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw->host_len,
0, tpcw->host, 0, &tpcw->host_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
UpdateBasketCosts(?,?)}", SQL_NTS);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);
if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
wsprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: %d %s",
tpcw->deadlock,
ShopBasket->sc_shopping_id,
tpcw->host);
ErrorLog( tpcw->odbcerrmsg," SQLUpdateBasketCosts");
#endif
Sleep( DEADLOCK_SLEEP);
}
}
else
{

```

```

ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
else
{
break;
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,& id, 0,
&tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2, SQL_C_DOUBLE,& srp,
0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,
SQL_C_DOUBLE,& cost, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
while (TRUE)
{
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
break;
}
q= ShopBasket->Item->next;

```

```

while( q!= ShopBasket->Item&& q->scl_i_id!= id)
{
q= q->next;
}
if( q== ShopBasket->Item)
{
SQLCloseCursor( tpcw->hstmt);
sprintf( tpcw->odbcerrmsg," Can't find item in the list: %d look for:
%d c_id: %I64d session: %d", q->scl_i_id, id, tpcw->ID.C_ID, tpcw-
>ID.SessionID);
return FALSE;
}
q->scl_i_srp=( long)( srp* 100);
q->scl_i_cost=( long)( cost* 100);
}
tpcw->retcode= SQLMoreResults( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_TYPE_TIMESTAMP,& btime, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
NoData( tpcw);
return FALSE;
}
newtime.tm_year= btime.year- 1900;
newtime.tm_mon= btime.month- 1; //0- 11
newtime.tm_mday= btime.day;
newtime.tm_hour= btime.hour;
newtime.tm_min= btime.minute;
newtime.tm_sec= btime.second;
newtime.tm_isdst= DayLightSaving;
ShopBasket->sc_date= mktime(& newtime);

```

```

SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLUnamePwdCheck( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA * cust)
{
    int uname_len;
    int passwd_len;
    uname_len= strlen( cust->c_uname);
    passwd_len= strlen( cust->c_passwd);
    cust->c_id= 0;
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, uname_len,
            0, cust->c_uname, 0,
            &uname_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, passwd_len,
            0, cust->c_passwd, 0,
            &passwd_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
UnamePwdCheck(?,?)", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1,
tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                wprintf( tpcw->odbcerrmsg," Deadlock retry: %d
Parameters: %s %s",
                    tpcw->deadlock,
                    tpcw->c_uname,
                    tpcw->c_passwd);
                ErrorLog( tpcw->odbcerrmsg," SQLUnamePwdCheck");
            }
        }
        #endif
        Sleep( DEADLOCK_SLEEP);
    }
    else
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    else
    {
        break;
    }
}while( tpcw->deadlock<DEADLOCK_RETRY);

    tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_UBIGINT,&cust->c_id, 0, &tpcw->cbValue);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {
        NoData( tpcw);
        return FALSE;
    }
    if( cust->c_id== 0)
    {
        return FALSE;
    }
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

```

```

BOOL SQLDeleteBasket( TPCW * tpcw)
{
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
        SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
        &tpcw->ID.SessionID, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
        tpcw->retcode== SQL_SUCCESS_WITH_INFO)

        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw->host_len,
        0, tpcw->host, 0, &tpcw->host_len);

        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
        tpcw->retcode== SQL_SUCCESS_WITH_INFO)

        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt, "{ call DeleteBasket(?,?)}",
        SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
        tpcw->retcode== SQL_SUCCESS_WITH_INFO)

        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
            >SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
            >MsgLen);

            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                wsprintf( tpcw->odbcerrmsg, " Deadlock retry: %d Parameters: %d %s",
                tpcw->deadlock,
                tpcw->ID.SessionID,
                tpcw->host);
                ErrorLog( tpcw->odbcerrmsg, " SQLDeleteBasket");
            }

            #endif
            Sleep( DEADLOCK_SLEEP);

```

```

    }
    else
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
}
else
{
    break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLAddNewOrder( TPCW * tpcw, ORDER_ENQ_DATA * order,
                    long co_id, double discount, char
                    auth_id[], struct BASKET_LIST * ShopBasket)
{
    int shiptype_len;
    int cc_type_len;
    int cc_name_len;
    int cc_num_len;
    int auth_len;
    TIMESTAMP_STRUCT btime;
    struct tm newtime;
    shiptype_len= strlen( order->o_ship_type);
    cc_type_len= strlen( order->o_cc_type);
    cc_name_len= strlen( order->o_cc_name);
    cc_num_len= strlen( order->o_cc_num);
    auth_len= 16;
    tpcw->deadlock= 0;

    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
        SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
        0,
        &order->o_id, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
        SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
        shiptype_len,
        0, order->o_ship_type, 0,

```

```

        &shiptype_len);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 3,
        SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
        &order->o_bill_addr_id, 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 4,
        SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
        &order->o_ship_addr_id, 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 5,
        SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
        &co_id, 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 6,
        SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
        &tpcw->ID.SessionID, 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 7,

```

```

        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
cc_type_len,
        0, order->o_cc_type, 0,
        &cc_type_len);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 8,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
cc_num_len,
        0, order->o_cc_num, 0,
        &cc_num_len);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 9,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
cc_name_len,
        0, order->o_cc_name, 0,
        &cc_name_len);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 10,
        SQL_PARAM_INPUT, SQL_C_TYPE_TIMESTAMP,
SQL_TYPE_TIMESTAMP, 1,0,
        &order->o_cc_expiration_date, 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 11,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, auth_len,
0, auth_id, 0, &auth_len);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);

```

```

        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 12,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw-
>host_len,
        0, tpcw->host, 0, &tpcw->host_len);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLExecDirect( tpcw->hstmt, "{ call
AddNewOrder(?,?,?,?,?,?,?,?,?,?)", SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);

        if( strcmp( tpcw->SqlState,"40001")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            #ifdef RETRY_LOGGING
            sprintf( tpcw->odbcerrmsg,"Deadlock
retry: %d Parameters: %I64d %s %I64d %I64d %d %d %s %s %s %2d/%2d/%4d
%s %s",

                tpcw->deadlock,
                order->o_id,
                order->o_ship_type,
                order->o_bill_addr_id,
                order->o_ship_addr_id,
                co_id,
                tpcw->ID.SessionID,
                order->o_cc_type,
                order->o_cc_num,
                order->o_cc_name,
                order->o_cc_expiration_date.day,
                order-
>o_cc_expiration_date.month,

                order->o_cc_expiration_date.year,
                auth_id,
                tpcw->host);
            ErrorLog( tpcw->odbcerrmsg,"

        #endif
        Sleep( DEADLOCK_SLEEP);
    }

    }
    else
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    }
    else
    {
        break;
    }
} while( tpcw->deadlock<DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_TYPE_TIMESTAMP,& btime, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
    NoData( tpcw);
    return FALSE;
}
newtime.tm_year= btime.year- 1900;
newtime.tm_mon= btime.month- 1; //0- 11
newtime.tm_mday= btime.day;
newtime.tm_hour= btime.hour;
newtime.tm_min= btime.minute;
newtime.tm_sec= btime.second;
newtime.tm_isdst= DayLightSaving;
ShopBasket->sc_date= mktime(& newtime);
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLAddNewCustomer( TPCW * tpcw, COUNTRY_ENQ_DATA *

```

```

country, ADDRESS_ENQ_DATA * addr, CUSTOMER_INFO_ENQ_DATA *
cust)
{
    int st1_len;
    int st2_len;
    int city_len;
    int state_len;
    int zip_len;
    int uname_len;
    int passwd_len;
    int fname_len;
    int lname_len;
    int phone_len;
    int email_len;
    int data_len;
    double discount;
    st1_len= strlen( addr->a_street1);
    st2_len= strlen( addr->a_street2);
    city_len= strlen( addr->a_city);
    state_len= strlen( addr->a_state);
    zip_len= strlen( addr->a_zip);
    uname_len= strlen( cust->c_uname);
    passwd_len= strlen( cust->c_passwd);
    fname_len= strlen( cust->c_fname);
    lname_len= strlen( cust->c_lname);
    phone_len= strlen( cust->c_phone);
    email_len= strlen( cust->c_email);
    data_len= strlen( cust->c_data);
    tpcw->deadlock= 0;

    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
        SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
        &country->co_id, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
        SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

        {
            tpcw->errcode = 1;
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, st1_len,
        0, addr->a_street1, 0,
        &st1_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
        SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

        {

```

```

tpcw->errcode = 2;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 3,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, st2_len,
0, addr->a_street2, 0,
&st2_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    tpcw->errcode = 3;
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 4,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, city_len,
0, addr->a_city, 0, &city_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    tpcw->errcode = 4;
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 5,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, state_len,
0, addr->a_state, 0,
&state_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    tpcw->errcode = 5;
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 6,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, zip_len,
0, addr->a_zip, 0, &zip_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    tpcw->errcode = 6;
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);

```

```

return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 7,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
&cust->c_id, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->errcode = 7;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 8,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, uname_len,
0, cust->c_uname, 0,
&uname_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->errcode = 8;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 9,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, passwd_len,
0, cust->c_passwd, 0,
&passwd_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->errcode = 9;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 10,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, fname_len,
0, cust->c_fname, 0,
&fname_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->errcode = 10;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}

```

```

tpcw->retcode = SQLBindParameter( tpcw->hstmt, 11,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, lname_len,
0, cust->c_lname, 0,
&lname_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->errcode = 11;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 12,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, phone_len,
0, cust->c_phone, 0,
&phone_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->errcode = 12;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 13,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, email_len,
0, cust->c_email, 0,
&email_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->errcode = 13;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
discount=( double)( cust->c_discount/100.0);
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 14,
SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_DOUBLE, 0, 0,
&discount, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->errcode = 14;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}

```

```

    tpcw->len= sprintf( tpcw->szBuffer,"%d-%d-%d", cust-
>c_birthdate.month, cust->c_birthdate.day, cust->c_birthdate.year);

    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 15,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw->len,
0, tpcw->szBuffer, 0, &tpcw->len);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
    tpcw->errcode = 15;
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 16,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, data_len,
0, cust->c_data, 0, &data_len);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
    tpcw->errcode = 16;
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 17,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
&addr->a_id, 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
    tpcw->errcode = 17;
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 18,
SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
&tpcw->ID.SessionID, 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
    tpcw->errcode = 18;
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 19,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw->host_len,

```

```

0, tpcw->host, 0, &tpcw->host_len);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
    tpcw->errcode = 19;
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
    }
    tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
AddNewCustomer(?,?,?,?,?,?,?,?,?,?,?,?,?,?), SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
    tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1,
tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

    if( strcmp( tpcw->SqlState,"40001")== 0)
    {
    SQLCloseCursor( tpcw->hstmt);
    tpcw->deadlock++;
    #ifdef RETRY_LOGGING
    sprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: %d
%s %s %s %s %s %I64d %s %s %s %s %s %s %s %I64d %d %s",
tpcw->deadlock,
country->co_id,
addr->a_street1,
addr->a_street2,
addr->a_city,
addr->a_state,
addr->a_zip,
cust->c_id,
cust->c_uname,
cust->c_passwd,
cust->c_fname,
cust->c_lname,
cust->c_phone,
cust->c_email,
cust->c_discount,
tpcw->szBuffer,
cust->c_data,
addr->a_id,
tpcw->ID.SessionID,
tpcw->host);
    ErrorLog( tpcw->odbcerrmsg," SQLAddNewCustomer");
    #endif
    Sleep( DEADLOCK_SLEEP);
    }
}

```

```

else if( strcmp( tpcw->SqlState,"23000")== 0)
{
break;
}
else
{
tpcw->errcode = 20;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,&
cust->c_id, 0, &tpcw->cbValue);

```

```

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

{
tpcw->errcode = 21;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}

```

```

tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

{
tpcw->errcode = 22;
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
NoData( tpcw);
return FALSE;
}
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

```

```

BOOL SQLDeleteItem( TPCW * tpcw, struct BASKET_ITEM * item, struct
BASKET_LIST * ShopBasket)

```

```

{
TIMESTAMP_STRUCT btime;

```

```

struct tm newtime;
double subtotal;
tpcw->deadlock= 0;
do
{
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
&item->scl_shopping_id, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}

```

```

tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
&item->scl_i_id, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}

```

```

subtotal=( double)( ShopBasket->sc_c_sub_total/100.0);
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 3,
SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_DOUBLE, 0, 0,
&subtotal, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}

```

```

tpcw->retcode = SQLBindParameter( tpcw->hstmt, 4,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw->host_len,
0, tpcw->host, 0, &tpcw->host_len);

```

```

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}

```

```

tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
DeleteBasketItem(?,?,?,?)}", SQL_NTS);

```

```

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
sprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: %d %I64d
%.2f %s",
tpcw->deadlock,
item->scl_shopping_id,
item->scl_i_id,
subtotal,
tpcw->host);
ErrorLog( tpcw->odbcerrmsg," SQLDeleteItem");

#endif
Sleep( DEADLOCK_SLEEP);
}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_TYPE_TIMESTAMP,& btime, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);

```

```

SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
NoData( tpcw);
return FALSE;
}
newtime.tm_year= btime.year- 1900;
newtime.tm_mon= btime.month- 1; //0- 11
newtime.tm_mday= btime.day;
newtime.tm_hour= btime.hour;
newtime.tm_min= btime.minute;
newtime.tm_sec= btime.second;
newtime.tm_isdst= DayLightSaving;
ShopBasket->sc_date= mktime(& newtime);
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

```

```

BOOL SQLUpdateItem( TPCW * tpcw, struct BASKET_ITEM * item, struct
BASKET_LIST * ShopBasket)
{
TIMESTAMP_STRUCT btime;
struct tm newtime;
double subtotal;
tpcw->deadlock= 0;
do
{
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
&item->scl_shopping_id, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
&item->scl_i_id, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 3,

```

```

SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
&item->scl_qty, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
subtotal=( double)( ShopBasket->sc_c_sub_total/100.0);
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 4,
SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_DOUBLE, 0, 0,
&subtotal, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 5,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw->host_len,
0, tpcw->host, 0, &tpcw->host_len);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLExecDirect( tpcw->hstmt, "{ call
UpdateBasketItem(?,?,?,?)}", SQL_NTS);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw->
SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw->
MsgLen);

if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
sprintf( tpcw->odbcerrmsg, " Deadlock retry: %d Parameters: %d %I64d %d
%.2f %s",
item->scl_shopping_id,

```

```

item->scl_i_id,
item->scl_qty,
subtotal,
tpcw->host);
ErrorLog( tpcw->odbcerrmsg, " SQLUpdateItem");

#endif
Sleep( DEADLOCK_SLEEP);
}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_TYPE_TIMESTAMP,& btime, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
NoData( tpcw);
return FALSE;
}
}
newtime.tm_year= btime.year- 1900;
newtime.tm_mon= btime.month- 1; //0- 11
newtime.tm_mday= btime.day;
newtime.tm_hour= btime.hour;
newtime.tm_min= btime.minute;
newtime.tm_sec= btime.second;
newtime.tm_isdst= DayLightSaving;
ShopBasket->sc_date= mktime(& newtime);

```

```

SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLAddToBasket( TPCW * tpcw, struct BASKET_ITEM * item, struct
BASKET_LIST * ShopBasket)
{
int title_len;
int backing_len;
int comment_len;
TIMESTAMP_STRUCT btime;
struct tm newtime;
double srp;
double cost;
double subtotal;
title_len= strlen( item->scl_i_title);
backing_len= strlen( item->scl_i_backing);
comment_len= strlen( item->scl_i_comments);
tpcw->deadlock= 0;
do
{
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
&item->scl_shopping_id, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
&item->scl_i_id, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 3,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, title_len,
0, item->scl_i_title, 0,
&title_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);

```

```

SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
srp=( double)( item->scl_i_srp/100.0);
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 4,
SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_DOUBLE, 0, 0,
&srp, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
cost=( double)( item->scl_i_cost/100.0);
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 5,
SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_DOUBLE, 0, 0,
&cost, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 6,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, backing_len,
0, item->scl_i_backing, 0,
&backing_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 7,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, comment_len,
0, item->scl_i_comments, 0,
&comment_len);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
subtotal=( double)( ShopBasket->sc_c_sub_total/100.0);
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 8,

```

```

SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_DOUBLE, 0, 0,
&subtotal, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 9,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw->host_len,
0, tpcw->host, 0, &tpcw->host_len);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
AddToBasket(?,?,?,?,?,?,?,?)", SQL_NTS);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
sprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: %d %I64d %s
%.2f %.2f %s %s %.2f %s",
item->scl_shopping_id,
item->scl_i_id,
item->scl_i_title,
srp,
cost,
item->scl_i_backing,
item->scl_i_comments,
subtotal,
tpcw->host);
ErrorLog( tpcw->odbcerrmsg," SQLAddToBasket");
#endif
Sleep( DEADLOCK_SLEEP);
}
}

```

```

else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_TYPE_TIMESTAMP,& btime, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
NoData( tpcw);
return FALSE;
}
newtime.tm_year= btime.year- 1900;
newtime.tm_mon= btime.month- 1; //0- 11
newtime.tm_mday= btime.day;
newtime.tm_hour= btime.hour;
newtime.tm_min= btime.minute;
newtime.tm_sec= btime.second;
newtime.tm_isdst= DayLightSaving;
ShopBasket->sc_date= mktime(& newtime);
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLBookDetailsProc( TPCW * tpcw, BOOK_DETAILS_ENQ_DATA * book)
{
// unsigned int image;
char * chrptr;
}

```

```

double srp;
double cost;
tpcw->deadlock= 0;

do
{
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
    SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0, 0,
    &book->i_id, 0, 0);
    if( tpcw->retcode== SQL_ERROR
        || tpcw->retcode== SQL_INVALID_HANDLE
        || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLExecDirect( tpcw->hstmt, "{call
BookDetailsProc(?)}", SQL_NTS);
    if( tpcw->retcode== SQL_ERROR
        || tpcw->retcode== SQL_INVALID_HANDLE
        || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg), &tpcw->MsgLen);

        if( strcmp( tpcw->SqlState, "40001")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            #ifdef RETRY_LOGGING
            sprintf( tpcw->odbcerrmsg, " Deadlock
retry: %d Parameters: %I64d",
                    tpcw->deadlock,
                    book->i_id);
            ErrorLog( tpcw->odbcerrmsg, "
SQLBookDetailsProc");

            #endif
            Sleep( DEADLOCK_SLEEP);
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
    else
    {
        break;
    }
}

```

```

}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT, &
book->i_id, 0, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR
    || tpcw->retcode== SQL_INVALID_HANDLE
    || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2, SQL_C_CHAR, book-
>i_title, TITLE_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,
    SQL_C_TYPE_TIMESTAMP, & book->i_pub_date, 0, &tpcw-
>cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 4, SQL_C_CHAR, book-
>i_publisher, PUB_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR
    || tpcw->retcode== SQL_INVALID_HANDLE
    || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 5, SQL_C_CHAR, book-
>i_subject, SUBJ_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR
    || tpcw->retcode== SQL_INVALID_HANDLE
    || tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 6, SQL_C_CHAR, book-
>i_desc, DESC_LEN, &tpcw->cbValue);

        if( tpcw->retcode== SQL_ERROR
            || tpcw->retcode== SQL_INVALID_HANDLE
            || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 7, SQL_C_CHAR, book-
>i_image, IMAGE_LEN, &tpcw->cbValue);

        if( tpcw->retcode== SQL_ERROR
            || tpcw->retcode== SQL_INVALID_HANDLE
            || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 8, SQL_C_DOUBLE,&srp,
0, &tpcw->cbValue);
        if( tpcw->retcode== SQL_ERROR
            || tpcw->retcode== SQL_INVALID_HANDLE
            || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 9,
SQL_C_DOUBLE,& cost, 0, &tpcw->cbValue);
        if( tpcw->retcode== SQL_ERROR
            || tpcw->retcode== SQL_INVALID_HANDLE
            || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 10,
SQL_C_TYPE_TIMESTAMP,& book->i_avail, 0, &tpcw->cbValue);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 11, SQL_C_CHAR, book-
>i_isbn, ISBN_LEN, &tpcw->cbValue);

        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 12, SQL_C_LONG,& book-
>i_page, 0, &tpcw->cbValue);

        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 13, SQL_C_CHAR, book-
>i_backing, BACKING_LEN, &tpcw->cbValue);

        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 14, SQL_C_CHAR, book-
>i_dimensions, DIMENSIONS_LEN, &tpcw->cbValue);

        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 15, SQL_C_CHAR, book-
>a_fname, AUTH_LEN, &tpcw->cbValue);

        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
}

```

```

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 16, SQL_C_CHAR, book-
>a_mname, AUTH_LEN, &tpcw->cbValue);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 17, SQL_C_CHAR, book-
>a_lname, AUTH_LEN, &tpcw->cbValue);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 18, SQL_C_CHAR, book-
>i_thumbnail, IMAGE_LEN, &tpcw->cbValue);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }

    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }

    if( tpcw->retcode== SQL_NO_DATA)
    {
        NoData( tpcw);
        return FALSE;
    }

```

```

}

book->i_srp=( long)(0.5+ srp* 100);
book->i_cost=( long)(0.5+ cost* 100);
strcpy( tpcw->szBuffer, book->i_image+ 1);
chrptr= tpcw->szBuffer;

chrptr = book->i_image;
chrptr+=strcspn(book->i_image, " \");
*chrptr = '\0';

tpcw->len= wsprintf(tpcw->endptr,
    "<H2>Title: %s</H2>"
    "<P>Author: %s %s<BR>"
    "Subject: %s</P>"
    "<P><IMG SRC=\"http://image.tpcw.net/tpcw/%s\"
WIDTH=\"200\" HEIGHT=\"200\" ALIGN=\"RIGHT\" BORDER=\"0\">"
    "Description: <I>%s",
    book->i_title,
    book->a_fname,
    book->a_lname,
    book->i_subject,
    book->i_image,
    book->i_desc);
tpcw->endptr= tpcw->endptr+ tpcw->len;

floatprint( book->i_srp,&tpcw->flt1);
floatprint( book->i_cost,&tpcw->flt2);
floatprint( book->i_srp- book->i_cost,&tpcw->flt3);
tpcw->len= sprintf( tpcw->endptr,
    "</I></P>"
    "<BLOCKQUOTE><P><B> Suggested Retail: $%d.%2d</B><BR>"
    "<B>Our Price:</B><font
color=\"#DD0000\"><B>%d.%2d</B></FONT><BR>"
    "<B>You Save:</B><font
color=\"#DD0000\"><B>%d.%2d</B></FONT></blockquote>"
    "<DL><DT><FONT SIZE=\" 2\">"
    "Backing: %s, %d pages<BR>"
    "Published by %s<BR>"
    "Publication date: %d-%d-%d<BR>"
    "Avail date: %d-%d-%d<BR>"
    "Dimensions (in inches): %s<BR>"
    "ISBN: %s<br>",
    tpcw->flt1.x, tpcw->flt1.y,
    tpcw->flt2.x, tpcw->flt2.y,
    tpcw->flt3.x, tpcw->flt3.y,
    book->i_backing,
    book->i_page,
    book->i_publisher,
    book->i_pub_date.month,
    book->i_pub_date.day,

```

```

        book->i_pub_date.year,
        book->i_avail.month,
        book->i_avail.day,
        book->i_avail.year,
        book->i_dimensions,
        book->i_isbn);
tpcw->endptr= tpcw->endptr+ tpcw->len;
tpcw->len= sprintf( tpcw->endptr,
        "</FONT></DT></DL><P ALIGN=\\" CENTER\\">"

        "<A
HREF=\\"http://s/tpcw/tpcw.dll?b&ADD_FLAG=Y&I_ID=%I64d&QTY=1\\"><IMG
SRC=\\"http://tpcwww.tpcw.net/tpcw/gifs/add.gif\\" ALT=\\" Add to Basket\\"
BORDER=\\"0\\" WIDTH=\\"120\\" HEIGHT=\\"30\\"></A>"
        "<A HREF=\\"http://s/tpcw/tpcw.dll?h\\"><IMG
SRC=\\"http://tpcwww.tpcw.net/tpcw/gifs/search.gif\\" ALT=\\" Search\\"
BORDER=\\"0\\" WIDTH=\\"120\\" HEIGHT=\\"30\\"></A>"
        "<A HREF=\\"http://s/tpcw/tpcw.dll?a\\"><IMG
SRC=\\"http://tpcwww.tpcw.net/tpcw/gifs/home.gif\\" ALT=\\" Home\\"
BORDER=\\"0\\" WIDTH=\\"120\\" HEIGHT=\\"30\\"></A>"
        "<A HREF=\\"http://s/tpcw/tpcw.dll?m&I_ID=%I64d\\"><IMG
SRC=\\"http://tpcwww.tpcw.net/tpcw/gifs/update.gif\\" ALT=\\" Update\\"
BORDER=\\"0\\" WIDTH=\\"120\\" HEIGHT=\\"30\\"></A>",
        domainname,
        book->i_id,
        domainname,
        domainname,
        domainname,
        domainname,
        book->i_id);
tpcw->endptr= tpcw->endptr+ tpcw->len;
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

/*=====
-----
-- Code changed on 11/08/00 to meet clause 2.4.3.3 of 1.2 spec
-- This portion of code added on 11/08/00 to meet clause 2.4.3.3 of
1.2 spec
-- It was not used during the measurement intervals of the 11/6 thru
11/10 benchmark but tests were run 11/08/00 to
demonstrate to the auditor that it had no significant impact on
performance.
-- See auditor's attestation letter for details.
Before code changed
=====
=====
=====
*/
//After code changed

```

```

//=====
//=====
BOOL SQLGetRelated1( TPCW * tpcw, BOOK_DETAILS_ENQ_DATA * book)
{
    double srp;
    double cost;
    tpcw->deadlock= 0;

    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
            &book->i_id, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
GetRelated1(?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);

            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                sprintf( tpcw->odbcerrmsg," Deadlock
retry: %d Parameters: %I64d",
                    tpcw->deadlock,
                    book->i_id);
                SQLBookDetailsProc");
                ErrorLog( tpcw->odbcerrmsg,"

                #endif
                Sleep( DEADLOCK_SLEEP);
            }
            else
            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                return FALSE;
            }
        }
    }
}

```

```

        else
        {
            break;
        }
    }while( tpcw->deadlock<DEADLOCK_RETRY);

    tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,&
book->i_id, 0, &tpcw->cbValue);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,
SQL_C_DOUBLE,& cost, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 3, SQL_C_DOUBLE,& srp,
0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 4, SQL_C_CHAR, book-
>i_title, TITLE_LEN, &tpcw->cbValue);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 5, SQL_C_CHAR, book-
>i_backing, BACKING_LEN, &tpcw->cbValue);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {
        NoData( tpcw);
        return FALSE;
    }
    book->i_srp=( long)( srp* 100);
    book->i_cost=( long)( cost* 100);
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

//=====
//=====
BOOL SQLProductUpdated( TPCW * tpcw, BOOK_DETAILS_ENQ_DATA * book)
{
    //    unsigned int image;
    //    unsigned int thumbnail;
    char * chrptr;
    double srp;
    double cost;
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
        &book->i_id, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
BookDetailsProc(?)}", SQL_NTS);

```

```

        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);

        if( strcmp( tpcw->SqlState,"40001")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            #ifdef RETRY_LOGGING
            sprintf( tpcw->odbcerrmsg," Deadlock
retry: %d Parameters: %I64d",
tpcw->deadlock,
book->i_id);
            ErrorLog( tpcw->odbcerrmsg,"
SQLProductUpdated");

            #endif
            Sleep( DEADLOCK_SLEEP);
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
    else
    {
        break;
    }
}
while( tpcw->deadlock<DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,&
book->i_id, 0, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2, SQL_C_CHAR, book-
>i_title, TITLE_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,
SQL_C_TYPE_TIMESTAMP,& book->i_pub_date, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
tpcw->retcode = SQLBindCol( tpcw->hstmt, 4, SQL_C_CHAR, book-
>i_publisher, PUB_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
tpcw->retcode = SQLBindCol( tpcw->hstmt, 5, SQL_C_CHAR, book-
>i_subject, SUBJ_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
tpcw->retcode = SQLBindCol( tpcw->hstmt, 6, SQL_C_CHAR, book-
>i_desc, DESC_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
tpcw->retcode = SQLBindCol( tpcw->hstmt, 7, SQL_C_CHAR, book-
>i_image, IMAGE_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 8, SQL_C_DOUBLE,& srp,
0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 9,
SQL_C_DOUBLE,& cost, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 10,
SQL_C_TYPE_TIMESTAMP,& book->i_avail, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 11, SQL_C_CHAR, book-
>i_isbn, ISBN_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 12, SQL_C_LONG,& book-
>i_page, 0, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 13, SQL_C_CHAR, book-
>i_backing, BACKING_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 14, SQL_C_CHAR, book-
>i_dimensions, DIMENSIONS_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 15, SQL_C_CHAR, book-
>a_fname, AUTH_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 16, SQL_C_CHAR, book-
>a_mname, AUTH_LEN, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 17, SQL_C_CHAR, book-
>a_lname, AUTH_LEN, &tpcw->cbValue);

```

```

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 18, SQL_C_CHAR, book-
>i_thumbnail, IMAGE_LEN, &tpcw->cbValue);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {
        NoData( tpcw);
        return FALSE;
    }
    book->i_srp=( long)(0.5+ srp* 100);
    book->i_cost=( long)(0.5+ cost* 100);

    chrptr = book->i_thumbnail;
    chrptr+=strcspn(book->i_thumbnail, " \");
    *chrptr = '\0';
    chrptr = book->i_image;
    chrptr+=strcspn(book->i_image, " \");
    *chrptr = '\0';
    tpcw->len= wsprintf( tpcw->endptr,
"<H2> Title: %s</H2>"
"<P> Author: %s %s</P>"
"Description: <I>%s",
book->i_title,
book->a_fname,
book->a_lname,
book->i_desc);
    tpcw->endptr= tpcw->endptr+ tpcw->len;
    floatprint( book->i_srp,&tpcw->flt1);

```

```

floatprint( book->i_cost,&tpcw->flt2);
floatprint( book->i_srp- book->i_cost,&tpcw->flt3);
tpcw->len= wsprintf( tpcw->endptr,
    "</I></P>"
    "<IMG SRC=\"http://image.tpcw.net/tpcw/%s\" BORDER=\"0\"
ALIGN=\" RIGHT\" WIDTH=\"200\" HEIGHT=\"200\">"
    "<IMG SRC=\"http://image.tpcw.net/tpcw/%s\" BORDER=\"0\"
ALIGN=\" RIGHT\" WIDTH=\"100\" HEIGHT=\"150\">"
    "<B>Our Price:</B><font
color=\"#DD0000\"><B>${d}%.2d</B></FONT><BR>"
    "<B>You Save:</B><font
color=\"#DD0000\"><B>${d}%.2d</B></FONT></blockquote>"
    "<DL><DT><FONT SIZE=\"2\">"
    "%s, %d pages<BR>"
    "Published by %s<BR>"
    "Publication date: %d-%d-%d<BR>"
    "Dimensions (in inches): %s<BR>"
    "ISBN: %s",
    book->i_image,
    book->i_thumbnail,
    tpcw->flt1.x, tpcw->flt1.y,
    tpcw->flt2.x, tpcw->flt2.y,
    tpcw->flt3.x, tpcw->flt3.y,
    book->i_backing,
    book->i_page,
    book->i_publisher,
    book->i_pub_date.month,
    book->i_pub_date.day,
    book->i_pub_date.year,
    book->i_dimensions,
    book->i_isbn
    );
    tpcw->endptr= tpcw->endptr+ tpcw->len;
    memcpy( tpcw->endptr, produpdate1, produpdate1len);
    tpcw->endptr= tpcw->endptr+ produpdate1len;
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

```

```

BOOL SQLGetMaxItem (TPCW * tpcw)
{
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call GetMaxItem}",
SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {

```

```

tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
wsprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: 0",
tpcw->deadlock);
ErrorLog( tpcw->odbcerrmsg," SQLGetMaxItem");
#endif
Sleep( DEADLOCK_SLEEP);
}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_LONG,&tpcw->hash, 0,
&tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
NoData( tpcw);
return FALSE;
}

```

```

SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLGetMaxO_ID (TPCW * tpcw, ORDER_ENQ_DATA * order, long range)
{
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
&range, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call GetMaxO_ID
(?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1,
tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                wsprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters:
%d",
                    tpcw->deadlock,
                    range);
                ErrorLog( tpcw->odbcerrmsg," SQLGetMaxO_ID");
            }
            #endif
            Sleep( DEADLOCK_SLEEP);
        }
    }
    else
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
}
else

```

```

    {
    break;
    }
    }while( tpcw->deadlock<DEADLOCK_RETRY);
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,&
order->o_id, 0, &tpcw->cbValue);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
    }
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {
    NoData( tpcw);
    return FALSE;
    }
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

BOOL SQLGetMaxADDR_ID (TPCW * tpcw, ADDRESS_ENQ_DATA * addr, long
range)
{
tpcw->deadlock= 0;
do
{
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
&range, 0, 0);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLExecDirect( tpcw->hstmt, "{ call GetMaxADDR_ID (?)}",
SQL_NTS);

```

```

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
wsprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: %d",
tpcw->deadlock,
range);
ErrorLog( tpcw->odbcerrmsg," SQLGetMaxADDR_ID");
#endif
Sleep( DEADLOCK_SLEEP);
}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,& addr->a_id,
0, &tpcw->cbValue);

if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}

```

```

if( tpcw->retcode== SQL_NO_DATA)
{
NoData( tpcw);
return FALSE;
}
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLCreateBasket (TPCW * tpcw, struct BASKET_LIST * ShopBasket)
{
    long c_fname_len;
    long c_lname_len;
    double discount;

    c_fname_len= strlen( ShopBasket->sc_c_fname);
    c_lname_len= strlen( ShopBasket->sc_c_lname);
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
            &tpcw->ID.SessionID, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                return FALSE;
            }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
            SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
            &tpcw->ID.C_ID, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                return FALSE;
            }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 3,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, c_fname_len,
0, ShopBasket->sc_c_fname, 0,
            &c_fname_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                return FALSE;
            }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 4,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, c_lname_len,
0, ShopBasket->sc_c_lname, 0,
            &c_lname_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                return FALSE;
            }
        discount=( double)( ShopBasket->sc_c_discount/100.0);
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 5,
            SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_DOUBLE, 0, 0,
            &discount, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                return FALSE;
            }
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 6,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw->host_len,
0, tpcw->host, 0, &tpcw->host_len);

        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                return FALSE;
            }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
CreateBasket (?,?,?,?,?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

            {
                tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw-
>hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw-
>Msg),&tpcw->MsgLen);

                if( strcmp( tpcw->SqlState,"40001")== 0)
                {
                    SQLCloseCursor( tpcw->hstmt);
                    tpcw->deadlock++;
                }
            }
    }
}

```

```

        #ifdef RETRY_LOGGING
        sprintf( tpcw->odbcerrmsg, " Deadlock retry: %d
Parameters: %d %I64d %s %s %.2f %s",
        tpcw->deadlock,
        tpcw->ID.SessionID,
        tpcw->ID.C_ID,
        ShopBasket->sc_c_fname,
        ShopBasket->sc_c_lname,
        discount,
        tpcw->host);
        ErrorLog( tpcw->odbcerrmsg, " SQLCreateBasket");

        #endif
        Sleep( DEADLOCK_SLEEP);
    }
    else
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    else
    {
        break;
    }
}while( tpcw->deadlock<DEADLOCK_RETRY);

SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLGetMaxSessionID (TPCW * tpcw, long range)
{
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
            &range, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt, "{ call
GetMaxSC_ID (?)", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {

```

```

        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);
        if( strcmp( tpcw->SqlState, "40001")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            #ifdef RETRY_LOGGING
            wsprintf( tpcw->odbcerrmsg, " Deadlock
retry: %d Parameters: %d",
            tpcw->deadlock,
            range);
            ErrorLog( tpcw->odbcerrmsg, "
SQLGetMaxSessionID");
        }
        #endif
        Sleep( DEADLOCK_SLEEP);
    }
    else
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    else
    {
        break;
    }
} while( tpcw->deadlock<DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_LONG,&tpcw-
>hash, 0, &tpcw->cbValue);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {

```

```

        NoData( tpcw);
        return FALSE;
    }
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

BOOL SQLSwitchBasketItem( TPCW * tpcw, struct BASKET_LIST *
Basket_List[])
{
    BOOL rc;
    struct BASKET_ITEM item;
    struct BASKET_LIST ShopBasket;
    struct BASKET_LIST * pShopBasket= NULL;
    TIMESTAMP_STRUCT btime;
    TPCW * db;
    BOOL ret= TRUE;
    double srp;
    double cost;
    double subtotal;
    double tax;
    double ship_cost;
    double total;
    double discount;
    db= malloc( sizeof( TPCW));
    if( db== NULL)
    {
        return FALSE;
    }
    strcpy( db->host, hostname);
    db->host_len= strlen( db->host);
    rc= SQLInit( db);
    if( db== FALSE)
    {
        free( db);
        return FALSE;
    }
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
&tpcw->ID.SessionID, 0, 0);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            ret= FALSE;
            goto Done;
        }
    }

```

```

        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw->host_len,
0, tpcw->host, 0, &tpcw->host_len);

        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
GetItemWithSC_ID(?,?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1,
tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                wsprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: %d
%s",
                    tpcw->deadlock,
                    tpcw->ID.SessionID,
                    tpcw->host);
                ErrorLog( tpcw->odbcerrmsg," SQLSwitchBasketItem Step 1");
                #endif
                Sleep( DEADLOCK_SLEEP);
            }
            else
            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                ret= FALSE;
                goto Done;
            }
        }
        else
        {
            break;
        }
    }while( tpcw->deadlock<DEADLOCK_RETRY);
    item.scl_shopping_id= 0;
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_LONG,& item.scl_shopping_id, 0, &tpcw->cbValue);

```

```

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,
SQL_C_UBIGINT,& item.scl_i_id, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,
SQL_C_LONG,& item.scl_qty, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 4,
SQL_C_DOUBLE,& cost, 0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 5, SQL_C_DOUBLE,& srp,
0, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 6,

```

```

SQL_C_CHAR, item.scl_i_title, TITLE_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 7,
SQL_C_CHAR, item.scl_i_backing, BACKING_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    while (TRUE)
    {
        tpcw->retcode = SQLFetch( tpcw->hstmt);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            ret= FALSE;
            goto Done;
        }
        if( tpcw->retcode== SQL_NO_DATA)
        {
            break;
        }
        item.scl_i_cost=( long)( cost* 100);
        item.scl_i_srp=( long)( srp* 100);
        pShopBasket= NULL;
        db->ID.C_ID= 0;
        db->hash= item.scl_shopping_id& num_to_and;
        AddToSessionForBasketSwitch( db,& item, Basket_List,&
pShopBasket);
    }
    SQLCloseCursor( tpcw->hstmt);
    if( item.scl_shopping_id!= tpcw->ID.SessionID||
item.scl_shopping_id== 0)
    {
        goto Done;
    }

```

```

tpcw->deadlock= 0;
do
{
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
    SQL_PARAM_INPUT, SQL_C_LONG, SQL_INTEGER, 0, 0,
    &tpcw->ID.SessionID, 0, 0);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    tpcw->retcode = SQLBindParameter( tpcw->hstmt, 2,
    SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, tpcw->host_len,
    0, tpcw->host, 0, &tpcw->host_len);

    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }

    tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
GetBasketWithSC_ID(?,?)}", SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),&tpcw->MsgLen);

        if( strcmp( tpcw->SqlState,"40001")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            #ifdef RETRY_LOGGING
            wsprintf( tpcw->odbcerrmsg," Deadlock
retry: %d Parameters: %d %s",
tpcw->deadlock,
tpcw->ID.SessionID,
tpcw->host);
            ErrorLog( tpcw->odbcerrmsg,"
SQLSwitchBasketItem Step 2");
            #endif
            Sleep( DEADLOCK_SLEEP);
        }
    }
}
else
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
}
else
{
    break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,
SQL_C_LONG,& ShopBasket.sc_shopping_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,
SQL_C_UBIGINT,& ShopBasket.sc_c_id, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,
SQL_C_TYPE_TIMESTAMP,& btime, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 4,
SQL_C_DOUBLE,& subtotal, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

```

```

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 5, SQL_C_DOUBLE,& tax,
0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 6,
SQL_C_DOUBLE,& ship_cost, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 7,
SQL_C_DOUBLE,& total, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 8,
SQL_C_CHAR, ShopBasket.sc_c_fname, FNAME_LEN, &tpcw-
>cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 9,
SQL_C_CHAR, ShopBasket.sc_c_lname, LNAME_LEN, &tpcw-
>cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 10,
SQL_C_DOUBLE,& discount, 0, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    ret= FALSE;
    goto Done;
}
while (TRUE)
{
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)

    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        ret= FALSE;
        goto Done;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {
        break;
    }
    ShopBasket.sc_c_sub_total=( long)( subtotal* 100);
    ShopBasket.sc_c_tax=( long)( tax* 100);
    ShopBasket.sc_c_ship_cost=( long)( ship_cost* 100);
    ShopBasket.sc_c_total=( long)( total* 100);
    ShopBasket.sc_c_discount=( long)( discount* 100);

    db->hash= ShopBasket.sc_shopping_id & num_to_and;
    pShopBasket= Basket_List[ db->hash]->next;
    while( pShopBasket->next!= Basket_List[ db->hash]
        && pShopBasket->sc_shopping_id!=
ShopBasket.sc_shopping_id)
    {
        pShopBasket= pShopBasket->next;
    }
}

```

```

        if( pShopBasket->sc_shopping_id==
ShopBasket.sc_shopping_id)
        {
            struct tm newtime;
            strcpy( pShopBasket->sc_c_fname,
ShopBasket.sc_c_fname);

            strcpy( pShopBasket->sc_c_lname,
ShopBasket.sc_c_lname);

            pShopBasket->sc_c_discount=
ShopBasket.sc_c_discount;

            pShopBasket->sc_c_id= ShopBasket.sc_c_id;
            pShopBasket->sc_c_ship_cost=
ShopBasket.sc_c_ship_cost;

            pShopBasket->sc_c_sub_total=
ShopBasket.sc_c_sub_total;

            pShopBasket->sc_c_tax= ShopBasket.sc_c_tax;
            pShopBasket->sc_c_total= ShopBasket.sc_c_total;
            newtime.tm_year= btime.year- 1900;
            newtime.tm_mon= btime.month- 1; //0- 11
            newtime.tm_mday= btime.day;
            newtime.tm_hour= btime.hour;
            newtime.tm_min= btime.minute;
            newtime.tm_sec= btime.second;
            newtime.tm_isdst= DayLightSaving;
            pShopBasket->sc_date= mktime(& newtime);
            pShopBasket->valid=1;
        }
    }
    SQLCloseCursor( tpcw->hstmt);

Done:
    SQLFreeHandle( SQL_HANDLE_STMT, db->hstmt);
    SQLDisconnect( db->hdbc);
    SQLFreeHandle( SQL_HANDLE_DBC, db->hdbc);
    free( db);
    SQLCloseCursor( tpcw->hstmt);
    return ret;
}

```

```

BOOL SQLBeginTran( TPCW * tpcw)
{
    tpcw->deadlock= 0;
    do
    {

```

```

        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call BeginTran}",
SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                #ifdef RETRY_LOGGING
                wprintf( tpcw->odbcerrmsg," Deadlock retry: %d Parameters: 0",
tpcw->deadlock);
                ErrorLog( tpcw->odbcerrmsg," SQLBeginTran");
            }
            #endif
            Sleep( DEADLOCK_SLEEP);
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
    else
    {
        break;
    }
}
while( tpcw->deadlock<DEADLOCK_RETRY);
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

```

```

BOOL SQLCommitTran( TPCW * tpcw)
{
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call CommitTran}",
SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw->hstmt, 1, tpcw-
>SqlState,&tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

```

```

if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
wsprintf( tpcw->odbcerrmsg, " Deadlock retry: %d Parameters: 0",
tpcw->deadlock);
ErrorLog( tpcw->odbcerrmsg, " SQLCommitTran");
#endif
Sleep( DEADLOCK_SLEEP);
}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else
{
break;
}
}while( tpcw->deadlock<DEADLOCK_RETRY);
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

/*=====
-- Code changed on 11/13/00 to meet clause 2.6.3.2 of 1.2 spec
-- This portion of code added on 11/13/00 to meet clause 2.6.3.2 of
1.2 spec
-- It was not used during the measurement intervals of the 11/6 thru
11/10 benchmark but tests were run 11/08/00 to
demonstrate to the auditor that it had no significant impact on
performance.
-- See auditor's attestation letter for details.
=====
//Before code changed
//=====
//=====
//=====
//After code changed
//=====
//=====
BOOL SQLGetCustUname( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA * cust)
{
    tpcw->deadlock= 0;

```

```

do
{
tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
SQL_PARAM_INPUT, SQL_C_UBIGINT, SQL_BIGINT, 0,
0,
&cust->c_id, 0, 0);
if( tpcw->retcode== SQL_ERROR
|| tpcw->retcode== SQL_INVALID_HANDLE
|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
tpcw->retcode = SQLExecDirect( tpcw->hstmt, "{ call
GetCustUname(?)}", SQL_NTS);
if( tpcw->retcode== SQL_ERROR||
tpcw->retcode== SQL_INVALID_HANDLE||
tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT, tpcw-
>hstmt, 1,
tpcw->SqlState,&tpcw->NativeError, tpcw->Msg,
sizeof( tpcw->Msg),&tpcw->MsgLen);
if( strcmp( tpcw->SqlState,"40001")== 0)
{
SQLCloseCursor( tpcw->hstmt);
tpcw->deadlock++;
#ifdef RETRY_LOGGING
sprintf( tpcw->odbcerrmsg, " Deadlock
retry: %d Parameters: %I64d",
tpcw->deadlock,
cust->c_id);
ErrorLog( tpcw->odbcerrmsg, "
SQLGetCustName");
#endif
Sleep( DEADLOCK_SLEEP);
}
else
{
ODBCErrorHandleSTMT( tpcw);
SQLCloseCursor( tpcw->hstmt);
return FALSE;
}
}
else
{
break;
}
} while( tpcw->deadlock<DEADLOCK_RETRY);
tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_CHAR, cust-
>c_urname,
C_UNAME_LEN, &tpcw->cbValue);

```

```

if( tpcw->retcode== SQL_ERROR
    || tpcw->retcode== SQL_INVALID_HANDLE
    || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLFetch( tpcw->hstmt);
if( tpcw->retcode== SQL_ERROR
    || tpcw->retcode== SQL_INVALID_HANDLE
    || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
if( tpcw->retcode== SQL_NO_DATA)
{
    NoData( tpcw);
    return FALSE;
}
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

```

#### tpcw\_db.h

```

// tpcw_db.h
//
// Copyright Unisys, 1998,1999,2000
#include <windows.h>
#include <sql.h>
#define AUTH_LEN 21
#define SUBJ_LEN 61
#define DESC_LEN 550
#define IMAGE_LEN 101
#define TITLE_LEN 61
#define PUB_LEN 61
#define ISBN_LEN 14
#define BACKING_LEN 16
#define DIMENSIONS_LEN 26
#define COMMENT_LEN 101
#define FNAME_LEN 16
#define LNAME_LEN 16
#define C_UNAME_LEN 21
#define C_PHONE_LEN 17
#define C_EMAIL_LEN 51
#define C_DATA_LEN 600
#define A_STREET_LEN 41
#define A_CITY_LEN 31
#define A_STATE_LEN 21
#define A_ZIP_LEN 11

```

```

#define CO_NAME_LEN 51
#define CO_CURRENCY_LEN 11
#define O_CC_TYPE_LEN 11
#define O_CC_AUTH_LEN 16
#define O_CC_NUM_LEN 17
#define O_CC_NAME_LEN 32
#define O_SHIP_TYPE_LEN 11
#define O_STATUS_LEN 16
#define DATE_LEN 11

```

typedef struct

```

{
    unsigned char b_title[TITLE_LEN];
    unsigned char category[SUBJ_LEN];
    long b_id;
    unsigned char b_fname[AUTH_LEN];
    unsigned char b_lname[AUTH_LEN];
    char author[AUTH_LEN];
} BOOK_LIST_ENQ_DATA;

```

typedef struct

```

{
    __int64 i_id;
    unsigned char i_thumbnail [IMAGE_LEN];
} ALSO;

```

typedef struct

```

{
    __int64 i_id;
    ALSO i_also[5];
    unsigned char i_title [TITLE_LEN];
    __int64 i_a_id;
    TIMESTAMP_STRUCT i_pub_date;
    unsigned char i_publisher [PUB_LEN];
    unsigned char i_subject [SUBJ_LEN];
    unsigned char i_desc [DESC_LEN];
    unsigned char i_image [IMAGE_LEN];
    unsigned char i_thumbnail [IMAGE_LEN];
    long i_srp;
    long i_cost;
    TIMESTAMP_STRUCT i_avail;
    unsigned char i_isbn [ISBN_LEN];
    unsigned long i_page;
    unsigned char i_backing [BACKING_LEN];
    unsigned char i_dimensions [DIMENSIONS_LEN];
    unsigned char a_fname [AUTH_LEN];
    unsigned char a_lname [AUTH_LEN];
    unsigned char a_mname [AUTH_LEN];
    TIMESTAMP_STRUCT a_dob;
    unsigned char a_bio [DESC_LEN];
} BOOK_DETAILS_ENQ_DATA;

```

typedef struct

```

{
    __int64 a_id;
    __int64 a_c_id;
    unsigned char a_street1[A_STREET_LEN];
    unsigned char a_street2[A_STREET_LEN];
    unsigned char a_city[A_CITY_LEN];
    unsigned char a_state[A_STATE_LEN];
    unsigned char a_zip[A_ZIP_LEN];
    long a_co_id;
} ADDRESS_ENQ_DATA;

typedef struct
{
    long co_id;
    unsigned char co_name[CO_NAME_LEN];
    double co_exchange;
    unsigned char co_currency[CO_CURRENCY_LEN];
} COUNTRY_ENQ_DATA;

typedef struct
{
    __int64 c_id;
    unsigned char c_uname [C_UNAME_LEN];
    unsigned char c_passwd [C_UNAME_LEN];
    unsigned char c_lname [FNAME_LEN];
    unsigned char c_fname [LNAME_LEN];
    __int64 c_addr_id;
    unsigned char c_phone[C_PHONE_LEN];
    unsigned char c_email[C_EMAIL_LEN];
    TIMESTAMP_STRUCT c_since;
    TIMESTAMP_STRUCT c_last_visit;
    TIMESTAMP_STRUCT c_login;
    TIMESTAMP_STRUCT c_expiration;
    long c_discount;
    long c_balance;
    long c_ytd_pmt;
    TIMESTAMP_STRUCT c_birthdate;
    unsigned char c_data [C_DATA_LEN];
} CUSTOMER_INFO_ENQ_DATA;

typedef struct
{
    __int64 o_id;
    __int64 o_c_id;
    TIMESTAMP_STRUCT o_date;
    long o_subtotal;
    long o_tax;
    long o_total;
    unsigned char o_ship_type[O_SHIP_TYPE_LEN];
    TIMESTAMP_STRUCT o_ship_date;
    __int64 o_bill_addr_id;
    __int64 o_ship_addr_id;
    unsigned char o_cc_type[O_CC_TYPE_LEN];

    unsigned char o_cc_auth[O_CC_AUTH_LEN];
    char o_cc_num[O_CC_NUM_LEN];
    unsigned char o_cc_name[FNAME_LEN+ LNAME_LEN];
    TIMESTAMP_STRUCT o_cc_expiration_date;
    unsigned char o_status[O_STATUS_LEN];
} ORDER_ENQ_DATA;

typedef struct
{
    __int64 o_id;
    __int64 i_id;
    long qty;
    unsigned char comments[COMMENT_LEN];
    double discounts;
} NEW_ORDER_LINE_DATA;

struct Item
{
    unsigned char string[128];
};

typedef struct
{
    struct Item List[50];
} Subject;

BOOL SQLInit( TPCW * tpcw);
BOOL SQLInitSearch( TPCW * tpcw);
BOOL SQLCountryCheck( TPCW * tpcw, COUNTRY_ENQ_DATA * country);
BOOL SQLGetAD( TPCW * tpcw, BOOK_DETAILS_ENQ_DATA * book);

/*=====
-----
-- Code changed on 11/ 08/ 00 to meet clause 2.4.3.3 of 1.2 spec
-- This portion of code added on 11/ 08/ 00 to meet clause 2.4.3.3 of
1.2 spec
-- It was not used during the measurement intervals of the 11/ 6 thru
11/ 10 benchmark but tests were run 11/ 08/ 00 to
demonstrate to the auditor that it had no significant impact on
performance.
-- See auditor's attestation letter for details.
Before code changed
=====
=====
=====
*/
// After code changed
//=====
//=====
BOOL SQLGetRelated1( TPCW * tpcw, BOOK_DETAILS_ENQ_DATA * book);
//=====
//=====

```

```

BOOL SQLGetBasketItem( TPCW * tpcw, struct BASKET_LIST *
    Basket_List[]);
BOOL SQLGetCountryID( TPCW * tpcw, COUNTRY_ENQ_DATA * country);
BOOL SQLGetAddr( TPCW * tpcw, ADDRESS_ENQ_DATA * addr, COUNTRY_ENQ_DATA
    * country);
BOOL SQLItemChange( TPCW * tpcw, BOOK_DETAILS_ENQ_DATA * book);
BOOL SQLAddNewAddress (TPCW * tpcw, ADDRESS_ENQ_DATA *
    shipaddress, COUNTRY_ENQ_DATA * shipcountry);
BOOL SQLGetOrderLine( TPCW * tpcw, __int64 o_id);
BOOL SQLGetCustOrderInfo( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA *
    cust, ADDRESS_ENQ_DATA * addr, COUNTRY_ENQ_DATA * country,
    ORDER_ENQ_DATA * order);
BOOL SQLGetCustInfo( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA *
    cust, ADDRESS_ENQ_DATA * addr, COUNTRY_ENQ_DATA * country);
BOOL SQLGetCustAddrID( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA * cust);
BOOL SQLAddressCheck( TPCW * tpcw, COUNTRY_ENQ_DATA *
    country, ADDRESS_ENQ_DATA * addr);
BOOL SQLGetCustName( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA * cust);
BOOL SQLUpdateBasketCosts( TPCW * tpcw, struct BASKET_LIST * basket);
BOOL SQLUnamePwdCheck( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA * cust);
BOOL SQLDeleteBasket( TPCW * tpcw);
BOOL SQLAddNewOrder( TPCW * tpcw, ORDER_ENQ_DATA * order, long
    co_id, double discount, char auth_id[], struct BASKET_LIST *
    ShopBasket);
BOOL SQLAddNewCustomer( TPCW * tpcw, COUNTRY_ENQ_DATA *
    country, ADDRESS_ENQ_DATA * addr, CUSTOMER_INFO_ENQ_DATA *
    cust);
/*=====
-- Code changed on 11/ 13/ 00 to meet clause 2.6.3.2 of 1.2 spec
-- This portion of code added on 11/ 13/ 00 to meet clause 2.6.3.2 of
1.2 spec
-- It was not used during the measurement intervals of the 11/ 6 thru
11/ 10 benchmark but tests were run 11/ 08/ 00 to
demonstrate to the auditor that it had no significant impact on
performance.
-- See auditor's attestation letter for details.
=====
=====*/
// Before code changed
//=====
//=====
//=====
//=====
// After code changed
//=====
//=====
BOOL SQLGetCustUname( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA * cust);
//=====
BOOL SQLBookListProcATC001( TPCW * tpcw, BOOK_LIST_ENQ_DATA * book);

```

```

BOOL SQLBookListProcATC010( TPCW * tpcw, BOOK_LIST_ENQ_DATA * book);
BOOL SQLBookListProcATC100( TPCW * tpcw, BOOK_LIST_ENQ_DATA * book);
BOOL SQLBookDetailsProc( TPCW * tpcw, BOOK_DETAILS_ENQ_DATA * book);
BOOL SQLProductUpdated( TPCW * tpcw, BOOK_DETAILS_ENQ_DATA * book);
BOOL SQLGetMaxItem( TPCW * tpcw);
BOOL SQLGetCountryID( TPCW * tpcw, COUNTRY_ENQ_DATA * country);
BOOL SQLGetMaxC_ID( TPCW * tpcw, CUSTOMER_INFO_ENQ_DATA * cust, long
    range);
BOOL SQLGetMaxO_ID( TPCW * tpcw, ORDER_ENQ_DATA * order, long range);
BOOL SQLGetMaxADDR_ID( TPCW * tpcw, ADDRESS_ENQ_DATA * addr, long
    range);
BOOL SQLCreateBasket( TPCW * tpcw, struct BASKET_LIST * ShopBasket);
BOOL SQLAddToBasket( TPCW * tpcw, struct BASKET_ITEM * item, struct
    BASKET_LIST * ShopBasket);
BOOL SQLUpdateItem( TPCW * tpcw, struct BASKET_ITEM * item, struct
    BASKET_LIST * ShopBasket);
BOOL SQLDeleteItem( TPCW * tpcw, struct BASKET_ITEM * item, struct
    BASKET_LIST * ShopBasket);
BOOL SQLGetMaxSessionID( TPCW * tpcw, long range);
BOOL SQLSwitchBasketItem( TPCW * tpcw, struct BASKET_LIST *
    Basket_List[]);
void ODBCErrorHandleSTMT( TPCW * tpcw);
void trim( char string[]);
BOOL SQLBeginTran( TPCW * tpcw);
BOOL SQLCommitTran( TPCW * tpcw);

```

----- list.dll

## List.dll Source Code

### Display.c

```

// display.c
#include <windows.h>
#include <sys/timeb.h>
#include <time.h>
#include "list.h"
#include "display.h"
#include "tpcw_db.h"
#include "encode.h"

extern char domainname[ 30];
extern char cachehost[30];
extern char footer[ 500];
extern long footerlen;
extern char header[ 500];
extern long headerlen;
extern long thread;
extern char result1[ 1000];
extern char result2[ 1000];
extern long result1len;

```

```

extern long result2len;

#ifdef SCREEN_INFO
long bs;
long np;
long sr;
CRITICAL_SECTION cs_bs;
CRITICAL_SECTION cs_np;
CRITICAL_SECTION cs_sr;
#endif

static char * subject[] = {
    "ARTS", // 0
    "BIOGRAPHIES", // 1
    "BUSINESS", // 2
    "CHILDREN", // 3
    "COMPUTERS", // 4
    "COOKING", // 5
    "HEALTH", // 6
    "HISTORY", // 7
    "HOME", // 8
    "HUMOR", // 9
    "LITERATURE", // 10
    "NON-FICTION", // 11
    "PARENTING", // 12
    "POLITICS", // 13
    "REFERENCE", // 14
    "RELIGION", // 15
    "ROMANCE", // 16
    "SELF-HELP", // 17
    "SCIENCE-NATURE", // 18
    "SCIENCE-FICTION", // 19
    "SPORTS", // 20
    "MYSTERY", // 21
    "TRAVEL", // 22
    "YOUTH" // 23
};

/* static char *
day[]={ "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };

static char *
mon[]={ "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };

*/
void Parse( char param[], char * querystring)
{
    char buffer[ 1024];
    char * chptr;
    int i= 0;

```

```

    buffer[ 0]= '\0';
    chptr= querystring;
    querystring = strchr( querystring, '=')+ 1;
    while(* querystring!= '&' &&* querystring!= '\0')
    {
        buffer[ i]=* querystring;
        querystring++;
        i++;
    }
    buffer[ i]= '\0';
    if(* querystring!= '\0')
    {
        querystring++;
    }
    strcpy( param, buffer);
    strcpy( chptr, querystring);
}
void DecodeParse( char param[], char * querystring)
{
    char buffer[ 1500];
    char * chptr;
    int i= 0;
    buffer[ 0]= '\0';
    chptr= querystring;
    querystring = strchr( querystring, '=')+ 1;
    while(* querystring!= '&' && * querystring!= '\0')
    {
        buffer[ i]=* querystring;
        querystring++;
        i++;
    }
    buffer[ i]= '\0';
    if(* querystring!= '\0')
    {
        querystring++;
    }
    URL_Decode( buffer);
    strcpy( param, buffer);
    strcpy( chptr, querystring);
}
void EnterInteraction( char interaction)
{
#ifdef SCREEN_INFO
switch( interaction)
{
    case 'n':
        EnterCriticalSection(& cs_np);
        np++;
        LeaveCriticalSection(& cs_np);
        break;
    case 'b':
        EnterCriticalSection(& cs_bs);
        bs++;

```

```

LeaveCriticalSection(& cs_bs);
break;
case 's':
EnterCriticalSection(& cs_sr);
sr++;
LeaveCriticalSection(& cs_sr);
break;
default:
ErrorLog(" Unknown interacation"," EnterInteraction()");
break;
}
#endif
}

```

```
void LeaveInteraction( char interaction)
```

```

{
#ifdef SCREEN_INFO
switch( interaction)
{
case 'n':
EnterCriticalSection(& cs_np);
np--;
LeaveCriticalSection(& cs_np);
break;
case 'b':
EnterCriticalSection(& cs_bs);
bs--;
LeaveCriticalSection(& cs_bs);
break;
case 's':
EnterCriticalSection(& cs_sr);
sr--;
LeaveCriticalSection(& cs_sr);
break;
default:
ErrorLog(" Unknown interaction"," LeaveInteraction()");
break;
}
#endif
}

```

```
BOOL process( TPCW * tpcw)
```

```

{
char chr;
char subj[ 20];
char value[ 128];
BOOL rc;
char type;
struct _timeb ltime;
struct tm *newtime,*loctime;
long len;

```

```

tpcw-> endptr= tpcw-> szContent;
tpcw-> szContent[ 0]= '\0';
tpcw-> expiresflag= LATER;

```

```

type= tpcw->pecb->lpszQueryString[0];
if( type == 'w')
{
tpcw-> expiresflag= NOW;
ShowStat( tpcw);
goto next;
}

```

```

if( tpcw-> pecb-> lpszQueryString[ 0]== '\0')
{
ErrorLog(" Querystring is NULL"," process()");
goto next;
}

```

```

if( type== 'n')
{
chr= tpcw->pecb->lpszQueryString[1] - 'a';
strcpy( subj, subject[chr]);
EnterInteraction('n');
NewProductList( subj, tpcw);
LeaveInteraction('n');
}

```

```

else if( type== 'b')
{
chr= tpcw->pecb->lpszQueryString[1] - 'a';
strcpy( subj, subject[chr]);
EnterInteraction('b');
BestSellerList( subj, tpcw);
LeaveInteraction('b');
}

```

```

else if( type== 'i')
{
EnterInteraction( 's');

```

```
strcpy( tpcw-> querystring, tpcw->pecb->lpszQueryString
```

```
Parse( subj, tpcw-> querystring);
```

```
DecodeParse( value, tpcw-> querystring);
```

```
memcpy( tpcw-> endptr, result1, result1len);
```

```
tpcw-> endptr= tpcw-> endptr+ result1len;
```

```
if( strcmp( subj,"author")== 0)
```

```

{
tpcw-> expiresflag= NOT;
rc= SQLBookListProcATC100( tpcw, value);
if( rc== FALSE)
{

```

```

        strcpy( tpcw-> err, " Result:
SQLBookListProcATC100()");
        ErrorMessage( tpcw);
        return FALSE;
    }
}
else if( strcmp( subj, "title")== 0)
{
    tpcw-> expiresflag= NOT;
    rc= SQLBookListProcATC010( tpcw, value);
    if( rc== FALSE)
    {
        strcpy( tpcw-> err, " Result:
SQLBookListProcATC010()");
        ErrorMessage( tpcw);
        goto next;
    }
}
else if( strcmp( subj, "subject")== 0)
{
    rc= SQLBookListProcATC001( tpcw, value);
    if( rc== FALSE)
    {
        strcpy( tpcw-> err, " Result:
SQLBookListProcATC001()");
        ErrorMessage( tpcw);
        goto next;
    }
}
else
{
    ErrorLog(" Unable to properly parse
Querystring.", " process()");
    goto next;
}

memcpy( tpcw-> endptr, result2, result2len);
tpcw-> endptr= tpcw-> endptr+ result2len;
LeaveInteraction('s');
}
else
{
    ErrorLog(" Unknown interaction", " process()");
    goto next;
}

next:
    _ftime(& ltime);

    loctime= localtime(&(ltime.time));

    len= wsprintf( tpcw-> expires, "<TABLE BORDER=\"0\"
CELLPADDING=\"0\" CELLSPACING=\"0\" ALIGN=\"CENTER\">"

```

```

        "<TR><TD><I> Timestamp %.4d-%.2d-%.2d %.2d:%.2d:%.2d.%d
</TD></TR></TABLE>",
        loctime-> tm_year+ 1900,
        loctime-> tm_mon+ 1,
        loctime-> tm_mday,
        loctime-> tm_hour,
        loctime-> tm_min,
        loctime-> tm_sec,
        ltime. millitm);
        strcpy( tpcw-> endptr, tpcw-> expires);
        tpcw-> endptr= tpcw-> endptr+ len;

        newtime= gmtime(&(ltime.time));

        len= wsprintf( tpcw-> expires, "%s, %.2d %s %.4d %.2d:%.2d:%.2d
GMT",
        day[ newtime-> tm_wday],
        newtime-> tm_mday,
        mon[ newtime-> tm_mon],
        newtime-> tm_year+ 1900,
        newtime-> tm_hour,
        newtime-> tm_min,
        newtime-> tm_sec);

        return TRUE;
}

BOOL NewProductList( char subject[], TPCW * tpcw)
{
    int rc;
    strcpy( tpcw->endptr, header);
    tpcw->endptr = tpcw->endptr + headerlen;
    rc= SQLNewProducts( tpcw, subject);
    if( rc== FALSE)
    {
        strcpy( tpcw->err, "NewProductList()");
        ErrorMessage( tpcw);
        return FALSE;
    }
    strcpy( tpcw->endptr, footer);
    tpcw->endptr= tpcw->endptr + footerlen;
    return TRUE;
}

BOOL BestSellerList( char subject[], TPCW * tpcw)
{
    int rc;
    strcpy( tpcw->endptr, header);
    tpcw->endptr = tpcw->endptr + headerlen;

    rc= SQLBestSellers( tpcw, subject);

```

```

if( rc== FALSE)
{
    strcpy( tpcw-> err," BestSellerList()");
    ErrorMessage( tpcw);
    return FALSE;
}
strcpy( tpcw->endptr, footer);
tpcw->endptr = tpcw->endptr + footerlen;
return TRUE;
}

```

```

BOOL ShowStat( TPCW * tpcw)

```

```

{
    #ifdef SCREEN_INFO
    long total= 0;
    tpcw->len= wprintf( tpcw->endptr,
"<table border= 0><tr valign=top><td>"
"<table border= 1>"
"<tr><td> number of best sellers<td>%d"
"<tr><td> number of new products<td>%d"
"<tr><td> number of search result<td>%d"
"<tr><td> number thread local storage<td>%d"
"</table>",
    bs,
    np,
    sr,
    thread
    );
    tpcw->endptr= tpcw->endptr + tpcw->len;
    #endif
    return TRUE;
}
// display. h

```

```

static char *
day[]={ "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };

```

```

static char *
mon[]={ "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };

```

```

BOOL process( TPCW * tpcw);
BOOL NewProductList( char subject[], TPCW * tpcw);
BOOL BestSellerList( char subject[], TPCW * tpcw);
BOOL ShowStat( TPCW * tpcw);

```

#### Encode.c

```

// encode.c

```

```

//
// Copyright Unisys, 1999,2000
#include <windows.h>
#include "encode.h"
const char sp_char[27]= {'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '_', '-', '=', '+', '{', '}', '[', ']', '|', ':', ';', ',', '.', '?', '/', '~', ' '};
const char sp_hex[27][3]={{ '%', '2', '1'}, /*!
{'%', '4', '0'}, //@
{'%', '2', '3'}, //#
{'%', '2', '4'}, //$
{'%', '2', '5'}, //%
{'%', '5', 'E'}, //^
{'%', '2', '6'}, //&
{'%', '2', 'A'}, //*
{'%', '2', '8'}, //(
{'%', '2', '9'}, //(
{'%', '5', 'F'}, //_
{'%', '2', 'D'}, //-
{'%', '3', 'D'}, //=
{'%', '2', 'B'}, //+
{'%', '7', 'B'}, //{
{'%', '7', 'D'}, //{
{'%', '5', 'B'}, //[
{'%', '5', 'D'}, //[
{'%', '7', 'C'}, //|
{'%', '3', 'A'}, //:
{'%', '3', 'B'}, //;
{'%', '2', 'C'}, //,
{'%', '2', 'E'}, //.
{'%', '3', 'F'}, //?
{'%', '2', 'F'}, //|
{'%', '7', 'E'}, //~
{'%', '2', '0'}
};

```

```

int URL_Decode( char *target)
{
    char temp[1500];
    char c1, c2;
    int k;
    int j;
    strcpy( temp, target);
    for( j= 0, k= 0; temp[j]!='\0'; j++, k++)
    {
        switch( temp[j])
        {
            case '+':
                target[k]= ' ';
                break;
            case '%':
                c1= tolower( temp[++ j]);
                c2= tolower( temp[++ j]);

```

```

        if( isdigit( c1))
        {
            c1-= '0';
        }
        else
        {
            c1= c1- 'a'+ 10;
        }
        if( isdigit( c2))
        {
            c2-= '0';
        }
        else
        {
            c2= c2- 'a'+ 10;
        }
        target[k]= c1* 16+ c2;
        break;
    default:
        target[k]= temp[j];
        break;
    }
}
target[k]= '\0';
return (k);
}

```

#### encode.h

```

// encode.h
//
// Copyright Unisys, 1999,2000
int URL_Decode( char * target);

```

#### htmltext.h

```

// htmltext.h

void FillHTML();

```

#### init.c

```

//      init.c

#define _WIN32_WINNT 0x0500
#define CSDLSPINCOUNT 200
#include <windows.h>
#include <stdio.h>
#include "list.h"
#include "init.h"
char hostname[ 20];
char domainname[ 30];
char cachehost[30];

```

```

char footer[ 700];
long footerlen;
char header[ 500];
long headerlen;
char result1[ 1000];
char result2[ 1000];
long result1len;
long result2len;
#ifdef SCREEN_INFO
extern long bs;
extern long np;
extern long sr;
extern CRITICAL_SECTION cs_bs;
extern CRITICAL_SECTION cs_np;
extern CRITICAL_SECTION cs_sr;
#endif
BOOL ReadHostName()
{
    int i;
    i= gethostname( hostname, sizeof( hostname));
    if( i!= 0)
    {
        ErrorLog(" Fail on gethostname", " ReadHostName()");
        return FALSE;
    }
    return TRUE;
}
BOOL PrintHeader()
{
    headerlen= wsprintf( header, "<TABLE BORDER=\"1\" CELLPADDING=\"1\"
CELLSPACING=\"1\" ALIGN=\"CENTER\">"
        "<TR><TD><FONT
        SIZE=\"+1\">Author</FONT></TD>"
        "<TD><FONT SIZE=\"+1\">"
        Title</FONT></TD></TR>");
    return TRUE;
}
BOOL PrintFooter()
{
    footerlen= wsprintf( footer,
        "</TABLE><P ALIGN=\"CENTER\">"
        "<A
        HREF=\"http://%s/tpcw/tpcw.dll?b&ADD_FLAG=N\" target=\"_top\">"
        "<IMG
        SRC=\"http://tpcwww.tpcw.net/tpcw/gifs/cart.gif\" "
        "ALT=\"Shopping Cart\"
        BORDER=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>"
        "<A
        HREF=\"http://%s/tpcw/tpcw.dll?h\" target=\"_top\">"
        "<IMG
        SRC=\"http://tpcwww.tpcw.net/tpcw/gifs/search.gif\" "

```

```

        "ALT=\"Search\" BORDER=\"0\"
WIDTH=\"120\" HEIGHT=\"30\"></A>
        "<A
HREF=\"http://%s/tpcw/tpcw.dll?a\" target=\"_top\">\"
        "<IMG
SRC=\"http://tpcwww.tpcw.net/tpcw/gifs/home.gif\"
        "ALT=\"Home\" BORDER=\"0\"
WIDTH=\"120\" HEIGHT=\"30\"></A>
        "<BR>\",
        domainname, domainname,
domainname);
return TRUE;
}

BOOL ReadDomainName()
{
    HKEY hkTPCW;
    DWORD dwType= REG_SZ;
    DWORD dwSize;
    domainname[ 0]= '\0';
    if( RegOpenKeyEx( HKEY_LOCAL_MACHINE,
        "SOFTWARE\\DELL\\TPCW", 0,KEY_READ,& hkTPCW)==
ERROR_SUCCESS)
    {
        dwSize= sizeof( domainname);
        if( RegQueryValueEx( hkTPCW,"DomainName", NULL,&
dwType,( unsigned char *) domainname,& dwSize)!= ERROR_SUCCESS)
        {
            ErrorLog("DomainName key value does not exist in
registry","ReadDomainName()");
            return FALSE;
        }
    }
    else
    {
        ErrorLog("Cannot open registry key","ReadDomainName()");
        return FALSE;
    }
    if( RegCloseKey( hkTPCW)!= ERROR_SUCCESS)
    {
        return FALSE;
    };
    if( strlen( domainname)> 0)
    {
        return TRUE;
    }
    else
    {

```

```

        ErrorLog(" DomainName length is less than zero","
ReadDomainName()");
        return FALSE;
    }
}

BOOL ReadCacheHost()
{
    HKEY hkTPCW;
    DWORD dwType= REG_SZ;
    DWORD dwSize;
    cachehost[ 0]= '\0';
    if( RegOpenKeyEx( HKEY_LOCAL_MACHINE,"SOFTWARE\\DELL\\TPCW", 0
,KEY_READ,&hkTPCW)== ERROR_SUCCESS)
    {
        dwSize= sizeof( cachehost);
        if( RegQueryValueEx( hkTPCW,"CacheHost", NULL,&dwType,(
unsigned
        char *) cachehost,&dwSize)!= ERROR_SUCCESS)
        {
            ErrorLog("CacheHost key value does not exist in
registry","ReadCacheHost()");
            return FALSE;
        }
    }
    else
    {
        ErrorLog(" Cannot open register key"," ReadCacheHost()");
        return FALSE;
    }
    if( RegCloseKey( hkTPCW)!= ERROR_SUCCESS)
    {
        return FALSE;
    };
    if( strlen( cachehost)>0)
    {
        return TRUE;
    }
    else
    {
        ErrorLog(" CacheHost length is less than zero",
ReadCacheHost()");
        return FALSE;
    }
}

void SearchHTML()
{
    resultllen= sprintf( result1,"<TABLE BORDER=\"1\" CELLPADDING=\"1\"
CELLSPACING=\"1\">\"

```

```

                "<TR><TD><FONT
SIZE=\"+1\">Author</FONT></TD>"
                "<TD><FONT
SIZE=\"+1\">Title</FONT></TD></TR>");
result2len= sprintf( result2,

                "</TABLE><P ALIGN=\"CENTER\">"
                "<A
HREF=\"http://%s/tpcw/tpcw.dll?b&ADD_FLAG=N\" target=\"_top\">"
                "<IMG
SRC=\"http://tpcwww.tpcw.net/tpcw/gifs/cart.gif\" ALT=\"Shopping Cart\"
BORDER=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>"
                "<A
HREF=\"http://%s/tpcw/tpcw.dll?h\" target=\"_top\">"
                "<IMG
SRC=\"http://tpcwww.tpcw.net/tpcw/gifs/search.gif\" ALT=\"Search\"
BORDER=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>"
                "<A
HREF=\"http://%s/tpcw/tpcw.dll?a\" target=\"_top\">"
                "<IMG
SRC=\"http://tpcwww.tpcw.net/tpcw/gifs/home.gif\" ALT=\"Home\"
BORDER=\"0\" WIDTH=\"120\" HEIGHT=\"30\"></A>"
                "</P></BODY></HTML>", domainname,
domainname, domainname);
}

BOOL Terminate()
{
#ifdef SCREEN_INFO
DeleteCriticalSection(& cs_bs);
DeleteCriticalSection(& cs_np);
#endif
return TRUE;
}
#define CSSPINCOUNT 200

BOOL Initialize()
{
    BOOL rc;

    #ifdef SCREEN_INFO
    InitializeCriticalSectionAndSpinCount(& cs_bs, CSSPINCOUNT);
    InitializeCriticalSectionAndSpinCount(& cs_np, CSSPINCOUNT);
    InitializeCriticalSectionAndSpinCount(& cs_sr, CSSPINCOUNT);
    bs= 0;
    np= 0;
    sr= 0;
    #endif

    rc= ReadHostName();
    if( rc== FALSE)

```

```

{
    #ifdef SCREEN_INFO
    DeleteCriticalSection(& cs_bs);
    DeleteCriticalSection(& cs_np);
    DeleteCriticalSection(& cs_sr);
    #endif
    ErrorLog(" ReadHostName Error", " Initialize()");
    return FALSE;
}

rc= ReadDomainName();
if( rc== FALSE)
{
    #ifdef SCREEN_INFO
    DeleteCriticalSection(& cs_bs);
    DeleteCriticalSection(& cs_np);
    DeleteCriticalSection(& cs_sr);
    #endif
    ErrorLog(" ReadDomainName Error", " Initialize()");
    return FALSE;
}

rc= ReadCacheHost();
if( rc== FALSE)
{
    #ifdef SCREEN_INFO
    DeleteCriticalSection(& cs_bs);
    DeleteCriticalSection(& cs_np);
    DeleteCriticalSection(& cs_sr);
    #endif
    ErrorLog(" ReadCacheHost Error", " Initialize()");
    return FALSE;
}

PrintHeader();
PrintFooter();
SearchHTML();
return TRUE;
}

init.h
//      init.h

BOOL Initialize();
BOOL Terminate();

List.c
//      list.c

#define WIN32_LEAN_AND_MEAN
#define _WIN32_WINNT 0x0500

```

```

#define CSDLSPINCOUNT 200
#define TLS_NULL 0xFFFFFFFF
#define TLSRETRY 5
#define FILE_LOG 666
#include <windows.h>
#include <time.h>
#include <malloc.h>
#include <stdio.h>
#include "list.h"
#include "init.h"
#include "display.h"
#include "tpcw_db.h"
#include <sys/timeb.h>

DWORD TlsIndex;
SQLHENV henv;
extern char hostname[ 20];
long thread;
CRITICAL_SECTION cs_thread;
FILE * flog;
CRITICAL_SECTION csFile;

void ErrorLog( char ErrMsg[], char func[])
{
    #ifdef FILE_LOG
    char dbuffer[ 9];
    char tbuffer[ 9];
    EnterCriticalSection(&csFile);
    _strdate( dbuffer );
    _strtime( tbuffer );
    fprintf( flog,"%s %s: ThreadID: %x %s - %s\n", dbuffer,
tbuffer, GetCurrentThreadId(), func, ErrMsg);
    LeaveCriticalSection(&csFile);
    fflush( flog);
    #endif
}

BOOL APIENTRY DllMain( HINSTANCE hModule, DWORD ul_reason_for_call,
LPVOID lpReserved)
{
    TPCW * tpcw= NULL;
    RETCODE retcode;
    switch (ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
            if(( flog= fopen("c:\\cache.log","w"))== NULL)
            {
                DWORD err;
                err= GetLastError();
                return FALSE;
            }
            InitializeCriticalSectionAndSpinCount(&csFile,
CSDLSPINCOUNT);
            retcode= SQLAllocHandle( SQL_HANDLE_ENV,
NULL,&henv);
            if( retcode== SQL_ERROR|| retcode==
SQL_INVALID_HANDLE)
            {
                DeleteCriticalSection(&csFile);
                fclose( flog);
                ErrorLog(" SQLAllocHandle return
SQL_ERROR or SQL_INVALID_HANDLE",
"
SQLAllocHandle()");
                return FALSE;
            }
            retcode = SQLSetEnvAttr( henv,
SQL_ATTR_ODBC_VERSION,
(SQLPOINTER) SQL_OV_ODBC3, SQL_IS_INTEGER);
            if( retcode== SQL_ERROR|| retcode==
SQL_INVALID_HANDLE)
            {
                DeleteCriticalSection(&csFile);
                fclose( flog);
                ErrorLog(" SQLSetEnvAttr return SQL_ERROR
or SQL_INVALID_HANDLE",
"
SQLSetEnvAttr()");
                return FALSE;
            }
            #ifdef SCREEN_INFO
            InitializeCriticalSectionAndSpinCount(&cs_thread,
CSDLSPINCOUNT);
            thread= 0;
            #endif
            if(! Initialize())
            {
                DeleteCriticalSection(&csFile);
                #ifdef SCREEN_INFO
                DeleteCriticalSection(&cs_thread);
                #endif
                fclose( flog);
                return FALSE;
            }
            TlsIndex= TlsAlloc();
            if( TlsIndex== TLS_NULL)
            {
                ErrorLog("TlsIndex is NULL","DllMain()");
                DeleteCriticalSection(&csFile);
                fclose( flog);
            }
    }
}

```

```

        #ifdef SCREEN_INFO
        DeleteCriticalSection(&cs_thread);
        #endif
        return FALSE;
    }
    DisableThreadLibraryCalls( hModule);
    break;

case DLL_THREAD_ATTACH:
    break;

case DLL_THREAD_DETACH:
    ThreadDetach( tpcw);
    break;

case DLL_PROCESS_DETACH:
    #ifdef SCREEN_INFO
    DeleteCriticalSection(&cs_thread);
    #endif
    Terminate();
    TlsFree( TlsIndex);
    TlsIndex= TLS_NULL;
    SQLFreeHandle( SQL_HANDLE_ENV, henv);
    fclose( flog);
    DeleteCriticalSection(&csFile);
    break;

    default:
        break;
}
return TRUE;
}

```

```

BOOL WINAPI GetExtensionVersion (OUT HSE_VERSION_INFO * pVer)
{
    pVer->dwExtensionVersion = MAKELONG( HSE_VERSION_MINOR,
        HSE_VERSION_MAJOR );
    lstrcpy(pVer->lpszExtensionDesc,"List ISAPI, v1.0",
        HSE_MAX_EXT_DLL_NAME_LEN );
    return TRUE;
}

```

```

BOOL ThreadAttach( TPCW * * tpcw)
{
    BOOL rc;
    HANDLE ThreadID;
    *tpcw= malloc( sizeof( TPCW));
    if(* tpcw== NULL)
    {
        ErrorLog(" HeapAlloc is NULL"," ThreadAttach()");
        return FALSE;
    }
}

```

```

    }
    else
    {
        rc= TlsSetValue(TlsIndex,* tpcw);
        if( rc== FALSE)
        {
            ErrorLog("Cannot set TLS value","
ThreadAttach()");
            return FALSE;
        }
        rc= SQLInit(* tpcw);
        if( rc== FALSE)
        {
            free(*tpcw);
            *tpcw= NULL;
            TlsSetValue( TlsIndex, *tpcw);

            ErrorLog("Cannot connect to database",
ThreadAttach()");
            return FALSE;
        }
        (* tpcw)->reconnect= FALSE;
        ThreadID= GetCurrentThread();
        if( SetThreadPriority( ThreadID,
THREAD_PRIORITY_HIGHEST)== 0)
        {
            ErrorLog(" SetThreadPriority Error",
ThreadAttach()");
            return FALSE;
        }
    }
    #ifdef SCREEN_INFO
    EnterCriticalSection(&cs_thread);
    thread++;
    LeaveCriticalSection(&cs_thread);
    #endif
}
return TRUE;
}

```

```

BOOL ThreadDetach( TPCW * tpcw)
{
    tpcw= TlsGetValue( TlsIndex);
    if( tpcw!= NULL)
    {
        SQLFreeHandle( SQL_HANDLE_STMT, tpcw->hstmt);
        SQLDisconnect( tpcw->hdbc);
        SQLFreeHandle( SQL_HANDLE_DBC, tpcw->hdbc);
        free( tpcw);
    }
}

```

```

        tpcw= NULL;
        TlsSetValue( TlsIndex, tpcw);
        #ifdef SCREEN_INFO
            EnterCriticalSection(&cs_thread);
            thread--;
            LeaveCriticalSection(&cs_thread);
        #endif
    }
    return TRUE;
}

#define TLSERR1 "<TITLE>TPC-W ERROR</TITLE><H1> TPCW ERROR: <BR> Cannot
allocate TLS<BR></H1>"
#define TLSERR2 "<TITLE>TPC-W ERROR</TITLE><H1> TPCW ERROR: <BR> Error
in process()<BR></H1>"
#define TLSERR3 "<TITLE>TPC-W ERROR</TITLE><H1> TPCW ERROR: <BR> Error
in SendToClient<BR></H1>"
#define TLSERRDEF "<TITLE>TPC-W ERROR</TITLE><H1> TPCW ERROR: <BR> UH?<
BR></H1>"

BOOL TLSERROR( EXTENSION_CONTROL_BLOCK * pecb, long num)
{
    TPCW tpcw;
    tpcw.pecb= pecb;
    tpcw.expiresflag= QUICK;
    tpcw.endptr= tpcw.szContent;
    switch( num)
    {
        case 1:
            strcpy( tpcw.szContent, TLSERR1); break;
        case 2:
            strcpy( tpcw.szContent, TLSERR2); break;
        case 3:
            strcpy( tpcw.szContent, TLSERR3); break;
        default:
            strcpy( tpcw.szContent, TLSERRDEF); break;
    }
    tpcw.endptr= tpcw.endptr+ strlen( tpcw.szContent);
    SendToClient(&tpcw);
    return TRUE;
}

#define WAITDELAY 1000

DWORD WINAPI HttpExtensionProc( EXTENSION_CONTROL_BLOCK * pecb)
{
    TPCW * tpcw= NULL;
    BOOL rc;
    int i= 0;

    rc= TRUE;

```

```

#ifdef SCREEN_INFO
    if( pecb->lpszQueryString[0]== 'w')
    {
        TPCW stat;
        stat.endptr= stat.szContent;
        stat.szContent[0]= '\0';
        stat.expiresflag= NOW;
        ShowStat(&stat);
        stat.pecb= pecb;
        rc= SendToClient(&stat);
        if( rc== FALSE)
        {
            ErrorLog( tpcw-
>szContent, "HttpExtensionProc");
            rc= TLSERROR( pecb, 3);
            if( rc== FALSE)
            {
                return HSE_STATUS_ERROR;
            }
            goto exit;
        }
        goto exit;
    }
#endif

while( tpcw==NULL && i<TLSRETRY)
{
    tpcw= TlsGetValue( TlsIndex);
    if( tpcw== NULL)
    {
        rc= ThreadAttach(&tpcw);
        if( rc== FALSE)
        {
            Sleep(WAITDELAY);
        }
    }
    i++;
}

if( tpcw== NULL)
{
    ErrorLog("TPCW= NULL Error", "HttpExtensionProc()");
    TLSERROR( pecb, 1);
    goto exit;
}

if( rc== FALSE)
{
    ErrorLog("ThreadAttach Error", "HttpExtensionProc()");
    TLSERROR( pecb, 1);
    goto exit;
}

```

```

tpcw->pcb= pcb;
rc= process( tpcw);
if( rc== FALSE)
{
    TLSERROR( pcb, 2);
    goto exit;
}
rc= SendToClient( tpcw);
if( rc== FALSE)
{
    ErrorLog( tpcw->szContent, " HttpExtensionProc");
    TLSERROR( pcb, 3);
    goto exit;
}
exit:
if( tpcw->reconnect== TRUE)
{
    ThreadDetach( tpcw);
}

return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}

BOOL SendToClient( TPCW * tpcw)
{
    HSE_SEND_HEADER_EX_INFO HeaderExInfo;
    BOOL ret;
    long len;
    DWORD err;

    len= tpcw->endptr- tpcw->szContent;
    tpcw->endptr= tpcw->szBuffer;

    if( tpcw->expiresflag== LATER)
    {
        tpcw->len= sprintf( tpcw->szBuffer, HEADEREXPIRE, len,
tpcw->expires);
    }
    else if( tpcw->expiresflag== NOT)
    {
        tpcw->len= sprintf( tpcw->szBuffer, HEADERNOTEXPIRE,
len);
    }
    else if( tpcw->expiresflag== NOW)
    {
        tpcw->len= sprintf( tpcw->szBuffer, HEADEREXPIRENOW,
len);
    }
    else if( tpcw->expiresflag== QUICK)
    {

```

```

struct _timeb ltime;
struct tm *newtime;
_ftime(& ltime);
newtime= gmtime(&(ltime.time));
wsprintf( tpcw-> expires,"%s, %d %s %.4d %.2d:%.2d:%.2d
GMT",

        day[ newtime-> tm_wday],
        newtime-> tm_mday,
        mon[ newtime-> tm_mon],
        newtime-> tm_year+ 1900,
        newtime-> tm_hour,
        newtime-> tm_min,
        newtime-> tm_sec);

        tpcw->len= sprintf( tpcw->szBuffer, HEADEREXPIRE5,
len);
    }
    tpcw->endptr= tpcw->endptr+ tpcw->len;
    strcpy( tpcw->endptr, tpcw->szContent);
    tpcw->len= tpcw->len+ len;
    HeaderExInfo.pszHeader = tpcw->szBuffer;
    HeaderExInfo.cchHeader = tpcw->len;
    HeaderExInfo.pszStatus = "200 OK";
    HeaderExInfo.cchStatus = 6;
    HeaderExInfo.fKeepConn = TRUE;
    ret= tpcw->pcb->ServerSupportFunction(tpcw->pcb-
>ConnID,HSE_REQ_SEND_RESPONSE_HEADER_EX,&HeaderExInfo,NULL, NULL);
    if( ret== FALSE)
    {
        err= 0;
        err= GetLastError();
        sprintf( tpcw->szContent,"SendToClient Error: %d", err);
    }
    return ret;
}

void ErrorMessage( TPCW * tpcw)
{
    tpcw->endptr= tpcw->szContent;
    tpcw->len= sprintf( tpcw->szContent,"<TITLE>TPC-W
ERROR</TITLE>"

        "<H1>TPCW ERROR:<BR>"
        "%s<BR>"
        "%s</H1>",
        tpcw->err,
        tpcw->odbcerrmsg);

    tpcw->endptr= tpcw->endptr+ tpcw->len;
    tpcw->expiresflag= QUICK;
}
LIBRARY list

```

```
DESCRIPTION 'TPC-W Search ISAPI'
```

#### EXPORTS

```
HttpExtensionProc  
GetExtensionVersion
```

#### List.h

```
// list.h  
#define LEN_BUFFER 16000  
#define SCREEN_INFO 1  
#include <httpext.h>  
#include <sqlext.h>  
#include <odbcss.h>  
#define HEADEREXPIRENOW "Content-Length: %lu\r\nContent-Type:  
text/html\r\nCache-Control: private\r\n\r\n"  
#define HEADEREXPIRE "Content-Length: %lu\r\nContent-Type:  
text/html\r\nLast-Modified: %s\r\nCache-control: max-age=30\r\n\r\n"  
#define HEADEREXPIRE5 "Content-Length: %lu\r\nContent-Type:  
text/html\r\nLast-Modified: %s\r\nCache-control: max-age=5\r\n\r\n"
```

```
#define HEADERNOTEXPIRE "Content-Length: %lu\r\nContent-Type:  
text/html\r\nCache-Control: public\r\n\r\n"
```

```
enum EXP {NOW, LATER, NOT, QUICK};
```

```
typedef struct  
{  
EXTENSION_CONTROL_BLOCK * pecb;  
char szContent[ LEN_BUFFER];  
char * endptr;  
long len;  
char szBuffer[ LEN_BUFFER];  
char expires[ 1500];  
enum EXP expiresflag;  
char querystring[ 700];  
BOOL reconnect;  
SQLHDBC hdbc;  
SQLHSTMT hstmt;  
RETCODE retcode;  
SQLINTEGER cbValue;  
int deadlock;  
SQLCHAR SqlState[ 6];  
SQLCHAR Msg[ SQL_MAX_MESSAGE_LENGTH];  
SQLINTEGER NativeError;  
SQLSMALLINT MsgLen;  
char odbcerrmsg[ 1024];  
char err[ 512];  
} TPCW;
```

```
BOOL ThreadDetach( TPCW * tpcw);  
BOOL SendToClient( TPCW * tpcw);  
void ErrorMessage( TPCW * tpcw);
```

```
void ErrorLog( char ErrMsg[], char func[]);
```

#### pagesize.h

```
// pagesize.h  
//  
// Copyright Unisys, 2000  
#define HOMESIZE 7172  
#define SHOPSIZE 2056  
#define CUSTREGSIZE 2769  
#define BUYREQSIZE 3690  
#define BUYCONFSIZE 1530  
#define ORDERINQ 953  
#define ORDERDISP 1715  
#define SEARCHREQ 2004  
#define SEARCHRES 1891  
#define NEWPROD 8095  
#define BESTPROD 6640  
#define PRODDET 1726  
#define ADMINREQ 1551  
#define ADMINCONF 1349
```

#### pge.h

```
// pge.h  
//  
// Copyright Unisys, 1999,2000  
  
#define SECURITY_WIN32  
#define PGEHost "pge"  
#define MAX_CONN_OUT 1024  
#define O_CC_NUM_LEN 17  
#define MAX_SECURE_RETRIES 10  
#define MAX_SSL_HANDSHAKE_RETRIES 10  
#define MAX_OPEN_SOCKET_RETRIES 5
```

```
#include <windows.h>  
#include <httpext.h>  
#include <sql.h>  
#include <time.h>  
#include "basetd.h"  
#include "sspi.h"
```

```
extern HMODULE g_hSecurity;  
extern SecurityFunctionTable g_SecurityFunc;  
extern CredHandle hClientCreds;
```

```
//  
// HtmlBuffer  
// A buffer to receive/ contain HTML code.  
typedef struct {  
PBYTE pbBuffer; // Pointer to the buffer.  
LONG cbBuffer; // Size of buffer in (contiguous) bytes.
```

```

// (This is that max available storage in the buffer.
LONG cbOffset; // An offset into the buffer.
// This is a utility var that is used for various
// purposes depending on context.
// (E.g., Returned by a receive data function to
// use it to point to the next byte past the last
// byte received into the buffer.)
int i;
} HtmlBuffer;

//
// HtmlHeader
// Contains fields for certain components that can occur in the
// header of an HTML Response.
typedef struct {
    INT RespStatusCode; // The Response Status Code
    CHAR RespStatusMsg[ 16]; // First 15 chars of RespStatusMsg
subfield
    LONG ContentLength; // Content- Length header field.
    CHAR SetCookie[ 64]; // Set- Cookie field
    CHAR Location[ 128]; // Location field
} HtmlHeader;

#define MAX_BUFF_SZ 8000
typedef struct
{
    UINT uSocketNo;
    SOCKET sSocket;
    BOOL bRenegotiateConnection;
    // If TRUE then indicates that it is desired that a new full
    // handshake be performed between the caller and the server.
    BOOL ConnMade;
    // Flag that is set TRUE when a connection is established via
this socket.
    ULONG ulSendBuffSz;
    ULONG ulRecvBuffSz;
    CHAR szIPAddr[ 20];
    USHORT uPort;
    CHAR SendBuffer[ MAX_BUFF_SZ];
    CHAR RecvBuffer[ MAX_BUFF_SZ];
    ULONG SendMsgSz;
    PBYTE SendMsg;
    ULONG RecvMsgSz;
    PBYTE RecvMsg;
} SOCKET_STATE;

typedef struct
{
    DWORD PGETHreadID;
    // SOCKET Socket;
    int TransactionCount;
    long o_id;
    char cc_num[ O_CC_NUM_LEN];

```

```

long lTLSeed;
char request[ 8000];
char embedmsg[ 60];
char padstring[ 6000];
char pgemsg[ 7000];
char auth[ 16];
time_t starttime;
SOCKET_STATE ssSecureWebConn;
CtxtHandle hContext;
SecPkgContext_StreamSizes Sizes;
LONG ServerNameGenerator;
CHAR MyIDString[ 16];
CHAR szServerName[ 32];
HtmlBuffer SendBuffer;
HtmlBuffer RecvBuffer;
HtmlHeader RecvHeader;
} PGE;

char pgehost[ 20];
BOOL PGEInit();
BOOL CleanPGE();
BOOL PGERequest( PGE * pge, int o_id, char cc_num[], char auth_id[]);
BOOL PGETHreadAttach( PGE * pge);
BOOL PGETHreadDetach( PGE * pge);
HRESULT SecureSendThenRecvHtml(
    HtmlBuffer * pSendBuffer,
    HtmlBuffer * pRecvBuffer,
    HtmlHeader * pRecvHeader,
    SOCKET_STATE * pss,
    PGE * pts,
    SecurityFunctionTable * g_pSecurityFunc,
    CredHandle * phClientCreds,
    CtxtHandle * phContext,
    SecPkgContext_StreamSizes * pSizes
);
INT
DisconnectFromServer(
    SOCKET Socket,
    SecurityFunctionTable *g_pSecurityFunc,
    PCredHandle phCreds,
    CtxtHandle * phContext,
    BOOL fVerbose
);
#define TPCW_E_OK ((HRESULT)0x00000000L)
#define TPCW_E_SECURE_SEND_THEN_RECV_HTML_BAD_HTTP_PAGE_STATUS
((HRESULT)0xE0010031L)
#define TPCW_E_SECURE_SEND_THEN_RECV_HTML_BAD_HTTP_IMAGE_STATUS
((HRESULT)0xE0010032L)
// ParseHtmlHeader codes
#define TPCW_E_PARSE_HTML_HEADER_FAILED_TO_CONVERT_STATUS_TO_INT
((HRESULT)0xE0010201L)

```

```

#define
TPCW_E_PARSE_HTML_HEADER_FAILED_TO_CONVERT_CONTENTLENGTH_TO_INT
((HRESULT)0xE0010202L)
#define TPCW_E_PARSE_HTML_HEADER_HEADER_DOESNOT_BEGIN_WITH_HTTP_SLASH
((HRESULT)0xE0010203L)
#define TPCW_E_PARSE_HTML_HEADER_FIRST_BLANK_CHAR_NOT_FOUND
((HRESULT)0xE0010204L)
#define
TPCW_E_PARSE_HTML_HEADER_DIDNOT_FIND_BLANK_CHAR_AFTER_STATUSCODE
((HRESULT)0xE0010205L)
#define
TPCW_E_PARSE_HTML_HEADER_DIDNOT_FIND_CRLF_TO_TERMINATE_STATUSCODE_LINE
((HRESULT)0xE0010206L)
#define
TPCW_E_PARSE_HTML_HEADER_NOT_ENOUGH_ROOM_IN_HEADER_LOCATION_FIELD
((HRESULT)0xE0010207L)
// SecureSendThenRecvHtml codes
#define TPCW_E_SECURE_SEND_THEN_RECV_HTML_SOCKIOINIT_FAILED
((HRESULT)0xE0010301L)
#define
TPCW_E_SECURE_SEND_THEN_RECV_HTML_MAX_SSL_HANDSHAKE_RETRIES_FAILED
((HRESULT)0xE0010302L)
#define TPCW_E_SECURE_SEND_THEN_RECV_HTML_MAX_SECURE_RETRIES_FAILED
((HRESULT)0xE0010303L)
#define
TPCW_E_SECURE_SEND_THEN_RECV_HTML_FATAL_ERR_FROM_SECURE_SEND_THEN_RECV
((HRESULT)0xE0010304L)
#define TPCW_E_SECURE_SEND_THEN_RECV_HTML_HTTP_STATUS_NOT_200
((HRESULT)0xE0010305L)
// SecureSendThenRecv codes
#define TPCW_E_SECURE_SEND_THEN_RECV_PARSE_HTML_HEADER_FAILED
((HRESULT)0xE0010501L)
#define TPCW_E_SECURE_SEND_THEN_RECV_CONTENT_LENGTH_EQ_MINUS_1
((HRESULT)0xE0010502L)
#define TPCW_E_SECURE_SEND_THEN_RECV_NOT_ENOUGH_ROOM_IN_RECVBUFFER
((HRESULT)0xE0010503L)
#define TPCW_E_SECURE_SEND_THEN_RECV_BUFFER_ALLOC_FAILED
((HRESULT)0xE0010504L)
// PerformClientHandShake codes
#define TPCW_E_PERFORMCLIENTHANDSHAKE_ERROR_SENDING_DATA_TO_SERVER_1
((HRESULT)0xE0010601L)
// ClientHandShakeLoop codes
#define
TPCW_E_CLIENTHANDSHAKELOOP_SERVER_REQUESTED_CLIENT_AUTHORIZATION
((HRESULT)0xE0010701L)
#define TPCW_E_CLIENTHANDSHAKELOOP_ERROR_SENDING_DATA_TO_SERVER_2
((HRESULT)0xE0010702L)
#define TPCW_E_CLIENTHANDSHAKELOOP_OUT_OF_MEMORY_2
((HRESULT)0xE0010703L)
#define TPCW_E_CLIENTHANDSHAKELOOP_BUFFER_ALLOC_FAILED
((HRESULT)0xE0010704L)

```

#### readreg.h

```

// readreg.h
//
// Copyright Unisys, 1999,2000

BOOL ReadHostName();
BOOL ReadPGEHost();
BOOL ReadDomainName();
BOOL ReadCacheHost();
BOOL ReadMaxItem(TPCW * tpcw);
long ReadMaxO_ID(TPCW * tpcw, long range);
long ReadMaxC_ID(TPCW * tpcw, long range);
long ReadMaxADDR_ID(TPCW * tpcw, long range);
long ReadMaxSession_ID(TPCW * tpcw, long range);

strings.h

// strings.h
//
// Copyright Unisys, 1999,2000
// Defines
#define A 16807
#define M 2147483647
#define Q 127773 /* M div A */
#define R 2836 /* M mod A */
#define Thread __declspec( thread)

BOOL DigSyl(long lNum, char *pszOut);
int MakeAlphaString( int x, int y, int z, char *str, long * lTLSeed);
long irand(long * Seed);
long RandomNumber(long lower, long upper, long * Seed);
long InitRnd();// thread.h
//
// Copyright Unisys, 1998,1999,2000
#define DIG_TO_MATCH 14
#define AR_SIZE 16384
struct BASKET_ITEM
{
    long scl_shopping_id;
    long scl_i_id;
    long scl_qty;
    char scl_i_title [TITLE_LEN+ 1];
    long scl_i_srp;
    long scl_i_cost;
    char scl_i_backing [BACKING_LEN+ 1];
    char scl_i_comments[ COMMENT_LEN+ 1];
    struct BASKET_ITEM * next;
    struct BASKET_ITEM * prev;
};

struct BASKET_LIST
{

```

```

    long sc_shopping_id; // shopping id
    long sc_c_id; // sc_c_id
    time_t sc_date; // sc_date
    char sc_c_lname [FNAME_LEN+ 1];
    char sc_c_fname [LNAME_LEN+ 1];
    long sc_c_discount;
    long sc_c_sub_total;
    long sc_c_tax;
    long sc_c_ship_cost;
    long sc_c_total;
    struct BASKET_ITEM * Item;
    struct BASKET_LIST * prev;
    struct BASKET_LIST * next;
};

```

```

BOOL Initialize();
BOOL process( TPCW * tpcw);
void Terminate();
BOOL DeallocBasket();

```

#### **Tpcw.h**

```

//tpcw.h
//
//Copyright Unisys, 1998,1999,2000

#define SCREEN_INFO 1

#define FILE_LOG 666
#define LEN_BUFFER 20000
#define TLS_NULL 0xFFFFFFFF
#define HEADER "Content-Length: %lu\r\nContent-Type: text/html\r\n\r\n"
#define HEADERLEN 49

#include <windows.h>
#include <winsock.h>
#include <stdio.h>
#include <process.h>
#include <malloc.h>
#include <sql.h>
#include <sqlext.h>
#include <odbcss.h>
#include <time.h>
#include <httpext.h>

#ifndef PGEHost
#include "pge.h"
#endif

struct C_Session_ID
{
    long C_ID;
    long SessionID;

```

```

};

struct FLOATPRINT
{
    long x;
    long y;
};

typedef struct
{
    EXTENSION_CONTROL_BLOCK * pecb;
    long lTLSeed;
    long hash;
    struct C_Session_ID ID;
    char querystring[700];
    long len;
    char szContent[LEN_BUFFER];
    char * endptr;
    char szHeader[128];
    char szBuffer[LEN_BUFFER];
    char host[20];
    int host_len;
    SQLHDBC hdbc;
    SQLHSTMT hstmt;
    BOOL reconnect;
    RETCODE retcode;
    SQLINTEGER cbValue;
    int deadlock;
    SQLCHAR SqlState[6];
    SQLCHAR Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLINTEGER NativeError;
    SQLSMALLINT MsgLen;
    char odbcerrmsg[512];
    char err[128];
    struct FLOATPRINT flt1;
    struct FLOATPRINT flt2;
    struct FLOATPRINT flt3;
    struct FLOATPRINT flt4;
    struct FLOATPRINT flt5;
    PGE pge;
} TPCW;

BOOL ThreadDetach( TPCW * tpcw);
BOOL ThreadAttach( TPCW * * tpcw);
BOOL SendDebug( TPCW * tpcw, char msg[]);
void Message( char msg[]);
void ErrorMessage( TPCW * tpcw);
void ErrorLog( char ErrMsg[], char func[]);

```

#### **tpcw\_db.c**

```

// tpcw_db.c

```

```

#define DEADLOCK_RETRY 5
#define DEADLOCK_SLEEP 500
#define MAX_CONN_OUT 1024
#define TITLE_LEN 61
#define AUTH_LEN 21
#include <windows.h>
#include <sql.h>
#include <sqlext.h>
#include <stdio.h>
#include <sys/timeb.h>
#include <time.h>
#include "list.h"
#include "tpcw_db.h"

extern SQLHENV henv;
extern char hostname[ 20];
extern char domainname[ 30];
extern char cachehost[30];

void ODBCErrorHandleSTMT( TPCW * tpcw)
{
    tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
        tpcw->hstmt, 1, tpcw->SqlState,
        &tpcw->NativeError, tpcw->Msg,
        sizeof( tpcw->Msg),& tpcw->MsgLen);
    if( tpcw->retcode== SQL_NO_DATA)
    {
        wsprintf( tpcw->odbcerrmsg,"No Error code returned");
    }
    else
    {
        wsprintf( tpcw->odbcerrmsg,"SQL State: %s<br>
NativeError: %d<br>Msg: %s<br>"
            "Deadlock retry: %d",
            tpcw->SqlState,
            tpcw->NativeError,
            tpcw->Msg,
            tpcw->deadlock);
        ErrorLog( tpcw->odbcerrmsg,"SQL");
        tpcw->reconnect= TRUE;
        tpcw->expiresflag= NOW;
    }
}

void ODBCErrorHandleDBC( TPCW * tpcw)
{
    tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_DBC, tpcw->hdbc, 1,
tpcw->SqlState,& tpcw->NativeError, tpcw->Msg, sizeof( tpcw->Msg),&
tpcw->MsgLen);
    if( tpcw->retcode== SQL_NO_DATA)
    {
        wsprintf(tpcw->odbcerrmsg,"No Error code returned");
    }
}

```

```

    }
    else
    {
        wsprintf( tpcw->odbcerrmsg,"SQL State: %s<br>
NativeError: %d<br>Msg: %s<br>"
            "Deadlock retry: %d",
            tpcw->SqlState,
            tpcw->NativeError,
            tpcw->Msg,
            tpcw->deadlock);
        ErrorLog( tpcw->odbcerrmsg,"SQL");
        tpcw->reconnect= TRUE;
        tpcw->expiresflag= NOW;
    }
}

BOOL SQLInit( TPCW * tpcw)
{
    struct _timeb timeBuffer1;
    struct _timeb timeBuffer2;

    char string[ 128];
    SQLCHAR szOutConn[ MAX_CONN_OUT];
    SQLSMALLINT cbOutConn;
    tpcw->retcode = SQLAllocHandle( SQL_HANDLE_DBC, henv, &tpcw-
>hdbc);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE)
    {
        return FALSE;
    }
    strcpy( string,"DSN=tpcwDSN;UID=sa;PWD=;APP=");

    strcat( string, hostname);
    _ftime(&timeBuffer1);
    tpcw->retcode = SQLDriverConnect( tpcw->hdbc, NULL, (SQLTCHAR*)
string, sizeof(
string),szOutConn,MAX_CONN_OUT,&cbOutConn,SQL_DRIVER_NOPROMPT);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE)
    {
        ODBCErrorHandleDBC( tpcw);

        SQLFreeHandle( SQL_HANDLE_DBC, tpcw->hdbc);
        _ftime(&timeBuffer2);
    }
}

```

```

        wsprintf( tpcw->odbcerrmsg,"Elapsed SQLDriverConnect
time is %ds %dms",
                timeBuffer2.time - timeBuffer1.time,
                timeBuffer2.millitm - timeBuffer1.millitm);
        ErrorLog(tpcw->odbcerrmsg,"SQLInit");

        return FALSE;
    }
    tpcw->retcode = SQLAllocHandle( SQL_HANDLE_STMT, tpcw-
>hdbc,&tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE)
    {
        return FALSE;
    }
    tpcw->retcode = SQLExecDirect( tpcw->hstmt,"set nocount
on",SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE)
    {
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLExecDirect( tpcw->hstmt,"set XACT_ABORT ON",
SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE)
    {
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

```

```

BOOL SQLBestSellers( TPCW * tpcw, char subject[])
{
    int subject_len;
    __int64 id;
    char title[ TITLE_LEN];
    char fname[ AUTH_LEN];
    char lname[ AUTH_LEN];
    subject_len= strlen( subject);
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter(
            tpcw->hstmt,
            1,
            SQL_PARAM_INPUT,
            SQL_C_CHAR,
            SQL_CHAR,

```

```

            subject_len,
            0,
            subject,
            0,
            &subject_len);
        if( tpcw->retcode== SQL_ERROR
            || tpcw->retcode== SQL_INVALID_HANDLE
            || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{call
BestSellers(?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR
            || tpcw->retcode== SQL_INVALID_HANDLE
            || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,&tpcw-
>NativeError,
            tpcw->Msg, sizeof( tpcw->Msg),&tpcw-
>MsgLen);

            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                Sleep( DEADLOCK_SLEEP);
            }
            else
            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                return FALSE;
            }
        }
    }
    else
    {
        break;
    }
}
while( tpcw->deadlock< DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,SQL_C_CHAR, fname,
AUTH_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}

```

```

    tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,SQL_C_CHAR, lname,
AUTH_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 3, SQL_C_UBIGINT,& id,
0,&tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 4,SQL_C_CHAR, title,
TITLE_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    while (TRUE)
    {
        tpcw->retcode = SQLFetch( tpcw->hstmt);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        if( tpcw->retcode== SQL_NO_DATA)
        {
            break;
        }
        tpcw->len= sprintf( tpcw->endptr,
            "<TR>"
            "<TD><I>%s %s</I></TD>"
            "<TD><A HREF=\"http://%s/tpcw/tpcw.dll?l=%I64d\"
target=\"_top\">%s</A></TD></TR>",
            fname,lname,cachehost,id,title);
        tpcw->endptr= tpcw->endptr+ tpcw->len;
    }
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

```

```

BOOL SQLNewProducts( TPCW * tpcw, char subject[])
{
    int subject_len;
    __int64 id;
    char title[ TITLE_LEN];
    char fname[ AUTH_LEN];
    char lname[ AUTH_LEN];
    subject_len= strlen( subject);
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt,
1,SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, subject_len,0, subject, 0,
&subject_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQExecDirect( tpcw->hstmt,"{call
NewProducts(?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR
            || tpcw->retcode== SQL_INVALID_HANDLE
            || tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1,
                tpcw->SqlState,&tpcw->NativeError, tpcw-
>Msg, sizeof( tpcw->Msg),& tpcw->MsgLen);
            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                Sleep( DEADLOCK_SLEEP);
            }
            else
            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                return FALSE;
            }
        }
        else
        {
            break;
        }
    }
    }while( tpcw->deadlock< DEADLOCK_RETRY);

    tpcw->retcode = SQLBindCol( tpcw->hstmt, 1,SQL_C_CHAR, fname,
AUTH_LEN, &tpcw->cbValue);

```

```

        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,SQL_C_CHAR, lname,
AUTH_LEN, &tpcw->cbValue);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 3, SQL_C_UBIGINT,& id,
0,&tpcw->cbValue);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLBindCol( tpcw->hstmt, 4, SQL_C_CHAR, title,
TITLE_LEN, &tpcw->cbValue);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        while (TRUE)
        {
            tpcw->retcode = SQLFetch( tpcw->hstmt);
            if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
                return FALSE;
            }
            if( tpcw->retcode== SQL_NO_DATA)
            {
                break;
            }
            tpcw->len= sprintf(tpcw->endptr,
                                "<TR>"
                                "<TD><I>%s

```

```

                                "<TD><A
                                HREF=\"http://%s/tpcw/tpcw.dll?l=%I64d\"
                                target=\"_top\">%s</A></TD></TR>",
                                fname,lname,cachehost,id,title);
                                tpcw->endptr= tpcw->endptr+ tpcw->len;
                            }
                            SQLCloseCursor( tpcw->hstmt);
                            return TRUE;
                        }
                    }
                }
            }
        }
    }
}

BOOL SQLBookListProcATC001( TPCW * tpcw, char * value)
{
    int subject_len;
    long i;
    __int64 id;
    char title[ TITLE_LEN];
    char fname[ AUTH_LEN];
    char lname[ AUTH_LEN];
    subject_len= strlen( value);
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt, 1,
            SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
            subject_len,
            0, value, 0, &subject_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{call
BookListProcATC001(?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,& tpcw->NativeError, tpcw->Msg,
sizeof(tpcw->Msg),& tpcw->MsgLen);
            if( strcmp( tpcw->SqlState,"40001")== 0)
            {
                SQLCloseCursor( tpcw->hstmt);
                tpcw->deadlock++;
                Sleep( DEADLOCK_SLEEP);
            }
            else
            {
                ODBCErrorHandleSTMT( tpcw);
                SQLCloseCursor( tpcw->hstmt);
            }
        }
    }
}

```

```

        return FALSE;
    }
    else
    {
        break;
    }
}
while( tpcw->deadlock< DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,& id,
0,&tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,SQL_C_CHAR, title,
TITLE_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,SQL_C_CHAR, fname,
AUTH_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 4,SQL_C_CHAR, lname,
AUTH_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
}
i= 1;
while (TRUE)
{
    tpcw->retcode = SQLFetch( tpcw->hstmt);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {

```

```

        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    if( tpcw->retcode== SQL_NO_DATA)
    {
        break;
    }
    tpcw->len= sprintf( tpcw->endptr,"<TR>"
        "<TD><I>%s %s</I></TD>"
        "<TD><A
        HREF=\"http://%s/tpcw/tpcw.dll?l=%I64d\"
        target=\"_top\">%s</A></TD></TR>",
        fname,lname,cachehost,id,title);
    tpcw->endptr= tpcw->endptr+ tpcw->len;
}
SQLCloseCursor( tpcw->hstmt);
return TRUE;
}

BOOL SQLBookListProcATC010( TPCW * tpcw, char * value)
{
    __int64 id;
    char title[ TITLE_LEN];
    char fname[ AUTH_LEN];
    char lname[ AUTH_LEN];
    int subject_len;
    int i;
    subject_len= strlen( value);
    tpcw->deadlock= 0;
    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt,
1,SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, subject_len,0, value, 0,
&subject_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{call
BookListProcATC010(?)}", SQL_NTS);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,& tpcw->NativeError, tpcw->Msg,
sizeof(tpcw->Msg),& tpcw->MsgLen);
            if( strcmp( tpcw->SqlState,"40001")== 0)

```

```

        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            Sleep( DEADLOCK_SLEEP);
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
    else
    {
        break;
    }
}
while( tpcw->deadlock< DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,& id,
0,&tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,
SQL_C_CHAR, title, TITLE_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,
SQL_C_CHAR, fname, AUTH_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 4,
SQL_C_CHAR, lname, AUTH_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);

```

```

        return FALSE;
    }
    i= 1;
    while (TRUE)
    {
        tpcw->retcode = SQLFetch( tpcw->hstmt);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        if( tpcw->retcode== SQL_NO_DATA)
        {
            break;
        }
        tpcw->len= sprintf( tpcw->endptr,"<TR>"
            "<TD><I>%s
            %s</I></TD>"
            "<TD><A
            HREF=\"http://%s/tpcw/tpcw.dll?l=%I64d\"
            target=\"_top\">%s</A></TD></TR>",
            fname,lname,cachehost,id,title);
        tpcw->endptr= tpcw->endptr+ tpcw->len;
    }
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

BOOL SQLBookListProcATC100( TPCW * tpcw, char * value)
{
    __int64 id;
    char title[ TITLE_LEN];
    char fname[ AUTH_LEN];
    char lname[ AUTH_LEN];
    int subject_len;
    int i;
    subject_len= strlen( value);
    tpcw->deadlock= 0;

    do
    {
        tpcw->retcode = SQLBindParameter( tpcw->hstmt,
1,SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, subject_len,0, value, 0,
&subject_len);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);

```

```

        return FALSE;
    }
    tpcw->retcode = SQLExecDirect( tpcw->hstmt,"{ call
BookListProcATC100(?)", SQL_NTS);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        tpcw->retcode= SQLGetDiagRec( SQL_HANDLE_STMT,
tpcw->hstmt, 1, tpcw->SqlState,& tpcw->NativeError, tpcw->Msg, sizeof(
tpcw->Msg),& tpcw->MsgLen);
        if( strcmp( tpcw->SqlState,"40001")== 0)
        {
            SQLCloseCursor( tpcw->hstmt);
            tpcw->deadlock++;
            Sleep( DEADLOCK_SLEEP);
        }
        else
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
    }
    else
    {
        break;
    }
}
while( tpcw->deadlock< DEADLOCK_RETRY);

tpcw->retcode = SQLBindCol( tpcw->hstmt, 1, SQL_C_UBIGINT,& id,
0,&tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 2,SQL_C_CHAR, title,
TITLE_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{
    ODBCErrorHandleSTMT( tpcw);
    SQLCloseCursor( tpcw->hstmt);
    return FALSE;
}
tpcw->retcode = SQLBindCol( tpcw->hstmt, 3,
SQL_C_CHAR, fname, AUTH_LEN, &tpcw->cbValue);
if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
{

```

```

        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    tpcw->retcode = SQLBindCol( tpcw->hstmt, 4,SQL_C_CHAR, lname,
AUTH_LEN, &tpcw->cbValue);
    if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
    {
        ODBCErrorHandleSTMT( tpcw);
        SQLCloseCursor( tpcw->hstmt);
        return FALSE;
    }
    i= 1;
    while (TRUE)
    {
        tpcw->retcode = SQLFetch( tpcw->hstmt);
        if( tpcw->retcode== SQL_ERROR|| tpcw->retcode==
SQL_INVALID_HANDLE|| tpcw->retcode== SQL_SUCCESS_WITH_INFO)
        {
            ODBCErrorHandleSTMT( tpcw);
            SQLCloseCursor( tpcw->hstmt);
            return FALSE;
        }
        if( tpcw->retcode== SQL_NO_DATA)
        {
            break;
        }
        tpcw->len= sprintf( tpcw->endptr,"<TR>"
" <TD><I>%s
"s</I></TD>"
" <TD><A
HREF=\"http://%s/tpcw/tpcw.dll?l=%I64d\"
target=\"_top\">%s</A></TD></TR>",
        fname,lname,cachehost,id,title);
        tpcw->endptr= tpcw->endptr+ tpcw->len;
    }
    SQLCloseCursor( tpcw->hstmt);
    return TRUE;
}

```

#### tpcw\_db.h

```

//      tpcw_db. h
//
// Copyright Unisys, 1998,1999,2000
/* #include <windows.h>
*/ #include <sql. h>
#define AUTH_LEN 21
#define SUBJ_LEN 61
#define DESC_LEN 550
#define IMAGE_LEN 101

```

```

#define TITLE_LEN 61
#define PUB_LEN 61
#define ISBN_LEN 14
#define BACKING_LEN 16
#define DIMENSIONS_LEN 26
#define COMMENT_LEN 101
#define FNAME_LEN 16
#define LNAME_LEN 16
#define C_UNAME_LEN 21
#define C_PHONE_LEN 17
#define C_EMAIL_LEN 51
#define C_DATA_LEN 600
#define A_STREET_LEN 41
#define A_CITY_LEN 31
#define A_STATE_LEN 21
#define A_ZIP_LEN 11
#define CO_NAME_LEN 51
#define CO_CURRENCY_LEN 11
#define O_CC_TYPE_LEN 11
#define O_CC_NUM_LEN 17
#define O_CC_NAME_LEN 32
#define O_SHIP_TYPE_LEN 11
#define O_STATUS_LEN 16
#define DATE_LEN 11

typedef struct
{
    unsigned char b_title[TITLE_LEN];
    unsigned char category[SUBJ_LEN];
    long b_id;
    unsigned char b_fname[AUTH_LEN];
    unsigned char b_lname[AUTH_LEN];
    char author[AUTH_LEN];
} BOOK_LIST_ENQ_DATA;

typedef struct
{
    long i_id;
    unsigned char i_thumbnail [IMAGE_LEN];
} ALSO;

typedef struct
{
    long i_id;
    ALSO i_also[5];
    unsigned char i_title [TITLE_LEN];
    long i_a_id;
    TIMESTAMP_STRUCT i_pub_date;
    unsigned char i_publisher [PUB_LEN];
    unsigned char i_subject [SUBJ_LEN];
    unsigned char i_desc [DESC_LEN];
    unsigned char i_image [IMAGE_LEN];
    unsigned char i_thumbnail [IMAGE_LEN];

```

```

    long i_srp;
    long i_cost;
    TIMESTAMP_STRUCT i_avail;
    unsigned char i_isbn [ISBN_LEN];
    unsigned long i_page;
    unsigned char i_backing [BACKING_LEN];
    unsigned char i_dimensions [DIMENSIONS_LEN];
    unsigned char a_fname [AUTH_LEN];
    unsigned char a_lname [AUTH_LEN];
    unsigned char a_mname [AUTH_LEN];
    TIMESTAMP_STRUCT a_dob;
    unsigned char a_bio [DESC_LEN];
} BOOK_DETAILS_ENQ_DATA;

typedef struct
{
    long a_id;
    long a_c_id;
    unsigned char a_street1[A_STREET_LEN];
    unsigned char a_street2[A_STREET_LEN];
    unsigned char a_city[A_CITY_LEN];
    unsigned char a_state[A_STATE_LEN];
    unsigned char a_zip[A_ZIP_LEN];
    long a_co_id;
} ADDRESS_ENQ_DATA;

typedef struct
{
    long co_id;
    unsigned char co_name[CO_NAME_LEN];
    double co_exchange;
    unsigned char co_currency[CO_CURRENCY_LEN];
} COUNTRY_ENQ_DATA;

typedef struct
{
    long c_id;
    unsigned char c_uname [C_UNAME_LEN];
    unsigned char c_passwd [C_UNAME_LEN];
    unsigned char c_lname [FNAME_LEN];
    unsigned char c_fname [LNAME_LEN];
    long c_addr_id;
    unsigned char c_phone[C_PHONE_LEN];
    unsigned char c_email[C_EMAIL_LEN];
    TIMESTAMP_STRUCT c_since;
    TIMESTAMP_STRUCT c_last_visit;
    TIMESTAMP_STRUCT c_login;
    TIMESTAMP_STRUCT c_expiration;
    long c_discount;
    long c_balance;
    long c_ytd_pmt;
    TIMESTAMP_STRUCT c_birthdate;
    unsigned char c_data [C_DATA_LEN];

```

```

} CUSTOMER_INFO_ENQ_DATA;

typedef struct
{
long o_id;
long o_c_id;
TIMESTAMP_STRUCT o_date;
long o_subtotal;
long o_tax;
long o_total;
unsigned char o_ship_type[O_SHIP_TYPE_LEN];
TIMESTAMP_STRUCT o_ship_date;
long o_bill_addr_id;
long o_ship_addr_id;
unsigned char o_cc_type[O_CC_TYPE_LEN];
char o_cc_num[O_CC_NUM_LEN];
unsigned char o_cc_name[FNAME_LEN+ LNAME_LEN];
TIMESTAMP_STRUCT o_cc_expiration_date;
unsigned char o_status[O_STATUS_LEN];
} ORDER_ENQ_DATA;

typedef struct
{
long o_id;
long i_id;
long qty;
unsigned char comments[COMMENT_LEN];
double discounts;
} NEW_ORDER_LINE_DATA;

struct Item
{
unsigned char string[128];
};

typedef struct
{
struct Item List[50];
} Subject;

//BOOL SQLInit( TPCW *tpcw);
//BOOL SQLInitSearch( TPCW *tpcw);
//BOOL SQLCountryCheck( TPCW *tpcw, COUNTRY_ENQ_DATA *country);
//BOOL SQLGetAD( TPCW *tpcw, BOOK_DETAILS_ENQ_DATA *book,
TIMESTAMP_STRUCT *btime);

*/
/*=====
-- Code changed on 11/ 08/ 00 to meet clause 2.4.3.3 of 1.2 spec
-- This portion of code added on 11/ 08/ 00 to meet clause 2.4.3.3 of
1.2 spec
-- It was not used during the measurement intervals of the 11/ 6 thru
11/ 10 benchmark but tests were run 11/ 08/ 00 to
demonstrate to the auditor that it had no significant impact on
performance.
-- See auditor's attestation letter for details.
=====*/
// Before code changed
//=====

```

```

11/ 10 benchmark but tests were run 11/ 08/ 00 to
demonstrate to the auditor that it had no significant impact on
performance.
-- See auditor's attestation letter for details.
Before code changed
=====
=====
=====
*/
// After code changed
//=====
/* BOOL SQLGetRelated1( TPCW *tpcw, BOOK_DETAILS_ENQ_DATA *book);
//=====
BOOL SQLGetBasketItem( TPCW *tpcw, struct BASKET_LIST *Basket_List[]);
BOOL SQLGetCountryID( TPCW *tpcw, COUNTRY_ENQ_DATA *country);
BOOL SQLGetAddr( TPCW *tpcw, ADDRESS_ENQ_DATA *addr, COUNTRY_ENQ_DATA
*country);
BOOL SQLItemChange( TPCW *tpcw, BOOK_DETAILS_ENQ_DATA *book);
BOOL SQLAddNewAddress( TPCW *tpcw, ADDRESS_ENQ_DATA *shipaddress,
COUNTRY_ENQ_DATA *shipcountry);
BOOL SQLGetOrderLine( TPCW *tpcw, long o_id);
BOOL SQLGetCustOrderInfo( TPCW *tpcw, CUSTOMER_INFO_ENQ_DATA *cust,
ADDRESS_ENQ_DATA *addr, COUNTRY_ENQ_DATA *country, ORDER_ENQ_DATA
*order);
BOOL SQLGetCustInfo( TPCW *tpcw, CUSTOMER_INFO_ENQ_DATA *cust,
ADDRESS_ENQ_DATA *addr, COUNTRY_ENQ_DATA *country);
BOOL SQLGetCustAddrID( TPCW *tpcw, CUSTOMER_INFO_ENQ_DATA *cust);
BOOL SQLAddressCheck( TPCW *tpcw, COUNTRY_ENQ_DATA *country,
ADDRESS_ENQ_DATA *addr);
BOOL SQLGetCustName( TPCW *tpcw, CUSTOMER_INFO_ENQ_DATA *cust);
BOOL SQLUpdateBasketCosts( TPCW *tpcw, struct BASKET_LIST *basket);
BOOL SQLUnamePwdCheck( TPCW *tpcw, CUSTOMER_INFO_ENQ_DATA *cust);
BOOL SQLDeleteBasket( TPCW *tpcw);
BOOL SQLAddNewOrder( TPCW *tpcw, ORDER_ENQ_DATA *order, long co_id,
double discount, char auth_id[], struct BASKET_LIST *ShopBasket);
BOOL SQLAddNewCustomer( TPCW *tpcw, COUNTRY_ENQ_DATA *country,
ADDRESS_ENQ_DATA *addr, CUSTOMER_INFO_ENQ_DATA *cust);
/*=====
-- Code changed on 11/ 13/ 00 to meet clause 2.6.3.2 of 1.2 spec
-- This portion of code added on 11/ 13/ 00 to meet clause 2.6.3.2 of
1.2 spec
-- It was not used during the measurement intervals of the 11/ 6 thru
11/ 10 benchmark but tests were run 11/ 08/ 00 to
demonstrate to the auditor that it had no significant impact on
performance.
-- See auditor's attestation letter for details.
=====
=====*/
// Before code changed
//=====

```

```

// After code changed
//=====
// BOOL SQLGetCustUname( TPCW *tpcw, CUSTOMER_INFO_ENQ_DATA *cust);
//=====
//BOOL SQLBookListProcATC001(TPCW *tpcw, BOOK_LIST_ENQ_DATA *book);
//BOOL SQLBookListProcATC010(TPCW *tpcw, BOOK_LIST_ENQ_DATA *book);
//BOOL SQLBookListProcATC100(TPCW *tpcw, BOOK_LIST_ENQ_DATA *book);
/* BOOL SQLBookDetailsProc(TPCW *tpcw, BOOK_DETAILS_ENQ_DATA *book);
BOOL SQLProductUpdated(TPCW *tpcw, BOOK_DETAILS_ENQ_DATA *book);
BOOL SQLGetMaxItem (TPCW *tpcw);
BOOL SQLGetCountryID( TPCW *tpcw, COUNTRY_ENQ_DATA *country);
BOOL SQLGetMaxC_ID( TPCW *tpcw, CUSTOMER_INFO_ENQ_DATA *cust, long
range);
BOOL SQLGetMaxO_ID (TPCW *tpcw, ORDER_ENQ_DATA *order, long range);
BOOL SQLGetMaxADDR_ID (TPCW *tpcw, ADDRESS_ENQ_DATA *addr, long range);
BOOL SQLCreateBasket (TPCW *tpcw, struct BASKET_LIST *ShopBasket);
BOOL SQLAddToBasket( TPCW *tpcw, struct BASKET_ITEM *item, struct
BASKET_LIST *ShopBasket);
BOOL SQLUpdateItem( TPCW *tpcw, struct BASKET_ITEM *item, struct
BASKET_LIST *ShopBasket);
BOOL SQLDeleteItem( TPCW *tpcw, struct BASKET_ITEM *item, struct
BASKET_LIST *ShopBasket);
BOOL SQLGetMaxSessionID (TPCW *tpcw, long range);
BOOL SQLSwitchBasketItem( TPCW *tpcw, struct BASKET_LIST
*Basket_List[]);
void ODBCErrorHandleSTMT( TPCW *tpcw);
void trim( char string[]);
BOOL SQLBeginTran( TPCW *tpcw);
BOOL SQLCommitTran( TPCW *tpcw);
*/

BOOL SQLInit( TPCW *tpcw);
BOOL SQLNewProducts( TPCW *tpcw, char subject[]);
BOOL SQLBestSellers( TPCW *tpcw, char subject[]);
BOOL SQLBookListProcATC001( TPCW *tpcw, char *value);
BOOL SQLBookListProcATC010( TPCW *tpcw, char *value);
BOOL SQLBookListProcATC100( TPCW *tpcw, char *value);

```

## Appendix B - Configuration and Tuning

### Mylex e2000 2INT/4EXT 64MB RAID Controllers

Bios: 6.00.00  
 Firmware: 6.00.02  
 Miniport Driver: mac2w2k.sys 9-00-04  
 Accelerated Driver: dac2w2k.sys 9-00-04

### Database

System Information report written at: 12/18/2001 11:46:19 AM  
 [System Information]

[ Following are sub-categories of this main category ]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Advanced Server
Version	5.0.2195 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	MERCHANTSERVER
System Manufacturer	Dell Computer Corporation
System Model	PowerEdge 6400/900
System Type	X86-based PC
Processor	x86 Family 6 Model 10 Stepping 4 GenuineIntel ~900 Mhz
Processor	x86 Family 6 Model 10 Stepping 4 GenuineIntel ~900 Mhz
Processor	x86 Family 6 Model 10 Stepping 4 GenuineIntel ~900 Mhz
Processor	x86 Family 6 Model 10 Stepping 4 GenuineIntel ~900 Mhz
BIOS Version	Phoenix ROM BIOS PLUS Version 1.10 A09
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	MERCHANTSERVER\Administrator
Time Zone	Central Standard Time
Total Physical Memory	3,669,544 KB
Available Physical Memory	3,447,876 KB
Total Virtual Memory	11,269,828 KB
Available Virtual Memory	10,985,556 KB
Page File Space	7,600,284 KB
Page File	G:\pagefile.sys

[Hardware Resources]

[ Following are sub-categories of this main category ]

[Conflicts/Sharing]

Resource	Device
No conflicted/shared resources	

[DMA]

Channel	Device	Status
4	Direct memory access controller	OK
2	Standard floppy disk controller	OK

[Forced Hardware]

Device	PNP Device ID
No	Forced Hardware

[I/O]

Address Range	Device	Status
0x0000-0x03AF	PCI bus	OK
0x0000-0x03AF	Direct memory access controller	OK
0x03B0-0x03DF	PCI bus	OK
0x03B0-0x03DF	ATI Technologies Inc. 3D RAGE IIC	PCI OK
0x03E0-0x0FFF	PCI bus	OK
0xD000-0xFFFF	PCI bus	OK
0xFC00-0xFCFF	ATI Technologies Inc. 3D RAGE IIC	PCI OK
0x03C0-0x03DF	ATI Technologies Inc. 3D RAGE IIC	PCI OK
0xF800-0xF8FF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0xF400-0xF4FF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0xF000-0xF0FF	Adaptec AIC-7880 PCI SCSI Controller	OK
0xECC0-0xECFF	Intel 8255x-based PCI Ethernet Adapter (10/100)	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x02F4-0x02F7	ISAPNP Read Data Port	OK
0x0080-0x009F	Direct memory access controller	OK
0x00C0-0x00DF	Direct memory access controller	OK
0x00F0-0x00FF	Numeric data processor	OK
0x0020-0x003F	Programmable interrupt controller	OK
0x00A0-0x00BF	Programmable interrupt controller	OK
0x04D0-0x04D1	Programmable interrupt controller	OK
0x0061-0x0061	System speaker	OK
0x0040-0x005F	System timer	OK
0x03F0-0x03F5	Standard floppy disk controller	OK
0x03F7-0x03F7	Standard floppy disk controller	OK
0x0060-0x0060	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK
0x0064-0x0064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK
0x03F8-0x03FF	Communications Port (COM1)	OK
0x02F8-0x02FF	Communications Port (COM2)	OK
0x0378-0x037F	ECP Printer Port (LPT1)	OK
0x0778-0x077F	ECP Printer Port (LPT1)	OK
0x0070-0x007F	System CMOS/real time clock	OK
0x0814-0x085B	System board	OK
0x08A0-0x08AF	System board	OK
0x0C00-0x0CD7	System board	OK
0x0F50-0x0F58	System board	OK
0x00E0-0x00EF	System board	OK
0x7000-0xCFFF	PCI bus	OK
0xC000-0xCFFF	DEC 21154 PCI to PCI bridge	OK
0xCC80-0xCFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK

0xB000-0xBFFF DEC 21154 PCI to PCI bridge OK  
 0xBC80-0xBCFF Mylex eXtremeRAID 2000 Disk Array Controller OK  
 0xA000-0xAFFF DEC 21154 PCI to PCI bridge OK  
 0xAC80-0xACFF Mylex eXtremeRAID 2000 Disk Array Controller OK  
 0x5000-0x6FFF PCI bus OK

[IRQs]

IRQ Number	Device
9	Microsoft ACPI-Compliant System
17	Adaptec AIC-7899 Ultra160/m PCI SCSI Card
18	Adaptec AIC-7899 Ultra160/m PCI SCSI Card
16	Adaptec AIC-7880 PCI SCSI Controller
26	Intel 8255x-based PCI Ethernet Adapter (10/100)
13	Numeric data processor
6	Standard floppy disk controller
1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12	PS/2 Compatible Mouse
4	Communications Port (COM1)
3	Communications Port (COM2)
8	System CMOS/real time clock
15	System board
11	Standard OpenHCD USB Host Controller
22	Mylex eXtremeRAID 2000 Disk Array Controller
23	Mylex eXtremeRAID 2000 Disk Array Controller
24	Mylex eXtremeRAID 2000 Disk Array Controller

[Memory]

Range	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	ATI Technologies Inc. 3D RAGE IIC	PCI OK
0xDC000-0xEFFFF	PCI bus	OK
0xFAD00000-0xFBFFFFFF	PCI bus	OK
0xFC000000-0xFD3FFFFFF	PCI bus	OK
0xFC000000-0xFD3FFFFFF	ATI Technologies Inc. 3D RAGE IIC	PCI OK
0xFE000000-0xFFFFFFFF	PCI bus	OK
0xFBFEFF000-0xFBFFFFFF	ATI Technologies Inc. 3D RAGE IIC	PCI OK
0xFBFEF000-0xFBFFFFFF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0xFBFEFD000-0xFBFDFFF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0xFBFEFC000-0xFBFCFFF	Adaptec AIC-7880 PCI SCSI Controller	OK
0xFBFEFB000-0xFBFBFFF	Intel 8255x-based PCI Ethernet Adapter (10/100)	OK
0xFBFD00000-0xFBDFFFF	Intel 8255x-based PCI Ethernet Adapter (10/100)	OK
0x0000-0x9FFFF	System board	OK
0x100000-0xE37FFFF	System board	OK
0xF0000-0xFFFF	System board	OK
0xFEC00000-0xFEC0FFF	System board	OK
0xFEE00000-0xFEE0FFF	System board	OK
0xFFE00000-0xFFFFFFFF	System board	OK
0xFBFEFA000-0xFBFAFFF	Standard OpenHCD USB Host Controller	OK
0xE3800000-0xE78FFFF	PCI bus	OK
0xE8000000-0xF7FFFF	PCI bus	OK
0xE6800000-0xE77FFFF	DEC 21154 PCI to PCI bridge	OK

0xE6800000-0xE77FFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xF4000000-0xF7FFFF	DEC 21154 PCI to PCI bridge	OK
0xF4000000-0xF7FFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xE5800000-0xE67FFFF	DEC 21154 PCI to PCI bridge	OK
0xE5800000-0xE67FFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xF0000000-0xF3FFFFFF	DEC 21154 PCI to PCI bridge	OK
0xF0000000-0xF3FFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xE4800000-0xE57FFFF	DEC 21154 PCI to PCI bridge	OK
0xE4800000-0xE57FFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xEC000000-0xEFFFFFF	DEC 21154 PCI to PCI bridge	OK
0xEC000000-0xEFFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xF9C00000-0xFACFFF	PCI bus	OK
0xFDC00000-0xFDFFFF	PCI bus	OK

[Components]

[ Following are sub-categories of this main category ]

[Multimedia]

[ Following are sub-categories of this main category ]

[Audio Codecs]

Codec	Manufacturer	Description	Status	File	Version	Size
		Creation Date				
c:\winnt\system32\msgsm32.acm	Microsoft Corporation		OK			
	C:\WINNT\System32\MSGSM32.ACM	5.00.2134.1		22.27 KB		(22,800 bytes)
		12/7/1999 6:00:00 AM				
c:\winnt\system32\iac25_32.ax	Intel Corporation				Indeo@ audio software	
	C:\WINNT\System32\IAC25_32.AX		OK		2.05.53	195.00 KB
		12/7/1999 6:00:00 AM				(199,680 bytes)
c:\winnt\system32\lhacm.acm	Microsoft Corporation		OK			
	C:\WINNT\System32\LHACM.ACM	4.4.3385		33.27 KB		(34,064 bytes)
		8/8/2001 3:53:41 PM				
c:\winnt\system32\tssoft32.acm	DSP GROUP, INC.		OK			
	C:\WINNT\System32\TSSOFT32.ACM	1.01		9.27 KB		(9,488 bytes)
		12/7/1999 6:00:00 AM				
c:\winnt\system32\msadp32.acm	Microsoft Corporation		OK			
	C:\WINNT\System32\MSADP32.ACM	5.00.2134.1		14.77 KB		(15,120 bytes)
		12/7/1999 6:00:00 AM				
c:\winnt\system32\msg711.acm	Microsoft Corporation		OK			
	C:\WINNT\System32\MSG711.ACM	5.00.2134.1		10.27 KB		(10,512 bytes)
		12/7/1999 6:00:00 AM				
c:\winnt\system32\msg723.acm	Microsoft Corporation		OK			
	C:\WINNT\System32\MSG723.ACM	4.4.3385		106.77 KB		(109,328 bytes)
		8/8/2001 3:53:40 PM				
c:\winnt\system32\imaadp32.acm	Microsoft Corporation		OK			
	C:\WINNT\System32\IMAADP32.ACM	5.00.2134.1		16.27 KB		(16,656 bytes)
		12/7/1999 6:00:00 AM				

[Video Codecs]

Codec	Manufacturer	Description	Status	File	Version	Size
		Creation Date				

```

c:\winnt\system32\ir50_32.dll Intel Corporation      Indeo® video 5.10
OK      C:\WINNT\System32\IR50_32.DLL R.5.10.15.2.55 737.50 KB
(755,200 bytes)      12/7/1999 6:00:00 AM
c:\winnt\system32\msh261.drv Microsoft Corporation      OK
C:\WINNT\System32\MSH261.DRV 4.4.3385      163.77 KB (167,696
bytes) 8/8/2001 3:53:41 PM
c:\winnt\system32\msh263.drv Microsoft Corporation      OK
C:\WINNT\System32\MSH263.DRV 4.4.3385      252.27 KB (258,320
bytes) 8/8/2001 3:53:17 PM
c:\winnt\system32\msvidc32.dll Microsoft Corporation      OK
C:\WINNT\System32\MSVIDC32.DLL      5.00.2134.1      27.27 KB
(27,920 bytes) 12/7/1999 6:00:00 AM
c:\winnt\system32\msrle32.dll Microsoft Corporation      OK
C:\WINNT\System32\MSRLE32.DLL 5.00.2134.1      10.77 KB (11,024
bytes) 12/7/1999 6:00:00 AM
c:\winnt\system32\ir32_32.dll Intel(R) Corporation      OK
C:\WINNT\System32\IR32_32.DLL Not Available 194.50 KB (199,168
bytes) 12/7/1999 6:00:00 AM
c:\winnt\system32\iccvide.dll Radius Inc.      OK
C:\WINNT\System32\ICCVID.DLL 1.10.0.6      108.00 KB (110,592
bytes) 12/7/1999 6:00:00 AM

```

[CD-ROM]

```

Item      Value
Drive D:
Description      CD-ROM Drive
Media Loaded      True
Media Type      CD-ROM
Name      NEC CD-ROM DRIVE:466 SCSI CdRom Device
Manufacturer      (Standard CD-ROM drives)
Status      OK
Transfer Rate      987.43 kbytes/sec
SCSI Target ID      5
PNP Device ID      SCSI\CDROM&VEN_NEC&PROD_CD-
ROM_DRIVE:466&REV_1.06\4&24A1F46A&0&050

```

[Sound Device]

```

Item      Value
No sound devices

```

[Display]

```

Item      Value
Name      ATI Technologies Inc. 3D RAGE IIC PCI
PNP Device ID
      PCI\VEN_1002&DEV_4759&SUBSYS_00000000&REV_7A\3&13C0B0C5&0&20
Adapter Type      ATI 3D RAGE IIC PCI (A21), ATI Technologies Inc.
compatible
Adapter Description      ATI Technologies Inc. 3D RAGE IIC PCI
Adapter RAM      4.00 MB (4,194,304 bytes)
Installed Drivers      atiraged.dll
Driver Version      5.00.2174.1

```

```

INF File      display.inf (atirage section)
Color Planes      1
Color Table Entries      256
Resolution      1024 x 768 x 60 hertz
Bits/Pixel      8

```

[Infrared]

```

Item      Value
No infrared devices

```

[Input]

[ Following are sub-categories of this main category ]

[Keyboard]

```

Item      Value
Description      Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name      Enhanced (101- or 102-key)
Layout      00000409
PNP Device ID      ACPI\PNP0303\4&14E2F907&0
NumberOfFunctionKeys      12

```

[Pointing Device]

```

Item      Value
Hardware Type      PS/2 Compatible Mouse
Number of Buttons      2
Status      OK
PNP Device ID      ACPI\PNP0F13\4&14E2F907&0
Power Management Supported      False
Double Click Threshold      6
Handedness      Right Handed Operation

```

[Modem]

```

Item      Value
No modems

```

[Network]

[ Following are sub-categories of this main category ]

[Adapter]

```

Item      Value
Name      [00000000] Intel 8255x-based PCI Ethernet Adapter (10/100)
Adapter Type      Ethernet 802.3
Product Name      Intel 8255x-based PCI Ethernet Adapter (10/100)
Installed      True

```

PNP Device ID  
PCI\VEN\_8086&DEV\_1229&SUBSYS\_009B1028&REV\_08\3&13C0B0C5&0&40  
Last Reset 12/18/2001 2:36:21 AM  
Index 0  
Service Name E100B  
IP Address 192.1.10.32  
IP Subnet 255.255.255.0  
Default IP Gateway Not Available  
DHCP Enabled False  
DHCP Server Not Available  
DHCP Lease Expires Not Available  
DHCP Lease Obtained Not Available  
MAC Address 00:B0:D0:D0:69:12  
Service Name E100B  
IRQ Number 26  
I/O Port 0xECC0-0xECFF  
Driver c:\winnt\system32\drivers\el00bnt5.sys (80144, 4.01.67.0000)

Name [00000001] RAS Async Adapter  
Adapter Type Not Available  
Product Name RAS Async Adapter  
Installed True  
PNP Device ID Not Available  
Last Reset 12/18/2001 2:36:21 AM  
Index 1  
Service Name AsyncMac  
IP Address Not Available  
IP Subnet Not Available  
Default IP Gateway Not Available  
DHCP Enabled False  
DHCP Server Not Available  
DHCP Lease Expires Not Available  
DHCP Lease Obtained Not Available  
MAC Address Not Available  
Service Name Not Available

Name [00000002] WAN Miniport (L2TP)  
Adapter Type Not Available  
Product Name WAN Miniport (L2TP)  
Installed True  
PNP Device ID ROOT\MS\_L2TPMINIPORT\0000  
Last Reset 12/18/2001 2:36:21 AM  
Index 2  
Service Name Rasl2tp  
IP Address Not Available  
IP Subnet Not Available  
Default IP Gateway Not Available  
DHCP Enabled False  
DHCP Server Not Available  
DHCP Lease Expires Not Available  
DHCP Lease Obtained Not Available  
MAC Address Not Available  
Service Name Rasl2tp  
Driver c:\winnt\system32\drivers\rasl2tp.sys (50800, 5.00.2179.1)

Name [00000003] WAN Miniport (PPTP)  
Adapter Type Wide Area Network (WAN)  
Product Name WAN Miniport (PPTP)  
Installed True  
PNP Device ID ROOT\MS\_PPTPMINIPORT\0000  
Last Reset 12/18/2001 2:36:21 AM  
Index 3  
Service Name PptpMiniport  
IP Address Not Available  
IP Subnet Not Available  
Default IP Gateway Not Available  
DHCP Enabled False  
DHCP Server Not Available  
DHCP Lease Expires Not Available  
DHCP Lease Obtained Not Available  
MAC Address 50:50:54:50:30:30  
Service Name PptpMiniport  
Driver c:\winnt\system32\drivers\raspptp.sys (47856, 5.00.2160.1)

Name [00000004] Direct Parallel  
Adapter Type Not Available  
Product Name Direct Parallel  
Installed True  
PNP Device ID ROOT\MS\_PTMINIPORT\0000  
Last Reset 12/18/2001 2:36:21 AM  
Index 4  
Service Name Raspti  
IP Address Not Available  
IP Subnet Not Available  
Default IP Gateway Not Available  
DHCP Enabled False  
DHCP Server Not Available  
DHCP Lease Expires Not Available  
DHCP Lease Obtained Not Available  
MAC Address Not Available  
Service Name Raspti  
Driver c:\winnt\system32\drivers\raspti.sys (16880, 5.00.2146.1)

Name [00000005] WAN Miniport (IP)  
Adapter Type Not Available  
Product Name WAN Miniport (IP)  
Installed True  
PNP Device ID ROOT\MS\_NDISWANIP\0000  
Last Reset 12/18/2001 2:36:21 AM  
Index 5  
Service Name NdisWan  
IP Address Not Available  
IP Subnet Not Available  
Default IP Gateway Not Available  
DHCP Enabled False  
DHCP Server Not Available  
DHCP Lease Expires Not Available  
DHCP Lease Obtained Not Available  
MAC Address Not Available  
Service Name NdisWan

Driver c:\winnt\system32\drivers\ndiswan.sys (90768, 5.00.2184.1)

Name [00000006] Intel(R) PRO/100+ PCI Adapter  
Adapter Type Not Available  
Product Name Intel(R) PRO/100+ PCI Adapter  
Installed True  
PNP Device ID Not Available  
Last Reset 12/18/2001 2:36:21 AM  
Index 6  
Service Name E100B  
IP Address Not Available  
IP Subnet Not Available  
Default IP Gateway Not Available  
DHCP Enabled False  
DHCP Server Not Available  
DHCP Lease Expires Not Available  
DHCP Lease Obtained Not Available  
MAC Address Not Available  
Service Name Not Available

[Protocol]

Item	Value
Name	MSAFD Tcpi [TCP/IP]
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	16 bytes
MaximumMessageSize	0 bytes
MessageOriented	False
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	True
SupportsGracefulClosing	True
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD Tcpi [UDP/IP]
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	16 bytes
MaximumMessageSize	65467 bytes
MessageOriented	True
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False

SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	True

Name	RSVP UDP Service Provider
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	16 bytes
MaximumMessageSize	65467 bytes
MessageOriented	True
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	True
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	True

Name	RSVP TCP Service Provider
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	16 bytes
MaximumMessageSize	0 bytes
MessageOriented	False
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	True
SupportsExpeditedData	True
SupportsGracefulClosing	True
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpi_{905B5DF9-FB59-4802-92F1-7B4E7D38C69B}] SEQPACKET 3
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False

SupportsExpeditedData False  
SupportsGracefulClosing False  
SupportsGuaranteedBandwidth False  
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT\_Tcpip\_{905B5DF9-FB59-4802-92F1-7B4E7D38C69B}] DATAGRAM 3

ConnectionlessService True  
GuaranteesDelivery False  
GuaranteesSequencing False  
MaximumAddressSize 20 bytes  
MaximumMessageSize 64000 bytes  
MessageOriented True  
MinimumAddressSize 20 bytes  
PseudoStreamOriented False  
SupportsBroadcasting True  
SupportsConnectData False  
SupportsDisconnectData False  
SupportsEncryption False  
SupportsExpeditedData False  
SupportsGracefulClosing False  
SupportsGuaranteedBandwidth False  
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT\_Tcpip\_{1F594BC2-CBFF-447C-8C57-D6493B7BF947}] SEQPACKET 0

ConnectionlessService False  
GuaranteesDelivery True  
GuaranteesSequencing True  
MaximumAddressSize 20 bytes  
MaximumMessageSize 64000 bytes  
MessageOriented True  
MinimumAddressSize 20 bytes  
PseudoStreamOriented False  
SupportsBroadcasting False  
SupportsConnectData False  
SupportsDisconnectData False  
SupportsEncryption False  
SupportsExpeditedData False  
SupportsGracefulClosing False  
SupportsGuaranteedBandwidth False  
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT\_Tcpip\_{1F594BC2-CBFF-447C-8C57-D6493B7BF947}] DATAGRAM 0

ConnectionlessService True  
GuaranteesDelivery False  
GuaranteesSequencing False  
MaximumAddressSize 20 bytes  
MaximumMessageSize 64000 bytes  
MessageOriented True  
MinimumAddressSize 20 bytes  
PseudoStreamOriented False  
SupportsBroadcasting True  
SupportsConnectData False

SupportsDisconnectData False  
SupportsEncryption False  
SupportsExpeditedData False  
SupportsGracefulClosing False  
SupportsGuaranteedBandwidth False  
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT\_Tcpip\_{C2B518C2-A45A-428A-B0D1-0BAAA4C8EFC9}] SEQPACKET 1

ConnectionlessService False  
GuaranteesDelivery True  
GuaranteesSequencing True  
MaximumAddressSize 20 bytes  
MaximumMessageSize 64000 bytes  
MessageOriented True  
MinimumAddressSize 20 bytes  
PseudoStreamOriented False  
SupportsBroadcasting False  
SupportsConnectData False  
SupportsDisconnectData False  
SupportsEncryption False  
SupportsExpeditedData False  
SupportsGracefulClosing False  
SupportsGuaranteedBandwidth False  
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT\_Tcpip\_{C2B518C2-A45A-428A-B0D1-0BAAA4C8EFC9}] DATAGRAM 1

ConnectionlessService True  
GuaranteesDelivery False  
GuaranteesSequencing False  
MaximumAddressSize 20 bytes  
MaximumMessageSize 64000 bytes  
MessageOriented True  
MinimumAddressSize 20 bytes  
PseudoStreamOriented False  
SupportsBroadcasting True  
SupportsConnectData False  
SupportsDisconnectData False  
SupportsEncryption False  
SupportsExpeditedData False  
SupportsGracefulClosing False  
SupportsGuaranteedBandwidth False  
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT\_Tcpip\_{DC1184C9-5A16-42D0-96E5-A7A1021E0015}] SEQPACKET 2

ConnectionlessService False  
GuaranteesDelivery True  
GuaranteesSequencing True  
MaximumAddressSize 20 bytes  
MaximumMessageSize 64000 bytes  
MessageOriented True  
MinimumAddressSize 20 bytes  
PseudoStreamOriented False

SupportsBroadcasting False  
 SupportsConnectData False  
 SupportsDisconnectData False  
 SupportsEncryption False  
 SupportsExpeditedData False  
 SupportsGracefulClosing False  
 SupportsGuaranteedBandwidth False  
 SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT\_Tcpip\_{DC1184C9-5A16-42D0-96E5-A7A1021E0015}] DATAGRAM 2  
 ConnectionlessService True  
 GuaranteesDelivery False  
 GuaranteesSequencing False  
 MaximumAddressSize 20 bytes  
 MaximumMessageSize 64000 bytes  
 MessageOriented True  
 MinimumAddressSize 20 bytes  
 PseudoStreamOriented False  
 SupportsBroadcasting True  
 SupportsConnectData False  
 SupportsDisconnectData False  
 SupportsEncryption False  
 SupportsExpeditedData False  
 SupportsGracefulClosing False  
 SupportsGuaranteedBandwidth False  
 SupportsMulticasting False

[WinSock]

Item Value  
 File c:\winnt\system32\winsock.dll  
 Version 3.10  
 Size 2.80 KB (2,864 bytes)  
 File c:\winnt\system32\wsock32.dll  
 Version 5.00.2152.1  
 Size 21.27 KB (21,776 bytes)

[Ports]

[ Following are sub-categories of this main category ]

[Serial]

Item Value  
 Name COM1  
 Status OK  
 PNP Device ID ACPI\PNP0501\1  
 Maximum Input Buffer Size 0  
 Maximum Output Buffer Size False  
 Settable Baud Rate True  
 Settable Data Bits True

Settable Flow Control True  
 Settable Parity True  
 Settable Parity Check True  
 Settable Stop Bits True  
 Settable RLSD True  
 Supports RLSD True  
 Supports 16 Bit Mode False  
 Supports Special Characters False  
 Baud Rate 9600  
 Bits/Byte 8  
 Stop Bits 1  
 Parity None  
 Busy 0  
 Abort Read/Write on Error 0  
 Binary Mode Enabled -1  
 Continue Xmit on XOff 0  
 CTS Outflow Control 0  
 Discard NULL Bytes 0  
 DSR Outflow Control 0  
 DSR Sensitivity 0  
 DTR Flow Control Type Enable  
 EOF Character 0  
 Error Replace Character 0  
 Error Replacement Enabled 0  
 Event Character 0  
 Parity Check Enabled 0  
 RTS Flow Control Type Enable  
 XOff Character 19  
 XOffXmit Threshold 512  
 XOn Character 17  
 XOnXmit Threshold 2048  
 XOnXOff InFlow Control 0  
 XOnXOff OutFlow Control 0  
 IRQ Number 4  
 I/O Port 0x03F8-0x03FF  
 Driver c:\winnt\system32\drivers\serial.sys (62448, 5.00.2134.1)

Name COM2  
 Status OK  
 PNP Device ID ACPI\PNP0501\2  
 Maximum Input Buffer Size 0  
 Maximum Output Buffer Size False  
 Settable Baud Rate True  
 Settable Data Bits True  
 Settable Flow Control True  
 Settable Parity True  
 Settable Parity Check True  
 Settable Stop Bits True  
 Settable RLSD True  
 Supports RLSD True  
 Supports 16 Bit Mode False  
 Supports Special Characters False  
 Baud Rate 9600  
 Bits/Byte 8  
 Stop Bits 1

```

Parity None
Busy 0
Abort Read/Write on Error 0
Binary Mode Enabled -1
Continue XMit on XOff 0
CTS Outflow Control 0
Discard NULL Bytes 0
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled 0
Event Character 0
Parity Check Enabled 0
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Number 3
I/O Port 0x02F8-0x02FF
Driver c:\winnt\system32\drivers\serial.sys (62448, 5.00.2134.1)

```

[Parallel]

```

Item Value
Name LPT1
PNP Device ID ACPI\PNP0401\4&14E2F907&0

```

[Storage]

[ Following are sub-categories of this main category ]

[Drives]

```

Item Value
Drive A:
Description 3 1/2 Inch Floppy Drive

Drive C:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 8.48 GB (9,105,350,656 bytes)
Free Space 6.56 GB (7,048,904,704 bytes)
Volume Name
Volume Serial Number 6C7DACD6
Partition Disk #0, Partition #0
Partition Size 8.48 GB (9,105,352,704 bytes)
Starting Offset 32256 bytes

```

```

Drive Description Disk drive
Drive Manufacturer (Standard disk drives)
Drive Model SEAGATE ST318203LC SCSI Disk Device
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 1
Drive SCSTargetId 1
Drive SectorsPerTrack 63
Drive Size 18202544640 bytes
Drive TotalCylinders 2213
Drive TotalSectors 35551845
Drive TotalTracks 564315
Drive TracksPerCylinder 255

```

```

Drive G:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 8.47 GB (9,097,158,656 bytes)
Free Space 3.16 GB (3,390,701,568 bytes)
Volume Name ntfs
Volume Serial Number 2CF1CDE1
Partition Disk #0, Partition #1
Partition Size 8.47 GB (9,097,159,680 bytes)
Starting Offset Not Available
Drive Description Disk drive
Drive Manufacturer (Standard disk drives)
Drive Model SEAGATE ST318203LC SCSI Disk Device
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 1
Drive SCSTargetId 1
Drive SectorsPerTrack 63
Drive Size 18202544640 bytes
Drive TotalCylinders 2213
Drive TotalSectors 35551845
Drive TotalTracks 564315
Drive TracksPerCylinder 255

```

```

Drive L:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available

```

Partition Disk #2, Partition #0  
 Partition Size 135.59 GB (145,587,423,744 bytes)  
 Starting Offset 32256 bytes  
 Drive Description \\.\PHYSICALDRIVE2  
 Drive Manufacturer Not Available  
 Drive Model Not Available  
 Drive BytesPerSector 512  
 Drive MediaLoaded True  
 Drive MediaType Fixed hard disk media  
 Drive Partitions 1  
 Drive SCSIBus 4  
 Drive SCSILogicalUnit 0  
 Drive SCSIPort 4  
 Drive SCSTargetId 0  
 Drive SectorsPerTrack 63  
 Drive Size 145587456000 bytes  
 Drive TotalCylinders 17700  
 Drive TotalSectors 284350500  
 Drive TotalTracks 4513500  
 Drive TracksPerCylinder 255

Drive S:  
 Description Local Fixed Disk  
 Compressed False  
 File System NTFS  
 Size 3.91 GB (4,194,828,288 bytes)  
 Free Space 3.88 GB (4,170,645,504 bytes)  
 Volume Name Search  
 Volume Serial Number 687985A0  
 Partition Disk #1, Partition #0  
 Partition Size 3.91 GB (4,194,860,544 bytes)  
 Starting Offset 32256 bytes  
 Drive Description \\.\PHYSICALDRIVE1  
 Drive Manufacturer Not Available  
 Drive Model Not Available  
 Drive BytesPerSector 512  
 Drive MediaLoaded True  
 Drive MediaType Fixed hard disk media  
 Drive Partitions 2  
 Drive SCSIBus 4  
 Drive SCSILogicalUnit 0  
 Drive SCSIPort 3  
 Drive SCSTargetId 0  
 Drive SectorsPerTrack 63  
 Drive Size 363721881600 bytes  
 Drive TotalCylinders 44220  
 Drive TotalSectors 710394300  
 Drive TotalTracks 11276100  
 Drive TracksPerCylinder 255

Drive T:  
 Description Local Fixed Disk  
 Compressed Not Available  
 File System Not Available  
 Size Not Available

Free Space Not Available  
 Volume Name Not Available  
 Volume Serial Number Not Available  
 Partition Disk #1, Partition #1  
 Partition Size 334.84 GB (359,526,988,800 bytes)  
 Starting Offset Not Available  
 Drive Description \\.\PHYSICALDRIVE1  
 Drive Manufacturer Not Available  
 Drive Model Not Available  
 Drive BytesPerSector 512  
 Drive MediaLoaded True  
 Drive MediaType Fixed hard disk media  
 Drive Partitions 2  
 Drive SCSIBus 4  
 Drive SCSILogicalUnit 0  
 Drive SCSIPort 3  
 Drive SCSTargetId 0  
 Drive SectorsPerTrack 63  
 Drive Size 363721881600 bytes  
 Drive TotalCylinders 44220  
 Drive TotalSectors 710394300  
 Drive TotalTracks 11276100  
 Drive TracksPerCylinder 255

Drive U:  
 Description Local Fixed Disk  
 Compressed Not Available  
 File System Not Available  
 Size Not Available  
 Free Space Not Available  
 Volume Name Not Available  
 Volume Serial Number Not Available  
 Partition Disk #4, Partition #0  
 Partition Size 338.74 GB (363,721,849,344 bytes)  
 Starting Offset 32256 bytes  
 Drive Description \\.\PHYSICALDRIVE4  
 Drive Manufacturer Not Available  
 Drive Model Not Available  
 Drive BytesPerSector 512  
 Drive MediaLoaded True  
 Drive MediaType Fixed hard disk media  
 Drive Partitions 1  
 Drive SCSIBus 4  
 Drive SCSILogicalUnit 0  
 Drive SCSIPort 5  
 Drive SCSTargetId 0  
 Drive SectorsPerTrack 63  
 Drive Size 363721881600 bytes  
 Drive TotalCylinders 44220  
 Drive TotalSectors 710394300  
 Drive TotalTracks 11276100  
 Drive TracksPerCylinder 255

Drive W:  
 Description Local Fixed Disk

Compressed False  
 File System NTFS  
 Size 305.08 GB (327,579,926,528 bytes)  
 Free Space 45.28 GB (48,618,799,104 bytes)  
 Volume Name BACKUP  
 Volume Serial Number A4AF9836  
 Partition Disk #3, Partition #0  
 Partition Size 305.08 GB (327,579,969,024 bytes)  
 Starting Offset 32256 bytes  
 Drive Description \\.\PHYSICALDRIVE3  
 Drive Manufacturer Not Available  
 Drive Model Not Available  
 Drive BytesPerSector 512  
 Drive MediaLoaded True  
 Drive MediaType Fixed hard disk media  
 Drive Partitions 1  
 Drive SCSI Bus 4  
 Drive SCSI Logical Unit 0  
 Drive SCSI Port 4  
 Drive SCISITargetId 1  
 Drive SectorsPerTrack 63  
 Drive Size 327580001280 bytes  
 Drive TotalCylinders 39826  
 Drive TotalSectors 639804690  
 Drive TotalTracks 10155630  
 Drive TracksPerCylinder 255

[SCSI]

Item Value  
 Name Adaptec AIC-7899 Ultra160/m PCI SCSI Card  
 Caption Adaptec AIC-7899 Ultra160/m PCI SCSI Card  
 Driver adpu160m  
 Status OK  
 PNP Device ID  
 PCI\VEN\_9005&DEV\_00CF&SUBSYS\_009C1028&REV\_01\3&13C0B0C5&0&28  
 Device ID  
 PCI\VEN\_9005&DEV\_00CF&SUBSYS\_009C1028&REV\_01\3&13C0B0C5&0&28  
 Device Map Not Available  
 Index Not Available  
 Max Number Controlled Not Available  
 IRQ Number 17  
 I/O Port 0xF800-0xF8FF  
 Driver c:\winnt\system32\drivers\adpu160m.sys (64432, v3.10a)

Name Adaptec AIC-7899 Ultra160/m PCI SCSI Card  
 Caption Adaptec AIC-7899 Ultra160/m PCI SCSI Card  
 Driver adpu160m  
 Status OK  
 PNP Device ID  
 PCI\VEN\_9005&DEV\_00CF&SUBSYS\_009C1028&REV\_01\3&13C0B0C5&0&29  
 Device ID  
 PCI\VEN\_9005&DEV\_00CF&SUBSYS\_009C1028&REV\_01\3&13C0B0C5&0&29  
 Device Map Not Available

Index Not Available  
 Max Number Controlled Not Available  
 IRQ Number 18  
 I/O Port 0xF400-0xF4FF  
 Driver c:\winnt\system32\drivers\adpu160m.sys (64432, v3.10a)

Name Adaptec AIC-7880 PCI SCSI Controller  
 Caption Adaptec AIC-7880 PCI SCSI Controller  
 Driver aic78xx  
 Status OK  
 PNP Device ID  
 PCI\VEN\_9004&DEV\_8078&SUBSYS\_009C1028&REV\_02\3&13C0B0C5&0&30  
 Device ID  
 PCI\VEN\_9004&DEV\_8078&SUBSYS\_009C1028&REV\_02\3&13C0B0C5&0&30  
 Device Map Not Available  
 Index Not Available  
 Max Number Controlled Not Available  
 IRQ Number 16  
 I/O Port 0xF000-0xF0FF  
 Driver c:\winnt\system32\drivers\aic78xx.sys (56848, v2.20b)

Name Mylex eXtremeRAID 2000 Disk Array Controller  
 Caption Mylex eXtremeRAID 2000 Disk Array Controller  
 Driver dac2w2k  
 Status OK  
 PNP Device ID  
 PCI\VEN\_1069&DEV\_BA56&SUBSYS\_00401069&REV\_00\4&254DAD54&0&4040  
 Device ID  
 PCI\VEN\_1069&DEV\_BA56&SUBSYS\_00401069&REV\_00\4&254DAD54&0&4040  
 Device Map Not Available  
 Index Not Available  
 Max Number Controlled Not Available  
 IRQ Number 22  
 I/O Port 0xCC80-0xCCFF  
 Driver c:\winnt\system32\drivers\dac2w2k.sys (185584, 9.00-04)

Name Mylex eXtremeRAID 2000 Disk Array Controller  
 Caption Mylex eXtremeRAID 2000 Disk Array Controller  
 Driver dac2w2k  
 Status OK  
 PNP Device ID  
 PCI\VEN\_1069&DEV\_BA56&SUBSYS\_00401069&REV\_00\4&94A037D&0&4048  
 Device ID  
 PCI\VEN\_1069&DEV\_BA56&SUBSYS\_00401069&REV\_00\4&94A037D&0&4048  
 Device Map Not Available  
 Index Not Available  
 Max Number Controlled Not Available  
 IRQ Number 23  
 I/O Port 0xBC80-0xBCFF  
 Driver c:\winnt\system32\drivers\dac2w2k.sys (185584, 9.00-04)

Name Mylex eXtremeRAID 2000 Disk Array Controller  
 Caption Mylex eXtremeRAID 2000 Disk Array Controller  
 Driver dac2w2k  
 Status OK

PNP Device ID  
 PCI\VEN\_1069&DEV\_BA56&SUBSYS\_00401069&REV\_00\4&15BEFF34&0&4050  
 Device ID  
 PCI\VEN\_1069&DEV\_BA56&SUBSYS\_00401069&REV\_00\4&15BEFF34&0&4050  
 Device Map Not Available  
 Index Not Available  
 Max Number Controlled Not Available  
 IRQ Number 24  
 I/O Port 0xAC80-0xACFF  
 Driver c:\winnt\system32\drivers\dac2w2k.sys (185584, 9.00-04)

[Printing]

Name Port Name Server Name  
 No printing information

[Problem Devices]

Device	PNP Device ID	Error Code
Dell 12 BAY U2W CU SCSI Processor Device	SCSI\PROCESSOR&VEN_DELL&PROD_12_BAY_U2W_CU&REV_0209\5&2CB960EC&0&2F0	28
Dell 12 BAY U2W CU SCSI Processor Device	SCSI\PROCESSOR&VEN_DELL&PROD_12_BAY_U2W_CU&REV_0209\5&3CAF76E&0&1F0	28
Dell 12 BAY U2W CU SCSI Processor Device	SCSI\PROCESSOR&VEN_DELL&PROD_12_BAY_U2W_CU&REV_0209\5&3CAF76E&0&2F0	28
Dell 12 BAY U2W CU SCSI Processor Device	SCSI\PROCESSOR&VEN_DELL&PROD_12_BAY_U2W_CU&REV_0209\5&3CAF76E&0&3F0	28

[USB]

Device	PNP Device ID
Standard OpenHCD USB Host Controller	PCI\VEN_1166&DEV_0220&SUBSYS_02201166&REV_04\3&13C0B0C5&0&7A
USB Root Hub	USB\ROOT_HUB\4&6EF0984&0

[Software Environment]

[ Following are sub-categories of this main category ]

[Drivers]

Name	Description	File	Type	Started	Start Mode	State
abiosdsk	Abiosdsk		Not Available	Kernel Driver	False	False
abp480n5	abp480n5		Not Available	Kernel Driver	False	False
acpi	Microsoft ACPI Driver	c:\winnt\system32\drivers\acpi.sys	Kernel Driver	True	Boot Running OK	Normal False True

acpiec	ACPIEC	c:\winnt\system32\drivers\acpiec.sys	Kernel Driver	False	Disabled	Stopped OK	Normal False False	
adpu160m	adpu160m	c:\winnt\system32\drivers\adpu160m.sys	Kernel Driver	True	Boot Running OK	Normal False True		
afd	AFD Networking Support Environment	c:\winnt\system32\drivers\afd.sys	Kernel Driver	True	Auto Running OK	Normal False True		
ahal154x	Ahal154x	Not Available	Kernel Driver	False	Disabled	Stopped OK	Normal False False	
aic116x	aic116x	Not Available	Kernel Driver	False	Disabled	Stopped OK	Normal False False	
aic78u2	aic78u2	Not Available	Kernel Driver	False	Disabled	Stopped OK	Normal False False	
aic78xx	aic78xx	c:\winnt\system32\drivers\aic78xx.sys	Kernel Driver	True	Boot Running OK	Normal False True		
ami0nt	ami0nt	Not Available	Kernel Driver	False	Disabled	Stopped OK	Normal False False	
amsint	amsint	Not Available	Kernel Driver	False	Disabled	Stopped OK	Normal False False	
asc	asc	Not Available	Kernel Driver	False	Disabled	Stopped OK	Normal False False	
asc3350p	asc3350p	Not Available	Kernel Driver	False	False	Disabled	Stopped OK	Normal False False
asc3550	asc3550	Not Available	Kernel Driver	False	Disabled	Stopped OK	Normal False False	
asynctac	RAS Asynchronous Media Driver	c:\winnt\system32\drivers\asynctac.sys	Kernel Driver	False	Manual Stopped OK	Normal False False		
atapi	atapi	c:\winnt\system32\drivers\atapi.sys	Kernel Driver	False	Disabled	Stopped OK	Normal False False	
atdisk	Atdisk	Not Available	Kernel Driver	False	Disabled	Stopped OK	Ignore False False	
atirage	atirage	c:\winnt\system32\drivers\atiragem.sys	Kernel Driver	True	Manual Running OK	Ignore False True		
atmarpc	ATM ARP Client Protocol	c:\winnt\system32\drivers\atmarpc.sys	Kernel Driver	False	Manual Stopped OK	Normal False False		
audstub	Audio Stub Driver	c:\winnt\system32\drivers\audstub.sys	Kernel Driver	True	Manual Running OK	Normal False True		
beep	Beep	c:\winnt\system32\drivers\beep.sys	Kernel Driver	True	System Running OK	Normal False True		
buslogic	BusLogic	Not Available	Kernel Driver	False	Disabled	Stopped OK	Normal False False	
cd20xrnt	cd20xrnt	Not Available	Kernel Driver	False	Disabled	Stopped OK	Normal False False	
cdaudio	Cdaudio	c:\winnt\system32\drivers\cdaudio.sys	Kernel Driver	False	System Stopped OK	Ignore False False		
cdfs	Cdfs	c:\winnt\system32\drivers\cdfs.sys	File System Driver	True	Disabled	Running OK	Normal False True	
cdrom	CD-ROM Driver	c:\winnt\system32\drivers\cdrom.sys	Kernel Driver	True	System Running OK	Normal False True		
changer	Changer	Not Available	Kernel Driver	False	System Stopped OK	Ignore False False		





```

serenum Serenum Filter Driver c:\winnt\system32\drivers\serenum.sys
Kernel Driver True Manual Running OK Normal False
True
serial Serial port driver c:\winnt\system32\drivers\serial.sys
Kernel Driver True System Running OK Ignore False
True
sfloppy Sfloppy c:\winnt\system32\drivers\sfloppy.sys Kernel Driver
False System Stopped OK Ignore False False
sglfb sglfb Not Available Kernel Driver False System Stopped OK
Normal False False
simbad Simbad Not Available Kernel Driver False Disabled
Stopped OK Normal False False
sparrow Sparrow Not Available Kernel Driver False Disabled
Stopped OK Normal False False
srv Srv c:\winnt\system32\drivers\srv.sys File System Driver
True Manual Running OK Normal False True
swenum Software Bus Driver c:\winnt\system32\drivers\swenum.sys
Kernel Driver True Manual Running OK Normal False
True
sync810 sync810 Not Available Kernel Driver False Disabled
Stopped OK Normal False False
sync8xx sync8xx Not Available Kernel Driver False Disabled
Stopped OK Normal False False
sym_hi sym_hi Not Available Kernel Driver False Disabled
Stopped OK Normal False False
tcPIP TCP/IP Protocol Driver c:\winnt\system32\drivers\tcPIP.sys
Kernel Driver True System Running OK Normal False
True
tdasync TDASYNC c:\winnt\system32\drivers\tdasync.sys Kernel Driver
False Manual Stopped OK Ignore False False
tdipx TDIPX c:\winnt\system32\drivers\tdipx.sys Kernel Driver
False Manual Stopped OK Ignore False False
tdnetb TDNETB c:\winnt\system32\drivers\tdnetb.sys Kernel Driver
False Manual Stopped OK Ignore False False
tdpipe TDPIPE c:\winnt\system32\drivers\tdpipe.sys Kernel Driver
False Manual Stopped OK Ignore False False
tdspX TDS PX c:\winnt\system32\drivers\tdspX.sys Kernel Driver
False Manual Stopped OK Ignore False False
tdtcp TDTCP c:\winnt\system32\drivers\tdtcp.sys Kernel Driver
False Manual Stopped OK Ignore False False
termdd Terminal Device Driver c:\winnt\system32\drivers\termdd.sys
Kernel Driver False Disabled Stopped OK Normal
False False
tga TGA Not Available Kernel Driver False System Stopped OK
Ignore False False
udfs Udfs c:\winnt\system32\drivers\udfs.sys File System Driver
False Disabled Stopped OK Normal False False
ultra66 ultra66 Not Available Kernel Driver False Disabled
Stopped OK Normal False False
update Microcode Update Driver
c:\winnt\system32\drivers\update.sys Kernel Driver True
Manual Running OK Normal False True
usbhub Microsoft USB Standard Hub Driver
c:\winnt\system32\drivers\usbhub.sys Kernel Driver True
Manual Running OK Normal False True

```

```

vgasave VgaSave c:\winnt\system32\drivers\vga.sys Kernel Driver
True System Running OK Ignore False True
wanarp Remote Access IP ARP Driver
c:\winnt\system32\drivers\wanarp.sys Kernel Driver True
Manual Running OK Normal False True
wdica WDICA Not Available Kernel Driver False Manual Stopped OK
Ignore False False

```

[Environment Variables]

```

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
HOME C:/ <SYSTEM>
NUMBER_OF_PROCESSORS 4 <SYSTEM>
OS Windows_NT <SYSTEM>
Os2LibPath %SystemRoot%\system32\os2\dll; <SYSTEM>
Path
C:\mks\mksnt;C:\WINNT\system32;C:\WINNT;C:\WINNT\System32\Wbem;
C:\Program Files\Microsoft SQL Server\MSSQL\Binn;. ;C:\Program
Files\Microsoft SQL Server\80\Tools\BINN <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 6 Model 10 Stepping 4, GenuineIntel
<SYSTEM>
PROCESSOR_LEVEL 6 <SYSTEM>
PROCESSOR_REVISION 0a04 <SYSTEM>
ROOTDIR C:/mks <SYSTEM>
SHELL C:/mks/mksnt/sh.exe <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
TMPDIR C:/WINNT/TEMP <SYSTEM>
windir %SystemRoot% <SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp MERCHANTSERVER\admin
TMP %USERPROFILE%\Local Settings\Temp MERCHANTSERVER\admin
TEMP %USERPROFILE%\Local Settings\Temp
MERCHANTSERVER\Administrator
TMP %USERPROFILE%\Local Settings\Temp
MERCHANTSERVER\Administrator

```

[Jobs]

[ Following are sub-categories of this main category ]

[Print]

Document	Size	Owner	Notify	Status	Time Submitted	Start Time
Until Time	Elapsed Time	Pages Printed	Job ID	Priority		
Queue	Data Type	Name	Print Processor	Host	Print	
Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown
Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown
Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown

[Network Connections]

Local Name Remote Name Type Status User Name  
 No network connections information

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Max Working Set
Working Set	Start Time	Version	Size	File Date	
system idle process	Not Available	0	0	Not Available	Not Available
system	Not Available	8	8	0	1413120 Not Available
Unknown Unknown Unknown					
smss.exe	c:\winnt\system32\smss.exe	172	11	204800	
1413120	12/18/2001 8:36:59 AM	5.00.2170.1	44.27 KB	(45,328 bytes)	12/7/1999 6:00:00 AM
csrss.exe	Not Available	196	13	Not Available	Not Available
12/18/2001 8:37:02 AM	Unknown	Unknown	Unknown		
winlogon.exe	c:\winnt\system32\winlogon.exe	216	13		
204800	1413120 12/18/2001 8:37:04 AM	5.00.2182.1	173.27 KB	(177,424 bytes)	12/7/1999 6:00:00 AM
services.exe	c:\winnt\system32\services.exe	244	9		
204800	1413120 12/18/2001 8:37:05 AM	5.00.2134.1	86.77 KB	(88,848 bytes)	12/7/1999 6:00:00 AM
lsass.exe	c:\winnt\system32\lsass.exe	256	13	204800	
1413120	12/18/2001 8:37:05 AM	5.00.2184.1	32.77 KB	(33,552 bytes)	12/7/1999 6:00:00 AM
svchost.exe	c:\winnt\system32\svchost.exe	376	8	204800	
1413120	12/18/2001 8:37:07 AM	5.00.2134.1	7.77 KB	(7,952 bytes)	12/7/1999 6:00:00 AM
msdtc.exe	c:\winnt\system32\msdtc.exe	420	8	204800	
1413120	12/18/2001 8:37:09 AM	1999.9.3421.3	6.77 KB	(6,928 bytes)	8/8/2001 10:51:23 AM
gamscm.exe	c:\winnt\system32\gamscm.exe	588	8		
204800	1413120 12/18/2001 8:37:12 AM	Not Available	134.72 KB	(137,956 bytes)	12/16/2001 10:26:18 AM
regsvc.exe	c:\winnt\system32\regsvc.exe	612	8	204800	
1413120	12/18/2001 8:37:12 AM	5.00.2155.1	65.27 KB	(66,832 bytes)	12/7/1999 6:00:00 AM
gamserv.exe	c:\winnt\system32\gamserv\gamserv.exe	640	13		
204800	1413120 12/18/2001 8:37:12 AM	Not Available	160.00 KB	(163,841 bytes)	12/16/2001 10:26:18 AM
gamevent.exe	c:\winnt\system32\gamserv\gamevent.exe	648	13		
204800	1413120 12/18/2001 8:37:12 AM	Not Available	105.33 KB	(107,858 bytes)	12/16/2001 10:26:18 AM
gamevlog.exe	c:\winnt\system32\gamserv\gamevlog.exe	656	13		
204800	1413120 12/18/2001 8:37:12 AM	Not Available	227.97 KB	(233,442 bytes)	12/16/2001 10:26:18 AM
winmgmt.exe	c:\winnt\system32\wbem\winmgmt.exe	712	8		
204800	1413120 12/18/2001 8:37:13 AM	1.50.1085.0001	188.05 KB	(192,567 bytes)	12/7/1999 6:00:00 AM
winvnc.exe	c:\program files\orl\vnc\winvnc.exe	724	8		
204800	1413120 12/18/2001 8:37:14 AM	3, 3, 3, 7	204.00 KB	(208,896 bytes)	10/30/2001 9:27:51 AM
mssearch.exe	c:\program files\common files\system\mssearch\bin\mssearch.exe	760	8	204800	

1413120	12/18/2001 8:37:18 AM	9.107.6223.2	64.00 KB	(65,536 bytes)	5/10/2001 12:32:30 PM
explorer.exe	c:\winnt\explorer.exe	816	8	204800	1413120
12/18/2001 8:37:21 AM	5.00.2920.0000	232.77 KB	(238,352 bytes)		12/7/1999 6:00:00 AM
svchost.exe	c:\winnt\system32\svchost.exe	920	8	204800	
1413120	12/18/2001 8:37:26 AM	5.00.2134.1	7.77 KB	(7,952 bytes)	12/7/1999 6:00:00 AM
sqlmangr.exe	c:\program files\microsoft sql server\80\tools\bin\sqlmangr.exe	956	8	204800	1413120
12/18/2001 8:37:26 AM	2000.080.0382.00	72.57 KB	(74,308 bytes)		10/22/2001 9:19:50 AM
svchost.exe	c:\winnt\system32\svchost.exe	1100	8	204800	
1413120	12/18/2001 8:37:35 AM	5.00.2134.1	7.77 KB	(7,952 bytes)	12/7/1999 6:00:00 AM
mmc.exe	c:\winnt\system32\mmc.exe	1088	8	204800	1413120
12/18/2001 11:45:07 AM	5.00.2153.1	589.27 KB	(603,408 bytes)		12/7/1999 6:00:00 AM
rsvp.exe	c:\winnt\system32\rsvp.exe	1208	8	204800	
1413120	12/18/2001 11:45:57 AM	5.00.2167.1	172.77 KB	(176,912 bytes)	12/7/1999 6:00:00 AM

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
traffic.dll	5.00.2139.1	30.77 KB	(31,504 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\traffic.dll				
rsvp.exe	5.00.2167.1	172.77 KB	(176,912 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\rsvp.exe				
wbemprox.dll	1.50.1085.0001	40.05 KB	(41,016 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\wbem\wbemprox.dll				
mlang.dll	5.00.2920.0000	510.77 KB	(523,024 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\mlang.dll				
rassapi.dll	5.00.2188.1	14.27 KB	(14,608 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\rassapi.dll				
adsnt.dll	5.00.2191.1	194.27 KB	(198,928 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\adsnt.dll				
dbghelp.dll	5.00.2195.1	159.27 KB	(163,088 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\dbghelp.dll				
localesec.dll	5.00.2134.1	227.27 KB	(232,720 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\localesec.dll				
devmgr.dll	5.00.2166.1	215.77 KB	(220,944 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\devmgr.dll				
filemgmt.dll	5.00.2134.1	287.27 KB	(294,160 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\filemgmt.dll				
pdh.dll	5.00.2174.1	143.27 KB	(146,704 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\pdh.dll				
smlogcfg.dll	5.00.2163.1	273.27 KB	(279,824 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\smlogcfg.dll				
cabinet.dll	5.00.2147.1	54.77 KB	(56,080 bytes)		12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\cabinet.dll				
msinfo32.dll	5.00.2177.1	312.27 KB	(319,760 bytes)		8/8/2001 3:53:38 PM
Microsoft Corporation	c:\program files\common files\microsoft shared\msinfo\msinfo32.dll				

riched20.dll	5.30.23.1200	421.27 KB (431,376 bytes)	12/7/1999	sqlsvr.dll	2000.080.0194.00	24.00 KB (24,576 bytes)	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\riched20.dll		10/22/2001 9:19:48 AM	Microsoft Corporation	c:\program	
riched32.dll	5.00.2134.1	3.77 KB (3,856 bytes)	12/7/1999 6:00:00 AM	files\microsoft sql server\80\tools\bin\resources\1033\sqlsvr.dll			
AM	Microsoft Corporation	c:\winnt\system32\riched32.dll		odbcint.dll	3.520.7326.0	88.00 KB (90,112 bytes)	8/16/2001
els.dll	5.00.2175.1	151.27 KB (154,896 bytes)	12/7/1999 6:00:00 AM	2:29:46 PM	Microsoft Corporation	c:\winnt\system32\odbcint.dll	
AM	Microsoft Corporation	c:\winnt\system32\els.dll		sqlresld.dll	2000.080.0382.00	28.56 KB (29,248 bytes)	
ntmsmgr.dll	1,0,0,1 427.77 KB (438,032 bytes)	12/7/1999 6:00:00 AM		10/22/2001 9:19:48 AM	Microsoft Corporation	c:\program	
AM	Microsoft Corporation and HighGround Systems, Inc.	c:\winnt\system32\ntmsmgr.dll		files\microsoft sql server\80\tools\bin\sqlresld.dll			
mmfutil.dll	1.50.1085.0000	32.06 KB (32,829 bytes)	12/7/1999	odbcbcpl.dll	2000.080.0380.00	28.57 KB (29,252 bytes)	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\mmfutil.dll		8/16/2001 2:29:56 PM	Microsoft Corporation		
logdrive.dll	1.50.1085.0000	200.06 KB (204,863 bytes)	12/7/1999	c:\winnt\system32\odbcbcpl.dll			
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\logdrive.dll		sqlsvr.dll	2000.080.0382.00	92.56 KB (94,784 bytes)	
dfrgres.dll	5.00.2150.1	27.50 KB (28,160 bytes)	12/7/1999	10/22/2001 9:19:48 AM	Microsoft Corporation	c:\program	
6:00:00 AM	Executive Software International, Inc.	c:\winnt\system32\dfrgres.dll		files\microsoft sql server\80\tools\bin\sqlsvr.dll			
dfrgsnap.dll	5.00.2150.1	41.77 KB (42,768 bytes)	12/7/1999	odbc32.dll	3.520.7326.0	216.27 KB (221,456 bytes)	8/16/2001
6:00:00 AM	Executive Software International, Inc.	c:\winnt\system32\dfrgsnap.dll		2:29:45 PM	Microsoft Corporation	c:\winnt\system32\odbc32.dll	
dmskres.dll	2191.1.296.2	119.00 KB (121,856 bytes)	12/7/1999	w95scm.dll	2000.080.0194.00	48.56 KB (49,728 bytes)	
6:00:00 AM	Microsoft Corp., VERITAS Software	c:\winnt\system32\dmskres.dll		10/22/2001 9:19:48 AM	Microsoft Corporation	c:\program	
dmutil.dll	2191.1.296.2	41.77 KB (42,768 bytes)	12/7/1999	files\microsoft sql server\80\tools\bin\w95scm.dll			
6:00:00 AM	VERITAS Software Corp.	c:\winnt\system32\dmutil.dll		comdlg32.dll	5.00.2920.0000	236.77 KB (242,448 bytes)	12/7/1999
ntmsapi.dll	5.00.1948.1	50.27 KB (51,472 bytes)	12/7/1999	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\comdlg32.dll	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntmsapi.dll		sqlunirl.dll	2000.080.0380.00	176.56 KB (180,800 bytes)	
dmskmgr.dll	2191.1.296.2	158.77 KB (162,576 bytes)	12/7/1999	4/9/2001 10:46:18 AM	Microsoft Corporation		
6:00:00 AM	Microsoft Corp., VERITAS Software	c:\winnt\system32\dmskmgr.dll		c:\winnt\system32\sqlunirl.dll			
mycomput.dll	5.00.2134.1	107.77 KB (110,352 bytes)	12/7/1999	sqlmgr.exe	2000.080.0382.00	72.57 KB (74,308 bytes)	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\mycomput.dll		10/22/2001 9:19:50 AM	Microsoft Corporation	c:\program	
mmcndmgr.dll	5.00.2178.1	815.27 KB (834,832 bytes)	12/7/1999	files\microsoft sql server\80\tools\bin\sqlmgr.exe			
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\mmcndmgr.dll		ipbootp.dll	5.00.2168.1	33.77 KB (34,576 bytes)	12/7/1999
mfc42u.dll	6.00.8665.0	972.05 KB (995,384 bytes)	12/7/1999	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\ipbootp.dll	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\mfc42u.dll		rastls.dll	5.00.2188.1	47.77 KB (48,912 bytes)	12/7/1999
mmc.exe	5.00.2153.1	589.27 KB (603,408 bytes)	12/7/1999 6:00:00 AM	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\rastls.dll	
AM	Microsoft Corporation	c:\winnt\system32\mmc.exe		raschap.dll	5.00.2175.1	34.77 KB (35,600 bytes)	12/7/1999
h323.tsp	5.00.2143.1	245.27 KB (251,152 bytes)	12/7/1999	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\raschap.dll	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\h323.tsp		ntlsapi.dll	5.00.2134.1	6.77 KB (6,928 bytes)	12/7/1999 6:00:00 AM
ipconf.tsp	5.00.2143.1	10.77 KB (11,024 bytes)	12/7/1999	AM	Microsoft Corporation	c:\winnt\system32\ntlsapi.dll	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\ipconf.tsp		rasppp.dll	5.00.2175.1	192.27 KB (196,880 bytes)	12/7/1999
ndptsp.tsp	5.00.2143.1	38.27 KB (39,184 bytes)	12/7/1999	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\rasppp.dll	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\ndptsp.tsp		rastapi.dll	5.00.2188.1	52.27 KB (53,520 bytes)	12/7/1999
kmddsp.tsp	5.00.2150.1	17.77 KB (18,192 bytes)	12/7/1999	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\rastapi.dll	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\kmddsp.tsp		rasdlg.dll	5.00.2194.1	514.27 KB (526,608 bytes)	12/7/1999
uniplat.dll	5.00.2151.1	13.77 KB (14,096 bytes)	12/7/1999	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\rasdlg.dll	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\uniplat.dll		netcfgx.dll	5.00.2175.1	533.77 KB (546,576 bytes)	12/7/1999
unimdm.tsp	5.00.2175.1	196.77 KB (201,488 bytes)	12/7/1999	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\netcfgx.dll	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\unimdm.tsp		sens.dll	5.00.2163.1	36.77 KB (37,648 bytes)	12/7/1999
tapisrv.dll	5.00.2186.1	168.77 KB (172,816 bytes)	12/7/1999	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\sens.dll	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\tapisrv.dll		rasmans.dll	5.00.2188.1	146.77 KB (150,288 bytes)	12/7/1999
sqlmgr.rll	2000.080.0194.00	96.00 KB (98,304 bytes)		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\rasmans.dll	
	10/22/2001 9:19:50 AM	Microsoft Corporation	c:\program	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\netman.dll	
files\microsoft sql server\80\tools\bin\resources\1033\sqlmgr.rll				es.dll	1999.9.3422.21	231.77 KB (237,328 bytes)	12/7/1999 6:00:00 AM
				AM	Microsoft Corporation	c:\winnt\system32\es.dll	
				wininet.dll	5.00.2920.0000	456.77 KB (467,728 bytes)	12/7/1999
				6:00:00 AM	Microsoft Corporation	c:\winnt\system32\wininet.dll	
				shdoclc.dll	5.00.2920.0000	324.50 KB (332,288 bytes)	12/7/1999
				6:00:00 AM	Microsoft Corporation	c:\winnt\system32\shdoclc.dll	

urlmon.dll	5.00.2920.0000	426.77 KB (437,008 bytes)	12/7/1999	mssrch.dll	9.107.6223.2	1.23 MB (1,290,240 bytes)	5/10/2001
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\urlmon.dll		12:32:28 PM	Microsoft Corporation		
browsei.c	5.00.2920.0000	34.50 KB (35,328 bytes)	12/7/1999		c:\progra~1\common~1\system\mssearch\bin\mssrch.dll		
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\browsei.c		mssws.dll	9.107.6223.2	36.00 KB (36,864 bytes)	5/10/2001
mydocs.dll	5.00.2920.0000	55.77 KB (57,104 bytes)	12/7/1999	12:32:28 PM	Microsoft Corporation	c:\program files\common	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\mydocs.dll		files\system\mssearch\bin\mssws.dll			
faxshell.dll	5.00.2134.1	8.27 KB (8,464 bytes)	12/7/1999	mssearch.exe	9.107.6223.2	64.00 KB (65,536 bytes)	5/10/2001
AM	Microsoft Corporation	c:\winnt\system32\faxshell.dll		12:32:30 PM	Microsoft Corporation	c:\program files\common	
msacm32.dll	5.00.2134.1	65.27 KB (66,832 bytes)	12/7/1999	files\system\mssearch\bin\mssearch.exe			
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\msacm32.dll		omnithread_rt.dll	Not Available	44.00 KB (45,056 bytes)	
avifil32.dll	5.00.2134.1	76.27 KB (78,096 bytes)	12/7/1999		10/30/2001	9:27:51 AM	Not Available
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\avifil32.dll		c:\winnt\system32\omnithread_rt.dll			
msvfw32.dll	5.00.2134.1	113.77 KB (116,496 bytes)	12/7/1999	vnchooks.dll	3, 3, 3, 6	32.00 KB (32,768 bytes)	10/30/2001
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\msvfw32.dll		9:27:51 AM	AT&T Research Labs Cambridge	c:\program	
docprop2.dll	5.00.2178.1	297.77 KB (304,912 bytes)	12/7/1999	files\orl\vnc\vnchooks.dll			
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\docprop2.dll		winvnc.exe	3, 3, 3, 7	204.00 KB (208,896 bytes)	10/30/2001
ntshrui.dll	5.00.2134.1	46.77 KB (47,888 bytes)	12/7/1999	9:27:51 AM	AT&T Research Labs Cambridge	c:\program	
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntshrui.dll		files\orl\vnc\winvnc.exe			
linkinfo.dll	5.00.2134.1	15.77 KB (16,144 bytes)	12/7/1999	netui1.dll	5.00.2134.1	210.27 KB (215,312 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\linkinfo.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\netui1.dll	
powrprof.dll	5.00.2920.0000	13.27 KB (13,584 bytes)	12/7/1999	netui0.dll	5.00.2134.1	70.27 KB (71,952 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\powrprof.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\netui0.dll	
batmeter.dll	5.00.2920.0000	20.27 KB (20,752 bytes)	12/7/1999	ntlanman.dll	5.00.2157.1	35.27 KB (36,112 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\batmeter.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntlanman.dll	
stobject.dll	5.00.2144.1	81.77 KB (83,728 bytes)	12/7/1999	wshnetbs.dll	5.00.2134.1	7.77 KB (7,952 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\stobject.dll		AM	Microsoft Corporation	c:\winnt\system32\wshnetbs.dll	
msi.dll	1.10.1029.0	1.71 MB (1,794,320 bytes)	12/7/1999	rapilib.dll	5.00.2167.1	25.27 KB (25,872 bytes)	12/7/1999
AM	Microsoft Corporation	c:\winnt\system32\msi.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\rapilib.dll	
webcheck.dll	5.00.2920.0000	251.77 KB (257,808 bytes)	12/7/1999	rsvsp.dll	5.00.2167.1	74.77 KB (76,560 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\webcheck.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\rsvsp.dll	
netshell.dll	5.00.2176.1	456.77 KB (467,728 bytes)	12/7/1999	ntmarta.dll	5.00.2158.1	98.77 KB (101,136 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\netshell.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntmarta.dll	
browseui.dll	5.00.2920.0000	793.27 KB (812,304 bytes)	12/7/1999	provthrd.dll	1.50.1085.0000	68.07 KB (69,708 bytes)	8/8/2001
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\browseui.dll		3:53:30 PM	Microsoft Corporation		
shdocvw.dll	5.00.2920.0000	1.05 MB (1,104,144 bytes)	12/7/1999	c:\winnt\system32\wbem\provthrd.dll			
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\shdocvw.dll		ntevt.dll	1.50.1085.0000	192.06 KB (196,669 bytes)	12/7/1999
explorer.exe	5.00.2920.0000	232.77 KB (238,352 bytes)	12/7/1999	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\ntevt.dll	
6:00:00 AM	Microsoft Corporation	c:\winnt\explorer.exe		wmi.dll	5.00.2191.1	6.27 KB (6,416 bytes)	12/7/1999
iprop.dll	5.00.2181.1	4.27 KB (4,368 bytes)	12/7/1999	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\wmi.dll	
AM	Microsoft Corporation	c:\winnt\system32\iprop.dll		psapi.dll	5.00.2134.1	28.27 KB (28,944 bytes)	12/7/1999
srchidx.dll	9.107.6223.2	380.00 KB (389,120 bytes)	5/10/2001	6:00:00 AM	Microsoft Corporation	c:\winnt\system32\psapi.dll	
12:32:28 PM	Microsoft Corporation			framedyn.dll	1.50.1085.0000	164.05 KB (167,992 bytes)	12/7/1999
		c:\progra~1\common~1\system\mssearch\bin\srchidx.dll		6:00:00 AM	Microsoft Corporation		
propdefs.dll	9.107.6223.2	136.00 KB (139,264 bytes)	5/10/2001	c:\winnt\system32\wbem\framedyn.dll			
12:32:28 PM	Microsoft Corporation			cimwin32.dll	1.50.1085.0000	1.03 MB (1,077,306 bytes)	12/7/1999
		c:\progra~1\common~1\system\mssearch\bin\propdefs.dll		6:00:00 AM	Microsoft Corporation		
lcdetect.dll	9.107.6223.2	28.00 KB (28,672 bytes)	5/10/2001	c:\winnt\system32\wbem\cimwin32.dll			
12:32:28 PM	Microsoft Corporation	c:\program files\common		wbemsvc.dll	1.50.1085.0000	140.07 KB (143,430 bytes)	12/7/1999
files\system\mssearch\bin\lcdetect.dll				6:00:00 AM	Microsoft Corporation		
tquery.dll	9.107.6223.2	1.45 MB (1,515,520 bytes)	5/10/2001	c:\winnt\system32\wbem\wbemsvc.dll			
12:34:40 PM	Microsoft Corporation	c:\program files\common		wbemess.dll	1.50.1085.0001	352.05 KB (360,503 bytes)	12/7/1999
files\system\mssearch\bin\tquery.dll				6:00:00 AM	Microsoft Corporation		
security.dll	5.00.2154.1	5.77 KB (5,904 bytes)	12/7/1999	c:\winnt\system32\wbem\wbemess.dll			
AM	Microsoft Corporation	c:\winnt\system32\security.dll					

fastprox.dll	1.50.1085.0001	144.08 KB (147,534 bytes)	12/7/1999	msdtcprx.dll	1999.9.3422.10	619.27 KB (634,128 bytes)	8/8/2001
6:00:00 AM	Microsoft Corporation			10:51:24 AM	Microsoft Corporation c:\winnt\system32\msdtcprx.dll		
	c:\winnt\system32\wbem\fastprox.dll			txfaux.dll	1999.9.3422.24	341.27 KB (349,456 bytes)	8/8/2001
wbemcore.dll	1.50.1085.0001	632.05 KB (647,224 bytes)	12/7/1999	10:51:23 AM	Microsoft Corporation c:\winnt\system32\txfaux.dll		
6:00:00 AM	Microsoft Corporation			msdtctm.dll	1999.9.3422.12	1.02 MB (1,070,864 bytes)	8/8/2001
	c:\winnt\system32\wbem\wbemcore.dll			10:51:24 AM	Microsoft Corporation c:\winnt\system32\msdtctm.dll		
wbemcomm.dll	1.50.1085.0001	684.05 KB (700,472 bytes)	12/7/1999	msdtc.exe	1999.9.3421.3	6.77 KB (6,928 bytes)	8/8/2001 10:51:23 AM
6:00:00 AM	Microsoft Corporation			AM	Microsoft Corporation c:\winnt\system32\msdtc.exe		
	c:\winnt\system32\wbem\wbemcomm.dll			rasadhlp.dll	5.00.2168.1	7.27 KB (7,440 bytes)	12/7/1999 6:00:00 AM
winmgmt.exe	1.50.1085.0001	188.05 KB (192,567 bytes)	12/7/1999	AM	Microsoft Corporation c:\winnt\system32\rasadhlp.dll		
6:00:00 AM	Microsoft Corporation			winrnr.dll	5.00.2160.1	18.77 KB (19,216 bytes)	12/7/1999
	c:\winnt\system32\wbem\winmgmt.exe			6:00:00 AM	Microsoft Corporation c:\winnt\system32\winrnr.dll		
gamevlog.exe	Not Available	227.97 KB (233,442 bytes)	12/16/2001	dhcpcsvc.dll	5.00.2153.1	88.77 KB (90,896 bytes)	12/7/1999
10:26:18 AM	Not Available c:\winnt\system32\gamserv\gamevlog.exe			6:00:00 AM	Microsoft Corporation c:\winnt\system32\dhcpcsvc.dll		
gamevent.exe	Not Available	105.33 KB (107,858 bytes)	12/16/2001	tapi32.dll	5.00.2182.1	123.27 KB (126,224 bytes)	12/7/1999
10:26:18 AM	Not Available c:\winnt\system32\gamserv\gamevent.exe			6:00:00 AM	Microsoft Corporation c:\winnt\system32\tapi32.dll		
gamserv.exe	Not Available	160.00 KB (163,841 bytes)	12/16/2001	rasman.dll	5.00.2188.1	54.77 KB (56,080 bytes)	12/7/1999
10:26:18 AM	Not Available c:\winnt\system32\gamserv\gamserv.exe			6:00:00 AM	Microsoft Corporation c:\winnt\system32\rasman.dll		
perfctrs.dll	5.00.2134.1	40.77 KB (41,744 bytes)	12/7/1999	rasapi32.dll	5.00.2188.1	189.77 KB (194,320 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation c:\winnt\system32\perfctrs.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\rasapi32.dll		
tapiperf.dll	5.00.2143.1	6.27 KB (6,416 bytes)	12/7/1999 6:00:00 AM	icmp.dll	5.00.2134.1	7.27 KB (7,440 bytes)	12/7/1999 6:00:00 AM
AM	Microsoft Corporation c:\winnt\system32\tapiperf.dll			AM	Microsoft Corporation c:\winnt\system32\icmp.dll		
rsvpperf.dll	5.00.2167.1	10.77 KB (11,024 bytes)	12/7/1999	iphlpapi.dll	5.00.2173.2	67.77 KB (69,392 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation c:\winnt\system32\rsvpperf.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\iphlpapi.dll		
rasctrs.dll	5.00.2188.1	12.27 KB (12,560 bytes)	12/7/1999	rnr20.dll	5.00.2152.1	35.77 KB (36,624 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation c:\winnt\system32\rasctrs.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\rnr20.dll		
perfos.dll	5.00.2155.1	21.27 KB (21,776 bytes)	12/7/1999	wshtcpip.dll	5.00.2134.1	17.27 KB (17,680 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation c:\winnt\system32\perfos.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\wshtcpip.dll		
sqlctr80.dll	2000.080.0194.00	32.06 KB (32,827 bytes)	10/22/2001 9:19:30 AM	msafd.dll	5.00.2153.1	54.27 KB (55,568 bytes)	12/7/1999
	Microsoft Corporation			6:00:00 AM	Microsoft Corporation c:\winnt\system32\msafd.dll		
	c:\progra~1\micros~2\mssql\bin\sqlctr80.dll			rpcss.dll	5.00.2181.1	229.27 KB (234,768 bytes)	12/7/1999
msscctrs.dll	9.107.6223.2	24.00 KB (24,576 bytes)	5/10/2001	6:00:00 AM	Microsoft Corporation c:\winnt\system32\rpcss.dll		
12:32:28 PM	Microsoft Corporation			svchost.exe	5.00.2134.1	7.77 KB (7,952 bytes)	12/7/1999 6:00:00 AM
	c:\progra~1\common~1\system\mssearch\bin\msscctrs.dll			AM	Microsoft Corporation c:\winnt\system32\svchost.exe		
faxperf.dll	5.00.2148.1	6.27 KB (6,416 bytes)	12/7/1999 6:00:00 AM	scecli.dll	5.00.2191.1	105.27 KB (107,792 bytes)	12/7/1999
AM	Microsoft Corporation c:\winnt\system32\faxperf.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\scecli.dll		
regsvc.exe	5.00.2155.1	65.27 KB (66,832 bytes)	12/7/1999	atl.dll	3.00.8449	57.56 KB (58,938 bytes)	12/7/1999 6:00:00 AM
6:00:00 AM	Microsoft Corporation c:\winnt\system32\regsvc.exe			AM	Microsoft Corporation c:\winnt\system32\atl.dll		
gamscm.exe	Not Available	134.72 KB (137,956 bytes)	12/16/2001	certcli.dll	5.00.2175.1	132.27 KB (135,440 bytes)	12/7/1999
10:26:18 AM	Not Available c:\winnt\system32\gamserv\gamscm.exe			6:00:00 AM	Microsoft Corporation c:\winnt\system32\certcli.dll		
mtxoci.dll	1999.9.3421.3	109.27 KB (111,888 bytes)	8/8/2001	msock.dll	5.00.2152.1	62.27 KB (63,760 bytes)	12/7/1999
10:51:24 AM	Microsoft Corporation c:\winnt\system32\mtxoci.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\msock.dll		
resutils.dll	5.00.2191.1	39.77 KB (40,720 bytes)	12/7/1999	ntdsatq.dll	5.00.2181.1	31.27 KB (32,016 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation c:\winnt\system32\resutils.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\ntdsatq.dll		
clusapi.dll	5.00.2179.1	50.27 KB (51,472 bytes)	12/7/1999	ntdsa.dll	5.00.2195.1	993.27 KB (1,017,104 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation c:\winnt\system32\clusapi.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\ntdsa.dll		
msvcp50.dll	5.00.7051	552.50 KB (565,760 bytes)	12/7/1999	kdcsvc.dll	5.00.2181.1	133.77 KB (136,976 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation c:\winnt\system32\msvcp50.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\kdcsvc.dll		
xolehlp.dll	1999.9.3421.3	17.27 KB (17,680 bytes)	8/8/2001	sfmapi.dll	5.00.2134.1	38.77 KB (39,696 bytes)	12/7/1999
10:51:23 AM	Microsoft Corporation c:\winnt\system32\xolehlp.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\sfmapi.dll		
msdtclog.dll	1999.9.3421.3	89.77 KB (91,920 bytes)	8/8/2001	rtutils.dll	5.00.2168.1	43.77 KB (44,816 bytes)	12/7/1999
10:51:23 AM	Microsoft Corporation c:\winnt\system32\msdtclog.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\rtutils.dll		
mtxclu.dll	1999.9.3421.3	50.27 KB (51,472 bytes)	12/7/1999	adslrpc.dll	5.00.2172.1	127.77 KB (130,832 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation c:\winnt\system32\mtxclu.dll			6:00:00 AM	Microsoft Corporation c:\winnt\system32\adslrpc.dll		

activeds.dll	5.00.2172.1	172.77 KB (176,912 bytes)	12/7/1999	umpnpmgr.dll	5.00.2182.1	86.27 KB (88,336 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\activeds.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\umpnpmgr.dll	
mprapi.dll	5.00.2181.1	79.27 KB (81,168 bytes)	12/7/1999	services.exe	5.00.2134.1	86.77 KB (88,848 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\mprapi.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\services.exe	
rassfm.dll	5.00.2168.1	21.27 KB (21,776 bytes)	12/7/1999	clbcatq.dll	1999.9.3422.14	479.27 KB (490,768 bytes)	8/8/2001
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\rassfm.dll		10:51:18 AM	Microsoft Corporation	c:\winnt\system32\clbcatq.dll	
mpr.dll	5.00.2146.1	53.27 KB (54,544 bytes)	12/7/1999	oleaut32.dll	2.40.4512	600.27 KB (614,672 bytes)	12/7/1999
AM	Microsoft Corporation	c:\winnt\system32\mpr.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\oleaut32.dll	
schannel.dll	5.00.2170.1	139.77 KB (143,120 bytes)	12/7/1999	cscui.dll	5.00.2172.1	227.27 KB (232,720 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\schannel.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\cscui.dll	
netlogon.dll	5.00.2182.1	347.77 KB (356,112 bytes)	12/7/1999	winspool.drv	5.00.2167.1	109.77 KB (112,400 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\netlogon.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\winspool.drv	
msv1_0.dll	5.00.2164.1	94.77 KB (97,040 bytes)	12/7/1999	winscard.dll	5.00.2134.1	77.27 KB (79,120 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\msv1_0.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\winscard.dll	
kerberos.dll	5.00.2181.1	196.77 KB (201,488 bytes)	12/7/1999	wlnotify.dll	5.00.2164.1	53.27 KB (54,544 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\kerberos.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\wlnotify.dll	
msprivs.dll	5.00.2154.1	41.50 KB (42,496 bytes)	12/7/1999	cscdll.dll	5.00.2189.1	98.27 KB (100,624 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\msprivs.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\cscdll.dll	
samsrv.dll	5.00.2192.1	357.77 KB (366,352 bytes)	12/7/1999	lz32.dll	5.00.2134.1	9.77 KB (10,000 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\samsrv.dll		AM	Microsoft Corporation	c:\winnt\system32\lz32.dll	
lsasrv.dll	5.00.2184.1	487.77 KB (499,472 bytes)	12/7/1999	version.dll	5.00.2134.1	15.77 KB (16,144 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\lsasrv.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\version.dll	
lsass.exe	5.00.2184.1	32.77 KB (33,552 bytes)	12/7/1999	rsabase.dll	5.00.2150.1	128.77 KB (131,856 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\lsass.exe		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\rsabase.dll	
esent.dll	6.0.3939.6	1.07 MB (1,120,016 bytes)	12/7/1999	mecat32.dll	5.131.2134.1	7.77 KB (7,952 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\esent.dll		AM	Microsoft Corporation	c:\winnt\system32\mecat32.dll	
wmicore.dll	5.00.2178.1	70.77 KB (72,464 bytes)	12/7/1999	ole32.dll	5.00.2181.1	966.27 KB (989,456 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\wmicore.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\ole32.dll	
alrsvc.dll	5.00.2134.1	17.77 KB (18,192 bytes)	12/7/1999	imagehlp.dll	5.00.2195.1	125.27 KB (128,272 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\alrsvc.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\imagehlp.dll	
psbase.dll	5.00.2146.1	111.77 KB (114,448 bytes)	12/7/1999	msasn1.dll	5.00.2134.1	51.27 KB (52,496 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\psbase.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\msasn1.dll	
seclogon.dll	5.00.2135.1	15.77 KB (16,144 bytes)	12/7/1999	crypt32.dll	5.131.2173.1	465.77 KB (476,944 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\seclogon.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\crypt32.dll	
cryptsvc.dll	5.00.2181.1	61.77 KB (63,248 bytes)	12/7/1999	wintrust.dll	5.131.2143.1	162.27 KB (166,160 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\cryptsvc.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\wintrust.dll	
cryptdll.dll	5.00.2135.1	41.27 KB (42,256 bytes)	12/7/1999	setupapi.dll	5.00.2183.1	554.27 KB (567,568 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\cryptdll.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\setupapi.dll	
wkssvc.dll	5.00.2181.1	95.27 KB (97,552 bytes)	12/7/1999	winmm.dll	5.00.2161.1	184.77 KB (189,200 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\wkssvc.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\winmm.dll	
srvsvc.dll	5.00.2178.1	79.27 KB (81,168 bytes)	12/7/1999	comctl32.dll	5.81	540.27 KB (553,232 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\srvsvc.dll		AM	Microsoft Corporation	c:\winnt\system32\comctl32.dll	
cfgmgr32.dll	5.00.2134.1	16.77 KB (17,168 bytes)	12/7/1999	shlwapi.dll	5.00.2920.0000	282.77 KB (289,552 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\cfgmgr32.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\shlwapi.dll	
dmserver.dll	2191.1.296.2	11.77 KB (12,048 bytes)	12/7/1999	shell32.dll	5.00.2920.0000	2.24 MB (2,352,400 bytes)	12/7/1999
6:00:00 AM	VERITAS Software Corp.	c:\winnt\system32\dmserver.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\shell32.dll	
winsta.dll	5.00.2134.1	36.27 KB (37,136 bytes)	12/7/1999	msgina.dll	5.00.2191.1	309.77 KB (317,200 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\winsta.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\msgina.dll	
eventlog.dll	5.00.2178.1	43.77 KB (44,816 bytes)	12/7/1999	wsock32.dll	5.00.2152.1	21.27 KB (21,776 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\eventlog.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\wsock32.dll	
ntdsapi.dll	5.00.2160.1	56.27 KB (57,616 bytes)	12/7/1999	dnsapi.dll	5.00.2181.1	129.77 KB (132,880 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntdsapi.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\dnsapi.dll	
scesrv.dll	5.00.2188.1	225.77 KB (231,184 bytes)	12/7/1999	wldap32.dll	5.00.2168.1	155.77 KB (159,504 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation	c:\winnt\system32\scesrv.dll		6:00:00 AM	Microsoft Corporation	c:\winnt\system32\wldap32.dll	

```

ws2help.dll 5.00.2134.1 17.77 KB (18,192 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\ws2help.dll
ws2_32.dll 5.00.2134.1 69.77 KB (71,440 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\ws2_32.dll
samlib.dll 5.00.2160.1 46.27 KB (47,376 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\samlib.dll
netrap.dll 5.00.2134.1 11.27 KB (11,536 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\netrap.dll
netapi32.dll 5.00.2194.1 302.77 KB (310,032 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\netapi32.dll
profmap.dll 5.00.2181.1 29.27 KB (29,968 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\profmap.dll
secur32.dll 5.00.2154.1 46.77 KB (47,888 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\secur32.dll
sfc.dll 5.00.2164.1 84.27 KB (86,288 bytes) 12/7/1999 6:00:00
AM Microsoft Corporation c:\winnt\system32\sfc.dll
nddeapi.dll 5.00.2137.1 15.27 KB (15,632 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\nddeapi.dll
userenv.dll 5.00.2185.1 361.27 KB (369,936 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\userenv.dll
user32.dll 5.00.2180.1 393.27 KB (402,704 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\user32.dll
gdi32.dll 5.00.2180.1 228.77 KB (234,256 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\gdi32.dll
rpcrt4.dll 5.00.2193.1 434.27 KB (444,688 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\rpcrt4.dll
advapi32.dll 5.00.2191.1 349.27 KB (357,648 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\advapi32.dll
kernel32.dll 5.00.2191.1 715.27 KB (732,432 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\kernel32.dll
msvcrt.dll 6.10.8637.0 288.09 KB (295,000 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\msvcrt.dll
winlogon.exe 5.00.2182.1 173.27 KB (177,424 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\winlogon.exe
sfcfiles.dll 5.00.2195.1 973.27 KB (996,624 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\sfcfiles.dll
ntdll.dll 5.00.2163.1 469.77 KB (481,040 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\ntdll.dll
smss.exe 5.00.2170.1 44.27 KB (45,328 bytes) 12/7/1999
6:00:00 AM Microsoft Corporation c:\winnt\system32\smss.exe

```

[Services]

Display Name	Name	State	Start Mode	Service Type	Path
Error Control	Error Control	Start	Name	Tag ID	
Alerter	Alerter	Running	Auto	Share Process	
Application Management	AppMgmt	Stopped	Manual	Share Process	
Computer Browser	Browser	Stopped	Manual	Share Process	
Indexing Service	cisvc	Stopped	Manual	Share Process	
ClipBook	ClipSrv	Stopped	Manual	Own Process	

```

Distributed File System Dfs Stopped Manual Own Process
c:\winnt\system32\dfssvc.exe Normal LocalSystem 0
DHCP Client Dhcp Stopped Manual Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Logical Disk Manager Administrative Service dmadm Stopped Manual
Share Process c:\winnt\system32\dmadmin.exe /com Normal
LocalSystem 0
Logical Disk Manager dmserver Running Auto Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
DNS Client Dnscache Stopped Manual Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Event Log Eventlog Running Auto Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
COM+ Event System EventSystem Running Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem
0
Fax Service Fax Stopped Manual Own Process
c:\winnt\system32\faxsvc.exe Normal LocalSystem 0
Mylex Global Array Manager Server gamscm Running Auto Own Process
system32\gamsserv\gamscm.exe Normal LocalSystem 0
Intersite Messaging IsmServ Stopped Disabled Own Process
c:\winnt\system32\ismserv.exe Normal LocalSystem 0
Kerberos Key Distribution Center kdc Stopped Disabled
Share Process c:\winnt\system32\lsass.exe Normal LocalSystem
0
Server lanmanserver Running Auto Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Workstation lanmanworkstation Running Auto Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
License Logging Service LicenseService Stopped Manual Own Process
c:\winnt\system32\llssrv.exe Normal LocalSystem 0
TCP/IP NetBIOS Helper Service LmHosts Stopped Manual Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Messenger Messenger Stopped Manual Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
NetMeeting Remote Desktop Sharing mnmsrvc Stopped Manual Own Process
c:\winnt\system32\mnmsrvc.exe Normal LocalSystem 0
Distributed Transaction Coordinator MSDTC Running Auto Own Process
c:\winnt\system32\msdtc.exe Normal LocalSystem 0
Windows Installer MSIServer Stopped Manual Share Process
c:\winnt\system32\msiexec.exe /v Normal LocalSystem 0
Microsoft Search MSSEARCH Running Auto Share Process
"c:\program files\common
files\system\mssearch\bin\mssearch.exe" Normal LocalSystem 0
MSSQLSERVER MSSQLSERVER Stopped Manual Own Process
c:\progra~1\microso~2\mssql\bin\sqlservr.exe Normal LocalSystem
0
MSSQLServerADHelper MSSQLServerADHelper Stopped Manual Own Process
c:\program files\microsoft sql
server\80\tools\bin\sqladhlp.exe Normal LocalSystem 0
Network DDE NetDDE Stopped Manual Share Process
c:\winnt\system32\netdde.exe Normal LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped Manual Share Process
c:\winnt\system32\netdde.exe Normal LocalSystem 0

```

```

Net Logon      Netlogon      Stopped Manual Share Process
              c:\winnt\system32\lsass.exe Normal LocalSystem 0
Network Connections Netman Running Manual Share Process
              c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem
              0
File Replication NtFrs Stopped Manual Own Process
              c:\winnt\system32\ntfrs.exe Ignore LocalSystem 0
NT LM Security Support Provider NtLmSsp Running Manual Share
Process c:\winnt\system32\lsass.exe Normal LocalSystem 0
Removable Storage NtmsSvc Stopped Manual Share Process
              c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem
              0
Plug and Play PlugPlay Running Auto Share Process
              c:\winnt\system32\services.exe Normal LocalSystem 0
IPSEC Policy Agent PolicyAgent Stopped Manual Share Process
              c:\winnt\system32\lsass.exe Normal LocalSystem 0
Protected Storage ProtectedStorage Running Auto Share
Process c:\winnt\system32\services.exe Normal LocalSystem 0
Remote Access Connection Manager RasAuto Stopped Manual Share
Process c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem
              0
Routing and Remote Access RemoteAccess Stopped Disabled
Share Process c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Remote Registry Service RemoteRegistry Running Auto Own Process
              c:\winnt\system32\regsvc.exe Normal LocalSystem 0
Remote Procedure Call (RPC) Locator RpcLocator Stopped Manual Own
Process c:\winnt\system32\locator.exe Normal LocalSystem 0
Remote Procedure Call (RPC) RpcSs Running Auto Share Process
              c:\winnt\system32\svchost -k rpcss Normal LocalSystem 0
QoS RSVP RSVP Running Manual Own Process
              c:\winnt\system32\rsvp.exe -s Normal LocalSystem 0
Security Accounts Manager SamSs Running Auto Share Process
              c:\winnt\system32\lsass.exe Normal LocalSystem 0
Smart Card Helper SCardDrv Stopped Manual Share Process
              c:\winnt\system32\scardsvr.exe Ignore LocalSystem 0
Smart Card SCardSvr Stopped Manual Share Process
              c:\winnt\system32\scardsvr.exe Ignore LocalSystem 0
Task Scheduler Schedule Stopped Manual Share Process
              c:\winnt\system32\mstask.exe Normal LocalSystem 0
RunAs Service seclogon Running Auto Share Process
              c:\winnt\system32\services.exe Ignore LocalSystem 0
System Event Notification SENS Stopped Manual Share Process
              c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem
              0
Internet Connection Sharing SharedAccess Stopped Manual Share
Process c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem
              0
Print Spooler Spooler Stopped Manual Own Process
              c:\winnt\system32\spoolsv.exe Normal LocalSystem 0

```

```

SQLSERVERAGENT SQLSERVERAGENT Stopped Manual Own Process
              c:\progra~1\microso~2\mssql\bin\sqlagent.exe Normal LocalSystem
              0
Performance Logs and Alerts SysmonLog Stopped Auto Own Process
              c:\winnt\system32\smlogsvc.exe Normal .\admin 0
Telephony Tapisrv Running Manual Share Process
              c:\winnt\system32\svchost.exe -k tapisrv Normal LocalSystem
              0
Terminal Services TermService Stopped Disabled Own Process
              c:\winnt\system32\termsrv.exe Normal LocalSystem 0
Telnet TlntSvr Stopped Manual Own Process
              c:\winnt\system32\tlntsvr.exe Normal LocalSystem 0
Distributed Link Tracking Server TrkSvr Stopped Manual Share
Process c:\winnt\system32\services.exe Normal LocalSystem 0
Distributed Link Tracking Client TrkWks Stopped Manual Share
Process c:\winnt\system32\services.exe Normal LocalSystem 0
Uninterruptible Power Supply UPS Stopped Manual Own Process
              c:\winnt\system32\ups.exe Normal LocalSystem 0
Utility Manager UtilMan Stopped Manual Own Process
              c:\winnt\system32\utilman.exe Normal LocalSystem 0
Windows Time W32Time Stopped Manual Share Process
              c:\winnt\system32\services.exe Normal LocalSystem 0
Windows Management Instrumentation WinMgmt Running Auto Own Process
              c:\winnt\system32\wbem\winmgmt.exe Ignore LocalSystem 0
VNC Server winvnc Running Auto Own Process "c:\program
files\orl\vnc\winvnc.exe" -service Normal LocalSystem 0
Windows Management Instrumentation Driver Extensions Wmi Running
Manual Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0

```

[Program Groups]

Group Name	Name	User Name		
Accessories	Default User:Accessories	Default User		
Accessories\Accessibility	Default User:Accessories\Accessibility	Default User		
Accessories\Entertainment	Default User:Accessories\Entertainment	Default User		
Accessories\System Tools	Default User:Accessories\System Tools	Default User		
Startup	Default User:Startup	Default User		
Accessories	All Users:Accessories	All Users		
Accessories\Accessibility	All Users:Accessories\Accessibility	All Users		
Accessories\Communications	All Users:Accessories\Communications	All Users		
Accessories\Entertainment	All Users:Accessories\Entertainment	All Users		
Accessories\Games	All Users:Accessories\Games	All Users		
Accessories\System Tools	All Users:Accessories\System Tools	All Users		
Administrative Tools	All Users:Administrative Tools	All Users		
Microsoft SQL Server	All Users:Microsoft SQL Server	All Users		
Microsoft SQL Server - Switch	All Users:Microsoft SQL Server - Switch	All Users		

```

MKS Toolkit All Users:MKS Toolkit All Users
Startup All Users:Startup All Users
VNC All Users:VNC All Users
VNC\Administrative Tools All Users:VNC\Administrative Tools All
Users
Accessories MERCHANTSERVER\admin:Accessories
    MERCHANTSERVER\admin
Accessories\Accessibility
    MERCHANTSERVER\admin:Accessories\Accessibility
    MERCHANTSERVER\admin
Accessories\Entertainment
    MERCHANTSERVER\admin:Accessories\Entertainment
    MERCHANTSERVER\admin
Accessories\System Tools MERCHANTSERVER\admin:Accessories\System
Tools MERCHANTSERVER\admin
Startup MERCHANTSERVER\admin:Startup MERCHANTSERVER\admin
Accessories MERCHANTSERVER\Administrator:Accessories
    MERCHANTSERVER\Administrator
Accessories\Accessibility
    MERCHANTSERVER\Administrator:Accessories\Accessibility
    MERCHANTSERVER\Administrator
Accessories\Entertainment
    MERCHANTSERVER\Administrator:Accessories\Entertainment
    MERCHANTSERVER\Administrator
Accessories\System Tools
    MERCHANTSERVER\Administrator:Accessories\System Tools
    MERCHANTSERVER\Administrator
Administrative Tools MERCHANTSERVER\Administrator:Administrative
Tools MERCHANTSERVER\Administrator
Startup MERCHANTSERVER\Administrator:Startup
    MERCHANTSERVER\Administrator

```

[Startup Programs]

Program Command	User Name	Location
Service Manager	c:\progra~1\microso~2\80\tools\binn\sqlmangr.exe	
/n	All Users	Common Startup
WinVNC	"c:\program files\orl\vnc\winvnc.exe" -servicehelper	All Users
	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	

[OLE Registration]

Object	Local Server
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid
Sound	Not Available
Media Clip	Not Available
Image Document	"C:\Program Files\Windows
NT\Accessories\ImageVue	KodakImg.exe"
WordPad Document	"%ProgramFiles%\Windows
NT\Accessories\WORDPAD.EXE"	
Windows Media Services DRM Storage object	Not Available
Bitmap Image	mspaint.exe

[Internet Explorer 5]

[ Following are sub-categories of this main category ]

[Summary]

Item	Value
Version	5.00.2920.0000
Build	52920
Product ID	51879-OEM-0000007-00000
Application Path	C:\Program Files\Internet Explorer
Language	English (United States)
Active Printer	Not Available

Cipher Strength	56-bit
Content Advisor	Disabled
IEAK Install	No

[File Versions]

File	Version	Size	Date	Path	Company
advapi32.dll	5.0.2191.1	349 KB	12/7/1999 6:00:00 AM	C:\WINNT\system32	Microsoft Corporation
advapi32.dll	5.0.2191.1	349 KB	12/7/1999 6:00:00 AM	Microsoft Corporation	
advpack.dll	5.0.2920.0	87 KB	12/7/1999 6:00:00 AM	C:\WINNT\system32	Microsoft Corporation
advpack.dll	5.0.2920.0	87 KB	12/7/1999 6:00:00 AM	Microsoft Corporation	
browsecl.dll	5.0.2920.0	35 KB	12/7/1999 6:00:00 AM	C:\WINNT\system32	Microsoft Corporation
browsecl.dll	5.0.2920.0	35 KB	12/7/1999 6:00:00 AM	Microsoft Corporation	
browseui.dll	5.0.2920.0	793 KB	12/7/1999 6:00:00 AM	C:\WINNT\system32	Microsoft Corporation
browseui.dll	5.0.2920.0	793 KB	12/7/1999 6:00:00 AM	Microsoft Corporation	
ckcnv.exe	5.0.2189.1	9 KB	12/7/1999 6:00:00 AM	C:\WINNT\system32	Microsoft Corporation
ckcnv.exe	5.0.2189.1	9 KB	12/7/1999 6:00:00 AM	Microsoft Corporation	
comctl32.dll	5.81.2920.0	540 KB	12/7/1999 6:00:00 AM	C:\WINNT\system32	Microsoft Corporation
comctl32.dll	5.81.2920.0	540 KB	12/7/1999 6:00:00 AM	Microsoft Corporation	
crypt32.dll	5.131.2173.1	466 KB	12/7/1999 6:00:00 AM	C:\WINNT\system32	Microsoft Corporation
crypt32.dll	5.131.2173.1	466 KB	12/7/1999 6:00:00 AM	Microsoft Corporation	
enhsig.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
iemigrat.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available

iesetup.dll	5.0.2920.0	57 KB	12/7/1999 6:00:00 AM		oleaut32.dll	2.40.4512.1	600 KB	12/7/1999 6:00:00 AM	.
	C:\WINNT\system32		Microsoft Corporation						
iesetup.dll	5.0.2920.0	57 KB	12/7/1999 6:00:00 AM	.	olepro32.dll	5.0.4512.1	160 KB	12/7/1999 6:00:00 AM	
	Microsoft Corporation					C:\WINNT\system32		Microsoft Corporation	
iexplore.exe	5.0.2920.0	59 KB	12/7/1999 6:00:00 AM	C:\Program	olepro32.dll	5.0.4512.1	160 KB	12/7/1999 6:00:00 AM	.
	Files\Internet Explorer		Microsoft Corporation					Microsoft Corporation	
imagehlp.dll	5.0.2195.1	125 KB	12/7/1999 6:00:00 AM		rsabase.dll	5.0.2150.1	129 KB	12/7/1999 6:00:00 AM	
	C:\WINNT\system32		Microsoft Corporation			C:\WINNT\system32		Microsoft Corporation	
imagehlp.dll	5.0.2195.1	125 KB	12/7/1999 6:00:00 AM	.	rsabase.dll	5.0.2150.1	129 KB	12/7/1999 6:00:00 AM	.
	Microsoft Corporation							Microsoft Corporation	
imghelp.dll	<File Missing>	Not Available	Not Available	Not	rsaenh.dll	<File Missing>	Not Available	Not Available	Not
	Available	Not Available				Available	Not Available		
inseng.dll	5.0.2920.0	72 KB	12/7/1999 6:00:00 AM		rsapi32.dll	<File Missing>	Not Available	Not Available	Not
	C:\WINNT\system32		Microsoft Corporation			Available	Not Available		
inseng.dll	5.0.2920.0	72 KB	12/7/1999 6:00:00 AM	.	rsasig.dll	<File Missing>	Not Available	Not Available	Not
	Microsoft Corporation					Available	Not Available		
jobexec.dll	5.0.0.1 47 KB	12/7/1999 6:00:00 AM	C:\WINNT\system32		schannel.dll	5.0.2170.0	140 KB	12/7/1999 6:00:00 AM	
	Microsoft Corporation					C:\WINNT\system32		Microsoft Corporation	
jobexec.dll	5.0.0.1 47 KB	12/7/1999 6:00:00 AM	.	Microsoft	schannel.dll	5.0.2170.0	140 KB	12/7/1999 6:00:00 AM	.
	Corporation							Microsoft Corporation	
jscript.dll	5.1.0.4615	476 KB	12/7/1999 6:00:00 AM		shdoc401.dll	<File Missing>	Not Available	Not Available	Not
	C:\WINNT\system32		Microsoft Corporation			Available	Not Available		
jscript.dll	5.1.0.4615	476 KB	12/7/1999 6:00:00 AM	.	shdocvw.dll	5.0.2920.0	1078 KB	12/7/1999 6:00:00 AM	
	Microsoft Corporation					C:\WINNT\system32		Microsoft Corporation	
jsproxy.dll	5.0.2920.0	13 KB	12/7/1999 6:00:00 AM		shdocvw.dll	5.0.2920.0	1078 KB	12/7/1999 6:00:00 AM	.
	C:\WINNT\system32		Microsoft Corporation					Microsoft Corporation	
jsproxy.dll	5.0.2920.0	13 KB	12/7/1999 6:00:00 AM	.	shell32.dll	5.0.2920.0	2297 KB	12/7/1999 6:00:00 AM	
	Microsoft Corporation					C:\WINNT\system32		Microsoft Corporation	
msaahtml.dll	<File Missing>	Not Available	Not Available	Not	shell32.dll	5.0.2920.0	2297 KB	12/7/1999 6:00:00 AM	.
	Available	Not Available						Microsoft Corporation	
mshtml.dll	5.0.2920.0	2302 KB	12/7/1999 6:00:00 AM		shlwapi.dll	5.0.2920.0	283 KB	12/7/1999 6:00:00 AM	
	C:\WINNT\system32		Microsoft Corporation			C:\WINNT\system32		Microsoft Corporation	
mshtml.dll	5.0.2920.0	2302 KB	12/7/1999 6:00:00 AM	.	shlwapi.dll	5.0.2920.0	283 KB	12/7/1999 6:00:00 AM	.
	Microsoft Corporation							Microsoft Corporation	
msjava.dll	5.0.3234.0	918 KB	12/7/1999 6:00:00 AM		url.dll	5.0.2920.0	82 KB	12/7/1999 6:00:00 AM	C:\WINNT\system32
	C:\WINNT\system32		Microsoft Corporation					Microsoft Corporation	
msjava.dll	5.0.3234.0	918 KB	12/7/1999 6:00:00 AM	.	url.dll	5.0.2920.0	82 KB	12/7/1999 6:00:00 AM	.
	Microsoft Corporation							Microsoft Corporation	Microsoft
msoss.dll	<File Missing>	Not Available	Not Available	Not	urlmon.dll	5.0.2920.0	427 KB	12/7/1999 6:00:00 AM	
	Available	Not Available				C:\WINNT\system32		Microsoft Corporation	
msxml.dll	5.0.2920.0	521 KB	12/7/1999 6:00:00 AM		urlmon.dll	5.0.2920.0	427 KB	12/7/1999 6:00:00 AM	.
	C:\WINNT\system32		Microsoft Corporation					Microsoft Corporation	
msxml.dll	5.0.2920.0	521 KB	12/7/1999 6:00:00 AM	.	vbscript.dll	5.1.0.4615	428 KB	12/7/1999 6:00:00 AM	
	Microsoft Corporation					C:\WINNT\system32		Microsoft Corporation	
occache.dll	5.0.2920.0	86 KB	12/7/1999 6:00:00 AM		vbscript.dll	5.1.0.4615	428 KB	12/7/1999 6:00:00 AM	.
	C:\WINNT\system32		Microsoft Corporation					Microsoft Corporation	
occache.dll	5.0.2920.0	86 KB	12/7/1999 6:00:00 AM	.	webcheck.dll	5.0.2920.0	252 KB	12/7/1999 6:00:00 AM	
	Microsoft Corporation					C:\WINNT\system32		Microsoft Corporation	
ole32.dll	5.0.2181.1	966 KB	12/7/1999 6:00:00 AM		webcheck.dll	5.0.2920.0	252 KB	12/7/1999 6:00:00 AM	.
	C:\WINNT\system32		Microsoft Corporation					Microsoft Corporation	
ole32.dll	5.0.2181.1	966 KB	12/7/1999 6:00:00 AM	.	win.com	5.0.2134.1	24 KB	12/7/1999 6:00:00 AM	C:\WINNT\system32
	Microsoft Corporation							Microsoft Corporation	
oleaut32.dll	2.40.4512.1	600 KB	12/7/1999 6:00:00 AM		win.com	5.0.2134.1	24 KB	12/7/1999 6:00:00 AM	.
	C:\WINNT\system32		Microsoft Corporation					Microsoft Corporation	Microsoft

```

wininet.dll 5.0.2920.0 457 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
wininet.dll 5.0.2920.0 457 KB 12/7/1999 6:00:00 AM .
Microsoft Corporation
winsock.dll 3.10.0.103 3 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
winsock.dll 3.10.0.103 3 KB 12/7/1999 6:00:00 AM .
Microsoft Corporation
wintrust.dll 5.131.2143.1 162 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
wintrust.dll 5.131.2143.1 162 KB 12/7/1999 6:00:00 AM .
Microsoft Corporation
wsock.vxd <File Missing> Not Available Not Available Not
Available Not Available
wsock32.dll 5.0.2152.1 21 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
wsock32.dll 5.0.2152.1 21 KB 12/7/1999 6:00:00 AM .
Microsoft Corporation
wsock32n.dll <File Missing> Not Available Not Available Not
Available Not Available

```

[Connectivity]

```

Item Value
Connection Preference Never dial
EnableHttp1.1 1
ProxyHttp1.1 0

```

LAN Settings

```

AutoConfigProxy Not Available
AutoProxyDetectMode Enabled
AutoConfigURL
Proxy Disabled
ProxyServer
ProxyOverride

```

[Cache]

[ Following are sub-categories of this main category ]

[Summary]

```

Item Value
Page Refresh Type Automatic
Temporary Internet Files Folder C:\Documents and
Settings\admin\Local Settings\Temporary Internet Files
Total Disk Space 8683 MB
Available Disk Space 6722 MB
Maximum Cache Size 271 MB
Available Cache Size 271 MB

```

[List of Objects]

```

Program File Status CodeBase

```

No cached object information available

[Content]

[ Following are sub-categories of this main category ]

[Summary]

```

Item Value
Content Advisor Disabled

```

[Personal Certificates]

```

Issued To Issued By Validity Signature Algorithm
Administrator Administrator 8/8/2001 to 7/15/2101 sha1RSA

```

[Other People Certificates]

```

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

```

[Publishers]

```

Name
No publisher information available

```

[Security]

```

Zone Security Level

```

## Database Server Tuning

### SQL Server Registry Settings

Windows Registry Editor Version 5.00

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSSQLServer]

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client]
"SharedMemoryOn"=dword:00000001

```

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\ConnectTo]
"DSQUERY"="DBNETLIB"

```

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\DB-Lib]
"AutoAnsiToOem"="ON"
"UseIntlSettings"="ON"

```

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNe
tLib]
"ProtocolOrder"=hex(7):74,00,63,00,70,00,00,00,00,00,00,00,00,00,00,00

```

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\LastConnect]
"(local)"="2113994760:ipc:MERCHANTSERVER"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\Np]
"DefaultPipe"="sql\query"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\Tcp]
"DefaultPort"=dword:00000599

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\VIA]
"Vendor"=""
"DefaultServerPort"="0:1433"
"DefaultClientNIC"="0"
"RecognizedVendors"="Giganet, ServerNet II"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\TDS]
"PE6400"="7.0"
"@="7.0"
"(local)"="7.0"
"MERCHANTSERVER"="7.0"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer]
"FullTextDefaultPath"="C:\Program Files\Microsoft SQL Server\MSSQL\FTDATA"
"ListenOn"=hex(7):53,00,53,00,4d,00,53,00,53,00,48,00,37,00,30,00,00,00,53,00,\
53,00,4e,00,45,00,54,00,4c,00,49,00,42,00,00,00,00,00
"ResourceMgrID"="{32312A0F-C279-462A-8B76-3CFE07DE75F9}"
"SetHostName"=dword:00000000
"AuditLevel"=dword:00000000
>LoginMode"=dword:00000002
"Tapeloadwaittime"=dword:ffffffff
"DefaultLogin"="guest"
"Map_"="\\"
"Map#"="-"
"Map$"=""
"BackupDirectory"="C:\Program Files\Microsoft SQL Server\MSSQL\BACKUP"
"DefaultDomain"="MERCHANTSERVER"
"DefaultCollationName"="Latin1_General_BIN"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\CurrentVersion]
"checksum"=hex:38,33,32,32,63,31,35,38,61,65,37,64,34,63,64,37,35,30,64,61,30,\
33,34,62,36,30,31,35,62,63,64,36,37,65,31,36,33,64,35,61,31,39,36,62,62,63,\

```

```

34,31,36,62,64,66,37,65,38,30,64,33,38,33,34,64,65,61,31,38,65,30,66,61,61,\
37,61,32,37,66,64,39,62,36,62,62,62,32,30,63,36,39,31,61,34,63,63,62,66,30,\
34,66,33,31,36,62,39,35,32,33,38,66,33,66,37,37,64,62,36,66,35,65,34,63,35,\
64,63,37,35,31,35,30,33,66,33,31,38,37,37,39,31,32,63,61,32,30,30,66,62,36,\
31,39,32,64,61,65,64,32,39,62,32,31,35,62,37,61,35,34,30,32,30,37,00
"RegisteredOwner"="TRACKS"
"SerialNumber"=dword:80e30040
"CurrentVersion"="8.00.194"
"Language"=dword:00000409
"CSDVersionNumber"=dword:00000100
"CSDVersion"="8.00.384"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\Parameters]
"SQLArg0"="-dC:\\Program Files\Microsoft SQL Server\MSSQL\data\master.mdf"
"SQLArg1"="-eC:\\Program Files\Microsoft SQL Server\MSSQL\log\ERRORLOG"
"SQLArg2"="-lC:\\Program Files\Microsoft SQL Server\MSSQL\data\mastlog.ldf"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\RPCNetLib]
"Security"=""

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib]
"ProtocolList"=hex(7):74,00,63,00,70,00,00,00,6e,00,70,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib\Np]
"PipeName"="\\\\.\\pipe\\sql\\query"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib\Tcp]
"TcpHideFlag"=dword:00000000
"TcpPort"="1433"
"TcpDynamicPorts"=""

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\ADSDSOObject]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\DB2OLEDB]

```

```

"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\Microsoft.
Jet.OLEDB.4.0]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\MSDAORA]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\MSDASQL]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\MSIDXS]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\MSQLImpPro
v]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\MSSEARCHSO
L]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\SQLOLEDB]
"AllowInProcess"=dword:00000001
"DisallowAdhocAccess"=dword:00000000

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Replication]
"WorkingDirectory"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL\\REPLDATA"
"DistributionDB"=""
"IsInstalled"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Setup]
"SQLPath"="C:\\Program Files\\Microsoft SQL Server\\MSSQL"
"SQLDataRoot"="C:\\Program Files\\Microsoft SQL Server\\MSSQL"
"SourcePath"="C:\\sql2ksp1"
"firststart"=dword:00000000
"Scripts"=hex(7):73,00,70,00,31,00,5f,00,73,00,65,00,72,00,76,00,5f,00,
75,00,\
6e,00,69,00,2e,00,73,00,71,00,6c,00,00,00,38,00,30,00,73,00,70,00,31,00
,2d,\
00,74,00,6f,00,6f,00,6c,00,73,00,2e,00,73,00,71,00,6c,00,00,00,73,00,70
,00,\
31,00,5f,00,72,00,65,00,70,00,6c,00,2e,00,73,00,71,00,6c,00,00,00,73,00
,70,\
00,5f,00,76,00,75,00,70,00,67,00,72,00,61,00,64,00,65,00,5f,00,72,00,65
,00,\
70,00,6c,00,69,00,63,00,61,00,74,00,69,00,6f,00,6e,00,3a,00,40,00,6c,00
,6f,\

00,67,00,69,00,6e,00,3d,00,4e,00,27,00,73,00,61,00,27,00,2c,00,40,00,70
,00,\
61,00,73,00,73,00,77,00,6f,00,72,00,64,00,3d,00,4e,00,27,00,27,00,2c,00
,40,\
00,76,00,65,00,72,00,5f,00,6f,00,6c,00,64,00,3d,00,31,00,39,00,34,00,00
,00,\
00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\SQLServerAgent]
"JobHistoryMaxRowsPerJob"=dword:00000064
"RestartServer"=dword:00000001
"DownloadedMaxRows"=dword:00000064
"JobHistoryMaxRows"=dword:000003e8
"ErrorLoggingLevel"=dword:00000003
"NonAlertableErrors"="1204,4002"
"MSXServerName"=""
"ServerHost"=""
"WorkingDirectory"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL\\JOBS"
"ErrorLogFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL\\LOG\\SQLAGENT.OUT"
"SysAdminOnly"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\SQLServerAgent\Subsy
stems]
"CmdExec"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL\\BINN\\SQLCMDSS.DLL,NULL,CmdExecStart,CmdEvent,CmdExecSto
p,10"
"Snapshot"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL\\BINN\\SQLREPSS.DLL,C:\\Program Files\\Microsoft SQL
Server\\80\\COM\\SNAPSHOT.EXE,ReplStart,ReplEvent,ReplStop,100"
"Distribution"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL\\BINN\\SQLREPSS.DLL,C:\\Program Files\\Microsoft SQL
Server\\80\\COM\\DISTRIB.EXE,ReplStart,ReplEvent,ReplStop,100"
"Merge"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL\\BINN\\SQLREPSS.DLL,C:\\Program Files\\Microsoft SQL
Server\\80\\COM\\REPLMERG.EXE,ReplStart,ReplEvent,ReplStop,100"
"ActiveScripting"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL\\BINN\\SQLATXSS.DLL,NULL,ActiveScriptStart,ActiveScriptEv
ent,ActiveScriptStop,10"
"QueueReader"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL\\BINN\\SQLREPSS.DLL,C:\\Program Files\\Microsoft SQL
Server\\80\\COM\\QRDRSVC.EXE,ReplStart,ReplEvent,ReplStop,100"
"LogReader"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL\\BINN\\SQLREPSS.DLL,C:\\Program Files\\Microsoft SQL
Server\\80\\COM\\LOGREAD.EXE,ReplStart,ReplEvent,ReplStop,25"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\SQLServerSCP]
@=""

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Tracking]
"{E07FDDA4-5A21-11d2-9DAD-00C04F79D434}"=""

```

```

"{E07FDDA5-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDA6-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDA7-5A21-11d2-9DAD-00C04F79D434}"=""
"{6DC86044-0C71-11d3-9E18-00C04F79D434}"=""
"{E07FDDA8-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDA9-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDAA-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDAB-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDAC-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDAD-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDAF-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDB2-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDBE-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDC0-5A21-11d2-9DAD-00C04F79D434}"=""
"{E07FDDBF-5A21-11d2-9DAD-00C04F79D434}"=""

```

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Tracking\Shortcuts]
"Enterprise Manager"=""
"Query Analyzer"=""
"Profiler"=""
"Client Network Utility"=""
"Configure SQL XML Support in IIS"=""
"Import and Export Data"=""
"Service Manager"=""
"Server Network Utility"=""
"Books Online"=""

```

## TCPIP Registry Settings

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip]
"Type"=dword:00000001
"Start"=dword:00000001
"ErrorControl"=dword:00000001
"Tag"=dword:00000004
"ImagePath"=hex(2):53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,44,00,\
52,00,49,00,56,00,45,00,52,00,53,00,5c,00,74,00,63,00,70,00,69,00,70,00,2e,\
00,73,00,79,00,73,00,00,00
"DisplayName"="TCP/IP Protocol Driver"
"Group"="PNP_TDI"
"Description"="TCP/IP Protocol Driver"

```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Linkage]
"Bind"=hex(7):5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,7b,00,39,00,30,\

```

```

00,35,00,42,00,35,00,44,00,46,00,39,00,2d,00,46,00,42,00,35,00,39,00,2d,00,\

```

```

34,00,38,00,30,00,32,00,2d,00,39,00,32,00,46,00,31,00,2d,00,37,00,42,00,34,\
00,45,00,37,00,44,00,33,00,38,00,43,00,36,00,39,00,42,00,7d,00,00,00,5c,00,\
44,00,65,00,76,00,69,00,63,00,65,00,5c,00,7b,00,31,00,46,00,35,00,39,00,34,\
00,42,00,43,00,32,00,2d,00,43,00,42,00,46,00,46,00,2d,00,34,00,34,00,37,00,\
43,00,2d,00,38,00,43,00,35,00,37,00,2d,00,44,00,36,00,34,00,39,00,33,00,42,\
00,37,00,42,00,46,00,39,00,34,00,37,00,7d,00,00,00,5c,00,44,00,65,00,76,00,\
69,00,63,00,65,00,5c,00,4e,00,64,00,69,00,73,00,57,00,61,00,6e,00,49,00,70,\
00,00,00,00,00
"Route"=hex(7):22,00,7b,00,39,00,30,00,35,00,42,00,35,00,44,00,46,00,39,00,2d,\
00,46,00,42,00,35,00,39,00,2d,00,34,00,38,00,30,00,32,00,2d,00,39,00,32,00,\
46,00,31,00,2d,00,37,00,42,00,34,00,45,00,37,00,44,00,33,00,38,00,43,00,36,\
00,39,00,42,00,7d,00,22,00,00,00,22,00,7b,00,31,00,46,00,35,00,39,00,34,00,\
42,00,43,00,32,00,2d,00,43,00,42,00,46,00,46,00,2d,00,34,00,34,00,37,00,43,\
00,2d,00,38,00,43,00,35,00,37,00,2d,00,44,00,36,00,34,00,39,00,33,00,42,00,\
37,00,42,00,46,00,39,00,34,00,37,00,7d,00,22,00,00,00,22,00,4e,00,64,00,69,\
00,73,00,57,00,61,00,6e,00,49,00,70,00,22,00,00,00,00,00
"Export"=hex(7):5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,54,00,63,00,70,\
00,69,00,70,00,5f,00,7b,00,39,00,30,00,35,00,42,00,35,00,44,00,46,00,39,00,\
2d,00,46,00,42,00,35,00,39,00,2d,00,34,00,38,00,30,00,32,00,2d,00,39,00,32,\
00,46,00,31,00,2d,00,37,00,42,00,34,00,45,00,37,00,44,00,33,00,38,00,43,00,\
00,\

```

```

36,00,39,00,42,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00
,5c,\
00,54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,31,00,46,00,35,00,39,00,34
,00,\
42,00,43,00,32,00,2d,00,43,00,42,00,46,00,46,00,2d,00,34,00,34,00,37,00
,43,\
00,2d,00,38,00,43,00,35,00,37,00,2d,00,44,00,36,00,34,00,39,00,33,00,42
,00,\
37,00,42,00,46,00,39,00,34,00,37,00,7d,00,00,00,5c,00,44,00,65,00,76,00
,69,\
00,63,00,65,00,5c,00,54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,43,00,32
,00,\
42,00,35,00,31,00,38,00,43,00,32,00,2d,00,41,00,34,00,35,00,41,00,2d,00
,34,\
00,32,00,38,00,41,00,2d,00,42,00,30,00,44,00,31,00,2d,00,30,00,42,00,41
,00,\
41,00,41,00,34,00,43,00,38,00,45,00,46,00,43,00,39,00,7d,00,00,00,5c,00
,44,\
00,65,00,76,00,69,00,63,00,65,00,5c,00,54,00,63,00,70,00,69,00,70,00,5f
,00,\
7b,00,44,00,43,00,31,00,31,00,38,00,34,00,43,00,39,00,2d,00,35,00,41,00
,31,\
00,36,00,2d,00,34,00,32,00,44,00,30,00,2d,00,39,00,36,00,45,00,35,00,2d
,00,\
41,00,37,00,41,00,31,00,30,00,32,00,31,00,45,00,30,00,30,00,31,00,35,00
,7d,\
    00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]
"NV Hostname"="merchantserver"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6
f,00,6f,\
00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c
,00,\
64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,00,00,00
"NameServer"=""
"ForwardBroadcasts"=dword:00000000
"IPEnableRouter"=dword:00000000
"Domain"=""
"Hostname"="merchantserver"

```

```

"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Adapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Adapters\NdisWanIp]
"LLInterface"="WANARP"
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00
,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00
,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,43,00,32,00,42,00,35,00,31,00,38
,00,\
43,00,32,00,2d,00,41,00,34,00,35,00,41,00,2d,00,34,00,32,00,38,00,41,00
,2d,\
00,42,00,30,00,44,00,31,00,2d,00,30,00,42,00,41,00,41,00,41,00,34,00,43
,00,\
38,00,45,00,46,00,43,00,39,00,7d,00,00,00,54,00,63,00,70,00,69,00,70,00
,5c,\
00,50,00,61,00,72,00,61,00,6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49
,00,\
6e,00,74,00,65,00,72,00,66,00,61,00,63,00,65,00,73,00,5c,00,7b,00,44,00
,43,\
00,31,00,31,00,38,00,34,00,43,00,39,00,2d,00,35,00,41,00,31,00,36,00,2d
,00,\
34,00,32,00,44,00,30,00,2d,00,39,00,36,00,45,00,35,00,2d,00,41,00,37,00
,41,\
00,31,00,30,00,32,00,31,00,45,00,30,00,30,00,31,00,35,00,7d,00,00,00,00
,00
"NumInterfaces"=dword:00000002
"IpInterfaces"=hex:c2,18,b5,c2,5a,a4,8a,42,b0,d1,0b,aa,a4,c8,ef,c9,c9,8
4,11,dc,\
    16,5a,d0,42,96,e5,a7,a1,02,1e,00,15

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Adapters\{1F594BC2-CBFF-447C-8C57-D6493B7BF947}]
"LLInterface"=""

```

```

"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,31,00,46,00,35,00,39,00,34,00,42,00,\
43,00,32,00,2d,00,43,00,42,00,46,00,46,00,2d,00,34,00,34,00,37,00,43,00,2d,\
00,38,00,43,00,35,00,37,00,2d,00,44,00,36,00,34,00,39,00,33,00,42,00,37,00,\
    42,00,46,00,39,00,34,00,37,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{905B5DF9-FB59-4802-92F1-7B4E7D38C69B}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,39,00,30,00,35,00,42,00,35,00,44,00,\
46,00,39,00,2d,00,46,00,42,00,35,00,39,00,2d,00,34,00,38,00,30,00,32,00,2d,\
00,39,00,32,00,46,00,31,00,2d,00,37,00,42,00,34,00,45,00,37,00,44,00,33,00,\
    38,00,43,00,36,00,39,00,42,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\DNSRegisteredAdapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{1F594BC2-CBFF-447C-8C57-D6493B7BF947}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,2e,00,31,00,30,00,2e,00,33,00,\
    32,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
    00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""

```

```

"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,\
    30,00,\
    32,00,00,00,00,00
"DhcpServer"="255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3b71acda
"T1"=dword:3b71b3e2
"T2"=dword:3b71b928
"LeaseTerminatesTime"=dword:3b71baea
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{905B5DF9-FB59-4802-92F1-7B4E7D38C69B}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,2e,00,31,00,2e,00,31,00,30,00,\
    31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
    00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,\
    30,00,\
    33,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{C2B518C2-A45A-428A-B0D1-0BAAA4C8EFC9}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,00,00,\
    00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,00,\
    00
"DefaultGateway"=hex(7):00,00

```

```

"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Interfaces\{DC1184C9-5A16-42D0-96E5-A7A1021E0015}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,0
0
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,
00
"DefaultGateway"=hex(7):00,00
"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
PersistentRoutes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Winsock]
"UseDelayedAcceptance"=dword:00000000
"HelperDllName"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,
6f,00,\
6f,00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00
,5c,\
00,77,00,73,00,68,00,74,00,63,00,70,00,69,00,70,00,2e,00,64,00,6c,00,6c
,00,\
00,00
"MaxSockAddrLength"=dword:00000010
"MinSockAddrLength"=dword:00000010
"Mapping"=hex:0b,00,00,00,03,00,00,00,02,00,00,00,01,00,00,00,06,00,00,
00,02,\
00,00,00,01,00,00,00,00,00,00,00,02,00,00,00,00,00,00,00,06,00,00,00,00
,00,\
00,00,00,00,00,06,00,00,00,00,00,00,00,00,01,00,00,00,06,00,00,00,02,00
,00,\
00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00,00,00,00,02,00,00
,00,\
00,00,00,00,11,00,00,00,00,00,00,00,00,00,00,00,11,00,00,00,00,00,00,00
,02,\
00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Performance
]
"Close"="CloseTcpIpPerformanceData"
"Collect"="CollectTcpIpPerformanceData"
"Library"="Perfctrs.dll"
"Open"="OpenTcpIpPerformanceData"
"WbemAdapFileSize"=hex:00,60,4e,96,aa,40,bf,01

```

```

"WbemAdapFileSize"=dword:0000a310
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00
,00,02,\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00
,00,\
00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00
,00,\
05,12,00,00,00,02,00,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00
,00,\
20,00,00,00,20,02,00,00,03,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00
,00,\
00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00
,00,\
00,05,20,00,00,00,23,02,00,00,03,00,00,00,01,01,00,00,00,00,05,12,00
,00,\
00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\ServiceProv
ider]
"Class"=dword:00000008
"DnsPriority"=dword:000007d0
"HostsPriority"=dword:000001f4
"LocalPriority"=dword:000001f3
"ProviderPath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6
f,00,6f,\
00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c
,00,\
77,00,73,00,6f,00,63,00,6b,00,33,00,32,00,2e,00,64,00,6c,00,6c,00,00,00
"NetbtPriority"=dword:000007d1
"Name"="TCP/IP"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Enum]
"0"="Root\LEGACY_TCPIP\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

```

## Web Server Tuning

### TCPIP Registry Settings



```

00,2d,00,42,00,37,00,44,00,39,00,2d,00,46,00,34,00,43,00,41,00,35,00,32
,00,\
44,00,46,00,30,00,33,00,35,00,45,00,7d,00,00,00,5c,00,44,00,65,00,76,00
,69,\
00,63,00,65,00,5c,00,54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,30,00,35
,00,\
30,00,34,00,34,00,34,00,34,00,45,00,2d,00,38,00,32,00,39,00,46,00,2d,00
,34,\
00,41,00,36,00,37,00,2d,00,38,00,46,00,37,00,45,00,2d,00,41,00,30,00,43
,00,\
41,00,33,00,39,00,42,00,43,00,34,00,34,00,46,00,38,00,7d,00,00,00,5c,00
,44,\
00,65,00,76,00,69,00,63,00,65,00,5c,00,54,00,63,00,70,00,69,00,70,00,5f
,00,\
7b,00,41,00,42,00,36,00,37,00,32,00,45,00,38,00,46,00,2d,00,44,00,39,00
,45,\
00,42,00,2d,00,34,00,42,00,34,00,37,00,2d,00,38,00,46,00,41,00,41,00,2d
,00,\
42,00,31,00,45,00,35,00,38,00,34,00,30,00,44,00,38,00,31,00,31,00,45,00
,7d,\
00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,54,00,63,00,70
,00,\
69,00,70,00,5f,00,7b,00,36,00,32,00,35,00,34,00,41,00,36,00,32,00,30,00
,2d,\
00,30,00,43,00,38,00,42,00,2d,00,34,00,44,00,41,00,41,00,2d,00,41,00,43
,00,\
41,00,31,00,2d,00,39,00,37,00,41,00,36,00,35,00,34,00,33,00,38,00,39,00
,34,\
    00,31,00,39,00,7d,00,00,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]
"NV Hostname"="tpcwwl"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6
f,00,6f,\
00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c
,00,\
64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,00,00,00
"NameServer"=""
"ForwardBroadcasts"=dword:00000000

```

```

"IPEnableRouter"=dword:00000000
"Domain"=""
"Hostname"="tpcwwl"
"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001
"DisableDHCPMediaSense"=dword:00000001
"IPAutoconfigurationEnabled"=dword:00000001
"MaxUserPort"=dword:0000ffff
"MaxFreeTcbs"=dword:00010000
"TcpTimedWaitDelay"=dword:0000003c
"TcpWindowSize"=dword:0000ffff
"TcpMaxDataRetransmissions"=dword:00000200
"TcpMaxConnectRetransmissions"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Adapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Adapters\NdisWanIp]
"LLInterface"="WANARP"
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00
,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00
,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,41,00,42,00,36,00,37,00,32,00,45
,00,\
38,00,46,00,2d,00,44,00,39,00,45,00,42,00,2d,00,34,00,42,00,34,00,37,00
,2d,\
00,38,00,46,00,41,00,41,00,2d,00,42,00,31,00,45,00,35,00,38,00,34,00,30
,00,\
44,00,38,00,31,00,31,00,45,00,7d,00,00,00,54,00,63,00,70,00,69,00,70,00
,5c,\
00,50,00,61,00,72,00,61,00,6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49
,00,\
6e,00,74,00,65,00,72,00,66,00,61,00,63,00,65,00,73,00,5c,00,7b,00,36,00
,32,\
00,35,00,34,00,41,00,36,00,32,00,30,00,2d,00,30,00,43,00,38,00,42,00,2d
,00,\
34,00,44,00,41,00,41,00,2d,00,41,00,43,00,41,00,31,00,2d,00,39,00,37,00
,41,\

```

```

00,36,00,35,00,34,00,33,00,38,00,39,00,34,00,31,00,39,00,7d,00,00,00,00
,00
"NumInterfaces"=dword:00000002
"IpInterfaces"=hex:8f,2e,67,ab,eb,d9,47,4b,8f,aa,b1,e5,84,0d,81,1e,20,a
6,54,62,\
    8b,0c,aa,4d,ac,a1,97,a6,54,38,94,19

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Adapters\{0504444E-829F-4A67-8F7E-A0CA39BC44F8}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00
,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00
,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,30,00,35,00,30,00,34,00,34,00,34
,00,\

34,00,45,00,2d,00,38,00,32,00,39,00,46,00,2d,00,34,00,41,00,36,00,37,00
,2d,\

00,38,00,46,00,37,00,45,00,2d,00,41,00,30,00,43,00,41,00,33,00,39,00,42
,00,\
    43,00,34,00,34,00,46,00,38,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Adapters\{F3CF45A3-9E47-459B-B7D9-F4CA52DF035E}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00
,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00
,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,46,00,33,00,43,00,46,00,34,00,35
,00,\

41,00,33,00,2d,00,39,00,45,00,34,00,37,00,2d,00,34,00,35,00,39,00,42,00
,2d,\

00,42,00,37,00,44,00,39,00,2d,00,46,00,34,00,43,00,41,00,35,00,32,00,44
,00,\
    46,00,30,00,33,00,35,00,45,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Adapters\{FA9291B1-5966-430C-B5F6-4B7FD4CB6470}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00
,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00
,66,\

```

```

00,61,00,63,00,65,00,73,00,5c,00,7b,00,46,00,41,00,39,00,32,00,39,00,31
,00,\

42,00,31,00,2d,00,35,00,39,00,36,00,36,00,2d,00,34,00,33,00,30,00,43,00
,2d,\

00,42,00,35,00,46,00,36,00,2d,00,34,00,42,00,37,00,46,00,44,00,34,00,43
,00,\
    42,00,36,00,34,00,37,00,30,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
DNSRegisteredAdapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Interfaces]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Interfaces\{0504444E-829F-4A67-8F7E-A0CA39BC44F8}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,2e,00,39,00,2e,00,31,0
0,33,00,\
    35,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,
00,35,\
    00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00
,30,00,\
    33,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3ae5fbd3
"T1"=dword:3ae602db
"T2"=dword:3ae60821
"LeaseTerminatesTime"=dword:3ae609e3
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Interfaces\{F3CF45A3-9E47-459B-B7D9-F4CA52DF035E}]
"UseZeroBroadcast"=dword:00000000

```

```

"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,2e,00,39,00,2e,00,33,0
0,35,00,\
    00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,
00,35,\
    00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTFContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00
,30,00,\
    34,00,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3ae5fbd1
"T1"=dword:3ae602d9
"T2"=dword:3ae6081f
"LeaseTerminatesTime"=dword:3ae609e1
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Interfaces\{FA9291B1-5966-430C-B5F6-4B7FD4CB6470}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,2e,00,31,00,30,00,2e,0
0,33,00,\
    35,00,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,
00,35,\
    00,35,00,2e,00,30,00,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTFContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00
,30,00,\

```

```

    32,00,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3ae606fc
"T1"=dword:3ae60e04
"T2"=dword:3ae6134a
"LeaseTerminatesTime"=dword:3ae6150c
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
PersistentRoutes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Winsock]
"UseDelayedAcceptance"=dword:00000000
"HelperDllName"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,
6f,00,\
    6f,00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00
,5c,\
    00,77,00,73,00,68,00,74,00,63,00,70,00,69,00,70,00,2e,00,64,00,6c,00,6c
,00,\
    00,00
"MaxSockAddrLength"=dword:00000010
"MinSockAddrLength"=dword:00000010
"Mapping"=hex:0b,00,00,00,03,00,00,00,02,00,00,00,01,00,00,00,06,00,00,
00,02,\
    00,00,00,01,00,00,00,00,00,00,00,00,02,00,00,00,00,00,00,00,06,00,00,00,00
,00,\
    00,00,00,00,00,00,06,00,00,00,00,00,00,00,01,00,00,00,06,00,00,00,02,00
,00,\
    00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00,00,00,00,02,00,00
,00,\
    00,00,00,00,11,00,00,00,00,00,00,00,00,00,00,00,11,00,00,00,00,00,00,00
,02,\
    00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Performance]
"Close"="CloseTcpIpPerformanceData"
"Collect"="CollectTcpIpPerformanceData"
"Library"="Perfctrs.dll"
"Open"="OpenTcpIpPerformanceData"
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:0000a310
"WbemAdapStatus"=dword:00000000

```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\
00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
05,12,00,00,00,00,00,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,\
20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
00,00,05,0b,00,00,00,20,02,00,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,\
00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\ServiceProvider]
"Class"=dword:00000008
"DnsPriority"=dword:000007d0
"HostsPriority"=dword:000001f4
"LocalPriority"=dword:000001f3
"ProviderPath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,\
00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,\
77,00,73,00,6f,00,63,00,6b,00,33,00,32,00,2e,00,64,00,6c,00,6c,00,00,00
"NetbtPriority"=dword:000007d1
"Name"="TCP/IP"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Enum]
"0"="Root\LEGACY_TCPIP\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

## NDIS Registry Settings

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS]
"DisplayName"="NDIS System Driver"
"ErrorControl"=dword:00000001
"Group"="NDIS Wrapper"
```

```
"Start"=dword:00000000
"Type"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\MediaTypes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA]
"BusType"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\3C592]
"Id"=dword:20596d50
"Mask"=dword:f0ffffff
"Token"="EISA*TCM5920"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\3C597]
"Id"=dword:70596d50
"Mask"=dword:f0ffffff
"Token"="EISA*TCM5970"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\579TP]
"Id"=dword:93506d50
"Mask"=dword:ffffffff
"Token"="*TCM5093"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\BONSAI]
"Id"=dword:0062110e
"Mask"=dword:00ffffff
"Token"="EISA_CPQ6200_TOPNET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\BONSAI\PORT2]
"Id"=dword:0062110e
"Mask"=dword:00ffffff
"Token"="EISA_CPQ6200_BOTTOMNET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\DEC300]
"Id"=dword:0230a310
"Mask"=dword:0ffffff
"Token"="EISA&DEFEA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\DEC422]
"Id"=dword:2042a310
"Mask"=dword:f0ffffff
"Token"="EISA&DEC422"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\DURANGO]
```

"Id"=dword:0260110e  
"Mask"=dword:0fffffff  
"Token"="\*CPQ6002"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\EE100E]  
"Id"=dword:6010d425  
"Mask"=dword:f0ffffff  
"Token"="EISA&INT1060"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\HP2577A]  
"Id"=dword:4019f022  
"Mask"=dword:f0ffffff  
"Token"="\*HWP1940"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\INTEL32E]  
"Id"=dword:0112d425  
"Mask"=dword:0fffffff  
"Token"="\*INT1201"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\MAPLE]  
"Id"=dword:0160110e  
"Mask"=dword:0fffffff  
"Token"="\*CPQ6001"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\MDGEB]  
"Id"=dword:03008734  
"Mask"=dword:ffffffff  
"Token"="\*MDG0003"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\MDGER1]  
"Id"=dword:01008734  
"Mask"=dword:ffffffff  
"Token"="\*MDG0001"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\MDGER2]  
"Id"=dword:02008734  
"Mask"=dword:ffffffff  
"Token"="\*MDG0002"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\MDGER3]  
"Id"=dword:04008734  
"Mask"=dword:ffffffff  
"Token"="\*MDG0004"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\NE3200]  
"Id"=dword:0007cc3a

"Mask"=dword:00ffffff  
"Token"="\*NVL0701"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\NETFLEX3]  
"Id"=dword:20f1110e  
"Mask"=dword:f0ffffff  
"Token"="\*CPQF120"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\NETFLEX3.1]  
"Id"=dword:40f1110e  
"Mask"=dword:f0ffffff  
"Token"="\*CPQF140"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\NETFLX]  
"Id"=dword:0061110e  
"Mask"=dword:00ffffff  
"Token"="\*CPQ6100"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\NPEISA]  
"Id"=dword:0000093a  
"Mask"=dword:0000ffff  
"Token"="\*NPEISA"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\OC3133]  
"Id"=dword:0209833d  
"Mask"=dword:0xffffffff  
"Token"="EISA&\*OLC0902"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\OCTK32]  
"Id"=dword:0112833d  
"Mask"=dword:0xffffffff  
"Token"="\*OLC1201"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\P1990]  
"Id"=dword:00604f42  
"Mask"=dword:00ffffff  
"Token"="EISA&\*PRO6000"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\RODAN]  
"Id"=dword:0063110e  
"Mask"=dword:00ffffff  
"Token"="EISA\_CPQ6300\_TOPNET"

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI  
SA\RODANPORT2]  
"Id"=dword:0063110e  
"Mask"=dword:00ffffff

```
"Token"="EISA_CPQ6300_BOTTOMNET"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\ISA]
```

```
"BusType"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\Parameters]
```

```
"ProcessorAffinityMask"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\Enum]
```

```
"0"="Root\LEGACY_NDIS\0000"
```

```
"Count"=dword:00000001
```

```
"NextInstance"=dword:00000001
```

## InetInfo Registry Settings

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
```

```
"ListenBackLog"=dword:00000064
```

```
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
```

```
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
```

```
"PoolThreadLimit"=dword:00000050
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
```

```
"Library"="infoctrs.dll"
```

```
"Open"="OpenINFOPerformanceData"
```

```
"Close"="CloseINFOPerformanceData"
```

```
"Collect"="CollectINFOPerformanceData"
```

```
"Last Counter"=dword:00000826
```

```
"Last Help"=dword:00000827
```

```
"First Counter"=dword:000007e6
```

```
"First Help"=dword:000007e7
```

```
"Library Validation
```

```
Code"=hex:fe,0e,e1,c2,45,47,c0,01,10,25,00,00,00,00,00,00
```

```
"WbemAdapFileTime"=hex:80,6c,db,91,46,f3,bf,01
```

```
"WbemAdapFileSize"=dword:00002510
```

```
"WbemAdapStatus"=dword:00000000
```

## ODBC Registry Settings

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC]
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI]
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\ODBC]
```

```
"TraceFile"="\\SQL.LOG"
```

```
"TraceDll"="C:\\WINNT\\System32\\odbetrac.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\ODBC Data Sources]
```

```
"tpcwsn"="SQL Server"
```

```
"tpcwsn2"="SQL Server"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\ODBC File DSN]
```

```
"DefaultDSNDir"="C:\\Program Files\\Common Files\\ODBC\\Data Sources"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\tpcwsn]
```

```
"Driver"="C:\\WINNT\\System32\\SQLSRV32.dll"
```

```
"Description"="TPCW"
```

```
"Server"="merchantserver"
```

```
"Database"="tpcw"
```

```
"LastUser"="sa"
```

```
"AutoTranslate"="No"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\tpcwsn2]
```

```
"Driver"="C:\\WINNT\\System32\\SQLSRV32.dll"
```

```
"Description"="tpcw-IA64"
```

```
"Server"="MERCHANT2"
```

```
"Database"="tpcw"
```

```
"LastUser"="sa"
```

```
"AutoTranslate"="No"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI]
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Conversor de página de código MS]
```

```
"Translator"="C:\\WINNT\\System32\\MSCPXL32.dll"
```

```
"Setup"="C:\\WINNT\\System32\\MSCPXL32.dll"
```

```
"UsageCount"=dword:00000003
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Driver da Microsoft para arquivos texto (*.txt; *.csv)]
```

```
"UsageCount"=dword:00000003
```

```
"Driver"="C:\\WINNT\\System32\\odbcjt32.dll"
```

```
"Setup"="C:\\WINNT\\System32\\odtext32.dll"
```

```
"APILevel"="1"
```

```
"ConnectFunctions"="YYN"
```

```
"DriverODBCVer"="02.50"
```

```
"FileUsage"="1"
```

```
"FileExtns"="*.asc,*.csv,*.tab,*.txt,*.csv"
```

```
"SQLLevel"="0"
```

```
"CPOutput"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Driver do Microsoft Access (*.mdb)]
```

```
"UsageCount"=dword:00000003
```

```
"Driver"="C:\\WINNT\\System32\\odbcjt32.dll"
```

```
"Setup"="C:\\WINNT\\System32\\odbcjt32.dll"
"APILevel"="1"
"ConnectFunctions"="YYN"
"DriverODBCVer"="02.50"
"FileUsage"="2"
"FileExtns"="*.mdb"
"SQLLevel"="0"
"CPTimeout"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\Driver do Microsoft
dBase (*.dbf)]
"UsageCount"=dword:00000003
"Driver"="C:\\WINNT\\System32\\odbcjt32.dll"
"Setup"="C:\\WINNT\\System32\\oddbse32.dll"
"APILevel"="1"
"ConnectFunctions"="YYN"
"DriverODBCVer"="02.50"
"FileUsage"="1"
"FileExtns"="*.dbf,*.ndx,*.mdx"
"SQLLevel"="0"
"CPTimeout"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\Driver do Microsoft
Excel (*.xls)]
"UsageCount"=dword:00000003
"Driver"="C:\\WINNT\\System32\\odbcjt32.dll"
"Setup"="C:\\WINNT\\System32\\odexl32.dll"
"APILevel"="1"
"ConnectFunctions"="YYN"
"DriverODBCVer"="02.50"
"FileUsage"="1"
"FileExtns"="*.xls"
"SQLLevel"="0"
"CPTimeout"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\Driver do Microsoft
Paradox (*.db )]
"UsageCount"=dword:00000003
"Driver"="C:\\WINNT\\System32\\odbcjt32.dll"
"Setup"="C:\\WINNT\\System32\\odpdx32.dll"
"APILevel"="1"
"ConnectFunctions"="YYN"
"DriverODBCVer"="02.50"
"FileUsage"="1"
"FileExtns"="*.db"
"SQLLevel"="0"
"CPTimeout"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\Driver para o Microsoft
Visual FoxPro]
"UsageCount"=dword:00000003
"Driver"="C:\\WINNT\\System32\\vfpodbc.dll"
"Setup"="C:\\WINNT\\System32\\vfpodbc.dll"
"APILevel"="0"
"ConnectFunctions"="YYN"
```

```
"DriverODBCVer"="02.50"
"FileUsage"="1"
"FileExtns"="*.dbf,*.cdx,*.idx,*.fpt"
"SQLLevel"="0"
"CPTimeout"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\Microsoft Access Driver
(*.mdb)]
"UsageCount"=dword:00000003
"Driver"="C:\\WINNT\\System32\\odbcjt32.dll"
"Setup"="C:\\WINNT\\System32\\odbcjt32.dll"
"APILevel"="1"
"ConnectFunctions"="YYN"
"DriverODBCVer"="02.50"
"FileUsage"="2"
"FileExtns"="*.mdb"
"SQLLevel"="0"
"CPTimeout"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\Microsoft Access-Treiber
(*.mdb)]
"UsageCount"=dword:00000003
"Driver"="C:\\WINNT\\System32\\odbcjt32.dll"
"Setup"="C:\\WINNT\\System32\\odbcjt32.dll"
"APILevel"="1"
"ConnectFunctions"="YYN"
"DriverODBCVer"="02.50"
"FileUsage"="2"
"FileExtns"="*.mdb"
"SQLLevel"="0"
"CPTimeout"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\Microsoft dBase Driver
(*.dbf)]
"UsageCount"=dword:00000003
"Driver"="C:\\WINNT\\System32\\odbcjt32.dll"
"Setup"="C:\\WINNT\\System32\\oddbse32.dll"
"APILevel"="1"
"ConnectFunctions"="YYN"
"DriverODBCVer"="02.50"
"FileUsage"="1"
"FileExtns"="*.dbf,*.ndx,*.mdx"
"SQLLevel"="0"
"CPTimeout"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\Microsoft dBase VFP
Driver (*.dbf)]
"UsageCount"=dword:00000003
"Driver"="C:\\WINNT\\System32\\vfpodbc.dll"
"Setup"="C:\\WINNT\\System32\\vfpodbc.dll"
"APILevel"="0"
"ConnectFunctions"="YYN"
"DriverODBCVer"="02.50"
"FileUsage"="1"
"FileExtns"="*.dbf,*.cdx,*.idx,*.fpt"
```

```
"SQLLevel"="0"  
"CPOutput"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Microsoft dBase-Treiber  
(* .dbf)]  
"UsageCount"=dword:00000003  
"Driver"="C:\WINNT\System32\odbcjt32.dll"  
"Setup"="C:\WINNT\System32\oddbse32.dll"  
"APILevel"="1"  
"ConnectFunctions"="YYN"  
"DriverODBCVer"="02.50"  
"FileUsage"="1"  
"FileExtns"="*.dbf,*.ndx,*.mdx"  
"SQLLevel"="0"  
"CPOutput"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Microsoft Excel Driver  
(* .xls)]  
"UsageCount"=dword:00000003  
"Driver"="C:\WINNT\System32\odbcjt32.dll"  
"Setup"="C:\WINNT\System32\odexl32.dll"  
"APILevel"="1"  
"ConnectFunctions"="YYN"  
"DriverODBCVer"="02.50"  
"FileUsage"="1"  
"FileExtns"="*.xls"  
"SQLLevel"="0"  
"CPOutput"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Microsoft Excel-Treiber  
(* .xls)]  
"UsageCount"=dword:00000003  
"Driver"="C:\WINNT\System32\odbcjt32.dll"  
"Setup"="C:\WINNT\System32\odexl32.dll"  
"APILevel"="1"  
"ConnectFunctions"="YYN"  
"DriverODBCVer"="02.50"  
"FileUsage"="1"  
"FileExtns"="*.xls"  
"SQLLevel"="0"  
"CPOutput"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Microsoft FoxPro VFP  
Driver (*.dbf)]  
"UsageCount"=dword:00000003  
"Driver"="C:\WINNT\System32\vfpodbc.dll"  
"Setup"="C:\WINNT\System32\vfpodbc.dll"  
"APILevel"="0"  
"ConnectFunctions"="YYN"  
"DriverODBCVer"="02.50"  
"FileUsage"="1"  
"FileExtns"="*.dbf,*.cdx,*.idx,*.fpt"  
"SQLLevel"="0"  
"CPOutput"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Microsoft ODBC for  
Oracle]  
"UsageCount"=dword:00000002  
"Driver"="C:\WINNT\System32\msorcl32.dll"  
"Setup"="C:\WINNT\System32\msorcl32.dll"  
"SQLLevel"="1"  
"FileUsage"="0"  
"DriverODBCVer"="02.50"  
"ConnectFunctions"="YYY"  
"APILevel"="1"  
"CPOutput"="60"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Microsoft Paradox Driver  
(* .db )]  
"UsageCount"=dword:00000003  
"Driver"="C:\WINNT\System32\odbcjt32.dll"  
"Setup"="C:\WINNT\System32\odpdx32.dll"  
"APILevel"="1"  
"ConnectFunctions"="YYN"  
"DriverODBCVer"="02.50"  
"FileUsage"="1"  
"FileExtns"="*.db"  
"SQLLevel"="0"  
"CPOutput"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Microsoft Paradox-  
Treiber (*.db )]  
"UsageCount"=dword:00000003  
"Driver"="C:\WINNT\System32\odbcjt32.dll"  
"Setup"="C:\WINNT\System32\odpdx32.dll"  
"APILevel"="1"  
"ConnectFunctions"="YYN"  
"DriverODBCVer"="02.50"  
"FileUsage"="1"  
"FileExtns"="*.db"  
"SQLLevel"="0"  
"CPOutput"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Microsoft Text Driver  
(* .txt; *.csv)]  
"UsageCount"=dword:00000003  
"Driver"="C:\WINNT\System32\odbcjt32.dll"  
"Setup"="C:\WINNT\System32\odtext32.dll"  
"APILevel"="1"  
"ConnectFunctions"="YYN"  
"DriverODBCVer"="02.50"  
"FileUsage"="1"  
"FileExtns"="*.asc,*.csv,*.tab,*.txt,*.csv"  
"SQLLevel"="0"  
"CPOutput"="<not pooled>"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Microsoft Text-Treiber  
(* .txt; *.csv)]  
"UsageCount"=dword:00000003  
"Driver"="C:\WINNT\System32\odbcjt32.dll"
```

```

"Setup"="C:\\WINNT\\System32\\odtext32.dll"
"APILevel"="1"
"ConnectFunctions"="YYN"
"DriverODBCVer"="02.50"
"FileUsage"="1"
"FileExtns"="*.asc,*.csv,*.tab,*.txt,*.csv"
"SQLLevel"="0"
"CPTimeout"="<not pooled>"

[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\Microsoft Visual FoxPro
Driver]
"UsageCount"=dword:00000003
"Driver"="C:\\WINNT\\System32\\vfpodbc.dll"
"Setup"="C:\\WINNT\\System32\\vfpodbc.dll"
"APILevel"="0"
"ConnectFunctions"="YYN"
"DriverODBCVer"="02.50"
"FileUsage"="1"
"FileExtns"="*.dbf,*.cdx,*.idx,*.fpt"
"SQLLevel"="0"
"CPTimeout"="<not pooled>"

[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\Microsoft Visual FoxPro-
Treiber]
"UsageCount"=dword:00000003
"Driver"="C:\\WINNT\\System32\\vfpodbc.dll"
"Setup"="C:\\WINNT\\System32\\vfpodbc.dll"
"APILevel"="0"
"ConnectFunctions"="YYN"
"DriverODBCVer"="02.50"
"FileUsage"="1"
"FileExtns"="*.dbf,*.cdx,*.idx,*.fpt"
"SQLLevel"="0"
"CPTimeout"="<not pooled>"

[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\MS Code Page Translator]
"Translator"="C:\\WINNT\\System32\\MSCPXL32.dll"
"Setup"="C:\\WINNT\\System32\\MSCPXL32.dll"
"UsageCount"=dword:00000004

[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\MS Code Page-Übersetzer]
"Translator"="C:\\WINNT\\System32\\MSCPXL32.dll"
"Setup"="C:\\WINNT\\System32\\MSCPXL32.dll"
"UsageCount"=dword:00000003

[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\ODBC Connection Pooling]
"PerfMon"="0"
"Retry Wait"="120"

[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\ODBC Core]
"UsageCount"=dword:00000001

[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\ODBC Drivers]
"SQL Server"="Installed"
"Microsoft Access Driver (*.mdb)"="Installed"

```

```

"Microsoft Text Driver (*.txt; *.csv)"="Installed"
"Microsoft Excel Driver (*.xls)"="Installed"
"Microsoft dBase Driver (*.dbf)"="Installed"
"Microsoft Paradox Driver (*.db )"="Installed"
"Microsoft Visual FoxPro Driver"="Installed"
"Microsoft FoxPro VFP Driver (*.dbf)"="Installed"
"Microsoft dBase VFP Driver (*.dbf)"="Installed"
"Microsoft Access-Treiber (*.mdb)"="Installed"
"Microsoft Text-Treiber (*.txt; *.csv)"="Installed"
"Microsoft Excel-Treiber (*.xls)"="Installed"
"Microsoft dBase-Treiber (*.dbf)"="Installed"
"Microsoft Paradox-Treiber (*.db )"="Installed"
"Microsoft Visual FoxPro-Treiber"="Installed"
"Driver do Microsoft Access (*.mdb)"="Installed"
"Driver da Microsoft para arquivos texto (*.txt; *.csv)"="Installed"
"Driver do Microsoft Excel (*.xls)"="Installed"
"Driver do Microsoft dBase (*.dbf)"="Installed"
"Driver do Microsoft Paradox (*.db )"="Installed"
"Driver para o Microsoft Visual FoxPro"="Installed"
"Microsoft ODBC for Oracle"="Installed"

[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\ODBC Translators]
"MS Code Page Translator"="Installed"
"MS Code Page-Übersetzer"="Installed"
"Conversor de página de código MS"="Installed"

[HKEY_LOCAL_MACHINE\\SOFTWARE\\ODBC\\ODBCINST.INI\\SQL Server]
"UsageCount"=dword:00000004
"Driver"="C:\\WINNT\\System32\\SQLSRV32.dll"
"Setup"="C:\\WINNT\\System32\\SQLSRV32.dll"
"SQLLevel"="1"
"FileUsage"="0"
"DriverODBCVer"="03.50"
"ConnectFunctions"="YYY"
"APILevel"="2"
"CPTimeout"="60"

```

## Dell Registry Settings

```

[HKEY_LOCAL_MACHINE\\SOFTWARE\\DELL\\TPCW]
"CacheHost"="search.tpcw.net"
"DomainName"="tpcwww.tpcw.net"
"PGEHost"="192.1.9.200"

```

## Image Server Tuning

## TCPIP Registry Settings

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip]
"Type"=dword:00000001
"Start"=dword:00000001
"ErrorControl"=dword:00000001
"Tag"=dword:00000003
"ImagePath"=hex(2):53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,0
0,44,00,\
52,00,49,00,56,00,45,00,52,00,53,00,5c,00,74,00,63,00,70,00,69,00,70,00
,2e,\
    00,73,00,79,00,73,00,00,00
"DisplayName"="TCP/IP Protocol Driver"
"Group"="PNP_TDI"
"Description"="TCP/IP Protocol Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Linkage]
"Bind"=hex(7):5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,7b,00,46,
00,42,\
00,45,00,35,00,44,00,42,00,43,00,43,00,2d,00,42,00,35,00,34,00,33,00,2d
,00,\
34,00,37,00,30,00,45,00,2d,00,38,00,44,00,39,00,46,00,2d,00,41,00,37,00
,35,\
00,34,00,35,00,44,00,34,00,41,00,38,00,35,00,41,00,32,00,7d,00,00,00,5c
,00,\
44,00,65,00,76,00,69,00,63,00,65,00,5c,00,7b,00,45,00,34,00,32,00,36,00
,37,\
00,33,00,31,00,36,00,2d,00,41,00,36,00,31,00,37,00,2d,00,34,00,46,00,37
,00,\
42,00,2d,00,38,00,36,00,33,00,31,00,2d,00,37,00,31,00,32,00,43,00,37,00
,36,\
00,34,00,46,00,35,00,42,00,36,00,35,00,7d,00,00,00,5c,00,44,00,65,00,76
,00,\
69,00,63,00,65,00,5c,00,7b,00,41,00,41,00,43,00,43,00,34,00,34,00,37,00
,43,\
00,2d,00,46,00,43,00,46,00,36,00,2d,00,34,00,44,00,34,00,44,00,2d,00,41
,00,\
38,00,46,00,45,00,2d,00,34,00,33,00,38,00,34,00,38,00,36,00,37,00,43,00
,43,\
00,33,00,45,00,31,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65
,00,\
5c,00,7b,00,42,00,38,00,42,00,32,00,43,00,44,00,38,00,43,00,2d,00,30,00
,43,\
00,30,00,44,00,2d,00,34,00,35,00,36,00,34,00,2d,00,38,00,35,00,35,00,38
,00,\
2d,00,43,00,37,00,36,00,33,00,34,00,35,00,44,00,31,00,30,00,31,00,31,00
,42,\
00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,7b,00,32
,00,\
43,00,32,00,41,00,46,00,31,00,46,00,39,00,2d,00,37,00,44,00,37,00,34,00
,2d,\
00,34,00,41,00,35,00,37,00,2d,00,42,00,45,00,33,00,41,00,2d,00,42,00,44
,00,\
44,00,46,00,33,00,38,00,32,00,44,00,38,00,46,00,35,00,41,00,7d,00,00,00
,5c,\
00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,7b,00,43,00,46,00,38,00,34
,00,\
35,00,39,00,44,00,37,00,2d,00,42,00,36,00,33,00,33,00,2d,00,34,00,33,00
,37,\
00,34,00,2d,00,39,00,33,00,31,00,34,00,2d,00,36,00,36,00,30,00,43,00,32
,00,\
39,00,37,00,36,00,30,00,46,00,45,00,36,00,7d,00,00,00,5c,00,44,00,65,00
,76,\
00,69,00,63,00,65,00,5c,00,4e,00,64,00,69,00,73,00,57,00,61,00,6e,00,49
,00,\
    70,00,00,00,00,00
"Route"=hex(7):22,00,7b,00,46,00,42,00,45,00,35,00,44,00,42,00,43,00,43
,00,2d,\
00,42,00,35,00,34,00,33,00,2d,00,34,00,37,00,30,00,45,00,2d,00,38,00,44
,00,\
39,00,46,00,2d,00,41,00,37,00,35,00,34,00,35,00,44,00,34,00,41,00,38,00
,35,\
00,41,00,32,00,7d,00,22,00,00,00,22,00,7b,00,45,00,34,00,32,00,36,00,37
,00,\
33,00,31,00,36,00,2d,00,41,00,36,00,31,00,37,00,2d,00,34,00,46,00,37,00
,42,\
00,2d,00,38,00,36,00,33,00,31,00,2d,00,37,00,31,00,32,00,43,00,37,00,36
,00,\
34,00,46,00,35,00,42,00,36,00,35,00,7d,00,22,00,00,00,22,00,7b,00,41,00
,41,\

```

00,43,00,43,00,34,00,34,00,37,00,43,00,2d,00,46,00,43,00,46,00,36,00,2d,00,\

34,00,44,00,34,00,44,00,2d,00,41,00,38,00,46,00,45,00,2d,00,34,00,33,00,38,\

00,34,00,38,00,36,00,37,00,43,00,43,00,33,00,45,00,31,00,7d,00,22,00,00,00,\

22,00,7b,00,42,00,38,00,42,00,32,00,43,00,44,00,38,00,43,00,2d,00,30,00,43,\

00,30,00,44,00,2d,00,34,00,35,00,36,00,34,00,2d,00,38,00,35,00,35,00,38,00,\

2d,00,43,00,37,00,36,00,33,00,34,00,35,00,44,00,31,00,30,00,31,00,31,00,42,\

00,7d,00,22,00,00,00,22,00,7b,00,32,00,43,00,32,00,41,00,46,00,31,00,46,00,\

39,00,2d,00,37,00,44,00,37,00,34,00,2d,00,34,00,41,00,35,00,37,00,2d,00,42,\

00,45,00,33,00,41,00,2d,00,42,00,44,00,44,00,46,00,33,00,38,00,32,00,44,00,\

38,00,46,00,35,00,41,00,7d,00,22,00,00,00,22,00,7b,00,43,00,46,00,38,00,34,\

00,35,00,39,00,44,00,37,00,2d,00,42,00,36,00,33,00,33,00,2d,00,34,00,33,00,\

37,00,34,00,2d,00,39,00,33,00,31,00,34,00,2d,00,36,00,36,00,30,00,43,00,32,\

00,39,00,37,00,36,00,30,00,46,00,45,00,36,00,7d,00,22,00,00,00,22,00,4e,00,\

64,00,69,00,73,00,57,00,61,00,6e,00,49,00,70,00,22,00,00,00,00,00

"Export "=hex(7):5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,54,00,63,00,70,\

00,69,00,70,00,5f,00,7b,00,46,00,42,00,45,00,35,00,44,00,42,00,43,00,43,00,\

2d,00,42,00,35,00,34,00,33,00,2d,00,34,00,37,00,30,00,45,00,2d,00,38,00,44,\

00,39,00,46,00,2d,00,41,00,37,00,35,00,34,00,35,00,44,00,34,00,41,00,38,00,\

35,00,41,00,32,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,\

00,54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,45,00,34,00,32,00,36,00,37,00,\

33,00,31,00,36,00,2d,00,41,00,36,00,31,00,37,00,2d,00,34,00,46,00,37,00,42,\

00,2d,00,38,00,36,00,33,00,31,00,2d,00,37,00,31,00,32,00,43,00,37,00,36,00,\

34,00,46,00,35,00,42,00,36,00,35,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,\

00,63,00,65,00,5c,00,54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,41,00,41,00,\

43,00,43,00,34,00,34,00,37,00,43,00,2d,00,46,00,43,00,46,00,36,00,2d,00,34,\

00,44,00,34,00,44,00,2d,00,41,00,38,00,46,00,45,00,2d,00,34,00,33,00,38,00,\

34,00,38,00,36,00,37,00,43,00,43,00,33,00,45,00,31,00,7d,00,00,00,5c,00,44,\

00,65,00,76,00,69,00,63,00,65,00,5c,00,54,00,63,00,70,00,69,00,70,00,5f,00,\

7b,00,42,00,38,00,42,00,32,00,43,00,44,00,38,00,43,00,2d,00,30,00,43,00,30,\

00,44,00,2d,00,34,00,35,00,36,00,34,00,2d,00,38,00,35,00,35,00,38,00,2d,00,\

43,00,37,00,36,00,33,00,34,00,35,00,44,00,31,00,30,00,31,00,31,00,42,00,7d,\

00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,54,00,63,00,70,00,\

69,00,70,00,5f,00,7b,00,32,00,43,00,32,00,41,00,46,00,31,00,46,00,39,00,2d,\

00,37,00,44,00,37,00,34,00,2d,00,34,00,41,00,35,00,37,00,2d,00,42,00,45,00,\

33,00,41,00,2d,00,42,00,44,00,44,00,46,00,33,00,38,00,32,00,44,00,38,00,46,\

00,35,00,41,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,\

54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,43,00,46,00,38,00,34,00,35,00,39,\

```

00,44,00,37,00,2d,00,42,00,36,00,33,00,33,00,2d,00,34,00,33,00,37,00,34
,00,\
2d,00,39,00,33,00,31,00,34,00,2d,00,36,00,36,00,30,00,43,00,32,00,39,00
,37,\
00,36,00,30,00,46,00,45,00,36,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69
,00,\
63,00,65,00,5c,00,54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,38,00,44,00
,41,\
00,30,00,43,00,30,00,45,00,35,00,2d,00,46,00,39,00,45,00,39,00,2d,00,34
,00,\
38,00,43,00,36,00,2d,00,38,00,46,00,45,00,37,00,2d,00,34,00,45,00,30,00
,43,\
00,39,00,33,00,30,00,38,00,45,00,46,00,37,00,32,00,7d,00,00,00,5c,00,44
,00,\
65,00,76,00,69,00,63,00,65,00,5c,00,54,00,63,00,70,00,69,00,70,00,5f,00
,7b,\
00,32,00,41,00,33,00,41,00,46,00,33,00,45,00,42,00,2d,00,38,00,44,00,41
,00,\
36,00,2d,00,34,00,35,00,44,00,32,00,2d,00,38,00,44,00,31,00,33,00,2d,00
,38,\
00,42,00,30,00,42,00,35,00,36,00,33,00,37,00,44,00,41,00,32,00,45,00,7d
,00,\
    00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]
"NV Hostname"="imagel"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6
f,00,6f,\
00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c
,00,\
64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,00,00,00
"NameServer"=""
"ForwardBroadcasts"=dword:00000000
"IPEnableRouter"=dword:00000000
"Domain"=""
"Hostname"="imagel"
"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000

```

```

"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001
"DisableDHCPMediaSense"=dword:00000001
"IPAutoconfigurationEnabled"=dword:00000001
"NV Domain"=""
"MaxUserPort"=dword:0000ffff
"MaxFreeTcbs"=dword:00008000
"TcpTimedWaitDelay"=dword:0000003c
"TcpWindowSize"=dword:00004470
"TcpMaxDataRetransmissions"=dword:00000200
"TcpMaxConnectRetransmissions"=dword:00000200
"MaxHashTableSize"=dword:00002000
"SyncDomainWithMembership"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Adapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Adapters\NdisWanIp]
"LLInterface"="WANARP"
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00
,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00
,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,38,00,44,00,41,00,30,00,43,00,30
,00,\
45,00,35,00,2d,00,46,00,39,00,45,00,39,00,2d,00,34,00,38,00,43,00,36,00
,2d,\
00,38,00,46,00,45,00,37,00,2d,00,34,00,45,00,30,00,43,00,39,00,33,00,30
,00,\
38,00,45,00,46,00,37,00,32,00,7d,00,00,00,54,00,63,00,70,00,69,00,70,00
,5c,\
00,50,00,61,00,72,00,61,00,6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49
,00,\
6e,00,74,00,65,00,72,00,66,00,61,00,63,00,65,00,73,00,5c,00,7b,00,32,00
,41,\
00,33,00,41,00,46,00,33,00,45,00,42,00,2d,00,38,00,44,00,41,00,36,00,2d
,00,\
34,00,35,00,44,00,32,00,2d,00,38,00,44,00,31,00,33,00,2d,00,38,00,42,00
,30,\
00,42,00,35,00,36,00,33,00,37,00,44,00,41,00,32,00,45,00,7d,00,00,00,00
,00
"NumInterfaces"=dword:00000002
"IpInterfaces"=hex:e5,c0,a0,8d,e9,f9,c6,48,8f,e7,4e,0c,93,08,ef,72,eb,f
3,3a,2a,\

```

a6,8d,d2,45,8d,13,8b,0b,56,37,da,2e

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{2C2AF1F9-7D74-4A57-BE3A-BDDF382D8F5A}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
```

```
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
```

```
00,61,00,63,00,65,00,73,00,5c,00,7b,00,32,00,43,00,32,00,41,00,46,00,31,00,\
```

```
46,00,39,00,2d,00,37,00,44,00,37,00,34,00,2d,00,34,00,41,00,35,00,37,00,2d,\
```

```
00,42,00,45,00,33,00,41,00,2d,00,42,00,44,00,44,00,46,00,33,00,38,00,32,00,\
44,00,38,00,46,00,35,00,41,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{AACCC47C-FCF6-4D4D-A8FE-4384867CC3E1}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
```

```
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
```

```
00,61,00,63,00,65,00,73,00,5c,00,7b,00,41,00,41,00,43,00,43,00,34,00,34,00,\
```

```
37,00,43,00,2d,00,46,00,43,00,46,00,36,00,2d,00,34,00,44,00,34,00,44,00,2d,\
```

```
00,41,00,38,00,46,00,45,00,2d,00,34,00,33,00,38,00,34,00,38,00,36,00,37,00,\
43,00,43,00,33,00,45,00,31,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{B8B2CD8C-0C0D-4564-8558-C76345D1011B}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
```

```
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
```

```
00,61,00,63,00,65,00,73,00,5c,00,7b,00,42,00,38,00,42,00,32,00,43,00,44,00,\
```

```
38,00,43,00,2d,00,30,00,43,00,30,00,44,00,2d,00,34,00,35,00,36,00,34,00,2d,\
```

```
00,38,00,35,00,35,00,38,00,2d,00,43,00,37,00,36,00,33,00,34,00,35,00,44,00,\
31,00,30,00,31,00,31,00,42,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{CF8459D7-B633-4374-9314-660C29760FE6}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
```

```
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
```

```
00,61,00,63,00,65,00,73,00,5c,00,7b,00,43,00,46,00,38,00,34,00,35,00,39,00,\
```

```
44,00,37,00,2d,00,42,00,36,00,33,00,33,00,2d,00,34,00,33,00,37,00,34,00,2d,\
```

```
00,39,00,33,00,31,00,34,00,2d,00,36,00,36,00,30,00,43,00,32,00,39,00,37,00,\
36,00,30,00,46,00,45,00,36,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{E4267316-A617-4F7B-8631-712C764F5B65}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
```

```
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
```

```
00,61,00,63,00,65,00,73,00,5c,00,7b,00,45,00,34,00,32,00,36,00,37,00,33,00,\
```

```
31,00,36,00,2d,00,41,00,36,00,31,00,37,00,2d,00,34,00,46,00,37,00,42,00,2d,\
```

```
00,38,00,36,00,33,00,31,00,2d,00,37,00,31,00,32,00,43,00,37,00,36,00,34,00,\
46,00,35,00,42,00,36,00,35,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{FBE5DBCC-B543-470E-8D9F-A7545D4A85A2}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
```

```
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
```

```
00,61,00,63,00,65,00,73,00,5c,00,7b,00,46,00,42,00,45,00,35,00,44,00,42,00,\
```



```
00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,
00,35,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"="127.0.0.1"
"Domain"=""
"DisableDynamicUpdate"=dword:00000001
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,
30,00,\
32,00,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3bc7655a
"T1"=dword:3bc76c62
"T2"=dword:3bc771a8
"LeaseTerminatesTime"=dword:3bc7736a
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Interfaces\{CF8459D7-B633-4374-9314-660C29760FE6}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,0
0
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,
00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Interfaces\{E4267316-A617-4F7B-8631-712C764F5B65}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,2e,00,39,00,2e,00,37,0
0,37,00,\
```

```
00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,
00,35,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"="127.0.0.1"
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3bcaeda7
"T1"=dword:3bcaf4af
"T2"=dword:3bc9f9f5
"LeaseTerminatesTime"=dword:3bc9fbb7
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Interfaces\{FBE5DBCC-B543-470E-8D9F-A7545D4A85A2}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,2e,00,39,00,2e,00,35,0
0,31,00,\
00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,
00,35,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"="192.1.9.52,192.1.9.51"
"Domain"=""
"DisableDynamicUpdate"=dword:00000001
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,
30,00,\
32,00,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3bd735e2
"T1"=dword:3bd73cea
"T2"=dword:3bd74230
```

```

"LeaseTerminatesTime"=dword:3bd743f2
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
PersistentRoutes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Winsock]
"UseDelayedAcceptance"=dword:00000000
"HelperDllName"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,
6f,00,\
6f,00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00
,5c,\
00,77,00,73,00,68,00,74,00,63,00,70,00,69,00,70,00,2e,00,64,00,6c,00,6c
,00,\
00,00
"MaxSockAddrLength"=dword:00000010
"MinSockAddrLength"=dword:00000010
"Mapping"=hex:0b,00,00,00,03,00,00,00,02,00,00,00,01,00,00,00,06,00,00,
00,02,\
00,00,00,01,00,00,00,00,00,02,00,00,00,00,00,00,00,06,00,00,00,00
,00,\
00,00,00,00,00,06,00,00,00,00,00,00,00,01,00,00,00,06,00,00,00,02,00
,00,\
00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00,00,00,02,00,00
,00,\
00,00,00,00,11,00,00,00,00,00,00,00,00,00,00,00,11,00,00,00,00,00,00
,02,\
00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Performance
]
"Close"="CloseTcpIpPerformanceData"
"Collect"="CollectTcpIpPerformanceData"
"Library"="Perfctrs.dll"
"Open"="OpenTcpIpPerformanceData"
"WbemAdapFileTime"=hex:00,db,3d,bd,c4,d4,c0,01
"WbemAdapFileSize"=dword:0000a310
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00
,00,02,\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00
,00,\

```

```

00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00
,00,\
05,12,00,00,00,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00
,05,\
20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00
,00,\
00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00
,00,\
00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,\
00,01,01,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\ServiceProv
ider]
"Class"=dword:00000008
"DnsPriority"=dword:000007d0
"HostsPriority"=dword:000001f4
"LocalPriority"=dword:000001f3
"ProviderPath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6
f,00,6f,\
00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c
,00,\
77,00,73,00,6f,00,63,00,6b,00,33,00,32,00,2e,00,64,00,6c,00,6c,00,00,00
"NetbtPriority"=dword:000007d1
"Name"="TCP/IP"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Enum]
"0"="Root\LEGACY_TCPIP\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

```

## NDIS Registry Settings

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS]
"DisplayName"="NDIS System Driver"
"ErrorControl"=dword:00000001
"Group"="NDIS Wrapper"
"Start"=dword:00000000
"Type"=dword:00000001

```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\MediaTypes]

```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect]

```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA]
"BusType"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\3C592]
"Id"=dword:20596d50
"Mask"=dword:f0ffffff
"Token"="EISA*TCM5920"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\3C597]
"Id"=dword:70596d50
"Mask"=dword:f0ffffff
"Token"="EISA*TCM5970"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\579TP]
"Id"=dword:93506d50
"Mask"=dword:ffffffff
"Token"="*TCM5093"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\BONSAI]
"Id"=dword:0062110e
"Mask"=dword:00ffffff
"Token"="EISA_CPQ6200_TOPNET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\BONSAI\PORT2]
"Id"=dword:0062110e
"Mask"=dword:00ffffff
"Token"="EISA_CPQ6200_BOTTOMNET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\DEC300]
"Id"=dword:0230a310
"Mask"=dword:0xffffffff
"Token"="EISA&DEFEA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\DEC422]
"Id"=dword:2042a310
"Mask"=dword:f0ffffff
"Token"="EISA&DEC422"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\DURANGO]
"Id"=dword:0260110e
"Mask"=dword:0xffffffff
"Token"="*CPQ6002"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\EE100E]
"Id"=dword:6010d425
"Mask"=dword:f0ffffff
"Token"="EISA&INT1060"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\HP2577A]
"Id"=dword:4019f022
"Mask"=dword:f0ffffff
"Token"="*HWP1940"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\INTEL32E]
"Id"=dword:0112d425
"Mask"=dword:0xffffffff
"Token"="*INT1201"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\MAPLE]
"Id"=dword:0160110e
"Mask"=dword:0xffffffff
"Token"="*CPQ6001"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\MDGEB]
"Id"=dword:03008734
"Mask"=dword:ffffffff
"Token"="*MDG0003"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\MDGER1]
"Id"=dword:01008734
"Mask"=dword:ffffffff
"Token"="*MDG0001"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\MDGER2]
"Id"=dword:02008734
"Mask"=dword:ffffffff
"Token"="*MDG0002"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\MDGER3]
"Id"=dword:04008734
"Mask"=dword:ffffffff
"Token"="*MDG0004"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\NE3200]
"Id"=dword:0007cc3a
"Mask"=dword:00ffffff
"Token"="*NVL0701"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\NETFLEX3]
"Id"=dword:20f1110e

```

```
"Mask"=dword:f0ffffff
"Token"="*CPQF120"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\NETFLX3.1]
"Id"=dword:40f1110e
"Mask"=dword:f0ffffff
"Token"="*CPQF140"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\NETFLX]
"Id"=dword:0061110e
"Mask"=dword:00ffffff
"Token"="*CPQ6100"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\NPEISA]
"Id"=dword:0000093a
"Mask"=dword:0000ffff
"Token"="*NPIEISA"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\OC3133]
"Id"=dword:0209833d
"Mask"=dword:0fffffff
"Token"="EISA*OLC0902"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\OCTK32]
"Id"=dword:0112833d
"Mask"=dword:0fffffff
"Token"="*OLC1201"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\P1990]
"Id"=dword:00604f42
"Mask"=dword:00ffffff
"Token"="EISA*PRO6000"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\RODAN]
"Id"=dword:0063110e
"Mask"=dword:00ffffff
"Token"="EISA_CPQ6300_TOPNET"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\RODANPORT2]
"Id"=dword:0063110e
"Mask"=dword:00ffffff
"Token"="EISA_CPQ6300_BOTTOMNET"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\IS
A]
"BusType"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\Parameters]
"ProcessorAffinityMask"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\Enum]
"0"="Root\LEGACY_NDIS\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

## InetInfo Registry Settings

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Paramete
rs]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,0
0,53,00,\
    4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"ObjectCacheTTL"=dword:0ffffff
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performa
nce]
"Library"="infcotrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000826
"Last Help"=dword:00000827
"First Counter"=dword:000007e6
"First Help"=dword:000007e7
"Library Validation
Code"=hex:20,70,fd,3c,f7,a5,be,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,db,3d,bd,c4,d4,c0,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
```

## DNS Server Tuning

## DNS Server Configuration

30,00,2d,00,39,00,38,00,36,00,35,00,2d,00,43,00,34,00,42,00,39,00,44,00,37,\

00,39,00,44,00,31,00,37,00,37,00,44,00,7d,00,00,00,5c,00,44,00,65,00,76,00,\

## Web Cache Configuration

69,00,63,00,65,00,5c,00,7b,00,30,00,38,00,45,00,41,00,41,00,30,00,30,00,30,\

00,2d,00,32,00,45,00,35,00,43,00,2d,00,34,00,39,00,46,00,37,00,2d,00,38,00,\

## RBE Tuning

30,00,38,00,36,00,2d,00,45,00,37,00,31,00,33,00,41,00,41,00,41,00,38,00,41,\

## TCPIP Registry Settings

00,32,00,38,00,37,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,\

Windows Registry Editor Version 5.00

5c,00,7b,00,31,00,44,00,34,00,36,00,45,00,35,00,45,00,30,00,2d,00,42,00,35,\

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip]
"Type"=dword:00000001
"Start"=dword:00000001
"ErrorControl"=dword:00000001
"Tag"=dword:00000004
"ImagePath"=hex(2):53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,44,00,\
```

00,31,00,41,00,2d,00,34,00,31,00,33,00,45,00,2d,00,41,00,42,00,36,00,45,00,\

2d,00,38,00,35,00,30,00,43,00,41,00,36,00,43,00,36,00,41,00,35,00,41,00,30,\

```
52,00,49,00,56,00,45,00,52,00,53,00,5c,00,74,00,63,00,70,00,69,00,70,00,2e,\
,2e,\
00,73,00,79,00,73,00,00,00
"DisplayName"="TCP/IP Protocol Driver"
"Group"="PNP_TDI"
"Description"="TCP/IP Protocol Driver"
```

00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,4e,00,64,00,\

69,00,73,00,57,00,61,00,6e,00,49,00,70,00,00,00,00,00
"Route"=hex(7):22,00,7b,00,34,00,35,00,34,00,37,00,35,00,35,00,31,00,45,00,2d,\

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Linkage]
"Bind"=hex(7):5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,7b,00,34,00,35,\
```

00,34,00,42,00,34,00,43,00,2d,00,34,00,30,00,30,00,43,00,2d,00,38,00,44,00,\

```
00,34,00,37,00,35,00,35,00,31,00,45,00,2d,00,34,00,42,00,34,00,43,00,2d,00,\
```

39,00,35,00,2d,00,46,00,43,00,34,00,43,00,32,00,45,00,36,00,42,00,30,00,42,\

00,44,00,45,00,7d,00,22,00,00,00,22,00,7b,00,41,00,32,00,30,00,34,00,34,00,\

```
34,00,30,00,30,00,43,00,2d,00,38,00,44,00,39,00,35,00,2d,00,46,00,43,00,34,\
```

33,00,31,00,45,00,2d,00,35,00,46,00,38,00,41,00,2d,00,34,00,43,00,42,00,30,\

```
00,43,00,32,00,45,00,36,00,42,00,30,00,42,00,44,00,45,00,7d,00,00,00,5c,00,\
```

00,2d,00,39,00,38,00,36,00,35,00,2d,00,43,00,34,00,42,00,39,00,44,00,37,00,\

```
44,00,65,00,76,00,69,00,63,00,65,00,5c,00,7b,00,41,00,32,00,30,00,34,00,34,\
```

39,00,44,00,31,00,37,00,37,00,44,00,7d,00,22,00,00,00,22,00,7b,00,30,00,38,\

```
00,33,00,31,00,45,00,2d,00,35,00,46,00,38,00,41,00,2d,00,34,00,43,00,42,00,\
```

00,45,00,41,00,41,00,30,00,30,00,30,00,2d,00,32,00,45,00,35,00,43,00,2d,00,\

34,00,39,00,46,00,37,00,2d,00,38,00,30,00,38,00,36,00,2d,00,45,00,37,00,31,\

00,33,00,41,00,41,00,41,00,38,00,41,00,32,00,38,00,37,00,7d,00,22,00,00,00,\

22,00,7b,00,31,00,44,00,34,00,36,00,45,00,35,00,45,00,30,00,2d,00,42,00,35,\

00,31,00,41,00,2d,00,34,00,31,00,33,00,45,00,2d,00,41,00,42,00,36,00,45,00,\

2d,00,38,00,35,00,30,00,43,00,41,00,36,00,43,00,36,00,41,00,35,00,41,00,30,\

00,7d,00,22,00,00,00,22,00,4e,00,64,00,69,00,73,00,57,00,61,00,6e,00,49,00,\

70,00,22,00,00,00,00,00

"Export"=hex(7):5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,54,00,63,00,70,\

00,69,00,70,00,5f,00,7b,00,34,00,35,00,34,00,37,00,35,00,35,00,31,00,45,00,\

2d,00,34,00,42,00,34,00,43,00,2d,00,34,00,30,00,30,00,43,00,2d,00,38,00,44,\

00,39,00,35,00,2d,00,46,00,43,00,34,00,43,00,32,00,45,00,36,00,42,00,30,00,\

42,00,44,00,45,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,\

00,54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,41,00,32,00,30,00,34,00,34,00,\

33,00,31,00,45,00,2d,00,35,00,46,00,38,00,41,00,2d,00,34,00,43,00,42,00,30,\

00,2d,00,39,00,38,00,36,00,35,00,2d,00,43,00,34,00,42,00,39,00,44,00,37,00,\

39,00,44,00,31,00,37,00,37,00,44,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,\

00,63,00,65,00,5c,00,54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,30,00,38,00,\

45,00,41,00,41,00,30,00,30,00,30,00,2d,00,32,00,45,00,35,00,43,00,2d,00,34,\

00,39,00,46,00,37,00,2d,00,38,00,30,00,38,00,36,00,2d,00,45,00,37,00,31,00,\

33,00,41,00,41,00,41,00,38,00,41,00,32,00,38,00,37,00,7d,00,00,00,5c,00,44,\

00,65,00,76,00,69,00,63,00,65,00,5c,00,54,00,63,00,70,00,69,00,70,00,5f,00,\

7b,00,31,00,44,00,34,00,36,00,45,00,35,00,45,00,30,00,2d,00,42,00,35,00,31,\

00,41,00,2d,00,34,00,31,00,33,00,45,00,2d,00,41,00,42,00,36,00,45,00,2d,00,\

38,00,35,00,30,00,43,00,41,00,36,00,43,00,36,00,41,00,35,00,41,00,30,00,7d,\

00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,54,00,63,00,70,00,\

69,00,70,00,5f,00,7b,00,38,00,30,00,30,00,43,00,38,00,32,00,37,00,39,00,2d,\

00,32,00,35,00,42,00,37,00,2d,00,34,00,41,00,42,00,32,00,2d,00,42,00,39,00,\

45,00,43,00,2d,00,31,00,36,00,37,00,41,00,41,00,43,00,43,00,46,00,46,00,45,\

00,38,00,36,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,00,\

54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,30,00,38,00,35,00,45,00,38,00,38,\

00,38,00,37,00,2d,00,36,00,31,00,30,00,45,00,2d,00,34,00,30,00,42,00,45,00,\

2d,00,38,00,33,00,31,00,42,00,2d,00,38,00,34,00,32,00,39,00,34,00,46,00,36,\

00,41,00,39,00,31,00,43,00,41,00,7d,00,00,00,00,00

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]

"NV Hostname"="tpcw1"

"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,\

00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,\

64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,00,00,00

"NameServer"=""

"ForwardBroadcasts"=dword:00000000

"IPEnableRouter"=dword:00000000

"Domain"=""

"Hostname"="tpcw1"

```

"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001
"MaxUserPort"=dword:0000ffff
"TcpTimedWaitDelay"=dword:0000003c
"MaxFreeTcbs"=dword:00010000
"TcpWindowSize"=dword:0000ffff
"TcpNumConnections"=dword:00004000
"DhcpNameServer"="143.166.223.127 143.166.216.237"
"DhcpDomain"="us.dell.com"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\NdisWanIp]
"LLInterface"="WANARP"
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,38,00,30,00,30,00,43,00,38,00,32,00,\
37,00,39,00,2d,00,32,00,35,00,42,00,37,00,2d,00,34,00,41,00,42,00,32,00,2d,\
00,42,00,39,00,45,00,43,00,2d,00,31,00,36,00,37,00,41,00,41,00,43,00,43,00,\
46,00,46,00,45,00,38,00,36,00,7d,00,00,00,54,00,63,00,70,00,69,00,70,00,5c,\
00,50,00,61,00,72,00,61,00,6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,00,\
6e,00,74,00,65,00,72,00,66,00,61,00,63,00,65,00,73,00,5c,00,7b,00,30,00,38,\
00,35,00,45,00,38,00,38,00,38,00,37,00,2d,00,36,00,31,00,30,00,45,00,2d,00,\
34,00,30,00,42,00,45,00,2d,00,38,00,33,00,31,00,42,00,2d,00,38,00,34,00,32,\
00,39,00,34,00,46,00,36,00,41,00,39,00,31,00,43,00,41,00,7d,00,00,00,00,00,00
"NumInterfaces"=dword:00000002

"IpInterfaces"=hex:79,82,0c,80,b7,25,b2,4a,b9,ec,16,7a,ac,cf,fe,86,87,88,5e,08,\
0e,61,be,40,83,1b,84,29,4f,6a,91,ca

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{08EAA000-2E5C-49F7-8086-E713AAA8A287}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,30,00,38,00,45,00,41,00,41,00,30,00,\
30,00,30,00,2d,00,32,00,45,00,35,00,43,00,2d,00,34,00,39,00,46,00,37,00,00,\
2d,\
00,38,00,30,00,38,00,36,00,2d,00,45,00,37,00,31,00,33,00,41,00,41,00,41,00,00,\
00,\
38,00,41,00,32,00,38,00,37,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{1D46E5E0-B51A-413E-AB6E-850CA6C6A5A0}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,31,00,44,00,34,00,36,00,45,00,35,00,\
00,\
45,00,30,00,2d,00,42,00,35,00,31,00,41,00,2d,00,34,00,31,00,33,00,45,00,00,\
2d,\
00,41,00,42,00,36,00,45,00,2d,00,38,00,35,00,30,00,43,00,41,00,36,00,43,00,\
00,\
36,00,41,00,35,00,41,00,30,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{4547551E-4B4C-400C-8D95-FC4C2E6B0BDE}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,34,00,35,00,34,00,37,00,35,00,35,00,00,\
00,\
00,00

```





```
00,00,00,00,00,00,06,00,00,00,00,00,00,00,01,00,00,00,06,00,00,00,02,00,00,\
00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00,00,00,02,00,00,00,\
00,00,00,00,11,00,00,00,00,00,00,00,00,00,00,11,00,00,00,00,00,00,00,02,\
00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Performance]
"Close"="CloseTcpIpPerformanceData"
"Collect"="CollectTcpIpPerformanceData"
"Library"="Perfctrs.dll"
"Open"="OpenTcpIpPerformanceData"
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:0000a310
"WbemAdapStatus"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
```

```
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
```

```
00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
```

```
05,12,00,00,00,02,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
```

```
20,00,00,00,20,02,00,00,03,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,00,\
```

```
00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
```

```
00,05,20,00,00,00,23,02,00,00,03,00,00,00,01,01,00,00,00,00,00,00,05,12,00,00,\
00,01,01,00,00,00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\ServiceProvider]
"Class"=dword:00000008
"DnsPriority"=dword:000007d0
"HostsPriority"=dword:000001f4
"LocalPriority"=dword:000001f3
"ProviderPath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,\
```

```
00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,\
```

```
77,00,73,00,6f,00,63,00,6b,00,33,00,32,00,2e,00,64,00,6c,00,6c,00,00,00,00
"NetbtPriority"=dword:000007d1
"Name"="TCP/IP"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Enum]
"0"="Root\LEGACY_TCPIP\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

## NDIS Registry Settings

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS]
"DisplayName"="NDIS System Driver"
"ErrorControl"=dword:00000001
"Group"="NDIS Wrapper"
"Start"=dword:00000000
"Type"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\MediaTypes]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA]
"BusType"=dword:00000002
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\3C592]
"Id"=dword:20596d50
"Mask"=dword:f0ffffff
"Token"="EISA&*TCM5920"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\3C597]
"Id"=dword:70596d50
"Mask"=dword:f0ffffff
"Token"="EISA&*TCM5970"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\579TP]
"Id"=dword:93506d50
"Mask"=dword:ffffff
"Token"="*TCM5093"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\BONSAI]
"Id"=dword:0062110e
"Mask"=dword:00ffffff
"Token"="EISA_CPQ6200_TOPNET"
```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\BONSAIPORT2]
"Id"=dword:0062110e
"Mask"=dword:00ffffff
"Token"="EISA_C PQ6200_BOTTOMNET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\DEC300]
"Id"=dword:0230a310
"Mask"=dword:0ffffff
"Token"="EISA&DEFEA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\DEC422]
"Id"=dword:2042a310
"Mask"=dword:f0ffffff
"Token"="EISA&DEC422"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\DURANGO]
"Id"=dword:0260110e
"Mask"=dword:0ffffff
"Token"="*CPQ6002"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\EE100E]
"Id"=dword:6010d425
"Mask"=dword:f0ffffff
"Token"="EISA&INT1060"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\HP2577A]
"Id"=dword:4019f022
"Mask"=dword:f0ffffff
"Token"="*HWP1940"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\INTEL32E]
"Id"=dword:0112d425
"Mask"=dword:0ffffff
"Token"="*INT1201"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\MAPLE]
"Id"=dword:0160110e
"Mask"=dword:0ffffff
"Token"="*CPQ6001"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\MDGEB]
"Id"=dword:03008734
"Mask"=dword:ffffff
"Token"="*MDG0003"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\MDGER1]
"Id"=dword:01008734
"Mask"=dword:ffffff
"Token"="*MDG0001"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\MDGER2]
"Id"=dword:02008734
"Mask"=dword:ffffff
"Token"="*MDG0002"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\MDGER3]
"Id"=dword:04008734
"Mask"=dword:ffffff
"Token"="*MDG0004"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\NE3200]
"Id"=dword:0007cc3a
"Mask"=dword:00ffffff
"Token"="*NVL0701"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\NETFLEX3]
"Id"=dword:20f1110e
"Mask"=dword:f0ffffff
"Token"="*CPQF120"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\NETFLEX3.1]
"Id"=dword:40f1110e
"Mask"=dword:f0ffffff
"Token"="*CPQF140"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\NETFLX]
"Id"=dword:0061110e
"Mask"=dword:00ffffff
"Token"="*CPQ6100"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\NPEISA]
"Id"=dword:0000093a
"Mask"=dword:0000ffff
"Token"="*NPIEISA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\OC3133]
"Id"=dword:0209833d
"Mask"=dword:0ffffff
"Token"="EISA&*OLC0902"

```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\OCTK32]
"Id"=dword:0112833d
"Mask"=dword:0fffffff
"Token"="*OLC1201"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\P1990]
"Id"=dword:00604f42
"Mask"=dword:0fffffff
"Token"="EISA&*PRO6000"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\RODAN]
"Id"=dword:0063110e
"Mask"=dword:0fffffff
"Token"="EISA_CPQ6300_TOPNET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EI
SA\RODANPORT2]
"Id"=dword:0063110e
"Mask"=dword:0fffffff
"Token"="EISA_CPQ6300_BOTTOMNET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\IS
A]
"BusType"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\Parameters]
"ProcessorAffinityMask"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\Enum]
"0"="Root\LEGACY_NDIS\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

## DNSCache Registry Settings

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dnscache]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,0
0,6f,00,\
74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00
,73,\
00,65,00,72,00,76,00,69,00,63,00,65,00,73,00,2e,00,65,00,78,00,65,00,00
,00
```

```
"DisplayName"="DNS Client"
"Group"="TDI"
"DependOnService"=hex(7):54,00,63,00,70,00,69,00,70,00,00,00,00,00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Resolves and caches Domain Name System (DNS) names."
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dnscache\Paramete
rs]
"NegativeCacheTime"=dword:0000012c
"NegativeSOACacheTime"=dword:00000078
"NetFailureCacheTime"=dword:0000001e
"AdapterTimeoutCacheTime"=dword:00000078
"CacheHashTableSize"=dword:000000d3
"CacheHashTableBucketSize"=dword:0000000a
"NetFailureErrorPopupLimit"=dword:00000000
"MaxCacheEntryTtlLimit"=dword:00000001
"MaxSOACacheEntryTtlLimit"=dword:0000012c
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dnscache\Security
]
"Security"=hex:01,00,14,80,90,00,00,00,9c,00,00,00,14,00,00,00,30,00,00
,00,02,\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00
,00,\
00,00,02,00,60,00,04,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00
,00,\
05,0b,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00
,00,\
23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00
,20,\
02,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01
,01,\
00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dnscache\Enum]
"0"="Root\LEGACY_DNSCACHE\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

## DNS Configuration

### Tpcw.net.dns

```
;
; Database file tpcw.net.dns for tpcw.net zone.
; Zone version: 95
;
@ IN SOA image2. administrator.tpcw.net. (
```

```

          95          ; serial number
          900        ; refresh
          600        ; retry
          86400     ; expire
          3600      ) ; minimum TTL

;
; Zone NS records
;

@         NS      image2.
image2.   A       192.1.10.52

;
; Zone records
;

dns       A       192.1.9.52
image     A       192.1.9.52
          A       192.1.9.53
          A       192.1.9.54
          A       192.1.9.51
search    A       192.1.9.34
tpcwww    A       192.1.9.43
          A       192.1.9.46
          A       192.1.9.143
          A       192.1.9.44
          A       192.1.9.144
          A       192.1.9.35
          A       192.1.9.135
          A       192.1.9.36
          A       192.1.9.136
          A       192.1.9.37
          A       192.1.9.45
          A       192.1.9.145
          A       192.1.9.137
          A       192.1.9.38
          A       192.1.9.138
          A       192.1.9.39
          A       192.1.9.139
          A       192.1.9.40
          A       192.1.9.140
          A       192.1.9.41
          A       192.1.9.146
          A       192.1.9.141
          A       192.1.9.42
          A       192.1.9.142

```

## Appendix C – Database Design

### SQL Server Settings

1> 2>

```
-----
SQL Server Configuration Dec  7 2001 11:17:19:013AM

Microsoft SQL Server 2000 - 8.00.384 (Intel X86)
May 23 2001 00:02:52
Copyright (c) 1988-2000 Microsoft Corporation
Standard Edition on WindowsNT 5.0 (Build 2195: )
```

(1 row affected)

```
1> 2> name          minimum      maximum      config_value  run_value
-----
affinity mask
-2147483648  2147483647      0            0
allow updates
0            1            0            0
awe enabled
0            1            0            0
c2 audit mode
0            1            0            0
cost threshold for parallelism
0            32767        5            5
cursor threshold
-1          2147483647    -1           -1
default full-text language
0            2147483647    1033        1033
default language
0            9999         0            0
fill factor (%)
0            100          0            0
index create memory (KB)
704         2147483647    0            0
lightweight pooling
0            1            1            1
locks
5000        2147483647    100000       100000
max degree of parallelism
0            32            1            1
max server memory (MB)
4            2147483647    2147483647   2147483647
max text repl size (B)
0            2147483647    65536        65536
max worker threads
32          32767        2048         2048
media retention
```

```
0            365            0            0
min memory per query (KB)
512         2147483647    1024         1024
min server memory (MB)
0            2147483647    0            0
nested triggers
0            1            1            1
network packet size (B)
512         65536         4096         4096
open objects
0            2147483647    0            0
priority boost
0            1            0            0
query governor cost limit
0            2147483647    0            0
query wait (s)
-1          2147483647    -1           -1
recovery interval (min)
0            32767        0            0
remote access
0            1            1            1
remote login timeout (s)
0            2147483647    20           20
remote proc trans
0            1            0            0
remote query timeout (s)
0            2147483647    600          600
scan for startup procs
0            1            0            0
set working set size
0            1            0            0
show advanced options
0            1            1            1
two digit year cutoff
1753        9999         2049         2049
user connections
0            32767        0            0
user options
0            32767        0            0
```

### SQL Server Startup

The following command was used to start SQL Server:  
“c:\program files\Microsoft sql server\mssql\binn\sqlservr” –T3502 –T1204

-T3502 means “print a message to the log at the beginning and end of each checkpoint”  
-T1204 means “return the type of locks participating in a deadlock and the current command affected”

## CreateDatabase.sql

```
-- CreateDatabase.sql
-- Create the database
--
if exists (select name from sysdatabases where name='tpcw')
    DROP DATABASE tpcw
go

CREATE DATABASE tpcw
ON PRIMARY
(
    NAME = tpcw_root,
    FILENAME = 'C:\tpcwDB.mdf',
    SIZE = 10MB,
    FILEGROWTH = 0),
FILEGROUP tpcw_fg
(
    NAME=tpcw_data,
    FILENAME='T:',
    SIZE = 200000MB,
    FILEGROWTH = 0),
(
    NAME=tpcw_data2,
    FILENAME='U:',
    SIZE = 200000MB,
    FILEGROWTH = 0)
LOG ON
(
    NAME =tpcw_log,
    FILENAME = 'L:',
    SIZE = 50000MB,
    FILEGROWTH = 0)
go

sp_dboption tpcw, 'trunc. log on chkpt',true
go

sp_dboption tpcw, 'select into', true
go

alter database [tpcw] set recovery BULK_LOGGED
go
```

## CreateTables.sql

```
-- CreateTables.sql
--
-- Copyright (c) 1999-2000 Intel Corporation
--
use tpcw
go
--
sp_dboption tpcw, 'trunc. log on chkpt',true
go
```

```
sp_dboption tpcw, 'select into', true
go

alter database [tpcw] set recovery BULK_LOGGED
go
```

```
CREATE TABLE ADDRESS (
[ADDR_ID] [numeric](10, 0) NOT NULL ,
[ADDR_STREET1] [char] (40) COLLATE Latin1_General_BIN NULL ,
[ADDR_STREET2] [char] (40) COLLATE Latin1_General_BIN NULL ,
[ADDR_CITY] [char] (30) COLLATE Latin1_General_BIN NULL ,
[ADDR_STATE] [char] (20) COLLATE Latin1_General_BIN NULL ,
[ADDR_ZIP] [char] (10) COLLATE Latin1_General_BIN NULL ,
[ADDR_CO_ID] [smallint] NOT NULL
) ON [tpcw_fg]
GO
```

```
CREATE TABLE ADDR_LOCK (
[NEXT_ADDR_ID] [numeric](10, 0) NULL
) ON [tpcw_fg]
GO
```

```
CREATE TABLE AUTHOR (
[A_ID] [numeric](10, 0) NOT NULL ,
[A_FNAME] [char] (20) COLLATE Latin1_General_BIN NULL ,
[A_LNAME] [char] (20) COLLATE Latin1_General_BIN NULL ,
[A_MNAME] [char] (20) COLLATE Latin1_General_BIN NULL ,
[A_DOB] [datetime] NULL ,
[A_BIO] [text] COLLATE Latin1_General_BIN NULL
) ON [tpcw_fg] TEXTIMAGE_ON [tpcw_fg]
GO
```

```
CREATE TABLE CC_XACTS (
[CX_O_ID] [numeric](10, 0) NOT NULL ,
[CX_TYPE] [char] (10) COLLATE Latin1_General_BIN NULL ,
[CX_NUM] [numeric](16, 0) NULL ,
[CX_NAME] [char] (31) COLLATE Latin1_General_BIN NULL ,
[CX_EXPIRY] [datetime] NULL ,
[CX_AUTH_ID] [char] (15) COLLATE Latin1_General_BIN NULL ,
[CX_XACT_AMT] [numeric](17, 2) NULL ,
[CX_XACT_DATE] [datetime] NULL ,
[CX_CO_ID] [smallint] NOT NULL
) ON [tpcw_fg]
GO
```

```
CREATE TABLE COUNTRY (
[CO_ID] [smallint] NOT NULL ,
[CO_NAME] [char] (50) COLLATE Latin1_General_BIN NULL ,
[CO_EXCHANGE] [numeric](18, 6) NULL ,
[CO_CURRENCY] [char] (18) COLLATE Latin1_General_BIN NULL
) ON [tpcw_fg]
GO
```

```
CREATE TABLE CUSTOMER (
```

```

[C_ID] [numeric](10, 0) NOT NULL ,
[C_UNAME] [char] (20) COLLATE Latin1_General_BIN NULL ,
[C_PASSWD] [char] (20) COLLATE Latin1_General_BIN NULL ,
[C_FNAME] [char] (15) COLLATE Latin1_General_BIN NULL ,
[C_LNAME] [char] (15) COLLATE Latin1_General_BIN NULL ,
[C_ADDR_ID] [numeric](10, 0) NULL ,
[C_PHONE] [char] (16) COLLATE Latin1_General_BIN NULL ,
[C_EMAIL] [char] (50) COLLATE Latin1_General_BIN NULL ,
[C_SINCE] [datetime] NULL ,
[C_LAST_VISIT] [datetime] NULL ,
[C_LOGIN] [datetime] NULL ,
[C_EXPIRATION] [datetime] NULL ,
[C_DISCOUNT] [numeric](5, 2) NULL ,
[C_BALANCE] [numeric](17, 2) NULL ,
[C_YTD_PMT] [numeric](17, 2) NULL ,
[C_BIRTHDATE] [datetime] NULL ,
[C_DATA] [text] COLLATE Latin1_General_BIN NULL
) ON [tpcw_fg] TEXTIMAGE_ON [tpcw_fg]
GO

```

```

CREATE TABLE CUST_LOCK (
[NEXT_C_ID] [numeric](10, 0) NULL
) ON [tpcw_fg]
GO

```

```

CREATE TABLE ITEM (
[I_ID] [numeric](10, 0) NOT NULL ,
[I_TITLE] [char] (60) COLLATE Latin1_General_BIN NULL ,
[I_A_ID] [numeric](10, 0) NULL ,
[I_PUB_DATE] [datetime] NULL ,
[I_PUBLISHER] [char] (60) COLLATE Latin1_General_BIN NULL ,
[I_SUBJECT] [char] (60) COLLATE Latin1_General_BIN NULL ,
[I_DESC] [text] COLLATE Latin1_General_BIN NULL ,
[I_RELATED1] [numeric](10, 0) NOT NULL ,
[I_RELATED2] [numeric](10, 0) NOT NULL ,
[I_RELATED3] [numeric](10, 0) NOT NULL ,
[I_RELATED4] [numeric](10, 0) NOT NULL ,
[I_RELATED5] [numeric](10, 0) NOT NULL ,
[I_THUMBNAI] [char] (100) COLLATE Latin1_General_BIN NULL ,
[I_IMAGE] [char] (100) COLLATE Latin1_General_BIN NULL ,
[I_SRP] [numeric](17, 2) NULL ,
[I_COST] [numeric](17, 2) NULL ,
[I_AVAIL] [datetime] NULL ,
[I_STOCK] [smallint] NULL ,
[I_ISBN] [char] (13) COLLATE Latin1_General_BIN NULL ,
[I_PAGE] [int] NULL ,
[I_BACKING] [char] (15) COLLATE Latin1_General_BIN NULL ,
[I_DIMENSIONS] [char] (25) COLLATE Latin1_General_BIN NULL
) ON [tpcw_fg] TEXTIMAGE_ON [tpcw_fg]
GO

```

```

CREATE TABLE ORDERS (
[O_ID] [numeric](10, 0) NOT NULL ,
[O_C_ID] [numeric](10, 0) NULL ,
[O_DATE] [datetime] NULL ,

```

```

[O_SUBTOTAL] [numeric](17, 2) NULL ,
[O_TAX] [numeric](17, 2) NULL ,
[O_TOTAL] [numeric](17, 2) NULL ,
[O_SHIP_TYPE] [char] (10) COLLATE Latin1_General_BIN NULL ,
[O_SHIP_DATE] [datetime] NULL ,
[O_BILL_ADDR] [numeric](10, 0) NULL ,
[O_SHIP_ADDR] [numeric](10, 0) NULL ,
[O_STATUS] [char] (15) COLLATE Latin1_General_BIN NULL
) ON [tpcw_fg]
GO

```

```

CREATE TABLE ORDER_LINE (
[OL_ID] [smallint] NOT NULL ,
[OL_O_ID] [numeric](10, 0) NOT NULL ,
[OL_I_ID] [numeric](10, 0) NOT NULL ,
[OL_QTY] [smallint] NULL ,
[OL_DISCOUNT] [numeric](5, 2) NULL ,
[OL_COMMENTS] [char] (100) COLLATE Latin1_General_BIN NULL
) ON [tpcw_fg]
GO

```

```

CREATE TABLE ORDER_LOCK (
[NEXT_O_ID] [numeric](10, 0) NULL
) ON [tpcw_fg]
GO

```

```

CREATE TABLE SC_LOCK (
[NEXT_SC_SHOPPING_ID] [numeric](10, 0) NULL
) ON [tpcw_fg]
GO

```

```

CREATE TABLE SEARCH_AUTHOR (
[A_ID] [numeric](10, 0) NOT NULL ,
[A_FNAME] [char] (20) COLLATE Latin1_General_BIN NULL ,
[A_LNAME] [char] (20) COLLATE Latin1_General_BIN NULL ,
[A_MNAME] [char] (20) COLLATE Latin1_General_BIN NULL ,
[A_DOB] [datetime] NULL ,
[A_BIO] [text] COLLATE Latin1_General_BIN NULL
) ON [tpcw_fg] TEXTIMAGE_ON [tpcw_fg]
GO

```

```

CREATE TABLE SEARCH_ITEM (
[I_ID] [numeric](10, 0) NOT NULL ,
[I_TITLE] [char] (60) COLLATE Latin1_General_BIN NULL ,
[I_A_ID] [numeric](10, 0) NULL ,
[I_PUB_DATE] [datetime] NULL ,
[I_PUBLISHER] [char] (60) COLLATE Latin1_General_BIN NULL ,
[I_SUBJECT] [char] (60) COLLATE Latin1_General_BIN NULL ,
[I_DESC] [text] COLLATE Latin1_General_BIN NULL ,
[I_RELATED1] [numeric](10, 0) NOT NULL ,
[I_RELATED2] [numeric](10, 0) NOT NULL ,
[I_RELATED3] [numeric](10, 0) NOT NULL ,
[I_RELATED4] [numeric](10, 0) NOT NULL ,
[I_RELATED5] [numeric](10, 0) NOT NULL ,
[I_THUMBNAI] [char] (100) COLLATE Latin1_General_BIN NULL ,

```

```

[I_IMAGE] [char] (100) COLLATE Latin1_General_BIN NULL ,
[I_SRP] [numeric](17, 2) NULL ,
[I_COST] [numeric](17, 2) NULL ,
[I_AVAIL] [datetime] NULL ,
[I_STOCK] [smallint] NULL ,
[I_ISBN] [char] (13) COLLATE Latin1_General_BIN NULL ,
[I_PAGE] [int] NULL ,
[I_BACKING] [char] (15) COLLATE Latin1_General_BIN NULL ,
[I_DIMENSIONS] [char] (25) COLLATE Latin1_General_BIN NULL
) ON [tpcw_fg] TEXTIMAGE_ON [tpcw_fg]
GO

CREATE TABLE SHOPPING_BASKET (
[SC_SHOPPING_ID] [numeric](10, 0) NOT NULL ,
[SC_C_ID] [numeric](10, 0) NOT NULL ,
[SC_DATE] [datetime] NOT NULL ,
[SC_SUB_TOTAL] [numeric](17, 2) NULL ,
[SC_TAX] [numeric](17, 2) NULL ,
[SC_SHIP_COST] [numeric](17, 2) NULL ,
[SC_TOTAL] [numeric](17, 2) NULL ,
[SC_C_FNAME] [char] (15) COLLATE Latin1_General_BIN NULL ,
[SC_C_LNAME] [char] (15) COLLATE Latin1_General_BIN NULL ,
[SC_C_DISCOUNT] [numeric](5, 2) NULL ,
[SC_HOST] [char] (20) COLLATE Latin1_General_BIN NOT NULL
) ON [tpcw_fg]
GO

CREATE TABLE SHOPPING_ITEM (
[SCL_SHOPPING_ID] [numeric](10, 0) NOT NULL ,
[SCL_I_ID] [numeric](10, 0) NOT NULL ,
[SCL_I_QTY] [smallint] NOT NULL ,
[SCL_I_COST] [numeric](17, 2) NOT NULL ,
[SCL_I_SRP] [numeric](17, 2) NOT NULL ,
[SCL_I_TITLE] [char] (60) COLLATE Latin1_General_BIN NOT NULL,
[SCL_I_BACKING] [char] (15) COLLATE Latin1_General_BIN NOT NULL ,
[SCL_OL_ID] [smallint] NOT NULL ,
[SCL_I_COMMENTS] [char] (100) COLLATE Latin1_General_BIN NOT NULL ,
[SCL_HOST] [char] (20) COLLATE Latin1_General_BIN NOT NULL
) ON [tpcw_fg]
GO

```

## CreateIndexes.sql

```

-- CreateIndexes.sql
--
-- Copyright (c) 1999-2000 Intel Corporation
--
-- Create the clustered indexes on the customer file group.
--
use tpcw
go
checkpoint
go

```

```

CREATE UNIQUE CLUSTERED INDEX [A_INDEX_1] ON
[AUTHOR]([A_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE CLUSTERED INDEX [IX_CC_XACTS] ON
[CC_XACTS]([CX_O_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE CLUSTERED INDEX [COUNTRY_Index_1] ON
[COUNTRY]([CO_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE CLUSTERED INDEX [ITEM_Index_1] ON
[ITEM]([I_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE CLUSTERED INDEX [O_INDX1] ON [ORDERS]([O_ID])
ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE CLUSTERED INDEX [OL_INDX] ON
[ORDER_LINE]([OL_O_ID], [OL_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE CLUSTERED INDEX [A_INDEX_1] ON
[SEARCH_AUTHOR]([A_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE CLUSTERED INDEX [ITEM_Index_1] ON
[SEARCH_ITEM]([I_ID]) ON [tpcw_fg]
GO

```

```

checkpoint
go

CREATE UNIQUE CLUSTERED INDEX [SC_INDEX] ON
[SHOPPING_BASKET]([SC_SHOPPING_ID], [SC_HOST]) WITH FILLFACTOR
= 99 ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE CLUSTERED INDEX [SCL_INDX] ON
[SHOPPING_ITEM]([SCL_SHOPPING_ID], [SCL_HOST], [SCL_I_ID]) WITH
FILLFACTOR = 99 ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [test] ON [AUTHOR]([A_ID], [A_LNAME],
[A_FNAME]) ON [tpcw_fg]
GO

checkpoint
go

CREATE INDEX [A_LNAME_INX] ON [AUTHOR]([A_LNAME], [A_FNAME],
[A_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [COUNTRY_Index_2] ON
[COUNTRY]([CO_NAME]) WITH FILLFACTOR = 100 ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [test2] ON [ITEM]([I_ID], [I_A_ID],
[I_SUBJECT], [I_TITLE]) WITH FILLFACTOR = 99 ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [test3] ON [ITEM]([I_A_ID], [I_ID],
[I_TITLE]) WITH FILLFACTOR = 100 ON [tpcw_fg]
GO

checkpoint
go

```

```

CREATE UNIQUE INDEX [ITEM13] ON [ITEM]([I_TITLE], [I_ID],
[I_A_ID]) WITH FILLFACTOR = 99 ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [bysubject] ON [ITEM]([I_TITLE], [I_ID],
[I_A_ID], [I_SUBJECT]) WITH FILLFACTOR = 99 ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [related] ON [ITEM]([I_ID],
[I_RELATED1], [I_RELATED2], [I_RELATED3], [I_RELATED4], [I_RELATED5])
ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [Subject_Inx] ON [ITEM]([I_SUBJECT],
[I_A_ID], [I_ID], [I_TITLE]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [I_SUBJECT_INX] ON [ITEM]([I_TITLE],
[I_SUBJECT], [I_ID], [I_A_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [I_TITLE_INX] ON [ITEM]([I_SUBJECT],
[I_TITLE], [I_ID], [I_A_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [fornewprods] ON [ITEM]([I_SUBJECT],
[I_PUB_DATE] DESC , [I_TITLE], [I_ID], [I_A_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [O_C_INDX] ON [ORDERS]([O_C_ID], [O_ID])
WITH FILLFACTOR = 97, PAD_INDEX ON [tpcw_fg]
GO

```

```

checkpoint
go

CREATE UNIQUE INDEX [Orders_3] ON [ORDERS]([O_DATE] DESC ,
[O_ID], [O_C_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [OL_INDX2] ON [ORDER_LINE]([OL_O_ID],
[OL_I_ID], [OL_QTY], [OL_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE INDEX [ol3] ON [ORDER_LINE]([OL_I_ID]) WITH
FILLFACTOR = 97, PAD_INDEX ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [test] ON [SEARCH_AUTHOR]([A_ID],
[A_LNAME], [A_FNAME]) ON [tpcw_fg]
GO

checkpoint
go

CREATE INDEX [A_LNAME_INX] ON [SEARCH_AUTHOR]([A_LNAME],
[A_FNAME], [A_ID]) ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [test2] ON [SEARCH_ITEM]([I_ID],
[I_A_ID], [I_SUBJECT], [I_TITLE]) WITH FILLFACTOR = 99 ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [test3] ON [SEARCH_ITEM]([I_A_ID],
[I_ID], [I_TITLE]) WITH FILLFACTOR = 100 ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [ITEM13] ON [SEARCH_ITEM]([I_TITLE],
[I_ID], [I_A_ID]) WITH FILLFACTOR = 99 ON [tpcw_fg]
GO

```

```

checkpoint
go

CREATE UNIQUE INDEX [bysubject] ON [SEARCH_ITEM]([I_TITLE],
[I_ID], [I_A_ID], [I_SUBJECT]) WITH FILLFACTOR = 99 ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [related] ON [SEARCH_ITEM]([I_ID],
[I_RELATED1], [I_RELATED2], [I_RELATED3], [I_RELATED4], [I_RELATED5])
ON [tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [Subject_Inx] ON
[SEARCH_ITEM]([I_SUBJECT], [I_A_ID], [I_ID], [I_TITLE]) ON
[tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [I_SUBJECT_INX] ON
[SEARCH_ITEM]([I_TITLE], [I_SUBJECT], [I_ID], [I_A_ID]) ON
[tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [I_TITLE_INX] ON
[SEARCH_ITEM]([I_SUBJECT], [I_TITLE], [I_ID], [I_A_ID]) ON
[tpcw_fg]
GO

checkpoint
go

CREATE UNIQUE INDEX [fornewprods] ON
[SEARCH_ITEM]([I_SUBJECT], [I_PUB_DATE] DESC , [I_TITLE],
[I_ID], [I_A_ID]) ON [tpcw_fg]
GO

checkpoint
go

create unique index customer_idx
on CUSTOMER(C_ID)
with fillfactor=80

```

```

on tpcw_fg
go

checkpoint
go

CREATE UNIQUE CLUSTERED INDEX [C_INDX_1] ON
[CUSTOMER]([C_ID]) ON [tpcw_fg]
GO

```

```

checkpoint
go

```

```

CREATE UNIQUE INDEX [C_INDX_2] ON [CUSTOMER]([C_UNAME],
[C_ID]) WITH FILLFACTOR = 97, PAD_INDEX ON [tpcw_fg]
GO

```

```

checkpoint
go

```

```

alter table ORDERS
ADD CONSTRAINT FK_ORDERS_3
FOREIGN KEY (O_C_ID)
REFERENCES CUSTOMER (C_ID)
go

```

```

checkpoint
go

```

## CreateAddressIndexes.sql

```

-- CreateAddressIndexes.sql
--
-- Copyright (c) 1999-2000 Intel Corporation
--
-- Create the clustered indexes on the customer file group.
--
use tpcw
go

```

```

CREATE UNIQUE CLUSTERED INDEX [ADDR_INDEX_1] ON
[ADDRESS]([ADDR_ID]) ON [tpcw_fg]
GO

```

```

checkpoint
go

```

```

CREATE INDEX [ADDR_INDX_2] ON [ADDRESS]([ADDR_CO_ID],
[ADDR_ZIP], [ADDR_STATE]) WITH FILLFACTOR = 97, PAD_INDEX
ON [tpcw_fg]
GO

```

```

checkpoint
go

```

## CreateStoredProcs.sql

```

-- CreateStoredProcs.sql

```

```

use tpcw
go

```

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure AddNewAddress
(@ADDR_ID numeric(10),
@CO_ID smallint,
@ADDR_STREET1 char(40),
@ADDR_STREET2 char(40),
@ADDR_CITY char(30),
@ADDR_STATE char(20),
@ADDR_ZIP char(10)
)

```

```

As

```

```

begin

```

```

declare @AA_ID numeric(10)

```

```

begin transaction na

```

```

SELECT @AA_ID = ADDR_ID FROM ADDRESS WHERE
ADDR_CO_ID = @CO_ID and
ADDR_ZIP=@ADDR_ZIP and
ADDR_STATE = @ADDR_STATE and
ADDR_CITY = @ADDR_CITY and
ADDR_STREET1 = @ADDR_STREET1 and
ADDR_STREET2 = @ADDR_STREET2

```

```

if @@rowcount = 0

```

```

begin

```

```

Insert into ADDRESS

```

```

values(@ADDR_ID,@ADDR_STREET1,@ADDR_STREET2,
@ADDR_CITY,@ADDR_STATE,@ADDR_ZIP,@CO_ID)

```

```

end

```

```

else select @ADDR_ID = @AA_ID

```

```

commit transaction na

```

```

SELECT @ADDR_ID

```

```

end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure AddNewCCX
(@O_ID numeric(10),
@CX_TYPE char(10),
@CX_NUM numeric(16),
@CX_NAME char(31),
@CX_EXP datetime,
@O_TOTAL numeric(17,2),
@CO_ID smallint,
@AUTH char(15)
)
As
begin
Insert into CC_XACTS values
(@O_ID,@CX_TYPE,@CX_NUM,@CX_NAME,@CX_EXP,@AUTH,@O_TOTAL,
getdate(),@CO_ID)
end
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure AddNewCustomer
(
@CO_ID smallint,
@ADDR_STREET1 char(40),
@ADDR_STREET2 char(40),
@ADDR_CITY char(30),
@ADDR_STATE char(20),
@ADDR_ZIP char(10),
@C_ID numeric(10),
@C_UNAME char(20),
@C_PASSWD char(20),
@C_FNAME char(15),
@C_LNAME char(15),
@C_PHONE char(16),
@C_EMAIL char(50),
@C_DISCOUNT numeric(5,2),
@C_BIRTHDATE datetime,
@C_DATA text,
@ADDR_ID numeric(10),
@SC_SHOPPING_ID numeric(10),
@SC_HOST char(20)

```

```

)
As
declare @current datetime
begin
select @current= getdate()
begin transaction add_customer

declare @AA_ID numeric(10)

SELECT @AA_ID = ADDR_ID FROM ADDRESS WHERE
ADDR_CO_ID = @CO_ID and
ADDR_ZIP=@ADDR_ZIP and
ADDR_STATE = @ADDR_STATE and
ADDR_CITY = @ADDR_CITY and
ADDR_STREET1 = @ADDR_STREET1 and
ADDR_STREET2 = @ADDR_STREET2
if @@rowcount = 0
begin
Insert into ADDRESS
values(@ADDR_ID,@ADDR_STREET1,@ADDR_STREET2,
@ADDR_CITY,@ADDR_STATE,@ADDR_ZIP,@CO_ID)

end
else select @ADDR_ID = @AA_ID

--Insert into ADDRESS
--
values(@ADDR_ID,@ADDR_STREET1,@ADDR_STREET2,@ADDR_CITY,@ADDR_STATE,@ADD
R_ZIP,@CO_ID)
--if @@rowcount = 0
--SELECT @ADDR_ID = ADDR_ID FROM ADDRESS WHERE
--ADDR_STREET1 = @ADDR_STREET1 and
--ADDR_STREET2 = @ADDR_STREET2 and
--ADDR_CITY = @ADDR_CITY and
--ADDR_STATE = @ADDR_STATE and
--ADDR_ZIP=@ADDR_ZIP and
--ADDR_CO_ID = @CO_ID
Insert into CUSTOMER
values(@C_ID,@C_UNAME,@C_PASSWD,@C_FNAME,@C_LNAME,@ADDR_ID,@C_PHONE,
@C_EMAIL,@current,@current,@current,
DATEADD(hour, 2,@current),@C_DISCOUNT, 0,0,@C_BIRTHDATE,@C_DATA
)
Update SHOPPING_BASKET set
SC_C_DISCOUNT=@C_DISCOUNT, SC_C_ID=@C_ID, SC_C_FNAME=@C_FNAME,
SC_C_LNAME=@C_LNAME
where SC_SHOPPING_ID=@SC_SHOPPING_ID and SC_HOST=@SC_HOST
commit transaction add_customer
select @C_ID
end
GO

SET QUOTED_IDENTIFIER OFF
GO

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure AddNewOrder
(@O_ID numeric(10),
@O_SHIP_TYPE char(10),
@O_BILL_ADDR numeric(10),
@O_SHIP_ADDR numeric(10),
@CO_ID smallint,
@SC_SHOPPING_ID numeric(10),
@CX_TYPE char(10),
@CX_NUM numeric(16,0),
@CX_NAME char(31),
@CX_EXP datetime,
@AUTH char(15),
@SC_HOST char(20)
)
As
declare @current datetime
declare @O_C_ID numeric(10)
declare @O_SUBTOTAL numeric(17,2)
declare @O_TAX numeric (17,2)
declare @O_TOTAL numeric (17,2)
declare @O_DISCOUNT numeric (5,2)
declare @temp table (SCL_I_ID bigint, SCL_I_QTY int)

select @current= getdate()
begin
begin transaction new_order
select
@O_C_ID= SC_C_ID,@O_SUBTOTAL= SC_SUB_TOTAL,@O_TAX= SC_TAX,
@O_TOTAL= SC_TOTAL,@O_DISCOUNT= SC_C_DISCOUNT
from SHOPPING_BASKET
where SC_SHOPPING_ID=@SC_SHOPPING_ID and SC_HOST=@SC_HOST
Insert into ORDERS with (rowlock) values
(@O_ID,@O_C_ID,@current,@O_SUBTOTAL,@O_TAX,@O_TOTAL,@O_SHIP_TYPE,
-----
--Code changed on 11/ 08/ 00 from 'rand()* 7' to 'rand()* 6' to meet
--clause 2.7.3.3 of 1.2 spec
--This line of code added on 11/ 08/ 00 to meet clause 2.7.3.3 of 1.2
--spec
--It was not used during the measurement intervals of the
--11/ 6 thru 11/ 10 benchmark but tests were run 11/ 08/ 00 to
--demonstrate
--to the auditor that it had no significant impact on performance.
--See auditor's attestation letter for details.

@current+ floor(rand()* 6)+ 1,@O_BILL_ADDR,@O_SHIP_ADDR, 'Pending
')
```

```

--End of changed code
-----
Insert into CC_XACTS values
(@O_ID,@CX_TYPE,@CX_NUM,@CX_NAME,@CX_EXP,@AUTH,@O_TOTAL,@current,@CO_ID
)

Insert ORDER_LINE
(OL_ID, OL_O_ID, OL_I_ID, OL_QTY, OL_DISCOUNT, OL_COMMENTS)

Select
SCL_OL_ID,@O_ID, SCL_I_ID, SCL_I_QTY,@O_DISCOUNT, SCL_I_COMMENTS
from SHOPPING_ITEM [rowlock]
where SCL_SHOPPING_ID= @SC_SHOPPING_ID and
SCL_HOST=@SC_HOST

insert @temp (SCL_I_ID, SCL_I_QTY)
select SCL_I_ID, SCL_I_QTY
from SHOPPING_ITEM
where SCL_SHOPPING_ID= @SC_SHOPPING_ID and SCL_HOST=@SC_HOST
order by SCL_I_ID

update SHOPPING_BASKET set SC_DATE = @current
where SC_SHOPPING_ID=@SC_SHOPPING_ID and SC_HOST=@SC_HOST

Delete from SHOPPING_ITEM
where SCL_SHOPPING_ID= @SC_SHOPPING_ID and SCL_HOST=@SC_HOST
update ITEM
set I_STOCK = I_STOCK -SCL_I_QTY +
case when (I_STOCK -SCL_I_QTY > 9) then 0 else 21 end
from @temp
where SCL_I_ID = I_ID
commit transaction new_order
select @current
end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure AddToBasket
(@SC_SHOPPING_ID numeric(10),
@SCL_I_ID numeric(10),
@SCL_I_TITLE char(60),
@SCL_I_SRP numeric(17,2),
@SCL_I_COST numeric(17,2),
```

```

@scl_i_backing char(15),
@scl_i_comments char(100),
@sc_sub_total numeric(17,2),
@scl_host char(20)
)
As
begin
declare @current datetime
select @current =getdate()
begin transaction atb
update SHOPPING_BASKET set
SC_SUB_TOTAL=@SC_SUB_TOTAL, SC_DATE=@current
where SC_SHOPPING_ID=@SC_SHOPPING_ID and SC_HOST=@SCL_HOST
Insert into SHOPPING_ITEM
values(@SC_SHOPPING_ID,@SCL_I_ID, 1,@SCL_I_COST,@SCL_I_SRP,
@scl_i_title,@scl_i_backing, 0,@scl_i_comments,@scl_host)
commit transaction atb
select @current
end

```

```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure AddressCheck
(@CO_ID smallint,
@ADDR_STREET1 char(40),
@ADDR_STREET2 char(40),
@ADDR_CITY char(30),
@ADDR_STATE char(20),
@ADDR_ZIP char(10)
)
As
begin
SELECT ADDR_ID FROM ADDRESS WHERE
ADDR_CO_ID = @CO_ID and
ADDR_ZIP=@ADDR_ZIP and
ADDR_STATE = @ADDR_STATE and
ADDR_CITY = @ADDR_CITY and
ADDR_STREET1 = @ADDR_STREET1 and
ADDR_STREET2 = @ADDR_STREET2
end
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure BeginTran as
Begin Transaction
select 1
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure BestSellers
(@Category char(60))
As
begin
set transaction isolation level read uncommitted
select top 50 A_FNAME, A_LNAME, I_ID, I_TITLE
from ORDER_LINE join ITEM[ nolock] on OL_I_ID = I_ID
join AUTHOR on I_A_ID = A_ID
where OL_O_ID in (select top 3333 O_ID from ORDERS order by
O_DATE desc)
and I_SUBJECT = @Category
group by I_ID, I_TITLE, A_FNAME, A_LNAME
order by sum(OL_QTY) desc
--select getdate()
set transaction isolation level read committed
end

```

```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure BookDetailsProc
(@BookID numeric(10))
As
begin
SELECT
I_ID, I_TITLE, I_PUB_DATE, I_PUBLISHER, I_SUBJECT, I_DESC, I_IMAGE,
I_SRP,
I_COST, I_AVAIL, I_ISBN, I_PAGE, I_BACKING, I_DIMENSIONS,
A_FNAME, A_MNAME, A_LNAME, I_THUMBNAI
FROM ITEM ,AUTHOR

```

```

WHERE AUTHOR.A_ID = ITEM.I_A_ID AND ITEM.I_ID = @BookID
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure BookListProcATC001
(@CateGoryID char(60))
As
begin
SELECT TOP 50
ITEM.I_ID, ITEM.I_TITLE, AUTHOR.A_FNAME, AUTHOR.A_LNAME
FROM ITEM join AUTHOR on I_A_ID = A_ID
WHERE ITEM.I_SUBJECT = @CateGoryID
ORDER by I_TITLE
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure BookListProcATC010
(@Title varchar(60))
As
begin
declare @SearchParm varchar(80)
set @SearchParm = ' ' + @Title + '*'
SELECT top 50 I_ID, I_TITLE, A_FNAME, A_LNAME
FROM SEARCH_AUTHOR(index= test) join SEARCH_ITEM on
I_A_ID= A_ID
WHERE contains (I_TITLE, @SearchParm)
order by I_TITLE
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

SET ANSI_NULLS ON
GO

```

```

CREATE Procedure BookListProcATC100
(@Author varchar(20))
As
begin
declare @SearchParm varchar(40)
set @SearchParm = ' ' + @Author + '*'
SELECT top 50 I_ID, I_TITLE, A_FNAME, A_LNAME
FROM SEARCH_AUTHOR(index= test) join SEARCH_ITEM on
I_A_ID= A_ID
WHERE contains(A_LNAME, @SearchParm)
order by I_TITLE
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure ChangeID
(@C_ID numeric(10),
@SC_SHOPPING_ID numeric(10),
@SC_HOST char(20)
)
As
Begin
begin transaction cid
Update SHOPPING_BASKET
set SC_C_ID=@C_ID where SC_SHOPPING_ID=@SC_SHOPPING_ID and
SC_HOST=@SC_HOST
commit transaction cid
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure CheckUsername
(@C_UNAME char(20))
As
begin
DECLARE @num numeric(10)
SELECT @num= count(*) FROM CUSTOMER WHERE C_UNAME = @C_UNAME

```

```

select @num
end
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE procedure Clear_Basket as
begin
truncate TABLE SHOPPING_BASKET
TRUNCATE TABLE SHOPPING_ITEM
update SC_LOCK set NEXT_SC_SHOPPING_ID = 1
select NEXT_SC_SHOPPING_ID from SC_LOCK
end
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure CommitTran as
Commit Transaction
select 0

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure CountryCheck
(@CO_NAME char(50))
As
begin
Select CO_ID from COUNTRY where CO_NAME=@CO_NAME
end
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure CreateBasket
(@SC_SHOPPING_ID numeric(10,0),
@SC_C_ID numeric(10,0) ,
@SC_C_FNAME char(15),
@SC_C_LNAME char(15),
@SC_C_DISCOUNT numeric (5,2),
@SC_HOST char(20)
)
As
Begin
declare @current datetime
select @current =getdate()
Insert into SHOPPING_BASKET
values(@SC_SHOPPING_ID,@SC_C_ID,@current,
0,0,0,0,@SC_C_FNAME,@SC_C_LNAME,@SC_C_DISCOUNT,@SC_HOST)
select @current
end
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure DeleteBasket
(@SC_SHOPPING_ID numeric(10),
@SC_HOST char(20)
)
As
begin
begin transaction db
Delete from SHOPPING_BASKET
where SC_SHOPPING_ID=@SC_SHOPPING_ID and SC_HOST=@SC_HOST
Delete from SHOPPING_ITEM
where SCL_SHOPPING_ID=@SC_SHOPPING_ID and SCL_HOST=@SC_HOST
commit transaction db
end
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
SET ANSI_NULLS ON
GO

CREATE Procedure DeleteBasketItem
(@SC_SHOPPING_ID numeric(10),
@SCL_I_ID numeric(10),
@SC_SUB_TOTAL numeric(17,2),
@SC_HOST char(20)
)
As
Begin
declare @current datetime
select @current= getdate()
--begin transaction dbi
Update SHOPPING_BASKET with (rowlock) set
SC_DATE=@current, SC_SUB_TOTAL=@SC_SUB_TOTAL
where SC_SHOPPING_ID=@SC_SHOPPING_ID and SC_HOST=@SC_HOST
Delete from SHOPPING_ITEM with (rowlock)
where SCL_SHOPPING_ID=@SC_SHOPPING_ID and SCL_I_ID=@SCL_I_ID
and SCL_HOST=@SC_HOST
--commit transaction dbi
SELECT @current
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

Create Procedure DeleteOldBaskets
As
begin
declare @Tab table (SHOPPING_ID numeric(10) )
declare @delete_date datetime
select @delete_date = DATEADD( hh,-2, getdate())
insert @Tab (SHOPPING_ID)
select SC_SHOPPING_ID
from SHOPPING_BASKET with (readpast)
where SC_DATE < @delete_date
order by SC_SHOPPING_ID

if @@ rowcount > 0
begin
begin transaction deb1
Delete SHOPPING_BASKET with (PAGLOCK) From @Tab
where SC_SHOPPING_ID= SHOPPING_ID
Delete SHOPPING_ITEM with (PAGLOCK) From @Tab
where SCL_SHOPPING_ID= SHOPPING_ID
commit transaction deb1

```

```

end
end

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetAD
(@I_ID numeric(10))
As
begin
declare @a1 numeric(10)
declare @a2 numeric(10)
declare @a3 numeric(10)
declare @a4 numeric(10)
declare @a5 numeric(10)
--declare @current datetime
--select @current= getdate()
select @a1 = I_RELATED1, @a2 = I_RELATED2, @a3 = I_RELATED3, @a4 =
I_RELATED4, @a5 = I_RELATED5 from ITEM [rowlock]
where I_ID = @I_ID
Select I_ID, I_THUMBNAIL from ITEM [rowlock] where I_ID in
(@a1,@a2,@a3,@a4,@a5)
order by I_ID
--select @current
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure GetAddr
(@A_ID numeric(10)
)
As
begin
select
ADDR_STREET1, ADDR_STREET2, ADDR_CITY, ADDR_STATE, ADDR_ZIP, CO_NAME
from ADDRESS ,COUNTRY
where ADDR_ID=@A_ID and ADDR_CO_ID= CO_ID
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetAlsos
(@I_ID numeric(10))
As
begin
declare @a1 numeric(10)
declare @a2 numeric(10)
declare @a3 numeric(10)
declare @a4 numeric(10)
declare @a5 numeric(10)
select @a1 = I_RELATED1, @a2 = I_RELATED2, @a3 = I_RELATED3, @a4 =
I_RELATED4, @a5 = I_RELATED5 from ITEM [rowlock]
where I_ID = @I_ID
select I_IMAGE from ITEM [rowlock] where I_ID in (@a1, @a2, @a3, @a4,
@a5)
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetBasket
(@SC_HOST char(20))
As
begin
select
SC_SHOPPING_ID, SC_C_ID, SC_DATE, SC_SUB_TOTAL,
SC_TAX, SC_SHIP_COST, SC_TOTAL,
SC_C_FNAME, SC_C_LNAME, SC_C_DISCOUNT
from SHOPPING_BASKET
where SC_HOST=@SC_HOST
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetBasketWithSC_ID
(@SC_ID numeric(10),
@SC_HOST char(20))
As
begin
select
SC_SHOPPING_ID, SC_C_ID, SC_DATE, SC_SUB_TOTAL,
SC_TAX, SC_SHIP_COST, SC_TOTAL,
SC_C_FNAME, SC_C_LNAME, SC_C_DISCOUNT
from SHOPPING_BASKET
where
SC_SHOPPING_ID=@SC_ID
Update SHOPPING_BASKET
SET SC_HOST=@SC_HOST
where
SC_SHOPPING_ID=@SC_ID
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetCountryID
(@C_ID numeric(10) )
As
begin
select ADDR_CO_ID from CUSTOMER , ADDRESS where C_ADDR_ID= ADDR_ID
and C_ID=@C_ID
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetCustAddrID
(@C_ID numeric(10))

```

```

)
As
begin
select C_ADDR_ID from CUSTOMER where C_ID=@C_ID
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetCustInfo
(@C_ID numeric(10),
@SC_SHOPPING_ID numeric(10),
@SCL_HOST char(20))
As
begin

```

```

    declare @current datetime
    declare @C_DISCOUNT numeric(5,2)
    declare @C_FNAME char(15)
    declare @C_LNAME char(15)

```

```

    select @current= getdate()

```

```

    begin transaction gci

```

```

        SELECT @C_FNAME=C_FNAME, @C_LNAME=C_LNAME,
@C_DISCOUNT=C_DISCOUNT
        FROM CUSTOMER
        where C_ID=@C_ID

```

```

    SELECT

```

```

        C_UNAME, C_PASSWD, C_FNAME,C_LNAME, C_PHONE, C_EMAIL,
C_BIRTHDATE, C_DATA,
        ADDR_STREET1, ADDR_STREET2, ADDR_CITY, ADDR_STATE,
ADDR_ZIP, CO_NAME, C_DISCOUNT
        FROM CUSTOMER ,ADDRESS ,COUNTRY
        where C_ADDR_ID= ADDR_ID and ADDR_CO_ID= CO_ID and
C_ID=@C_ID

```

```

    Update CUSTOMER

```

```

        set C_LOGIN=@current, C_EXPIRATION= DATEADD(hour,
2,@current)
        where C_ID=@C_ID

```

```

    Update SHOPPING_BASKET

```

```

        set SC_C_ID=@C_ID, SC_C_DISCOUNT=@C_DISCOUNT,
SC_C_FNAME=@C_FNAME, SC_C_LNAME=@C_LNAME
        where SC_HOST = @SCL_HOST and
SC_SHOPPING_ID=@SC_SHOPPING_ID

```

```

    commit transaction gci

```

```

end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetCustName
(@C_ID numeric(10))
As

```

```

begin

```

```

Select C_FNAME, C_LNAME, C_DISCOUNT from CUSTOMER where C_ID =
@C_ID

```

```

end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetCustOrderInfo
(@C_ID numeric(10))
As

```

```

begin

```

```

DECLARE @O_ID numeric(10)

```

```

-- gnd changed 04-09-01

```

```

--select @O_ID = max(O_ID) from ORDERS where O_C_ID=@C_ID
        select top 1 @O_ID = O_ID
        from ORDERS
        where O_C_ID=@C_ID
        order by O_DATE desc

```

```

SELECT C_FNAME, C_LNAME, C_EMAIL, C_PHONE,
        O_ID, O_DATE, O_SUBTOTAL, O_TAX, O_TOTAL, O_SHIP_TYPE,
O_SHIP_DATE,

```

```

O_BILL_ADDR, O_SHIP_ADDR, CX_TYPE, O_STATUS,
ADDR_STREET1, ADDR_STREET2, ADDR_CITY, ADDR_STATE, ADDR_ZIP,
CO_NAME, CX_AUTH_ID
FROM CUSTOMER ,ADDRESS ,COUNTRY ,ORDERS ,CC_XACTS
where O_ID=@O_ID

```

```

-- and C_ID=@C_ID
-- and O_BILL_ADDR= ADDR_ID
-- and CX_O_ID=@O_ID
-- and ADDR_CO_ID= CO_ID

```

```

and C_ID=O_C_ID
and ADDR_ID=O_BILL_ADDR
AND CO_ID=ADDR_CO_ID
AND CX_O_ID=O_ID

```

```

end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

```

```

CREATE Procedure GetCustUname
(@C_ID numeric(10))
As
begin
--Stored procedure added 11/ 10/ 2000 to comply with clause 2.8.3.1
--See auditor's attestation letter
Select C_UNAME from CUSTOMER where C_ID = @C_ID
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetItem
(@SC_HOST char(20))
As
begin
select

```

```

SCL_SHOPPING_ID, SCL_I_ID, SCL_I_QTY, SCL_I_COST, SCL_I_SRP,
SCL_I_TITLE, SCL_I_BACKING
from SHOPPING_ITEM
where SCL_HOST=@SC_HOST
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetItemWithSC_ID
(@SC_ID numeric(10),
@SCL_HOST char(20))
As
begin
select
SCL_SHOPPING_ID, SCL_I_ID, SCL_I_QTY, SCL_I_COST, SCL_I_SRP,
SCL_I_TITLE, SCL_I_BACKING
from SHOPPING_ITEM
where SCL_SHOPPING_ID=@SC_ID
Update SHOPPING_ITEM set SCL_HOST=@SCL_HOST where
SCL_SHOPPING_ID=@SC_ID
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure GetMaxADDR_ID
(@Range numeric(10))
As
begin
declare @ADDR_ID numeric(10)
update ADDR_LOCK with (TABLOCK)
set @ADDR_ID = NEXT_ADDR_ID,
NEXT_ADDR_ID = NEXT_ADDR_ID+@Range
select @ADDR_ID
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure GetMaxC_ID
(@Range numeric(10))
As
begin
declare @C_ID numeric(10)
update CUST_LOCK with (TABLOCK)
set @C_ID = NEXT_C_ID,
NEXT_C_ID = NEXT_C_ID+@Range
select @C_ID
end
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure GetMaxItem
As
begin
Select max(I_ID) from ITEM
end
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure GetMaxO_ID
(@Range numeric(10))

```

```

As
begin
declare @O_ID numeric(10)
update ORDER_LOCK with (TABLOCK)
set @O_ID = NEXT_O_ID,
NEXT_O_ID = NEXT_O_ID+@Range
select @O_ID
end
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure GetMaxSC_ID
(@Range numeric(10))
As
begin
declare @SC_ID numeric(10)
update SC_LOCK with (TABLOCK)
set @SC_ID = NEXT_SC_SHOPPING_ID,
NEXT_SC_SHOPPING_ID = NEXT_SC_SHOPPING_ID+@Range
select @SC_ID
end
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure GetMaxSession
(@host char(60))
As
begin
Select max(SC_SHOPPING_ID)+ 1 from SHOPPING_BASKET where
SC_HOST=@host
end
GO

SET QUOTED_IDENTIFIER OFF

```

```

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure GetNewRelated
(@I_ID numeric( 10))
As
begin
declare @Min_O_ID numeric( 10)
declare @Max_O_ID numeric( 10)

set transaction isolation level read uncommitted

select top 10000 O_C_ID, O_ID into #temp1
from ORDERS
order by O_DATE desc

select @Min_O_ID = min( O_ID), @Max_O_ID = max( O_ID) from
#temp1

select distinct O_C_ID into #temp2
from #temp1 join ORDER_LINE [nolock] on O_ID = OL_O_ID
where OL_I_ID = @I_ID
and OL_O_ID between @Min_O_ID and @Max_O_ID
if @@rowcount <> 0

select top 5 OL_I_ID
from (#temp2 as C join #temp1 as O on C. O_C_ID =
O. O_C_ID)
join ORDER_LINE [nolock] on
O. O_ID = OL_O_ID
where OL_I_ID <> @I_ID
and OL_O_ID between @Min_O_ID and @Max_O_ID
group by OL_I_ID
order by sum( OL_QTY) desc

set transaction isolation level read committed
end
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure GetOrderDetailProc (@O_ID numeric (10))
As
begin
Select OL_QTY, A_LNAME, A_FNAME, I_TITLE
FROM ORDER_LINE , AUTHOR , ITEM [rowlock]
WHERE OL_O_ID = @O_ID AND OL_I_ID = I_ID AND A_ID = I_A_ID
end
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure GetOrderLine
(@O_ID numeric(10))
As
begin
select
OL_I_ID, I_TITLE, I_COST, OL_QTY, OL_DISCOUNT, OL_COMMENTS, I_PUBLISHER
from ORDER_LINE ,ITEM [rowlock]
where OL_I_ID= I_ID and OL_O_ID=@O_ID
order by OL_ID
end
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

-----
--This stored procedure added on 11/ 08/ 00 to meet clause 2.4.3.3 of
--1.2 spec
--It was not used during the measurement intervals of the
--11/ 6 thru 11/ 10 benchmark but tests were run 11/ 08/ 00 to
-- demonstrate
--to the auditor that it had no significant impact on performance.
--See auditor's attestation letter and application code for details.
-----

CREATE Procedure GetRelated1

```

```

(@I_ID numeric(10))
As
begin
declare @ID numeric(10)
select @ID= I_RELATED1 from ITEM[ rowlock]
where I_ID = @I_ID
Select I_ID, I_COST, I_SRP, I_TITLE, I_BACKING from ITEM[ ROWLOCK]
where
I_ID =@ID
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE PROCEDURE InitializeLocks as
begin
declare @x numeric(10)
begin transaction il
delete from ADDR_LOCK
select @x= max(ADDR_ID)+ 1 from ADDRESS
insert into ADDR_LOCK values(@x)
select @x, 'Addresses'
delete from ORDER_LOCK
select @x= max(O_ID)+ 1 from ORDERS
insert into ORDER_LOCK values(@x)
select @x, 'Orders'
delete from CUST_LOCK
select @x= max(C_ID)+ 1 from CUSTOMER
insert into CUST_LOCK values(@x)
select @x, 'Customers'
delete from SC_LOCK
delete from SHOPPING_BASKET
delete from SHOPPING_ITEM
insert into SC_LOCK values(1)
select 'Shopping baskets cleared and sessions set to 1.'
commit transaction il
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

SET ANSI_NULLS ON
GO

```

```

CREATE Procedure ItemChange (@I_ID numeric(10), @I_COST numeric(17,2),
                             @I_IMAGE char(100), @I_THUMBNAIL char(100))

```

```

As
begin
declare @Subject char(60)
begin transaction ic
Update ITEM with (rowlock)
set I_COST=@I_COST, I_IMAGE=@I_IMAGE, @Subject = I_SUBJECT,
I_THUMBNAIL=@I_THUMBNAIL, I_PUB_DATE= GETDATE()
where I_ID=@I_ID
commit transaction ic
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure NewProducts
(@Category char(60))
As
begin
set transaction isolation level read uncommitted
select top 50 A_FNAME, A_LNAME, I_ID, I_TITLE
from ITEM [nolock], AUTHOR
where I_A_ID = A_ID AND I_SUBJECT= @Category
order by I_PUB_DATE desc, I_TITLE
set transaction isolation level read committed
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure PriceCheck
(@SC_SHOPPING_ID numeric(10),

```

```

@SC_HOST char(20)
)
As
begin
declare @OL_ID numeric (10)
set @OL_ID= 0
begin transaction pc
Update SHOPPING_ITEM
set
@OL_ID=@OL_ID+ 1,
SHOPPING_ITEM.SCL_OL_ID=@OL_ID,
SHOPPING_ITEM.SCL_I_COST= ITEM.I_COST,
SHOPPING_ITEM.SCL_I_SRP= ITEM.I_SRP
FROM SHOPPING_ITEM ,ITEM
WHERE
SHOPPING_ITEM.SCL_SHOPPING_ID= @SC_SHOPPING_ID and
SHOPPING_ITEM.SCL_HOST= @SC_HOST and
SHOPPING_ITEM.SCL_I_ID= ITEM.I_ID
SELECT SCL_I_ID, SCL_I_SRP, SCL_I_COST FROM SHOPPING_ITEM
WHERE SCL_SHOPPING_ID= @SC_SHOPPING_ID and SCL_HOST= @SC_HOST
commit transaction pc
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure UnamePwdCheck
(@C_UNAME varchar(20),
@C_PASSWD varchar(20)
)
As
begin
select C_ID from CUSTOMER where C_UNAME=@C_UNAME and
C_PASSWD=@C_PASSWD
end

```

```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure UpdateBasketCosts
(@SC_SHOPPING_ID numeric(10),
@SC_HOST char(20)
)
As
begin
declare @OL_ID numeric (10)
declare @SC_SUB_TOTAL numeric (17,2)
declare @SC_SHIP_COST numeric (17,2)
declare @SC_TAX numeric (17,2)
declare @Sum_SCL_I_QTY smallint
declare @current datetime
select @current =getdate()
set @OL_ID= 0

begin transaction ubc
Update SHOPPING_ITEM
set @OL_ID=@OL_ID+ 1,
SCL_OL_ID=@OL_ID,
SCL_I_COST= I_COST,
SCL_I_SRP= I_SRP
FROM SHOPPING_ITEM [INDEX= SCL_INDX], ITEM [rowlock]
WHERE SCL_SHOPPING_ID= @SC_SHOPPING_ID and
SCL_HOST= @SC_HOST and
SCL_I_ID= I_ID
Select @SC_SUB_TOTAL = sum(SCL_I_COST * SCL_I_QTY),
@Sum_SCL_I_QTY = sum(SCL_I_QTY)
FROM SHOPPING_ITEM [rowlock]
WHERE SCL_SHOPPING_ID= @SC_SHOPPING_ID and
SCL_HOST= @SC_HOST
Update SHOPPING_BASKET
Set @SC_SUB_TOTAL = @SC_SUB_TOTAL * (1.00 - SC_C_DISCOUNT),
SC_SUB_TOTAL = @SC_SUB_TOTAL,
@SC_TAX = (@SC_SUB_TOTAL * 0.0825),
SC_TAX = @SC_TAX,
@SC_SHIP_COST = 3.00 + (1.00 * @Sum_SCL_I_QTY),
SC_SHIP_COST = @SC_SHIP_COST,
SC_TOTAL = @SC_SUB_TOTAL + @SC_SHIP_COST + @SC_TAX,
SC_DATE=@current
Where SC_SHOPPING_ID=@SC_SHOPPING_ID and
SC_HOST=@SC_HOST
SELECT SCL_I_ID, SCL_I_SRP, SCL_I_COST
FROM SHOPPING_ITEM [rowlock]
WHERE SCL_SHOPPING_ID= @SC_SHOPPING_ID and SCL_HOST= @SC_HOST
commit transaction ubc
select @current
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE Procedure UpdateBasketItem
(@SC_SHOPPING_ID numeric(10),
@I_ID numeric(10),
@I_QTY smallint,
@SC_SUB_TOTAL numeric (17,2),
@SCL_HOST char(20)
)
As
Begin
declare @current datetime
select @current =getdate()
begin transaction ubi
Update SHOPPING_BASKET with (rowlock) set
SC_SUB_TOTAL=@SC_SUB_TOTAL, SC_DATE=@current
where SC_SHOPPING_ID=@SC_SHOPPING_ID and SC_HOST=@SCL_HOST
Update SHOPPING_ITEM with (rowlock) set SCL_I_QTY=@I_QTY
where SCL_SHOPPING_ID=@SC_SHOPPING_ID and SCL_I_ID=@I_ID and
SCL_HOST=@SCL_HOST
commit transaction ubi
select @current
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure UpdateCustInfo
(@C_ID numeric(10))
As
declare @current datetime
begin
select @current= getdate()
begin transaction uci
Update CUSTOMER
set C_LOGIN=@current,
C_EXPIRATION= DATEADD(hour, 2,@current)
where C_ID=@C_ID
commit transaction uci
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure UpdateRelated
(@I_ID numeric(10),
@BOOK1 numeric(10),
@BOOK2 numeric(10),
@BOOK3 numeric(10),
@BOOK4 numeric(10),
@BOOK5 numeric(10)
)
As
Begin
begin transaction ur
Update ITEM with (rowlock) set I_RELATED1=@BOOK1,
I_RELATED2=@BOOK2,
I_RELATED3=@BOOK3, I_RELATED4=@BOOK4, I_RELATED5=@BOOK5
where I_ID=@I_ID
commit transaction ur
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

CREATE Procedure UpdateTotal
(@SC_SHOPPING_ID numeric(10),
@SC_SUB_TOTAL numeric (17,2),
@SC_TAX numeric (17,2),
@SC_SHIP_COST numeric (17,2),
@SC_TOTAL numeric (17,2),
@SCL_HOST char(20)
)
As
Begin
declare @current datetime
select @current =getdate()
begin transaction ut

```

```

Update SHOPPING_BASKET set
SC_SUB_TOTAL=@SC_SUB_TOTAL,
SC_TAX=@SC_TAX,
SC_SHIP_COST=@SC_SHIP_COST,
SC_TOTAL=@SC_TOTAL,
SC_DATE=@current
where SC_SHOPPING_ID=@SC_SHOPPING_ID and
SC_HOST=@SCL_HOST
commit transaction ut
select @current
end
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

```

use msdb
go

```

```

exec sp_add_job @job_name = 'DeleteBaskets'
go
exec sp_add_jobstep @job_name = 'DeleteBaskets',
                    @step_name = 'basket delete',
                    @subsystem = 'TSQL',
                    @database_name = 'tpcw',
                    @command = 'exec DeleteOldBaskets'

```

```
go
```

```

exec sp_add_jobschedule @job_name = 'DeleteBaskets',
                        @name = 'Nightly expired basket deleting',
                        @freq_type = 4,
                        @freq_interval = 1,
                        @active_start_time = 20000

```

```
go
```

```

exec sp_add_jobserver @job_name = 'DeleteBaskets'
go

```

## EnableFullText.sql

```
-- EnableFullText.sql
```

```
--
```

```
-- enable fulltext search in duplicate tables
```

```
--
```

```
use tpcw
```

```
go
```

```
if (select DATABASEPROPERTY(DB_NAME(), N'IsFullTextEnabled')) <> 1
```

```
exec sp_fulltext_database 'enable'
```

```
GO
```

```

if not exists (select * from dbo.sysfulltextcatalogs where name =
'AUTHOR_CAT')
exec sp_fulltext_catalog 'AUTHOR_CAT', 'create' , 'S:\CATALOGS'
GO
EXEC sp_fulltext_table 'SEARCH_AUTHOR', 'create', 'AUTHOR_CAT',
'A_INDEX_1'
go
EXEC sp_fulltext_column 'SEARCH_AUTHOR', 'A_LNAME', 'add'
go
EXEC sp_fulltext_table 'SEARCH_AUTHOR', 'activate'
go
EXEC sp_fulltext_catalog 'AUTHOR_CAT', 'start_full'
go

```

```

if not exists (select * from dbo.sysfulltextcatalogs where name =
'ITEM_CAT')
exec sp_fulltext_catalog 'ITEM_CAT', 'create' , 'S:\CATALOGS'
GO
EXEC sp_fulltext_table 'SEARCH_ITEM', 'create', 'ITEM_CAT',
'ITEM_Index_1'
go
EXEC sp_fulltext_column 'SEARCH_ITEM', 'I_TITLE', 'add'
go
EXEC sp_fulltext_table 'SEARCH_ITEM', 'activate'
go
EXEC sp_fulltext_catalog 'ITEM_CAT', 'start_full'
go

```

## Appendix D- HTML Code

### Home page

```
<HTML><HEAD><TITLE>TPC-W Home Page</TITLE><BODY BGCOLOR="#FFFFFF"><H1
ALIGN="CENTER"> TPC Web Commerce Benchmark (TPC-W)</H1><P
ALIGN="CENTER"><IMG SRC="gifs/tpclogo.gif" WIDTH="288" HEIGHT="67"
ALIGN="BOTTOM" BORDER="0"naturalsizeflag="3"></CENTER><H2
ALIGN="CENTER"> Home Page</H2><H3 ALIGN="CENTER">Welcome back <FONT
COLOR="#770000">firstname      lastname      </FONT></H3><TABLE
BORDER="0" WIDTH="660" ALIGN="CENTER"><TR ALIGN="CENTER"
VALIGN="TOP"><TD COLSPAN="5"><B><FONT SIZE="+ 1" color="#ff0000"> Click
on one of our latest books to find out more!</FONT></B></TD><TR><TR
ALIGN="CENTER" VALIGN="TOP"><TD><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=1245" target="_top"><IMG
SRC="http://image.tpcw.net/tpcw/thumb/083/08255.jpg" ALT="Book 1"
WIDTH="100" HEIGHT="150"></A></TD><TD><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=2232" target="_top"><IMG
SRC="http://image.tpcw.net/tpcw/thumb/023/02232.jpg" ALT="Book 2"
WIDTH="100" HEIGHT="150"></A></TD><TD><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=6831" target="_top"><IMG
SRC="http://image.tpcw.net/tpcw/thumb/069/06831.jpg" ALT="Book 3"
WIDTH="100" HEIGHT="150"></A></TD><TD><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=8516" target="_top"><IMG
SRC="http://image.tpcw.net/tpcw/thumb/086/08516.jpg" ALT="Book 4"
WIDTH="100" HEIGHT="150"></A></TD><TD><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=9337" target="_top"><IMG
SRC="http://image.tpcw.net/tpcw/thumb/094/09337.jpg" ALT="Book 5"
WIDTH="100" HEIGHT="150"></A></TD></TABLE><TABLE BORDER="0"
WIDTH="700" ALIGN="CENTER" CELLPADDING="6" CELLSPACING="0" BGCOLOR="#
C0C0C0"><TR ALIGN="CENTER" VALIGN="TOP" BGCOLOR="#FFFFFF"><TD
COLSPAN="2" VALIGN="MIDDLE" WIDTH="300"><IMG SRC="gifs/new.gif"
WIDTH="300" ALT="New Product"></TD><TD BGCOLOR="#FFFFFF"
WIDTH="100"></TD><TD COLSPAN="2" WIDTH="300"><IMG SRC="gifs/best.gif"
WIDTH="300" ALT="Best Seller"></TD></TR><TR><TR><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_arts.htm">ARTS</A></TD><TD><P
ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_parenting.htm">PARENTING</A></TD>
<TD WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_arts.htm">ARTS</A></TD><TD><P
ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_parenting.htm">PARENTING</A></TD>
</TR><TR><TR><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_biographies.htm">BIOGRAPHIES</A><
/ TD><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_politics.htm">POLITICS</A></TD><T
D WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_biographies.htm">BIOGRAPHIES</A>
</TD><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_politics.htm">POLITICS</A></TD><
/ TR><TR><TR><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_business.htm">BUSINESS</A></TD><T
```

```
D><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_reference.htm">REFERENCE</A></TD>
<TD WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_business.htm">BUSINESS</A></TD><
TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_children.htm">CHILDREN</A></TD><T
D><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_religion.htm">RELIGION</A></TD><T
D WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_children.htm">CHILDREN</A></TD><
TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_religion.htm">RELIGION</A></TD><
/ TR><TR><TR><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_computers.htm">COMPUTERS</A></TD>
<TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_romance.htm">ROMANCE</A></TD><TD
WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_computers.htm">COMPUTERS</A></TD>
<TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_romance.htm">ROMANCE</A></TD></T
R><TR><TR><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_cooking.htm">COOKING</A></TD><TD>
<P ALIGN="CENTER"><A HREF="http://tpcwww.tpcw.net/tpcw/new_self-
help.htm">SELF-HELP</A></TD><TD WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P
ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_cooking.htm">COOKING</A></TD><TD
><P ALIGN="CENTER"><A HREF="http://tpcwww.tpcw.net/tpcw/best_self-
help.htm">SELF-HELP</A></TD></TR><TR><TR><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_health.htm">HEALTH</A></TD><TD><P
ALIGN="CENTER"><A HREF="http://tpcwww.tpcw.net/tpcw/new_science-
nature.htm">SCIENCE-NATURE</A></TD><TD
WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_health.htm">HEALTH</A></TD><TD>
<P ALIGN="CENTER"><A HREF="http://tpcwww.tpcw.net/tpcw/best_science-
nature.htm">SCIENCE-NATURE</A></TD></TR><TR><TR><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_history.htm">HISTORY</A></TD><TD>
<P ALIGN="CENTER"><A HREF="http://tpcwww.tpcw.net/tpcw/new_science-
fiction.htm">SCIENCE-FICTION</A></TD><TD
WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_history.htm">HISTORY</A></TD><TD
><P ALIGN="CENTER"><A HREF="http://tpcwww.tpcw.net/tpcw/best_science-
fiction.htm">SCIENCE-FICTION</A></TD><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_home.htm">HOME</A></TD><TD><P
ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_sports.htm">SPORTS</A></TD><TD
WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_home.htm">HOME</A></TD><TD><P
ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/best_sports.htm">SPORTS</A></TD></TR>
<TR><TR><TD><P ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_humor.htm">HUMOR</A></TD><TD><P
ALIGN="CENTER"><A
HREF="http://tpcwww.tpcw.net/tpcw/new_mystery.htm">MYSTERY</A></TD><TD
```

WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P ALIGN="CENTER"><A  
HREF="http://tpcwww.tpcw.net/tpcw/best\_humor.htm">HUMOR</A></TD><TD><P  
ALIGN="CENTER"><A  
HREF="http://tpcwww.tpcw.net/tpcw/best\_mystery.htm">MYSTERY</A></TD></TR><TR><TD><P ALIGN="CENTER"><A  
HREF="http://tpcwww.tpcw.net/tpcw/new\_literature.htm">LITERATURE</A></TD><TD><P ALIGN="CENTER"><A  
HREF="http://tpcwww.tpcw.net/tpcw/new\_travel.htm">TRAVEL</A></TD><TD  
WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P ALIGN="CENTER"><A  
HREF="http://tpcwww.tpcw.net/tpcw/best\_literature.htm">LITERATURE</A></TD><TD><P ALIGN="CENTER"><A  
HREF="http://tpcwww.tpcw.net/tpcw/best\_travel.htm">TRAVEL</A></TD></TR><TR><TD><P ALIGN="CENTER"><A HREF="http://tpcwww.tpcw.net/tpcw/new\_non-fiction.htm">NON-FICTION</A></TD><TD><P ALIGN="CENTER"><A  
HREF="http://tpcwww.tpcw.net/tpcw/new\_youth.htm">YOUTH</A></TD><TD  
WIDTH="50" BGCOLOR="#FFFFFF"></TD><TD><P ALIGN="CENTER"><A  
HREF="http://tpcwww.tpcw.net/tpcw/best\_non-fiction.htm">NON-FICTION</A></TD><TD><P ALIGN="CENTER"><A  
HREF="http://tpcwww.tpcw.net/tpcw/best\_youth.htm">YOUTH</A></TD></TR></TABLE><P><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?b&ADD\_FLAG=N"><IMG  
SRC="gifs/cart.gif" ALT="Shopping Cart" BORDER="0" WIDTH="120"  
HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?h"><IMG  
SRC="gifs/search.gif" ALT="Search" BORDER="0" WIDTH="120"  
HEIGHT="30"></A><A HREF="https://tpcwww.tpcw.net/tpcw/tpcw.dll?f"><IMG  
SRC="gifs/status.gif" ALT="Order Status" BORDER="0" WIDTH="120"  
HEIGHT="30"></A></P></CENTER></BODY></HTML>

#### Shopping Cart page

<HTML><HEAD><TITLE>TPC W Shopping Cart</TITLE></HEAD><BODY  
BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-  
W)</H1><CENTER><IMG SRC="gifs/tpclogo.gif" ALIGN="BOTTOM" BORDER="0"  
WIDTH="288" HEIGHT="67"></CENTER><H2 ALIGN="CENTER"> Shopping Cart  
Page</H2><TABLE BORDER="0" WIDTH="660" ALIGN="CENTER"><TR  
ALIGN="CENTER" VALIGN="TOP"><TD COLSPAN="5"><B><FONT SIZE="+ 1"  
color="#ff0000"> Click on one of our latest books to find out  
more!</FONT></B></TD></TR><TR ALIGN="CENTER" VALIGN="TOP"><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=6976" target="\_top"><IMG  
SRC="http://image.tpcw.net/tpcw/thumb/070/06976.jpg" ALT="Book 1"  
WIDTH="100" HEIGHT="150"></A></TD><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=7671" target="\_top"><IMG  
SRC="http://image.tpcw.net/tpcw/thumb/077/07671.jpg" ALT="Book 2"  
WIDTH="100" HEIGHT="150"></A></TD><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=8793" target="\_top"><IMG  
SRC="http://image.tpcw.net/tpcw/thumb/088/08793.jpg" ALT="Book 3"  
WIDTH="100" HEIGHT="150"></A></TD><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=8947" target="\_top"><IMG  
SRC="http://image.tpcw.net/tpcw/thumb/090/08947.jpg" ALT="Book 4"  
WIDTH="100" HEIGHT="150"></A></TD><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=9249" target="\_top"><IMG  
SRC="http://image.tpcw.net/tpcw/thumb/093/09249.jpg" ALT="Book 5"  
WIDTH="100" HEIGHT="150"></A></TD></TABLE><FORM  
ACTION="http://tpcwww.tpcw.net/tpcw/tpcw.dll?METHOD="GET"><INPUT

TYPE="HIDDEN" NAME="o" VALUE=""><CENTER><P><TABLE  
BORDER="0"><TR><TD><B> Qty</B></TD><TD><B> Product</B></TD></TR><TR><TD  
VALIGN="TOP"><INPUT TYPE="HIDDEN" NAME="ITEM1" VALUE="9337"><INPUT  
NAME="QTY1" VALUE="1" SIZE="3"></TD><TD VALIGN="TOP"><I>BABABAOGSEULAT  
Human, alone centres </I>- Backing: AUDIO  
<BR>\$.8107.14, <FONT COLOR="#AA0000"><B>Your Price:  
<B>\$5107.50</B></TD></TR></TABLE><B><I>Subtotal price: \$5107.50  
</B><BR>Fri Dec 07 14:27:10 2001  
</I></P><P><BR><A HREF="http://tpcwww.tpcw.net/tpcw/custreg.htm"><IMG  
SRC="gifs/checkout.gif" ALT="Check Out" BORDER="0" WIDTH="120"  
HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a"><IMG  
SRC="gifs/home.gif" ALT="Home" BORDER="0" WIDTH="120"  
HEIGHT="30"></A><P> If you have changed the quantities and/or taken  
anything out<BR>of your shopping cart, click here to refresh your  
shopping cart:</P><P><INPUT TYPE="IMAGE" NAME="Refresh"  
SRC="gifs/refresh.gif" WIDTH="120"  
HEIGHT="30"></CENTER></FORM></BODY></HTML>

#### Customer Registration page

<HTML><HEAD><TITLE>Customer Registration</TITLE></HEAD><BODY  
BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER"> TPC Web Commerce Benchmark (TPC-  
W)</H1><H1 ALIGN="CENTER"><IMG SRC="gifs/tpclogo.gif" ALIGN="BOTTOM"  
BORDER="0" WIDTH="288" HEIGHT="67"></H1><H2 ALIGN="CENTER"> Customer  
Registration Page</H2><FORM  
ACTION="https://tpcwww.tpcw.net/tpcw/tpcw.dll? METHOD="GET">  
<input TYPE="HIDDEN" NAME=d VALUE=>  
<BLOCKQUOTE><BLOCKQUOTE><HR><TABLE BORDER="0"><TR><TD><INPUT  
TYPE="RADIO" NAME="RETURNFLAG" VALUE="Y">I am an existing  
customer</TD></TR><TR><TD><INPUT TYPE="RADIO"  
NAME="RETURNFLAG" VALUE="N" CHECKED="CHECKED">I am a first time  
customer</TD></TR></TABLE><HR><P><B>If you're an existing customer,  
enter your Username and Password:</B><BR><BR></P><TABLE><TR  
ALIGN="LEFT"><TD>User ID: <INPUT NAME="USERNAME" SIZE="23"></TD></TR><TR  
ALIGN="LEFT"><TD> Password: <INPUT TYPE="PASSWORD"  
NAME="password" SIZE="14"></TD></TR></TABLE><HR><P><B>If you're a first  
time customer, enter the detail below:</B><BR><BR></P><TABLE><TR><TD>  
Enter your birth date (mm/dd/yyyy):</TD><TD><INPUT  
NAME="BIRTHDATE" SIZE="10" VALUE="01/01/1998"></TD></TR><TR><TD> Enter  
your First Name:</TD><TD><INPUT NAME="FIRSTNAME" SIZE="15"  
VALUE="firstname"></TD></TR><TR><TD> Enter your Last  
Name:</TD><TD><INPUT NAME="LASTNAME" SIZE="15"  
VALUE="lastname"></TD></TR><TR><TD> Enter your Address  
1:</TD><TD><INPUT NAME="ADDRESS1" SIZE="40"  
VALUE="street1"></TD></TR><TR><TD> Enter your Address 2:</TD><TD><INPUT  
NAME="ADDRESS2" SIZE="40" VALUE="street2"></TD></TR><TR><TD> Enter your  
City, State, Zip:</TD><TD><INPUT NAME="CITY" SIZE="30" VALUE="city">  
<INPUT NAME="STATE" SIZE="20" VALUE="state"> <INPUT  
NAME="ZIP" SIZE="10" VALUE="zip"></TD></TR><TR><TD> Enter your  
Country:</TD><TD><INPUT NAME="COUNTRY" SIZE="50" VALUE="United  
States"></TD></TR><TR><TD> Enter your Phone:</TD><TD><INPUT  
NAME="PHONE" SIZE="16" VALUE="123-456-7890"></TD></TR><TR><TD> Enter your  
E-mail:</TD><TD><INPUT NAME="EMAIL" SIZE="50"  
VALUE="email"></TD></TR></TABLE><HR><TABLE><TR><TD COLSPAN="2"> Other  
Customer Data: <TEXTAREA NAME="DATA" ROWS="4" COLS="65">

information</TEXTAREA></TD></TR></TABLE><HR></BLOCKQUOTE></BLOCKQUOTE><CENTER><INPUT TYPE="IMAGE" SRC="gifs/submit.gif"NAME="Submit" BORDER="0"VSPACE="0" WIDTH="120" HEIGHT="30"><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?h"><IMG SRC="gifs/search.gif" ALT="Search Item" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120" HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a"><IMG SRC="gifs/home.gif" ALT="Home Page" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120" HEIGHT="30"></A></P></FORM></BODY></HTML>

#### Buy Request page

<HTML><HEAD><TITLE> TPC-W Buy Request</TITLE></HEAD><BODY BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER"> TPC Web Commerce Benchmark (TPC-W) </H1><H2 ALIGN="CENTER"> Buy Request Page</H2><HR><FORM ACTION="https://tpcwww.tpcw.net/tpcw/tpcw.dll?" METHOD="GET"><INPUT TYPE="HIDDEN" NAME="e" VALUE=""><TABLE BORDER="0" WIDTH="90%" CELLSPACING="0" CELLPADDING="0"><TR ALIGN="LEFT" VALIGN="TOP"><TD VALIGN="TOP" WIDTH="45%"><H2>Billing Information:</H2><TABLE WIDTH="100%" BORDER="0" CELLSPACING="0" CELLPADDING="0"><TR><TD>Firstname:</TD><TD>firstname</TD></TR><TR><TD>Lastname:</TD><TD>lastname</TD></TR><TR><TD>Addr\_street\_1:</TD><TD>street1</TD></TR><TR><TD>Addr\_street\_2:</TD><TD>street2</TD></TR><TR><TD>City:</TD><TD>city</TD></TR><TR><TD>State:</TD><TD>state</TD></TR><TR><TD>Zip:</TD><TD>zip</TD></TR><TR><TD>Country:</TD><TD>United States</TD></TR><TR><TD>Email:</TD><TD>email</TD></TR><TR><TD>Phone:</TD><TD>123-456-7890</TD></TR><TR><TD>Username:</TD><TD>OGREREBABAREREULAT</TD></TR><TR><TD>User ID:</TD><TD>144004476</TD></TR></TABLE></TD><TD ALIGN="TOP" WIDTH="45%"><H2>Shipping Information:</H2><TABLE WIDTH="100%" BORDER="0" CELLSPACING="0" CELLPADDING="0"><TR><TD WIDTH="50%">Addr\_street\_1:</TD><TD WIDTH="50%"><INPUT NAME="SHIPADD1" SIZE="40" VALUE="street1"></TD></TR><TR><TD>Addr\_street\_2:</TD><TD><INPUT NAME="SHIPADD2" SIZE="40" VALUE="street2"></TD></TR><TR><TD>City:</TD><TD><INPUT NAME="SHIPCITY" SIZE="30" VALUE="city"></TD></TR><TR><TD>State:</TD><TD><INPUT NAME="SHIPSTATE" SIZE="20" VALUE="state"></TD></TR><TR><TD>Zip:</TD><TD><INPUT NAME="SHIPZIP" SIZE="10" VALUE="zip"></TD></TR><TR><TD>Country:</TD><TD><INPUT NAME="SHIPCOUNTRY" SIZE="40" VALUE="United States"></TD></TR></TABLE></TD></TR></TABLE><HR><H2>Order Information:</H2><TABLE BORDER="1" CELLSPACING="0" CELLPADDING="0"><TR><TD><B> Qty</B></TD><TD><B> Product</B></TD><TR><TD VALIGN="TOP">1</TD><TD VALIGN="TOP">Title: <I>BABABAOGSEULAT Human, alone centres </I>- Backing: AUDIO <BR>SRP.\$8107.14, <FONT COLOR="#AA0000"><B>Your Price: \$5107.50</B></FONT></TD></TR></TABLE><BR><TABLE BORDER="0"><TR><TD><B>Subtotal with discount (49%):</B></TD><TD ALIGN="RIGHT"><B>\$2604.83</B></TD></TR><TR><TD><B>Tax</B></TD><TD ALIGN="RIGHT"><B>\$214.90</B></TD></TR><TR><TD><B>Shipping & Handling</B></TD><TD ALIGN="RIGHT"><B>\$4.00</B></TD></TR><TR><TD><B>Total</B></TD><TD ALIGN="RIGHT"><B>\$2823.73</B></TD></TR></TABLE><HR><TABLE BORDER="1" CELLPADDING="5" CELLSPACING="0"><TR><TD> Credit Card Type</TD><TD><INPUT

TYPE="RADIO" NAME="CCTYPE" VALUE="Visa" checked> VISA<INPUT TYPE="RADIO"NAME="CCTYPE" VALUE="MASTERCARD"> MASTERCARD<INPUT TYPE="RADIO"NAME="CCTYPE" VALUE="DISCOVER"> DISCOVER<INPUT TYPE="RADIO"NAME="CCTYPE" VALUE="AMEX"> AMERICAN EXPRESS<INPUT TYPE="RADIO"NAME="CCTYPE" VALUE="DINERS"> DINERS</TD></TR><TR><TD> Name on Credit Card</TD><TD><INPUT NAME="CCNAME"SIZE="30" VALUE="John Doe"></TD></TR><TR><TD> Credit Credit Number</TD><TD><INPUT NAME="CCNUM"SIZE="16" VALUE="1234567890123456"></TD></TR><TR><TD> Credit Credit Expiration Date</TD><TD><INPUT NAME="CCDATE"SIZE="15" VALUE="02/02/1999"></TD></TR><TR><TD> Shipping Method</TD><TD><INPUT TYPE="RADIO" NAME="SHIPMETHOD" VALUE="AIR" checked>AIR<INPUT TYPE="RADIO" NAME="SHIPMETHOD" VALUE="UPS">UPS<INPUT TYPE="RADIO" NAME="SHIPMETHOD" VALUE="FEDEX">FEDEX<INPUT TYPE="RADIO" NAME="SHIPMETHOD" VALUE="SHIP">SHIP<INPUT TYPE="RADIO" NAME="SHIPMETHOD" VALUE="COURIER">COURIER<INPUT TYPE="RADIO" NAME="SHIPMETHOD" VALUE="MAIL">MAIL</TD></TR></TABLE><P ALIGN="CENTER"><INPUT TYPE="IMAGE"NAME="Submit" SRC="gifs/processorder.gif" WIDTH="120" HEIGHT="30"><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?b&ADD\_FLAG=N"><IMG SRC="gifs/cart.gif" ALT="Shopping Cart" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120" HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a"><IMG SRC="gifs/home.gif" ALT="Home" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120" HEIGHT="30"></A></P></BLOCKQUOTE></BLOCKQUOTE></BLOCKQUOTE></BODY></HTML>

#### Buy Confirm page

<HTML><HEAD><TITLE> Order Confirmation</TITLE></HEAD><BODY BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-W) </H1><H2 ALIGN="CENTER">Buy Confirm Page</H2><BLOCKQUOTE><BLOCKQUOTE><BLOCKQUOTE><H2 ALIGN="LEFT"> Order Information:</H2><TABLE BORDER="1" CELLSPACING="0" CELLPADDING="0"><TR><TD><B> Qty</B></TD><TD><B> Product</B></TD><TR><TD VALIGN="TOP">1</TD><TD VALIGN="TOP"><I>Title: BABABAOGSEULAT Human, alone centres </I>- Backing: AUDIO <BR>SRP.\$8107.14, <FONT COLOR="#AA0000"><B>Your Price: \$5107.50</B></FONT></TD></TR></TABLE><H2 ALIGN="LEFT"> Your Order has been processed.</H2><TABLE BORDER="1" CELLPADDING="5" CELLSPACING="0"><TR><TD><H4>Subtotal with discount (49%):</TD><TD><H4>\$2604.83</TD></TR><TR><TD><H4>Tax (8.25%):</TD><TD><H4>\$214.90</TD></TR><TR><TD><H4>Shipping:</TD><TD><H4>\$4.00</TD></TR><TR><TD><H4>Total:</TD><TD><H4>\$2823.73</TD></TR></TABLE><P><BR></P><H2>Confirmation Number: 129627831</H2><H1>Thank you for shopping at TPC-W</H1><P></P><CENTER><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?h"><IMG SRC="gifs/search.gif" ALT="Search Item" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120" HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a"><IMG SRC="gifs/home.gif" ALT="Home Page" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120" HEIGHT="30"></A></CENTER></BLOCKQUOTE></BLOCKQUOTE></BLOCKQUOTE></BODY></HTML>

#### Order Inquiry page

```
<HTML><HEAD><TITLE>TPC-W Order Inquiry Page</TITLE></HEAD><BODY
BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-
W)</H1><H2 ALIGN="CENTER">Order Inquiry Page</H2><FORM
ACTION="https://tpcwww.tpcw.net/tpcw/tpcw.dll?" METHOD="GET"><INPUT
TYPE="HIDDEN" NAME="g" VALUE=""><TABLE
ALIGN="CENTER"><TR><TD><H4>Username:</TD><TD><INPUT NAME="USERNAME"
SIZE="10" value="OGREREABABAREULAT
"></TD></TR><TR><TD><H4>Password:</TD><TD><INPUT TYPE="PASSWORD"
NAME="password" SIZE="10"></TD></TR></TABLE><P
ALIGN="CENTER"><BR><INPUT TYPE="IMAGE" NAME="Submit"
SRC="gifs/status.gif" WIDTH="120" HEIGHT="30"><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?h"><IMG
SRC="gifs/search.gif" ALT="Search" BORDER="0" WIDTH="120"
HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a"><IMG
SRC="gifs/home.gif" ALT="Home" BORDER="0" WIDTH="120"
HEIGHT="30"></A></FORM></BODY></HTML>
```

#### Order Display page

```
<HTML><HEAD><TITLE>TPC-W Order Display Page</TITLE></HEAD><BODY
BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-
W)</H1><H2 ALIGN="CENTER">Order Display
Page</H2><BLOCKQUOTE><BLOCKQUOTE><BLOCKQUOTE><HR><P> Order
ID: 129627831<BR>Authorization ID: d{ys+ AFUYPL?5<BR>Order Placed on
12/7/2001<BR>Shipping Type: AIR <BR>Ship Date:
12/12/2001<BR>Order Subtotal: $2604.83<BR>Order Tax: $214.90<BR>Order
Total: $2823.73<BR></P><TABLE BORDER="0" WIDTH="80%"><TR><TD><B> Bill
To:</B></TD><TD><B> Ship To:</B></TD></TR><TR><TD
COLSPAN="2"><H4>firstname lastname </TD></TR><TR><TD
WIDTH="50%"><ADDRESS>street1
<BR>street2 <BR>city
<BR>state zip <BR>United States
<BR><BR>Email: email
<BR>Phone: 123-456-7890 </ADDRESS><BR><P>Credit Card Type: Visa
<BR>Order Status: Pending
</P></TD><TD WIDTH="50%" VALIGN="TOP"><ADDRESS>street1
<BR>street2 <BR>city
<BR>state zip <BR>United States
</ADDRESS></TD></TR></TABLE></BLOCKQUOTE></BLOCKQUOTE></BL
OCKQUOTE><CENTER><TABLE BORDER="1"
CELLPADDING="5" CELLSPACING="0"><TR><TD><H4> Item #</TD><TD><H4>
Title</TD><TD><H4> Cost</TD><TD><H4> Qty</TD><TD><H4>
Discount</TD><TD><H4> Comment</TD></TR><TR><TD><H4>9337</TD><TD>
VALIGN=" TOP"><H4> Title: BABABAOGSEULAT Human, alone centres
<BR> Publisher: 4fZS;!_Kp WR:5-L#D@P41q)Q-%ReCUWX7#NoIUf65D}&#x26;tVL&)%#f
</H4></TD><TD><H4>$5107.50</H4></TD><TD><H4>1</H4></TD><TD><H4>49%</H4>
</TD><TD><H4>OLHYPP17MvrhDbcvVNx08FwCe5pcc
</H4></TD></TR></TABLE><BR><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?h"><IMG
SRC="gifs/search.gif" ALT="Search" BORDER="0" WIDTH="120"
HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a"><IMG
SRC="gifs/home.gif" ALT="Home" BORDER="0" WIDTH="120"
HEIGHT="30"></A></CENTER></BODY></HTML>
```

#### Search Request page

```
<HTML><HEAD><TITLE>Search Request Page</TITLE></HEAD><BODY><CENTER><H1
ALIGN="CENTER"> TPC W Commerce Benchmark (TPC-W)</H1><H2
ALIGN="CENTER"><IMG SRC="gifs/tpclogo.gif" ALIGN="BOTTOM" BORDER="0"
WIDTH="288" HEIGHT="67"></H2><H2 ALIGN="CENTER"> Search Request
Page</H2><TABLE BORDER="0" WIDTH="660" ALIGN="CENTER"><TR
ALIGN="CENTER" VALIGN="TOP"><TD COLSPAN="5"><B>FONT SIZE="+ 1"
color="#ff0000"> Click on one of our latest books to find out
more!</FONT></B></TD></TR><TR ALIGN="CENTER" VALIGN="TOP"><TD><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=1518" target="_top"><IMG
SRC="http://image.tpcw.net/tpcw/thumb/003/00234.jpg" ALT="Book 1"
WIDTH="100" HEIGHT="150"></A></TD><TD><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=3585" target="_top"><IMG
SRC="http://image.tpcw.net/tpcw/thumb/036/03585.jpg" ALT="Book 2"
WIDTH="100" HEIGHT="150"></A></TD><TD><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=3688" target="_top"><IMG
SRC="http://image.tpcw.net/tpcw/thumb/037/03688.jpg" ALT="Book 3"
WIDTH="100" HEIGHT="150"></A></TD><TD><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=9386" target="_top"><IMG
SRC="http://image.tpcw.net/tpcw/thumb/094/09386.jpg" ALT="Book 4"
WIDTH="100" HEIGHT="150"></A></TD><TD><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=9487" target="_top"><IMG
SRC="http://image.tpcw.net/tpcw/thumb/088/08773.jpg" ALT="Book 5"
WIDTH="100" HEIGHT="150"></A></TD></TABLE><FORM
ACTION="/tpcw/tpcw.dll?i" METHOD="GET"><TABLE ALIGN="CENTER"><TR><TD
ALIGN="RIGHT"><H3> Search by:</h3></TD><TD
WIDTH="100"></TD></TR><TR><TD ALIGN="RIGHT"><select NAME="i"
SIZE="1"><option VALUE="author" selected="selected">
Author</option><option VALUE="title"> Title</option><option
VALUE="subject"> Subject</option></select></TD><TD><INPUT
NAME="SEARCHSTRING" SIZE="30"></TD></TR></TABLE><P ALIGN="CENTER"><INPUT
TYPE="IMAGE" NAME="Submit" SRC="gifs/submit.gif" WIDTH="120"
HEIGHT="30"><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a"><IMG
SRC="gifs/home.gif" ALT="Home" BORDER="0" WIDTH="120"
HEIGHT="30"></A><A
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?b&ADD_FLAG=N"><IMG
SRC="gifs/cart.gif" ALT="Shopping Cart" BORDER="0" WIDTH="120"
HEIGHT="30"></A></P></FORM></BODY></HTML>
```

#### Search Result page

```
<HTML><HEAD><TITLE>Search Result</TITLE></HEAD><BODY
BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-
W)</H1><P ALIGN="CENTER"><IMG SRC="gifs/tpclogo.gif" WIDTH="288"
HEIGHT="67" ALIGN="BOTTOM" BORDER="0"></P><H2 ALIGN="CENTER">Search
Results Page - author: Baba</H2><IFRAME
SRC="http://search.tpcw.net/tpcw/tpcw.dll?p=5582" WIDTH=800 HEIGHT=200
MARGINWIDTH=0" MARGINHEIGHT=0" SCROLLING="NO" FRAMEBORDER="0">
</IFRAME><IFRAME
SRC="http://search.tpcw.net/tpcw/list.dll?i=author&value=Baba"
WIDTH=800 HEIGHT=400 MARGINWIDTH=0" MARGINHEIGHT=0" FRAMEBORDER="0">
</IFRAME>
```

#### Search Result frame 1

<TABLE BORDER="0" WIDTH="660" ALIGN="CENTER"><TR ALIGN="CENTER" VALIGN="TOP"><TD COLSPAN="5"><B><FONT SIZE="+ 1" color="#ff0000"> Click on one of our latest books to find out more!</FONT></B></TD></TR><TR ALIGN="CENTER" VALIGN="TOP"><TD><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=242" target="\_top"><IMG SRC="http://image.tpcw.net/tpcw/thumb/003/00242.jpg" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD><TD><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=1965" target="\_top"><IMG SRC="http://image.tpcw.net/tpcw/thumb/020/01965.jpg" ALT="Book 2" WIDTH="100" HEIGHT="150"></A></TD><TD><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=4948" target="\_top"><IMG SRC="http://image.tpcw.net/tpcw/thumb/050/04948.jpg" ALT="Book 3" WIDTH="100" HEIGHT="150"></A></TD><TD><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=5336" target="\_top"><IMG SRC="http://image.tpcw.net/tpcw/thumb/054/05336.jpg" ALT="Book 4" WIDTH="100" HEIGHT="150"></A></TD><TD><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=9188" target="\_top"><IMG SRC="http://image.tpcw.net/tpcw/thumb/092/09188.jpg" ALT="Book 5" WIDTH="100" HEIGHT="150"></A></TD></TABLE>

**Search Result frame 2**

<TABLE BORDER="1" CELLPADDING="1" CELLSPACING="1"><TR><TD><FONT SIZE="+1">Author</FONT></TD><TD><FONT SIZE="+1">Title</FONT></TD></TR><TR><TD><I>\_,%[H\_tlycKx\_xPZ\IV]M BABABABAINOGSE </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1582" target="\_top">Able BABABAOGSEINAL advantages want.</A></TD></TR><TR><TD><I>\_/Z~ps}E8qim9DB BABABABANGNGUL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5204" target="\_top">Able banks BABABAOGRINGOGfill.</A></TD></TR><TR><TD><I>fb-Pd BABABABAATULRE </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6009" target="\_top">Able terms BABABABAINATRI justify by the</A></TD></TR><TR><TD><I>\_hj5n=I/k&!3%6&U BABABABAOGSEBAPT9W </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2149" target="\_top">Able tools do sell with the certain, far BABABABARIATRI</A></TD></TR><TR><TD><I>-@R BABABABARIREAL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8866" target="\_top">Able, BABABABAALALUL japanese claims into the great,</A></TD></TR><TR><TD><I>?V;lm/4eX\*6E BABABABASEOGNG </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6947" target="\_top">Able, BABABABABABARE organic problems</A></TD></TR><TR><TD><I>+\_O BABABABABAOGALXEbh </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8753" target="\_top">Able, BABABABAINSERIempty resources</A></TD></TR><TR><TD><I>2#~\*Rd(1DcZ BABABABANGATBA </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=960" target="\_top">Able, BABABABANGATBAYellow visitors must meet.</A></TD></TR><TR><TD><I>0u-s0c 6vT1;xdMA BABABABAALRESE </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7984" target="\_top">Able, BABABABAOGINOG leading</A></TD></TR><TR><TD><I>\*M77v BABABABAOGRIRI </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3963" target="\_top">Able, BABABABAOGREBALikely years

</A></TD></TR><TR><TD><I>f\$L&DSQSx@L:+0 BABABABAOGBAAT </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3121" target="\_top">Able, BABABAOGALATAT able shares make. Little, net attempts </A></TD></TR><TR><TD><I>.(f3 BABABABABAOGBAIT </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4837" target="\_top">Able, BABABAOGATALRI additional eyes in the healthy </A></TD></TR><TR><TD><I>(upd%fxMyS]W8@ BABABABABABANG </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8400" target="\_top">Able, BABABAOGBAALNGrare women say </A></TD></TR><TR><TD><I>7X,]Q7^vVKPH BABABABAATATNG </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3856" target="\_top">Able, BABABAOGBABARI dead </A></TD></TR><TR><TD><I>-iePfc3fR-G1W BABABABAATRIRI </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3577" target="\_top">Able, BABABAOGINALOGsoft expectations in the cold, full </A></TD></TR><TR><TD><I>#:th BABABABAREULBA </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2384" target="\_top">Able, BABABAOGNINBA basic </A></TD></TR><TR><TD><I>fp].Ind@\_\$ BABABABARINGAL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3845" target="\_top">Able, BABABAOGREALULadequate </A></TD></TR><TR><TD><I>V{01 , -:0rRNIqh98: BABABABAREATAL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8648" target="\_top">Able, BABABAOGRERIRI available mothers could read. Papers </A></TD></TR><TR><TD><I>:~+rC[\$Pykh]]{F4L; BABABABAALALAT </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6511" target="\_top">Able, BABABAOGRIATSE vulnerable points try to the white, </A></TD></TR><TR><TD><I>x\*a/\*%OOLj.|wdZ BABABABAOGBASE4 </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1914" target="\_top">Able, dangerous materials should BABABAOGNNGOGREget.</A></TD></TR><TR><TD><I>G#[7LGQt%eFa0]u@ BABABABAALALAT </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2908" target="\_top">Able, different BABABABAREATRE rules might write groups.</A></TD></TR><TR><TD><I>m-zPH!E.!iMHDO.\_ BABABABAINRIAT </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3885" target="\_top">Able, different BABABAOGNNGNGNG needs try. Sad </A></TD></TR><TR><TD><I>?dGn BABABABAATBANG </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7089" target="\_top">Able, elderly BABABABARIRIUL fingers </A></TD></TR><TR><TD><I>|PZ?x\$d[&~]zd|MYB BABABABASEALAT </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2238" target="\_top">Able, european vehicles stand. BABABAOGOGALIN Rich, western </A></TD></TR><TR><TD><I>-lPsZs^e) BABABABAULRIIN </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4503" target="\_top">Able, extra BABABAOGINULOG meetings for the other, </A></TD></TR><TR><TD><I>=B@AQaQDDjw, BABABABAREALIN </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6247" target="\_top">Able, foreign BABABABANGBAALpatients will have to see </A></TD></TR><TR><TD><I>D&&LXw2/Q BABABABANGALRImfIzlu</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1332" target="\_top">Able, full others BABABAOGRIRIAL ought to cost-- Plans </A></TD></TR><TR><TD><I>Q10 BABABABAATOGAL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4093" target="\_top">Able, general BABABABASEREUL programmes could become!

</A></TD></TR><TR><TD><I>nfb\_~Sg@)cI7 BABABABARIRENG  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1064"  
target="\_top">Able, good prices want. BABABAOGBAATRE Positive,  
</A></TD></TR><TR><TD><I>t(5-q7\_(I]W\_&Ci++ BABABABAREOGRI  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6764"  
target="\_top">Able, important BABABABASESERI thanks display in the  
</A></TD></TR><TR><TD><I>78fhvMIB[o\$\$ BABABABASEBASE  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6932"  
target="\_top">Able, large BABABABARINGUL women could have to  
</A></TD></TR><TR><TD><I>KK@1MijR~7\_JN,\$H BABABABARISEUL  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4454"  
target="\_top">Able, married pupils might BABABAOGBAATBA take by the  
</A></TD></TR><TR><TD><I>OT]S9guvl5hRMWX BABABABARESEIN  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4082"  
target="\_top">Able, national years answer for BABABABAINBARE the  
</A></TD></TR><TR><TD><I>O#eQbqWQ]%%&2Y8B9 BABABABASEOGRE6  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6134"  
target="\_top">Able, old results might BABABABAINBAAL call. Other  
</A></TD></TR><TR><TD><I>}RJBkp\*CE~; BABABABARIRIOG  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=331"  
target="\_top">Able, political BABABABARIRIOG purposes  
</A></TD></TR><TR><TD><I>,F8eJJD\_;Gx:  
BABABABABAREOGUXX24I</I></TD><TD><A  
HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5953" target="\_top">Able,  
reliable cells in the available, BABABABAATALSE true  
</A></TD></TR><TR><TD><I>07RTRh=? We:\*UA BABABABAULREAT  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6405"  
target="\_top">Able, rural methods shall suit. Happy,  
BABABABAREBAOGcivil </A></TD></TR><TR><TD><I>j.uaTF4zmf  
BABABABAINRIUL </I></TD><TD><A  
HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7367" target="\_top">Able,  
severe BABABAOGSENGSEconsequences may  
</A></TD></TR><TR><TD><I>e\_ZRaV5owfT[( (8Dze BABABABAALSEBA  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5300"  
target="\_top">Able, specific BABABABAALULNGcharacters must  
</A></TD></TR><TR><TD><I>,%DBD]9Y2bZ1SBpo BABABABABAALNG  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2259"  
target="\_top">Able, young words can BABABAOGREATRE ensure  
</A></TD></TR><TR><TD><I>Ghl) BABABABASEINOGQHH  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=581"  
target="\_top">About BABABABASEINOG video-taped authorities make on the  
</A></TD></TR><TR><TD><I>d;!;xn{Owuo A\*1 BABABABAATOGRE  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2312"  
target="\_top">Above young times BABABABABAATINshould discuss;  
</A></TD></TR><TR><TD><I>JHpcu:\$fYj12ofjHce%  
BABABABAOGATREmWYr3qc</I></TD><TD><A  
HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3969" target="\_top">Abroad  
sweet BABABABAINATAT steps  
</A></TD></TR><TR><TD><I>BLh9Dmhwa&M BABABABABAATRE7jx  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=9928"  
target="\_top">Absolute arts consult BABABAOGRIRERE with  
</A></TD></TR><TR><TD><I>GD?tUWQpvB;\$ER  
BABABABAULOGAL8S6Q7Y</I></TD><TD><A  
HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4659"  
target="\_top">Absolute trees say. BABABABAULALOG Occasions remove

</A></TD></TR><TR><TD><I>4\*9j(E;3MzUrilj9&G^ BABABABAULSEAT  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3000"  
target="\_top">Absolute, BABABAOGULSESEgood parents could have  
</A></TD></TR><TR><TD><I>N\_:+d)K2B,Kx%P3a\$uQ. BABABABAINNGALfmsI3  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5224"  
target="\_top">Absolute, final teams BABABABAOGRENG could  
</A></TD></TR><TR><TD><I>Xa6AEI.fEwRQ@[kcXrg# BABABABAALREAL  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7568"  
target="\_top">Absolute, national BABABABANGNGUL types would take.  
Stores </A></TD></TR><TR><TD><I>R6y BABABABAULINAT  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6176"  
target="\_top">Absolute, supreme patients might result BABABAOGOGSEIN  
just </A></TD></TR><TR><TD><I>2#cT;~v NN. BABABABAATBAAL09n  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8217"  
target="\_top">Absolute, young problems play. BABABABANGINSE Strange,  
</A></TD></TR><TABLE><P ALIGN="CENTER"><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?b&ADD\_FLAG=N"  
target="\_top"><IMG SRC="http://tpcwww.tpcw.net/tpcw/gifs/cart.gif"  
ALT="Shopping Cart" BORDER="0" WIDTH="120" HEIGHT="30"></A><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?h" target="\_top"><IMG  
SRC="http://tpcwww.tpcw.net/tpcw/gifs/search.gif" ALT="Search"  
BORDER="0" WIDTH="120" HEIGHT="30"></A><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a" target="\_top"><IMG  
SRC="http://tpcwww.tpcw.net/tpcw/gifs/home.gif" ALT="Home" BORDER="0"  
WIDTH="120" HEIGHT="30"></A></P></BODY></HTML><TABLE BORDER="0"  
CELLPADDING="0" CELLSPACING="0" ALIGN="CENTER"><TR><TD><I> Timestamp  
2001-12-07 14:50:58.370 </TD></TR></TABLE>

#### New Products page

<HTML><HEAD><TITLE>New CHILDREN</TITLE>  
<META content="text/html; charset=windows-1252" http-equiv=Content-  
Type>  
<FRAMESET border=0 frameBorder=0 frameSpacing=0 rows=69%,31%>  
<FRAME SRC="http://tpcwww.tpcw.net/tpcw/tpcw.dll?r&l=j&s=d"  
border=0 marginHeight=0 marginWidth=0 scrolling=no>  
<FRAME SRC="http://search.tpcw.net/tpcw/list.dll?nd"  
border=0 marginHeight=0 marginWidth=0>  
</FRAMESET></HTML>

#### New Products frame 1

<BODY BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER">TPC Web Commerce Benchmark  
(TPC-W)</H1><P ALIGN="CENTER"><IMG SRC="gifs/tpclogo.gif" WIDTH="288"  
HEIGHT="67" ALIGN="BOTTOM" BORDER="0"></P><H2 ALIGN="CENTER">New  
Product Page - Subject: CHILDREN</H2><TABLE BORDER="0" WIDTH="660"  
ALIGN="CENTER"><TR ALIGN="CENTER" VALIGN="TOP"><TD COLSPAN="5"><B><FONT  
SIZE="+ 1" color="#ff0000"> Click on one of our latest books to find  
out more!</FONT></B></TD></TR><TR ALIGN="CENTER" VALIGN="TOP"><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=1342" target="\_top"><IMG  
SRC="http://image.tpcw.net/tpcw/thumb/034/03367.jpg" ALT="Book 1"  
WIDTH="100" HEIGHT="150"></A></TD><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=2924" target="\_top"><IMG  
SRC="http://image.tpcw.net/tpcw/thumb/075/07462.jpg" ALT="Book 2"  
WIDTH="100" HEIGHT="150"></A></TD><TD><A

HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=3692" target="\_top"><IMG SRC="http://image.tpcw.net/tpcw/thumb/037/03692.jpg" ALT="Book 3" WIDTH="100" HEIGHT="150"></A></TD><TD><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=5816" target="\_top"><IMG SRC="http://image.tpcw.net/tpcw/thumb/059/05816.jpg" ALT="Book 4" WIDTH="100" HEIGHT="150"></A></TD><TD><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=8831" target="\_top"><IMG SRC="http://image.tpcw.net/tpcw/thumb/089/08831.jpg" ALT="Book 5" WIDTH="100" HEIGHT="150"></A></TD></TABLE>

## New Products frame 2

<TABLE BORDER="1" CELLPADDING="1" CELLSPACING="1" ALIGN="CENTER"><TR><TD><FONT SIZE="+1">Author</FONT></TD><TD><FONT SIZE="+1"> Title</FONT></TD></TR><TR><TD><I>8S\*.ci5 BABABABAREINUL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4601" target="\_top">Eyes may make on the necessary, BABABABAREALUL powerful </A></TD></TR><TR><TD><I>WGET=,PfFJI8@f BABABABAREINBAQeo </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8594" target="\_top">International, BABABABAOGULBA annual cases </A></TD></TR><TR><TD><I>-&e:w BABABABAOGOGAL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=9275" target="\_top">BABABABAINOGOG Present, direct prices gain. Grey, legal </A></TD></TR><TR><TD><I>kM1DCZ].3o[,L) 2T0dt BABABABAATNGIN </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=9537" target="\_top">Hands may choose to the BABABABASEULRI years. Various, </A></TD></TR><TR><TD><I>(yK j BABABABAALRINGaFDh5 </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=9601" target="\_top">BABABABABANGUL Students should see. British, full </A></TD></TR><TR><TD><I>&twZ-2eo}@5 BABABABAALALOG </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2290" target="\_top">Extensive, BABABABAREINBAsmall </A></TD></TR><TR><TD><I>Y.d+M/~ BABABABAULATRE </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4528" target="\_top">Clients without the BABABABAALSERE real, important reports </A></TD></TR><TR><TD><I>2:~FGb.)d BABABABAATSEOG </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1340" target="\_top">Likely, BABABAOGRIREBA small hundreds ought to take. </A></TD></TR><TR><TD><I>CnKoxaY BABABABAATSEOG </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4592" target="\_top">BABABABAINOGRE Excellent kids would show. Affairs could pay.</A></TD></TR><TR><TD><I>C o BABABABAALNGAL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3048" target="\_top">BABABABANGREIN Red, busy </A></TD></TR><TR><TD><I>8@; BABABABARERENG </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4414" target="\_top">Areas BABABAOGNLURE get. Original, other </A></TD></TR><TR><TD><I>yC9m).~M7=m9-^ BABABABAULNGIN </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8481" target="\_top">Alone, french items see perfect, BABABAOGINULOG true </A></TD></TR><TR><TD><I>LsT BABABABABANGBA </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3212" target="\_top">Adequate ideas in the BABABABANGOGREcomplex, elderly dogs do</A></TD></TR><TR><TD><I>BqgxNYg-+@xvCDZ BABABABAREATUL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=467" target="\_top">BABABABAREATUL Economic, </A></TD></TR><TR><TD><I>xJC.Y055,~:z BABABABANGREBAv1 </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5047" target="\_top">BABABABASEULRE Standards like </A></TD></TR><TR><TD><I>\*(!+y)+}p\_-6gXe BABABABARISENGtdH9 </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=9405" target="\_top">BABABABAOGNGAL Vehicles might </A></TD></TR><TR><TD><I>#?@Kx6(Lly~Eo3(,^ BABABABAATRIIN </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2417" target="\_top">Systems shall have to exercise. Old, strong </A></TD></TR><TR><TD><I>W4~7Pji\*)3gH5E+It BABABABARIINUL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5664" target="\_top">BABABABAOGGRIASpects would find. Heavy problems say. </A></TD></TR><TR><TD><I>5K,C+(FkoDjizl{Gx BABABABAATNGSE </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7811" target="\_top">Essential BABABABAINSEAT newspapers </A></TD></TR><TR><TD><I>560 BABABABAULULBA </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=9443" target="\_top">Existing, empty methods do BABABABAALREIN construct </A></TD></TR><TR><TD><I>034=1T0|V5 IE BABABABAALRIAT </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5357" target="\_top">Blue meetings shall keep. Usual hours may BABABABARERIBA </A></TD></TR><TR><TD><I>+VK7Cv7KIKD7/ BABABABAULINNGYY </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6413" target="\_top">BABABABAINULOGHouses do measure such as the </A></TD></TR><TR><TD><I>bNB4HLNr?rI| BABABABARIINAL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5406" target="\_top">BABABABAALSENG Major, clear plans may make. Strange, normal </A></TD></TR><TR><TD><I>dc70C\_([ BABABABAREBAAT </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5225" target="\_top">Steps enter in BABABAOGRIBAOGthe sexual, </A></TD></TR><TR><TD><I>9M)txU BABABABABABAUl0fCU4y</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2638" target="\_top">Original figures find to BABABABAINININthe </A></TD></TR><TR><TD><I>[tngmY3Q;6.,YxJ|@f BABABABABANGUL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2164" target="\_top">BABABABAREBAOnly, long-term applications would require </A></TD></TR><TR><TD><I>bXp-S BABABABAATBARE9MUE7 </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6627" target="\_top">BABABABABABOGBA Double, single eyes use. </A></TD></TR><TR><TD><I>[VW~71lk0S\$y BABABABAREOGAT </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4210" target="\_top">BABABABAOGBAOGAL Cold, individual years play. Quiet, soviet </A></TD></TR><TR><TD><I>6GIOY+MnJu BABABABAOGATREQ </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3661" target="\_top">BABABABARIATINBritish, democratic months ought to avoid. </A></TD></TR><TR><TD><I>TV8%8%^xTDLbf@ ,X\*0o BABABABAREATIN </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5176" target="\_top">BABABABAREATSE Parties can know. Significant </A></TD></TR><TR><TD><I>4{7WIhpLp~| BABABABARIRIRIC </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6024" target="\_top">BABABABAINOGIN Real, old

</A></TD></TR><TR><TD><I>kHy5?A BABABABANGATOGFF  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8647"  
target="\_top">Active, good decisions may tell BABABAOGNBAGmeasures.  
</A></TD></TR><TR><TD><I>HJ2)rqEs4# uvZfw5W1Y BABABABANGNGOG  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=991"  
target="\_top">BABABABANGNGOG International, dark  
</A></TD></TR><TR><TD><I>nHJfPl]4 BABABABAOGOGRI  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7547"  
target="\_top">BABABABAOGULULOffices with the  
</A></TD></TR><TR><TD><I>{, #mlB}Z BABABABAULATSE  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2901"  
target="\_top">Slight, legal BABABABABASEBA systems  
</A></TD></TR><TR><TD><I>L}S[4e BABABABARIULRI  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=9067"  
target="\_top">Fair, large BABABABABAATUL prices  
</A></TD></TR><TR><TD><I>, !Dd}G\_/i, &tR}  
BABABABABAALREEy0J07</I></TD><TD><A  
HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4729"  
target="\_top">Students shall rescue good BABABABARIULsystems.  
</A></TD></TR><TR><TD><I>|4:JPdNvkheaMlBu BABABABAINNGIN  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3415"  
target="\_top">Clear, BABABAOGREINUL overseas  
</A></TD></TR><TR><TD><I>1AQ~|v\*sz|J BABABABAINNGAL  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=9271"  
target="\_top">Managers with the BABABABARINGUL councils remember at the  
</A></TD></TR><TR><TD><I>{n\$CZ\$R&l]n!Na\$C7& BABABABANGONGWnX  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2393"  
target="\_top">BABABABAULRIAT Courts under the buildings avoid  
additional </A></TD></TR><TR><TD><I>9S151oAC(D  
BABABABAATNGAT </I></TD><TD><A  
HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3778"  
target="\_top">Supreme, comprehensive BABABAOGULNGOG businesses for  
</A></TD></TR><TR><TD><I>Om2z(1C]Xy{= % BABABABARERIRI  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1512"  
target="\_top">BABABAOGSEOGALThings on the  
</A></TD></TR><TR><TD><I>\*\* { BABABABABANGAL  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=92"  
target="\_top">Literary, old values BABABABABANGAL increase. Common  
</A></TD></TR><TR><TD><I>5o!FEx BABABABAOGRENG  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3775"  
target="\_top">BABABAOGNNGAL Constant, big  
</A></TD></TR><TR><TD><I>++WScLMCi BABABABABAOGSEHjk  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=15"  
target="\_top">Main, BABABABABAOGSE historical shops shall  
</A></TD></TR><TR><TD><I>Dcf{|^ BABABABAREOGSE9ZrX  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=415"  
target="\_top">BABABABAREOGSELong, personal  
</A></TD></TR><TR><TD><I>6BDa\* BABABABAULNGBA  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3745"  
target="\_top">Administrative, major units prove. BABABAOGRIRIIN  
</A></TD></TR><TR><TD><I>R+Q5 BABABABAOGULSE  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5276"  
target="\_top">Rich, BABABAOGINBAATdifferent  
</A></TD></TR><TR><TD><I>uK\*8liEaghT=9j BABABABASEOGSE  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1532"

target="\_top">Results BABABAOGSERIAL might take.  
</A></TD></TR><TR><TD><I>+v6u n~Z89ap BABABABABAATNGq  
</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4441"  
target="\_top">Individuals draw BABABABASEBAIN available,  
</A></TD></TR></TABLE><P ALIGN="CENTER"><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?b&ADD\_FLAG=N"  
target="\_top"><IMG  
SRC="http://tpcwww.tpcw.net/tpcw/gifs/cart.gif"ALT="Shopping Cart"  
BORDER="0" WIDTH="120" HEIGHT="30"></A><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?h" target="\_top"><IMG  
SRC="http://tpcwww.tpcw.net/tpcw/gifs/search.gif"ALT="Search"  
BORDER="0" WIDTH="120" HEIGHT="30"></A><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a" target="\_top"><IMG  
SRC="http://tpcwww.tpcw.net/tpcw/gifs/home.gif"ALT="Home" BORDER="0"  
WIDTH="120" HEIGHT="30"></A><BR><TABLE BORDER="0" CELLPADDING="0"  
CELLSPACING="0" ALIGN="CENTER"><TR><TD><I> Timestamp 2001-12-07  
14:47:08.372 </TD></TR></TABLE>

#### Best Sellers page

<HTML><HEAD><TITLE>Best HISTORY</TITLE>  
<META content="text/html; charset=windows-1252" http-equiv=Content-  
Type>  
<FRAMESET border=0 frameBorder=0 frameSpacing=0 rows=69%,31%>  
<FRAME SRC="http://tpcwww.tpcw.net/tpcw/tpcw.dll?r&l=k&s=h"  
border=0 marginHeight=0 marginWidth=0 scrolling=no>  
<FRAME SRC="http://search.tpcw.net/tpcw/list.dll?bh"  
border=0 marginHeight=0 marginWidth=0>  
</FRAMESET></HTML>

#### Best Sellers frame 1

<BODY BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER">TPC Web Commerce Benchmark  
(TPC-W)</H1><P ALIGN="CENTER"><IMG SRC="gifs/tpclogo.gif" WIDTH="288"  
HEIGHT="67" ALIGN="BOTTOM" BORDER="0"><P><H2 ALIGN="CENTER">Best  
Sellers Page - Subject: HISTORY</H2><TABLE BORDER="0" WIDTH="660"  
ALIGN="CENTER"><TR ALIGN="CENTER" VALIGN="TOP"><TD COLSPAN="5"><B><FONT  
SIZE="+ 1" color="#ff0000"> Click on one of our latest books to find  
out more!</FONT></B></TD></TR><TR ALIGN="CENTER" VALIGN="TOP"><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.net/tpcw/thumb/064/06373.jpg" ALT="Book 1"  
WIDTH="100" HEIGHT="150"></A></TD><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=6373" target="\_top"><IMG  
SRC="http://image.tpcw.net/tpcw/thumb/064/06373.jpg" ALT="Book 2"  
WIDTH="100" HEIGHT="150"></A></TD><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=6810" target="\_top"><IMG  
SRC="http://image.tpcw.net/tpcw/thumb/069/06810.jpg" ALT="Book 3"  
WIDTH="100" HEIGHT="150"></A></TD><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=7063" target="\_top"><IMG  
SRC="http://image.tpcw.net/tpcw/thumb/036/03592.jpg" ALT="Book 4"  
WIDTH="100" HEIGHT="150"></A></TD><TD><A  
HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?z=8454" target="\_top"><IMG  
SRC="http://image.tpcw.net/tpcw/thumb/085/08454.jpg" ALT="Book 5"  
WIDTH="100" HEIGHT="150"></A></TD></TABLE>

Best Sellers frame 2

<TABLE BORDER="1" CELLPADDING="1" CELLSPACING="1" ALIGN="CENTER"><TR><TD><FONT SIZE="+1">Author</FONT></TD><TD><FONT SIZE="+1"> Title</FONT></TD></TR><TR><TD><I>~jnd8 BABABABANGULIN </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6617" target="\_top">BABABAOGINNGBA Great, level patients</A></TD></TR><TR><TD><I>\_Z+I&YR09 |bYlp BABABABAREATNG81</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1364" target="\_top">Parts can marry. BABABAOGRIATRE General, red languages may </A></TD></TR><TR><TD><I>vHH\*S)S# BABABABAULREAT</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1397" target="\_top">Records in the unemployed, large BABABAOGRINGULI teeth</A></TD></TR><TR><TD><I>dkVM o0+f/46s BABABABAULINRI</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=783" target="\_top">Political advantages BABABABAULINRI take. Other units meet.</A></TD></TR><TR><TD><I>!d&sIt%T BABABABASESERE</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=554" target="\_top">Police may BABABABASESERE take. Families will have to say</A></TD></TR><TR><TD><I>VO=:qaLusp7gr94qhQy BABABABAOGBAAL</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7441" target="\_top">BABABABARISENG Ultimate leaves may marry on the</A></TD></TR><TR><TD><I>#sen( ;F(@m BABABABANGATOG</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8501" target="\_top">BABABAOGSEALIN Large,</A></TD></TR><TR><TD><I>^bZ?N BABABABASEREALSBU</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=9642" target="\_top">Blue, useful BABABAOGINSENG times</A></TD></TR><TR><TD><I>O8c\_8|Q:%6U gqGrNUJ BABABABAATNGNG</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1555" target="\_top">BABABAOGSESESE Schemes might carry. Industrial, extra</A></TD></TR><TR><TD><I>4&N+90Kc.2J(Z BABABABABANGAL</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7813" target="\_top">Rural, BABABAOGULBAIN british students</A></TD></TR><TR><TD><I>6F,8YZy&5QzY}Xf0 BABABABAOGALAL</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2136" target="\_top">High, sure communications BABABABABAALNG stay social, large </A></TD></TR><TR><TD><I>G[+Q]JazVE8N? BABABABAALBIOG</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5618" target="\_top">BABABAOGALNGUL Global, statistical days may mean in the</A></TD></TR><TR><TD><I>2\$ n.a\_ a%8 BABABABARIULBATHmX4</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5634" target="\_top">Servants to the heads BABABAOGATULSES shall recruit</A></TD></TR><TR><TD><I>034=1T0|V5 IE BABABABAALRIAT</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7616" target="\_top">Free, natural models BABABABARISEOG help in</A></TD></TR><TR><TD><I>L+u/( :RO}%I3PYDU BABABABABAOGAL66OI3F</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8490" target="\_top">Results BABABABAINOGOG in the right</A></TD></TR><TR><TD><I>\_wbw%SBUW BABABABARINGRI</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8768" target="\_top">BABABAOGREALIN Differences with the</A></TD></TR><TR><TD><I>46eCH k BABABABAINULSE</I></TD><TD></TR></TABLE>

</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4330" target="\_top">BABABABANGULOG Joint, recent</A></TD></TR><TR><TD><I>fYu\*Vb) BABABABAINSESE</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5206" target="\_top">Alone, wide arguments BABABAOGINGORE by the</A></TD></TR><TR><TD><I>+\_O BABABABABAOGALXEbH</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8753" target="\_top">Able, BABABABAINSERI empty resources</A></TD></TR><TR><TD><I>e?I5WmI|\_ BABABABARIRESEJq064R</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6593" target="\_top">Clean, total pp. BABABAOGBASEOG like the different, prime</A></TD></TR><TR><TD><I>hE\$/i&73M}Uc%DkvquO BABABABAATBARE</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8911" target="\_top">Skills shall BABABAOGBAOGRE have to introduce for</A></TD></TR><TR><TD><I>aGtGbAaac- BABABABAREULOG</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=9400" target="\_top">Adequate, other legs BABABAOGSEULIN show in the</A></TD></TR><TR><TD><I>~%hHB{1}X9/\$\*,O BABABABABAREIN</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1341" target="\_top">BABABAOGRIREOG Small, effective classes could marry to the</A></TD></TR><TR><TD><I>g.# n BABABABABAATNG</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1478" target="\_top">Urgent, local values about the BABABAOGREULIN matters can</A></TD></TR><TR><TD><I>&pzUAM\*?hu~)^tbVF}vH BABABABAATULRE</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=1958" target="\_top">Dead BABABAOGNGSEIN patients might approve.</A></TD></TR><TR><TD><I>LE%0ul}&eA-}d BABABABARIRIBA5</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2119" target="\_top">Adults can marry. Parliamentary courses BABABAOGINATBA could</A></TD></TR><TR><TD><I>{.} BABABABABARIAL</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3217" target="\_top">BABABABAOGNGOG More small authorities</A></TD></TR><TR><TD><I>-D: , BABABABAINATRI</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4514" target="\_top">Windows do BABABAOGATATUL try. Red</A></TD></TR><TR><TD><I>2z+9fIP;Zt(=I\_]y BABABABARIIOGBA</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2934" target="\_top">Human, BABABABAALRIIN available companies</A></TD></TR><TR><TD><I>L0Tl;qGco%tQ8yW2j}g BABABABAATBAAL</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4957" target="\_top">Prices might make from BABABABAULBAAT the years.</A></TD></TR><TR><TD><I>d7o0muX BABABABARIRIAL</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7093" target="\_top">BABABABAULBABAContemporary,</A></TD></TR><TR><TD><I>ZfI2?arOjWj BABABABAULSEBA6ZSg4</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3148" target="\_top">Level, sure conditions might BABABAOGULNGSEhelp.</A></TD></TR><TR><TD><I>K0ge^w BABABABAREBAOG</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3541" target="\_top">BABABAOGOGALRI Products last by the great comments.</A></TD></TR><TR><TD><I>~jnd8 BABABABANGULIN</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4671" target="\_top">Western, final levels pay to the BABABAOGOGATRE local,</A></TD></TR><TR><TD><I>jk | BABABABAULNGREhJw</I></TD><TD></TR></TABLE>

</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7031" target="\_top">Services would BABABABAOGSENG go windows-- Conventional, </A></TD></TR><TR><TD><I>dbaGB%lf{lq=nZU BABABABAREINRE </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=484" target="\_top">Young, recent participants BABABABAREINRE will have to plan </A></TD></TR><TR><TD><I>4V77.h:sHj; BABABABARESEBA </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=450" target="\_top">Then BABABABARESEBA present origins on the other individuals </A></TD></TR><TR><TD><I>7q!f^0(&SARwLJG8}I BABABABANGULUL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=2084" target="\_top">Main, BABABABAOGREBASE entire industries to the outer words </A></TD></TR><TR><TD><I>HcGmgS&p9}f3 BABABABASERIBA </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=5926" target="\_top">BABABABAUARIORG Reports should enjoy enough. Grey classes </A></TD></TR><TR><TD><I>PLL0BLIo/DiX BABABABABAOGRI </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6687" target="\_top">Individual, key boys do BABABABASEOGSEdecide </A></TD></TR><TR><TD><I>J\*Z BABABABABAOGULOGq6I </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=6926" target="\_top">Years study BABABABAOGALATAT then. </A></TD></TR><TR><TD><I>jk[D~wS)S^5 BABABABABAATBAAL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=7515" target="\_top">BABABABAUULALPositive, able kinds deliver in </A></TD></TR><TR><TD><I>!/.w^=^l[=HSmT}Wd # BABABABABAATRIN </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8352" target="\_top">Present, BABABABAOGATINNGtypical areas need. </A></TD></TR><TR><TD><I>!z%CSzb8wPA BABABABABANGOGAL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=8526" target="\_top">Necessary cases BABABABABAOGSESEcould have </A></TD></TR><TR><TD><I>VXg{I#k ZPvVe BABABABABAOGOGOG </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4705" target="\_top">Final, BABABABAOGATALSE right </A></TD></TR><TR><TD><I>#?Kx6(Lly~Eo3(,;^ BABABABABAATRIIN </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=4606" target="\_top">BABABABABAOGREAT Revolutionary, public shares </A></TD></TR><TR><TD><I>Co0kq-\*?0 BABABABABAALOGUL </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3800" target="\_top">Services could expect in the BABABABABAOGINBA boys! Companies </A></TD></TR><TR><TD><I>.6k6 BABABABABAUABAREESl033</I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3540" target="\_top">BABABABAOGOGBARE Hence difficult subjects shall have </A></TD></TR><TR><TD><I>IsKla BABABABABAINNGIN </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3531" target="\_top">BABABABABAATATBA Appropriate, royal </A></TD></TR><TR><TD><I>Z3j BABABABABABAATRE </I></TD><TD><A HREF="http://search.tpcw.net/tpcw/tpcw.dll?l=3172" target="\_top">Other BABABABABAUINGNGchildren convert physically from </A></TD></TR></TABLE><P ALIGN="CENTER"><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?b&ADD\_FLAG=N" target="\_top"><IMG SRC="http://tpcwww.tpcw.net/tpcw/gifs/cart.gif"ALT="Shopping Cart" BORDER="0" WIDTH="120" HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?h" target="\_top"><IMG

SRC="http://tpcwww.tpcw.net/tpcw/gifs/search.gif"ALT="Search" BORDER="0" WIDTH="120" HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a" target="\_top"><IMG SRC="http://tpcwww.tpcw.net/tpcw/gifs/home.gif"ALT="Home" BORDER="0" WIDTH="120" HEIGHT="30"></A><BR><TABLE BORDER="0" CELLPADDING="0" CELLSPACING="0" ALIGN="CENTER"><TR><TD><I> Timestamp 2001-12-07 14:49:49.901 </TD></TR></TABLE>

#### Product Detail page

<HTML><HEAD><TITLE>TPC-W Product Detail Page</TITLE></HEAD><BODY BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER"> TPC Web Commerce Benchmark (TPC-W)</H1><CENTER><IMG SRC="http://tpcwww.tpcw.net/tpcw/gifs/tpclogo.gif" ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67"></CENTER><H2 ALIGN="CENTER"> Product Detail Page</H2><H2>Title: BABABABAOGSEULAT Human, alone centres </H2><P>Author: 2\$n.a\_ a%8 BABABABARIULBATHmX4 <BR>Subject: COMPUTERS </P><P><IMG SRC="http://image.tpcw.net/tpcw/images/094/09337.jpg" WIDTH="200" HEIGHT="200" ALIGN="RIGHT" BORDER="0">Description: <I>\*c0Uw.rYBvS-YYSuO{7rWXl;)FsnPn.\$yDkR[C%kxN(Y+zA paHcXes2#hdQg%5YP)Q\*i!k[s+NfSWjJPxLZ&xBbdTG?~qBKpyHyM-Dj4Bp--lhGGQ8/nvJ Ri@oV!Y^tb5c@er!mo\_gY7gL50Gdbb(# p-L:RcnuFw8=H\*\$iCdK8oX\_FvvaAwfxQ;\$q.jm;MzbD/cv- ;{7&ylMRk)0ToZ&#M2!hlX@:HTSfMiT?{||}{0\_/h3!sg\_.iRmRc0rX{\*vtcVMaSVjbxFG #[R=gsf6#4MmAhL)l|]qJg!~R%3+i7HJl-R{?#1H36R}S7s~M?6ss3|Q&}38,!GIG4;IwCO@y5P; ,OYER!dmR9{io+-Z\_(8G,)-rq2(LP?m. x[wa?j-Munmh0;Fs]{\$yvQV\*!+. .KXPwD;K)Pxe (lp&bk?jt\_.G|E)4Ji nfv/Ml6V)A~/D</I></P><BLOCKQUOTE><P><B> Suggested Retail: \$8107.14</B><BR><B>Our Price:</B><font color="#DD0000"><B>\$5107.50</B></FONT><BR><B>You Save:</B><font color="#DD0000"><B>\$2999.64</B></FONT></blockquote><DL><DT><FONT SIZE="2">Backing: AUDIO , 4832 pages<BR>Published by 4fZS;!\_Kp WR:5-L#D@P4lq)Q~%ReCUWX7#NoIUf65D)ttVL&}%+f <BR>Publication date: 6-18-1944<BR>Avail date: 7-5-1944<BR>Dimensions (in inches): 59.73 x 1.76 x 94.56 <BR>ISBN: Ot;bTe9H&6fjq<br></FONT></DT></DL><P ALIGN="CENTER"><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?b&ADD\_FLAG=Y&I\_ID=9337&QTY=1"><IMG SRC="http://tpcwww.tpcw.net/tpcw/gifs/add.gif" ALT=" Add to Basket" BORDER="0" WIDTH="120" HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?h"><IMG SRC="http://tpcwww.tpcw.net/tpcw/gifs/search.gif" ALT=" Search" BORDER="0" WIDTH="120" HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a"><IMG SRC="http://tpcwww.tpcw.net/tpcw/gifs/home.gif" ALT=" Home" BORDER="0" WIDTH="120" HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?m&I\_ID=9337"><IMG SRC="http://tpcwww.tpcw.net/tpcw/gifs/update.gif" ALT=" Update" BORDER="0" WIDTH="120" HEIGHT="30"></A></BODY></HTML>

#### Admin Request page

<HTML><HEAD><TITLE>TPC-W Product Update Page</TITLE></HEAD><BODY BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-W)</H1><H2 ALIGN="CENTER"><IMG SRC="gifs/tpclogo.gif" ALIGN="BOTTOM"

BORDER="0" WIDTH="288" HEIGHT="67"></H2><H2 ALIGN="CENTER"> Admin Request Page</H2><H2>Title: BABABABARIALINMetres can put in the certain, detailed </H2><P ALIGN="LEFT">Author t5Y:4q~ BABABABAINSERELW <BR></P><IMG SRC="http://image.tpcw.net/tpcw/images/034/03373.jpg" BORDER="0" ALIGN="RIGHT" WIDTH="200" HEIGHT="200"><IMG SRC="http://image.tpcw.net/tpcw/thumb/034/03373.jpg" BORDER="0" ALIGN="RIGHT" WIDTH="100" HEIGHT="150"><P><BR><BR></P><FORM ACTION="http://tpcwww.tpcw.net/tpcw/tpcw.dll?" METHOD="GET"><INPUT TYPE="HIDDEN" NAME="n" VALUE=""><INPUT TYPE="HIDDEN" NAME="I\_ID" VALUE="3373"><TABLE BORDER="0"><TR><TD><B>Suggested Retail:</B></TD><TD><B>\$5249.87</B></TD></TR><TR><TD><B>Our Current Price:</B></TD><TD><FONT COLOR="#dd0000"><B>\$3149.92</B></FONT></TD></TR><TR><TD><B> Enter New Price</B></TD><TD ALIGN="RIGHT"><input type="text" NAME="NEWPRICE"></TD></TR><TR><TD><B> Enter New Picture</B></TD><TD ALIGN="RIGHT"><input type="text" NAME="NEWIMAGE"></TD></TR><TR><TD><B> Enter New Thumbnail</B></TD><TD ALIGN="RIGHT"><input type="text" NAME="NEWTUMB"></TD></TR></TABLE><P ALIGN="CENTER"><INPUT TYPE="IMAGE" NAME="Submit" SRC="gifs/changes.gif" BORDER="0" HEIGHT="30" WIDTH="130"><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?h"><IMG SRC="gifs/search.gif" ALT="Search" BORDER="0" WIDTH="120" HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a"><IMG SRC="gifs/home.gif" ALT="Home" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120" HEIGHT="30"></A></P></FORM></BODY></HTML>

HEIGHT="30"></A><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?a"><IMG SRC="gifs/home.gif" ALT="Home" BORDER="0" WIDTH="120" HEIGHT="30"></A></BODY></HTML>

#### Admin Confirm page

<HTML><HEAD><TITLE>TPC-W Admin Response Page</TITLE></HEAD><BODY BGCOLOR="#FFFFFF"><H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-W)</H1><H2 ALIGN="CENTER"><IMG SRC="gifs/tpclogo.gif" ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67"></H2><H2 ALIGN="CENTER"> Admin Confirm Page</H2><H2> Product Updated</H2><H2> Title: BABABABARIALINMetres can put in the certain, detailed </H2><P> Author: t5Y:4q~ BABABABAINSERELW </P>Description: <I>(G&{FG%?h=4~?fj9ZQ%AV)LBp;uh8~l1Q;NaqTr^46}@|eLIqvuT59f@A]UY3k})~!8) -X0Gmbj5Ua%uL}FC#Qz0l\*9opC8@17)-|&3s- {M[i/0.dp9h/EG|rZSK9hOz1hJ?d]Gply0@KvNefe[NwEEoa|1%} 6w@PhjQ[XeeZUXi8w8 aNR.7L7W.{x|M.,F#Ce#F=yGoro(zqa?jFATsGh3m7q6sV^ jDTbkh3QV(P,q\_#cGeZQ1-vDkbgLLZo3t.\_G%hWy(]NB-srA\$9+?[ /9iRh[@9GCltmi@p]\_489W9gRtwZ6B=EO)J/] \*|9;TS.l%DLx#v@4e+f/N=3&cv sFP3{Uv@1\$bI,&0\*mu8%GRO?Q{0nr\_4@5{ot;3yUQir5.E\$@jKQ4b}PW(W%zhUs\_iL6WRs{ </I></P><IMG SRC="http://image.tpcw.net/tpcw/images/002/00157.jpg" BORDER="0" ALIGN="RIGHT" WIDTH="200" HEIGHT="200"><IMG SRC="http://image.tpcw.net/tpcw/thumb/002/00157.jpg" BORDER="0" ALIGN="RIGHT" WIDTH="100" HEIGHT="150"><BLOCKQUOTE><P><B> Suggested Retail: \$5249.87</B><BR><B>Our Price:</B><font color="#DD0000"><B>\$1977.04</B></FONT><BR><B>You Save:</B><font color="#DD0000"><B>\$3272.83</B></FONT></blockquote><DL><DT><FONT SIZE="2">USED , 434 pages<BR>Published by a,Jxrkfx[glNPuuFF%{N+?vM0hbi+CeZO\*8G!\$m9zn-8xLC}fs6 <BR>Publication date: 12-7-2001<BR>Dimensions (in inches): 75.62 x 56.01 x 23.82 <BR>ISBN: ?aE^GL8JgAqcM</FONT></DT></DL><P ALIGN="CENTER"><A HREF="http://tpcwww.tpcw.net/tpcw/tpcw.dll?h"><IMG SRC="gifs/search.gif" ALT="Search" BORDER="0" WIDTH="120"

## Appendix E- Space Calculations

### Clause 4.4 and Clause 4.5.2

WIPS	Initial (kB)	Final (kB)	Delta (kB)	Total Transactions
6622.4	3866517.412	9884627.448	6018110.036	79444328

7-Day DB Log Space	Disk Capacity	# of HDDs
1.07522E+11	34359738368	3.129300777

WIPS	Initial (kB)	Final (kB)	Delta = Final - Initial (kB)	Delta minus S.C. and S.I. Growths (kB)	Total Transactions
6622.4	270993912	272331384	1337472	1064192	79444328

Shopping Cart (S.C.) Growth (kB)	Shopping Item (S.I.) Growth (kB)	8-Hour Storage Space for S.C. Growth (kB)	8-Hour Storage Space for S.I. Growth (kB)
92096	181184	245946.4861	484719.5484

180-Day DB Storage Space w/o S.C. and S.I. Growth (kB)	180-Day DB Storage Space w/ S.C. and S.I. Growth (kB)	Disk Capacity	# of HDDs
730866471.3	731597137.4	17179869184	43.60658749

Note: The 180-Day DB Storage Space value is calculated with the knowledge that old shopping carts and old shopping items are deleted overnight by a SQL stored procedure. Therefore, their growth is calculated over the 8-hour day only and not over the 180-day period. Their 8-hour growth are then added to the other tables' 180-day growth to calculate the 180-day DB storage space.

### Clause 4.5.1 Web Server Access Log

	8-Hour Web Log				# of 18GB HDD	Total Transactions	WIPS
	Initial (kB)	Final (kB)	Delta (kB)	Space (kB)			
WS #1	14401	3282310	3267909	7845397.549	0.45666224	79444328	6622.4
WS #2	1857	3294406	3292549	7904551.765	0.460105469		
WS #3	8705	3310022	3301317	7925601.447	0.461330722		
WS #4	5761	3309894	3304133	7932361.929	0.461724233		
WS #5	13249	3308486	3295237	7911004.952	0.460481094		
WS #6	19009	3299526	3280517	7875666.07	0.4584241		
WS #7	4417	3266118	3261701	7830493.759	0.455794726		
WS #8	7041	3263622	3256581	7818201.974	0.455079249		
WS #9	15041	3318086	3303045	7929749.925	0.461572195		
WS #10	10689	3296902	3286213	7889340.681	0.459220067		
WS #11	17985	3303302	3285317	7887189.619	0.459094859		
WS #12	4097	3332166	3328069	7989826.025	0.465069084		
Image WS#1	18689	7202700	7184011	17246937.5	1.00390389		
Image WS#2	10241	7189516	7179275	17235567.6	1.003242075		
Image WS#3	12737	7155788	7143051	17148603.22	0.998180082		
Image WS#4	4097	7208460	7204363	17295797.35	1.006747907		

**Clause 6.7.1 Uninterrupted Execution**

	79444328 transactions	6622.4 WIPS		(change / 7944328) * WIPS * 3600 * 8 * 14	
	initial free	final free	change	14-day	free after 14 days
Database server					
C	7,049,179,136	7,049,179,136	0	0	7,049,179,136
G	3,396,329,472	3,396,329,472	0	0	3,396,329,472
S	4,143,775,744	4,143,775,744	0	0	4,143,775,744
W	49,794,056,192	49,781,735,424	12,320,768	414,105,326	49,379,950,866
tpcwww1					
C	418,500,608	418,480,128	20,480	688,340	417,812,268
E	1,171,050,496	1,171,050,496	0	0	1,171,050,496
F	12,789,805,056	9,443,446,784	3,346,358,272	112,472,273,186	-99,682,468,130 * See Note1
tpcwww2					
C	1,347,268,608	1,347,244,032	24,576	826,008	1,346,442,600
E	1,594,179,584	1,594,179,584	0	0	1,594,179,584
F	12,802,674,688	9,431,085,056	3,371,589,632	113,320,308,029	-100,517,633,341 * See Note1
tpcwww3					
C	1,345,855,488	1,345,830,912	24,576	826,008	1,345,029,480
E	1,594,179,584	1,594,179,584	0	0	1,594,179,584
F	12,795,662,336	9,415,090,176	3,380,572,160	113,622,213,941	-100,826,551,605 * See Note1
tpcwww4					
C	867,487,744	867,463,168	24,576	826,008	866,661,736
E	1,594,179,584	1,594,179,584	0	0	1,594,179,584
F	12,798,693,376	9,415,237,632	3,383,455,744	113,719,132,208	-100,920,438,832 * See Note1
tpcwww5					
C	1,316,339,712	1,316,315,136	24,576	826,008	1,315,513,704
E	1,594,179,584	1,594,179,584	0	0	1,594,179,584
F	12,790,734,848	9,416,388,608	3,374,346,240	113,412,958,590	-100,622,223,742 * See Note1
tpcwww6					
C	728,842,240	728,817,664	24,576	826,008	728,016,232
E	1,594,179,584	1,594,179,584	0	0	1,594,179,584
F	12,785,127,424	9,425,854,464	3,359,272,960	112,906,340,371	-100,121,212,947 * See Note1
tpcwww7					
C	1,346,560,000	1,346,539,520	20,480	688,340	1,345,871,660
E	1,594,179,584	1,594,179,584	0	0	1,594,179,584
F	12,800,053,248	9,460,047,872	3,340,005,376	112,258,750,127	-99,458,696,879 * See Note1
tpcwww8					
C	551,288,832	551,268,352	20,480	688,340	550,600,492
E	1,594,179,584	1,594,179,584	0	0	1,594,179,584
F	12,797,382,656	9,462,620,160	3,334,762,496	112,082,535,094	-99,285,152,438 * See Note1
tpcwww9					
C	1,312,489,472	1,312,468,992	20,480	688,340	1,311,801,132
E	1,594,179,584	1,594,179,584	0	0	1,594,179,584
F	12,789,207,040	9,406,865,408	3,382,341,632	113,681,686,514	-100,892,479,474 * See Note1
tpcwww10					
C	649,310,208	649,289,728	20,480	688,340	648,621,868
E	1,594,179,584	1,594,179,584	0	0	1,594,179,584
F	12,793,630,720	9,428,525,056	3,365,105,664	113,102,379,595	-100,308,748,875 * See Note1
tpcwww11					
C	1,315,262,464	1,315,241,984	20,480	688,340	1,314,574,124
E	1,594,179,584	1,594,179,584	0	0	1,594,179,584
F	12,783,587,328	9,419,403,264	3,364,184,064	113,071,404,296	-100,287,816,968 * See Note1
tpcwww12					
C	1,316,249,600	1,316,229,120	20,480	688,340	1,315,561,260
E	1,594,179,584	1,594,179,584	0	0	1,594,179,584
F	12,799,561,728	9,391,599,616	3,407,962,112	114,542,799,817	-101,743,238,089 * See Note1
image1					
C	168,385,024	170,941,952	-2,556,928	-85,939,245	254,324,269
E	3,812,573,184	3,812,573,184	0	0	3,812,573,184
F	24,718,442,496	17,361,969,152	7,356,473,344	247,253,644,821	-222,535,202,325 * See Note1
image2					
C	992,156,672	994,361,344	-2,204,672	-74,099,798	1,066,256,470
E	3,812,573,184	3,812,573,184	0	0	3,812,573,184
F	24,727,322,624	17,375,690,752	7,351,631,872	247,090,921,252	-222,363,598,628 * See Note1
image3					
C	173,211,648	173,202,944	8,704	292,544	172,919,104
E	3,812,573,184	3,812,573,184	0	0	3,812,573,184
F	24,724,766,720	17,410,228,224	7,314,538,496	245,844,199,897	-221,119,433,177 * See Note1
image4					
C	183,208,960	183,177,728	31,232	1,049,718	182,159,242
E	3,790,077,952	3,787,194,368	2,883,584	96,918,268	3,693,159,684
F	24,733,614,080	17,356,292,096	7,377,321,984	247,954,374,911	-223,220,760,831 * See Note1

Note1: web server log volumes archived to tape on database server during off-peak hours

## Appendix F- Price Quotes

[Home](#)[Network Cards](#)[Network Cables](#)[Crossover Cables](#)[Print Servers](#)[Barcode Readers](#)[Extension Cables](#)[Miscellaneous](#)[Shipping Estimate](#)[Show Order](#)[Privacy Policy](#)[Info](#)[Search](#)[Index](#)[Y! SHOPPING](#)

# LanAdapters.com



## *PWLA8490 Intel PRO/1000 Server Adapter Gigabit (SX Fiber)*

RETAIL BOX - PWLA8490 Intel® PRO/1000 Gigabit Server Adapter (SX Fiber) LSI chipset Server PCI Adapter SX Multimode Fiber.  
\*LIFETIME WARRANTY\*

**Availability:** Usually ships the same business day.

PWLA8490 **\$435.00**

[Order](#)

From: sales@compuvview-ms.com  
Sent: Friday, December 14, 2001 2:45 PM  
To: kong\_yang@exchange.dell.com  
Subject: Compuview Microsystems, Web-Order# 6718

Dear Kong Yang

Thank you for purchasing from Compuview Microsystems.  
This is a confirmation E-Mail for your order that  
you have placed through our web-site at www.compuview-ms.com

Your order will be processing in 24 hours.

Qty	ItemName	Unit-Price	Ext-Price	Shipping-Cost
5	CCD/E2000-4-64BD-BX	\$1,398.00	\$6,990.00	\$36.25
	Mylex PCI eXtremeRaid 2 Int 4 Ext channel Raid Controller 64Mb e2000-4-64bd New factory warranty			

Item(s)SubTotal \$6,990.00  
UPS 3 Day Select \$36.25  
Total \$7,026.25

Payment: Visa xxxx-xxxx-xxxx-  
\*\*\*\*\*

Customer Request:  
this is the quote for customer only,  
\*\*\*\*\*

Thank you for your business!

Compuview Sales Team  
Sales@compuvview-ms.com  
(877)846-5454

-----  
This is an automatically generated Delivery Status Notification.

Microsoft Corporation  
One Microsoft Way  
Redmond, WA 98052-6399

Tel 425 882 8080  
Fax 425 936 7329  
<http://www.microsoft.com/>

**Microsoft**

December 13, 2001

Dell Computer  
Corporation  
Kong Yang  
200 One Dell Way  
Round Rock 5  
Round Rock, TX 78682

Kong:

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-W benchmark testing.

All pricing shown is in US Dollars (\$).

Part Number	Description	Unit Price	Quantity	Price
228-00968	<b>SQL Server 2000 Standard Edition</b> <i>Per processor licensing</i>	\$ 4,271	4	\$ 17,084
C10-00475	<b>Windows 2000 Advanced Server</b> <i>Server license only - No CALs</i> <i>Discount schedule: Open Program - No Level</i>	\$ 2,399	1	\$ 2,399
048-00317	<b>Visual C++ Professional 6.0 Win32</b>	\$ 549	1	\$ 549
	<b>3-year maintenance for above software</b>	\$ 2,095	1	\$ 6,285

All products are currently orderable through Microsoft's normal distribution channels.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or [jamiere@microsoft.com](mailto:jamiere@microsoft.com).

Reference ID: Ppmmu0113125374

Please include this Reference ID in any correspondence regarding this price quote.