



Hewlett-Packard Company

TPC Benchmark™ C
Full Disclosure Report
for
HP ProLiant DL380 G5 5160
using
Microsoft SQL Server 2005 Enterprise x64 Edition SP1
and
Windows Server 2003 Enterprise x64 Edition SP1

**First Edition
Submitted for Review
June 26, 2006**

First Edition –June 2006

Hewlett-Packard Company (HP) believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. HP assumes no responsibility for any errors that may appear in this document. The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, HP provides no warranty of the pricing information in this document.

Benchmark results are highly dependent upon workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC Benchmark C 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. Results obtained in other operating environments may vary significantly. HP does not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute (tpmC) or normalized price/performance (\$/tpmC). No warranty of system performance or price/performance is expressed or implied in this report.

Copyright 2006 Hewlett-Packard Company.

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.

Printed in U.S.A., 2006

HP, NonStop, ProLiant DL380 G5, ProLiant DL360 G4p and ProLiant are registered trademarks of Hewlett-Packard Company.

Microsoft, Windows 2000, Windows Server 2003, Enterprise x64 Edition and SQL Server 2005 Enterprise x64 Edition are registered trademarks of Microsoft Corporation.

Xeon is a registered trademark of Intel.

TPC Benchmark is a trademark of the Transaction Processing Performance Council.

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

Preface

The TPC Benchmark C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark C Standard Specifications Version 5.7.

TPC Benchmark C Overview

The TPC describes this benchmark in Clause 0.1 of the specifications as follows:

TPC Benchmark™ C (TPC-C) is an OLTP workload. It is a mixture of read-only and update intensive transactions that simulate the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with such environments, which are characterized by:

- The simultaneous execution of multiple transaction types that span a breadth of complexity
- On-line and deferred transaction execution modes
- Multiple on-line terminal sessions
- Moderate system and application execution time
- Significant disk input/output
- Transaction integrity (ACID properties)
- Non-uniform distribution of data access through primary and secondary keys
- Databases consisting of many tables with a wide variety of sizes, attributes, and relationships
- Contention on data access and update

The performance metric reported by TPC-C is a "business throughput" measuring the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint. The performance metric for this benchmark is expressed in transactions-per-minute-C (tpmC). To be compliant with the TPC-C standard, all references to tpmC results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

Although these specifications express implementation in terms of a relational data model with conventional locking scheme, the database may be implemented using any commercially available database management system (DBMS), database server, file system, or other data repository that provides a functionally equivalent implementation. The terms "table", "row", and "column" are used in this document only as examples of logical data structures.

TPC-C uses terminology and metrics that are similar to other benchmarks, originated by the TPC or others. Such similarity in terminology does not in any way imply that TPC-C results are comparable to other benchmarks. The only benchmark results comparable to TPC-C are other TPC-C results conformant with the same revision.

Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

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

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted on the HP ProLiant DL380 G5. The operating system used for the benchmark was Windows Server 2003, Enterprise x64 Edition SP1. The DBMS used was Microsoft SQL Server 2005 Enterprise x64 Edition SP1.

TPC Benchmark C Metrics

The standard TPC Benchmark C metrics, tpmC (transactions per minute), price per tpmC (three year capital cost per measured tpmC), and the availability date are reported as:

140,264 tpmC

USD \$3.56per tpmC

The availability date is November 22, 2006.

Standard and Executive Summary Statements

The following pages contain executive summary of results for this benchmark.

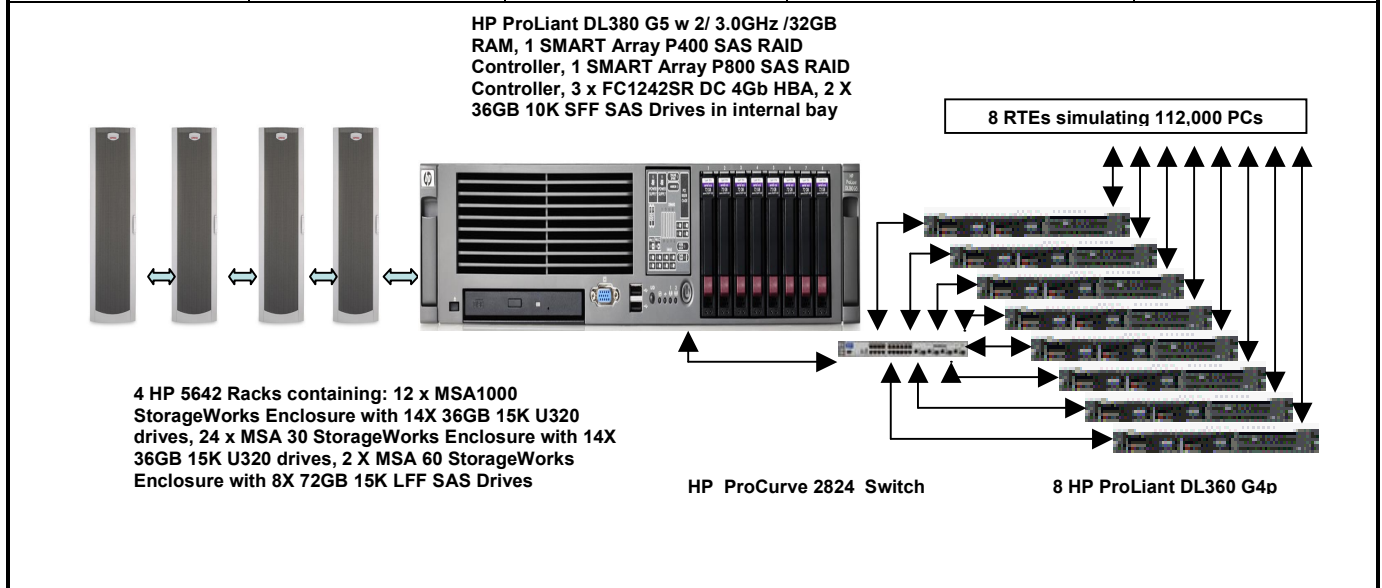
Auditor

The benchmark configuration, environment and methodology were audited by Lorna Livingtree of Performance Metrics, Inc. to verify compliance with the relevant TPC specifications.

Hewlett-Packard Company	HP ProLiant DL380 G5 5160 3.0 GHz/4MB Dual Core	TPC-C Rev. 5.7
	C/S with 8 HP ProLiant DL360 G4p	Report Date: June 26, 2006

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
USD \$499,207	140,264	USD \$3.56	Nov. 22, 2006

Database Server Processors /Cores/Threads	Database Manager	Operating System	Other Software	Number of Users
2/4/4 Intel 5160 3.0 GHz DC	Microsoft SQL Server 2005 Enterprise x64 Edition SP1	Windows Server 2003 Enterprise x64 Edition SP1	Microsoft Visual C++ Microsoft COM+	112,000



System Components	Server		Each Client	
	Quantity	Description	Quantity	Description
Processors/Cores/Threads	2/4/4	3.0 GHz DC Intel 5160 w/ 4M Cache	2/2/4	3.6 GHz Intel Xeon w/ 2MB cache
Memory	4	8 GB DDR (2 X 4 GB)	1	1024 MB
Disk Controllers	1 1 3	Smart P400 Controller Smart P800 Controller FC1242SR DC 4Gb HBA	1	Integrated Smart Array 6i Controller
Disk Drives	16 504 2	72GB 15K LFF SAS Drives (log) 36 GB 15K U320 Drives (data) 36 GB 10K SFF SAS Drives (internal, OS)	2	36 GB SCSI Drive
Total Storage		18,184.40 GB		72 GB

Hewlett-Packard	HP ProLiant DL380 G5 5160		TPC-C Rev. 5.7			
Company	3.0GHz/4MB DC Client/Server		Report Date	26-Jun-06		
Description	Part Number	Third Party	Unit Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware						
		Brand Pricing				
HP DL380G5 5160 2G Svr (p400/Dual Integrated Gigabit NIC)	417458-001	1	4,149	1	4,149	
HP X5160 DL380G5 Kit (3.0GHz/4MB dual-core processor option kit)	418324-B21	1	1,499	1	1,499	
8GB FBD PC2-5300 2 x 4GB Kit	397415-B21	1	10,499	4	41,996	
HP Smart Array P800/512MB SAS Controller	381513-B21	1*	1,299	1	1,299	
HP StorageWorks MSA-60 Storage	418408-B21	1*	3,250	2	6,500	
HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA	AE312A	1	2,200	3	6,600	
HP StorageWorks4gbSW SFP Transceiver	A7446B	1	199	18	3,582	
HP StorageWorks SAN Switch 4/32	A7393A	1	32,490	1	32,490	
Modular SAN Array 1000 (incl. 2 Gb SFP SW Transceiver Kit)	201723-B22	1	6,995	12	83,940	
HP StorageWorks MSA 30 SB Storage	302969-B21	1	2,829	24	67,896	
HP s7540 17in. CRT Monitor	PF997AA#ABA	1	139	1	139	
HP PS/2 Scroll Mouse carbonite	DG169AV	1	5	1	5	
HP Enhanced Keyboard	DG170AV#ABA	1	10	1	10	
HP 5642 Pallet Unassembled Rack	358254-B21	1	689	4	2,756	
UPS R1500 XR Low Voltage US	204404-001	1	866	1	866	
36GB 15Krpm U320 UNI HDD	286776-B22	1	269	504	135,576	
36GB 15Krpm U320 UNI HDD (10% spares)	286776-B22	1	269	51		13,719
HP 72GB 15K SAS 3.5 Hot Plug Hard Drive	375870-B21	1	399	16	6,384	
HP 72GB 15K SAS 3.5 Hot Plug Hard Drive (10% Spares)	375870-B21	1	399	2		798
HP 36GB 10K SAS 2.5 Hot Plug Hard Drive	375859-B21	1	299	2	598	
HP StorageWorks MSA 30 SB Storage (10% spares)	302969-B21	1	2,829	3		8,487
HP StorageWorks MSA-60 Storage (10% Spares)	418408-B21	1*	3,250	2		6,500
Modular SAN Array 1000 (incl. 2 Gb SFP SW Transceiver Kit) (10% spares)	201723-B22	1	6,995	2		13,990
HP StorageWorks4gbSW SFP Transceiver (10% spares)	A7446B	1	199	2		398
Storage Works LC/LC 15m Fibre Cable	221692-B23	1	103	18	1,854	
Storage Works LC/LC 15m Fibre Cable (10% spares)	221692-B23	1	103	2		206
HP 3y 4h 24x7 ProLiant DL38x HW Support ,ProLiant Server DL38x	U4545E	1	949	1		949
HP CP 3y 4h 24x7 HW B series 32 ports	UA260E	1	5,760	1		5,760
				Subtotal	398,139	50,807
Server Software						
Microsoft SQL Server 2005 Enterprise X64 Edition(per processor)	810-03150	Microsoft	23,911	2	47,822	Incl Below
Microsoft Visual C++ Standard	254-00170	Microsoft	109	1	109	Incl Below
Microsoft Windows 2003 Server, Enterprise Edition X64	P72-00274	Microsoft	2,334	1	2,334	Incl Below
Microsoft Problem Resolution Services		Microsoft	245	1		245
				Subtotal	50,265	245
Client Hardware						
HP DL360G4p X3.6GHz/2MB/1GB SCSI US Svr	376236-001	1	2,799	8	22,392	
Dual Integrated Gigabit NIC, Integrated Smart Array Controller 6i						
Intel Xeon 3.6GHz 2MB DL360G4 Processor	376242-B21	1	999	8	7,992	
36GB 15K U320 Pluggable Hard Drive	286776-B22	1	269	16	4,304	
HP CP 3Y 4H 24x7 HW Entry300 4-Hour 24 Hour x 7 Day Coverage 3 Years	162675-002	1	599	8		4,792
				Subtotal	34,688	4,792
Client Software						
Windows Server 2000, Standard Edition	C11-00821	Microsoft	738	8	5,904	Incl. Above
				Subtotal	5,904	0
User Connectivity						
HP ProCurve Switch 2824	J4903A#ABA	1	2499	1	2,499	
HP CP for HP ProCurve Networking products 3 Yr 4 hr/24x7	U2856E	1	1000	1		1,000
10 foot Cat5E Non Booted Network Patch Cables (plus 10% spares)	cb1c5ENB10	3	3	20	60	
				Subtotal	2,559	1,000
Large Purchase and Net 30 discount (See Note 1)	10.0%	1			(\$43,533)	(\$5,660)
				Total	\$448,022	\$51,184
Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark pricing specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.				Three-Year Cost of Ownership: USD \$499,207		
				tpmC Rating: 140,264		
				\$ / tpmC: USD \$3.56		
Pricing: 1=HP Direct 800-203-6748 2= Microsoft 3= LanAdapters.com						
Note 1 = Discount based on HP Direct guidance applies to all lines where pricing = 1						
* = These components are not immediately orderable. See the FDR for more information.						
Note 2 = The benchmark results were audited by Lorna Livingtree of Performance Metrics						

Numerical Quantities Summary

MQTH, Computed Maximum Qualified Throughput

140,264 tpmC

Response Times (in seconds)	Average	90%	Maximum
New-Order	0.32	0.53	9.08
Payment	0.29	0.50	9.77
Order-Status	0.30	0.52	8.48
Delivery (interactive portion)	0.10	0.11	0.24
Delivery (deferred portion)	0.18	0.30	4.98
Stock-Level	0.30	0.52	7.32
Menu	0.10	0.11	0.41

Transaction Mix, in percent of total transaction

New-Order	44.93%
Payment	43.03%
Order-Status	4.01%
Delivery	4.01%
Stock-Level	4.02%

Emulation Delay (in seconds)

	Resp.Time	Menu
New-Order	0.10	0.10
Payment	0.10	0.10
Order-Status	0.10	0.10
Delivery (interactive)	0.10	0.10
Stock-Level	0.10	0.10

Keying/Think Times (in seconds)

	Min.	Average	Max.
New-Order	18.02/0.00	18.03/12.12	18.40/121.13
Payment	3.02/0.00	3.03/12.12	3.41/121.13
Order-Status	2.02/0.00	2.03/10.12	2.39/101.03
Delivery (interactive)	2.02/0.00	2.03/5.09	2.23/50.72
Stock-Level	2.02/0.00	2.03/5.09	2.41/50.72

Test Duration

Ramp-up time	20 minutes
Measurement interval	120 minutes
Transactions (all types) completed during measurement interval	38,964,451
Ramp down time	29 minutes

Checkpointing

Number of checkpoints	4
Checkpoint interval	30 minutes

General Items

Test Sponsor

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

This benchmark was sponsored by Hewlett-Packard Company. The benchmark was developed and engineered by Hewlett-Packard Company. Testing took place at HP benchmarking laboratories in Houston, Texas.

Application Code and Definition Statements

The application program (as defined in clause 2.1.7) must be disclosed. This includes, but is not limited to, the code implementing the five transactions and the terminal input output functions.

Appendix A contains all source code implemented in this benchmark.

Parameter Settings

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

- *Database options*
- *Recover/commit options*
- *Consistency locking options*
- *Operating system and application configuration parameters*

This requirement can be satisfied by providing a full list of all parameters.

Appendix C contains the tunable parameters to for the database, the operating system, and the transaction monitor.

Configuration Items

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

The configuration diagram for both the tested and priced systems are included on the following page.

Figure 1. Benchmarked Configuration

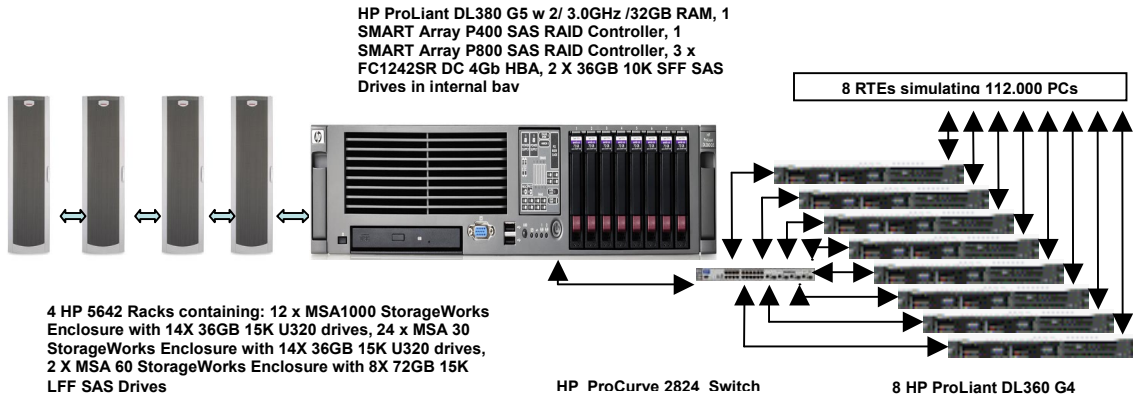
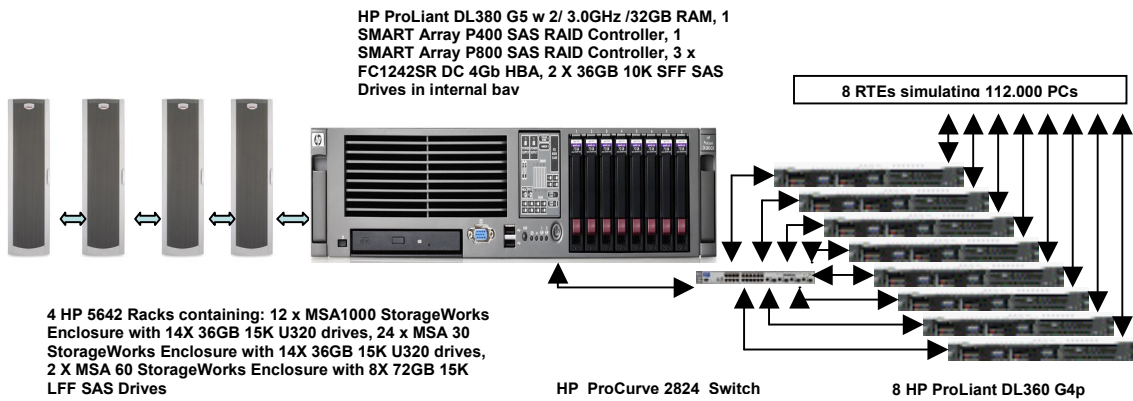


Figure 2. Priced Configuration



Clause 1 Related Items

Table Definitions

Listing must be provided for all table definition statements and all other statements used to set up the database.

Appendix B contains the code used to define and load the database tables.

Physical Organization of Database

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

The tested configuration consisted of 504 drives at 36GB for database data, two 36GB drives for the operating system, and 16 drives at 72GB for database log. There were 12 MSA1000 enclosures connected to two MSA30 enclosures. Two MSA1000 enclosures were attached to one port of a dual channel FC1242SR DC 4Gb HBA by means of the HP StorageWorks SAN Switch 4/32. Each MSA1000 and MSA30 enclosure contained (14) 36GB 15K U320 disk drives used for database data. There were 2 MSA60 enclosures with (8) 72GB LFF SAS disk drives each that were used for the database log and were connected to a P800 Smart Array Controller. The 2 36GB SFF SAS disk drives for the operating system were in the internal drive cage of the DL380 G5, which was connected to a P400 Smart Array Controller.

Benchmarked Configuration:

DL380 G5, Slot 1, P400 SAS Smart Array Controller, Array A

<u>LOGICAL DRIVE C:</u>	<u>Total Capacity = 33.88 GB</u>	<u>RAID 0+1</u>
-------------------------	----------------------------------	-----------------

Microsoft Windows Server 2003, Enterprise x64 Edition SP1

DL380 G5, Slot 2, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 1, Data 1 Array A

<u>LOGICAL DRIVE C:\cs\cs1:</u>	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
MSSQL_cs1		
<u>LOGICAL DRIVE C:\misc\misc1:</u>	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>
MSSQL_misc1		
<u>LOGICAL DRIVE W:</u>	<u>Total Capacity = 663.44 GB</u>	<u>RAID 0+1</u>
tpcback1		

DL380 G5, Slot 2, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 1, Data 2 Array A

<u>LOGICAL DRIVE C:\cs\cs2:</u>	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
MSSQL_cs2		
<u>LOGICAL DRIVE C:\misc\misc2:</u>	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>
MSSQL_misc2		
<u>LOGICAL DRIVE X:</u>	<u>Total Capacity = 663.44 GB</u>	<u>RAID 0+1</u>
tpcback2		

**DL380 G5, Slot 2, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 2, Data 3
Array A**

<u>LOGICAL DRIVE C:\cs\cs3:</u> MSSQL_cs3	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc3:</u> MSSQL_misc3	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>

**DL380 G5, Slot 2, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 2, Data 4
Array A**

<u>LOGICAL DRIVE C:\cs\cs4:</u> MSSQL_cs4	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc4:</u> MSSQL_misc4	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>

DL380 G5, Slot 3, P800 SAS Smart Array Controller, Port 1 & Port 2, Array A

<u>LOGICAL DRIVE E:</u> MSSQL_tpcc_log	<u>Total Capacity = 546.68 GB</u>	<u>RAID 0+1</u>
---	-----------------------------------	-----------------

**DL380 G5, Slot 4, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 1, Data 5
Array A**

<u>LOGICAL DRIVE C:\cs\cs5:</u> MSSQL_cs5	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc5:</u> MSSQL_misc5	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE Y:</u> tpccback3	<u>Total Capacity = 663.44 GB</u>	<u>RAID 0+1</u>

**DL380 G5, Slot 4, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 1, Data 6
Array A**

<u>LOGICAL DRIVE C:\cs\cs6:</u> MSSQL_cs6	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc6:</u> MSSQL_misc6	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE Z:</u> tpccback4	<u>Total Capacity = 663.44 GB</u>	<u>RAID 0+1</u>

**DL380 G5, Slot 4, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 2, Data 7
Array A**

<u>LOGICAL DRIVE C:\cs\cs7:</u> MSSQL_cs7	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc7:</u> MSSQL_misc7	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>

**DL380 G5, Slot 4, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 2, Data 8
Array A**

<u>LOGICAL DRIVE C:\cs\cs8:</u> MSSQL_cs8	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc8:</u> MSSQL_misc8	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>

**DL380 G5, Slot 5, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 1, Data 9
Array A**

<u>LOGICAL DRIVE C:\cs\cs9:</u> MSSQL_cs9	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc9:</u> MSSQL_misc9	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>

**DL380 G5, Slot 5, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 1, Data 10
Array A**

<u>LOGICAL DRIVE C:\cs\cs10:</u> MSSQL_cs10	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc10:</u> MSSQL_misc10	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>

**DL380 G5, Slot 5, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 2, Data 11
Array A**

<u>LOGICAL DRIVE C:\cs\cs11:</u> MSSQL_cs11	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc11:</u> MSSQL_misc11	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>

**DL380 G5, Slot 5, HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA, Port 2, Data 12
Array A**

<u>LOGICAL DRIVE C:\cs\cs12:</u> MSSQL_cs12	<u>Total Capacity = 63.47 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc12:</u> MSSQL_misc12	<u>Total Capacity = 34.17 GB</u>	<u>RAID 0</u>

Priced Configuration vs. Measured Configuration:

The client systems in the tested configuration were DL360G4 systems no longer available. In the priced configuration the client systems were substituted with DL360G4P servers currently available.

Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restrictions in the SUT database implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the minimum key value for these new rows.

All insert and delete functions were fully operational during the entire benchmark.

Partitioning

While there are a few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark, any such partitioning must be disclosed.

No partitioning was used in this benchmark.

Replication, Duplication or Additions

Replication of tables, if used, must be disclosed. Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance.

No replications, duplications or additional attributes were used in this benchmark.

Clause 2 Related Items

Random Number Generation

The method of verification for the random number generation must be described.

In the Benchcraft RTE from Microsoft, each driver engine uses an independent random number sequence. All of the users within a given driver draw from the same sequence.

The Benchcraft RTE computes random integers as described in "Random Numbers Generators: Good Ones Are Hard to Find." Communications of the ACM - October 1988 Volume 31 Number 10.

The seeds for each user were captured and verified by the auditor to be unique. In addition, the contents of the database were systematically searched, and randomly sampled by the auditor for patterns that would indicate the random number generator had affected any kind of a discernible pattern; none was found.

Input/Output Screen Layout

The actual layout of the terminal input/output screens must be disclosed.

All screen layouts followed the specifications exactly.

Priced Terminal Feature Verification

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained. Although not specifically priced, the type and model of the terminals used for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The terminal attributes were verified by the auditor. The auditor manually exercised each specification on a representative HP ProLiant web server.

Presentation Manager or Intelligent Terminal

Any usage of presentation managers or intelligent terminals must be explained.

Application code running on the client machines implemented the TPC-C user interface. No presentation manager software or intelligent terminal features were used. The source code for the forms applications is listed in Appendix A.

Transaction Statistics

Table 2.1 lists the numerical quantities that Clauses 8.1.3.5 to 8.1.3.11 require.

Table 2.1 Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse payments	85.00%
	Remote warehouse payments	15.00%
	Accessed by last name	60.02%
Order Status	Accessed by last name	60.07%
Transaction Mix	New Order	44.93%
	Payment	43.03%
	Order status	4.01%
	Delivery	4.01%
	Stock level	4.02%

Queuing Mechanism

The queuing mechanism used to defer the execution of the Delivery transaction must be disclosed.

Microsoft COM+ on each client machine served as the queuing mechanism to the database. Each delivery request was submitted to Microsoft COM+ asynchronously with control being returned to the client process immediately and the deferred delivery part completing asynchronously.

The source code is listed in Appendix A.

Clause 3 Related Items

Transaction System Properties (ACID)

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7.

All ACID property tests were successful. The executions are described below.

Atomicity

The system under test must guarantee that the database transactions are atomic; the system will either perform all individual operations on the data or will assure that no partially completed operations leave any effects on the data.

Completed Transactions

A row was selected in a script from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was committed and the rows were verified to contain correctly updated balances.

Aborted Transactions

A row was selected in a script from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was rolled back and the rows were verified to contain the original balances.

Consistency

Consistency is the property of the application that requires any execution of a database transaction to take the database from one consistent state to another, assuming that the database is initially in a consistent state.

Consistency conditions one through four were tested using a script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests.

A run was executed under full load lasting over two hours and included a checkpoint.

The script was executed again. The result of the same queries verified that the database remained consistent after the run.

Isolation

Sufficient conditions must be enabled at either the system or application level to ensure the required isolation defined above (clause 3.4.1) is obtained.

Isolation tests one through nine were executed using shell scripts to issue queries to the database. Each script included timestamps to demonstrate the concurrency of operations. The results of the queries were captured to files. The captured files were verified by the auditor to demonstrate the required isolation had been met.

In addition, the phantom tests and the stock level tests were executed and verified.

For Isolation test seven, case A was followed.

Durability

The tested system must guarantee durability: the ability to preserve the effects of committed transaction and insure database consistency after recovery from any one of the failures listed in Clause 3.5.3.

Durable Media Failure

Loss of Data and Log

To demonstrate recovery from a permanent failure of durable medium containing DBMS logs and TPC-C tables, the following steps were executed. This test was executed on a fully scaled database of 1240 warehouses under a load of 12400 users.

- The total number of New Orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
- The RTEs were started.
- The test was allowed to run for a minimum of 10 minutes.
- One log disk was removed from the MSA60 enclosure housing the log drives.
- Since the disk was mirrored, processing was not interrupted. This was verified by checking the user's status on the RTE.
- One of the data disks was removed from one MSA1000 drive cabinet.
- When Microsoft SQL Server recorded errors about not being able to access the database, the RTE was shut down.
- Microsoft SQL Server was shutdown, and the system rebooted after replacing the pulled drives with new drives.
- After the RAID recovery process finished Microsoft SQL Server was started, and a dump of the transaction log was taken.
- The database was restored from backup and the transaction log dump was applied.
- Consistency condition #3 was executed and verified.
- Step 2 was repeated and the difference between the first and second counts was noted.
- An RTE report was generated for the entire run time giving the number of NEW-ORDERS successfully returned to the RTE.
- The counts in steps 12 and 13 were compared and the results verified that all committed transactions had been successfully recovered.
- Samples were taken from the RTE files and used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table.

Instantaneous Interruption and Loss of Memory

Because loss of power erases the contents of memory, the instantaneous interruption and the loss of memory tests were combined into a single test. This test was executed on a fully scaled database of 11200 warehouses under a full load of 112,000 users. The following steps were executed:

- The total number of New Orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
- The RTE was started with 112,000 users.
- The test was allowed to run for a minimum of 10 minutes.
- Pulling the power cords from the SUT induced system crash and loss of memory. No battery backup or Uninterruptible Power Supply (UPS) were used to preserve the contents of memory.
- The RTE was paused then stopped.
- Power was restored and the system restarted.
- Microsoft SQL Server was restarted and performed an automatic recovery.
- Consistency condition #3 was executed and verified.
- Step 1 was repeated and the difference between the first and second counts was noted.
- An RTE report was generated for the entire run time giving the number of NEW-ORDERS successfully returned to the RTE.
- The counts in step 9 and 10 were compared and the results verified that all committed transactions had been successfully recovered.
- Samples were taken from the RTE files and used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table.

Clause 4 Related Items

Initial Cardinality of Tables

The cardinality (e.g. number of rows) of each table, as it existed at the start of the benchmark run, must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted, the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

Table 4.1 Number of Rows for Server

Table	Cardinality as built
Warehouse	12,400
District	124,000
Customer	372,000,000
History	372,000,000
Orders	372,000,000
New Order	111,600,000
Order Line	3,719,992,214
Stock	1,240,000,000
Item	100,000
Deleted Warehouses	0

Database Layout

The distribution of tables and logs across all media must be explicitly depicted for tested and priced systems.

The benchmarked configuration used 6 HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA's for database data storage, 1 P800 Smart Array Controller for database log storage, and 1 P400 Smart Array Controller for the OS and binaries. Each of the 2 ports of the 6 HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA's used for data was connected through the HP StorageWorks SAN Switch 4/32 to a HP StorageWorks MSA1000, which in-turn was connected to 2 MSA30s. There were a total of 12 sets of MSA1000 enclosures with connected MSA30s each enclosure contained (14) 36GB 15K U320 disk drives. Each MSA1000 set had all 42 disk drives configured into a single array with 2 logical disk drives for data. The 2 logical disk drives were used for database storage and configured as RAID 0. Two of the HP StorageWorks FC1242SR Dual Channel 4 Gb PCI-E HBA's also had a RAID 0+1 logical drive used for database backup. The P800 Smart Array Controller was connected to 2 MSA60 enclosures with (8) 72GB LFF SAS drives per enclosure used for the log. This was configured as one RAID 0+1 logical drive. The P400 Smart array controller connected to the internal drive cage of the DL380 G5 which contained (2)

36GB SFF SAS disk drives configured as RAID 0+1 for the operating system and binaries. The Array Accelerators in the MSA 1000's were configured as 100% write cache and were enabled for all RAID 0 volumes. The P800 used for the log had the cache disabled.

Section 1.2 of this report details the distribution of database tables across all disks. The code that creates the file groups and tables is included in Appendix B.

Type of Database

A statement must be provided that describes:

- *The data model implemented by DBMS used (e.g. relational, network, hierarchical).*
- *The database interface (e.g. embedded, call level) and access language (e.g. SQL, DL/I, COBOL read/write used to implement the TPC-C transaction. If more than one interface/access language is used to implement TPC-C, each interface/access language must be described and a list of which interface/access language is used with which transaction type must be disclosed.*

Microsoft SQL Server 2005 Enterprise x64 Edition SP1 is a relational DBMS.

The interface used was Microsoft SQL Server stored procedures accessed with Remote Procedure Calls embedded in C code.

Database Mapping

The mapping of database partitions/replications must be explicitly described.

The database was not replicated.

60 Day Space

Details of the 60-day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables (Order, Order-Line, and History) must be disclosed.

To calculate the space required to sustain the database log for 8 hours of growth at steady state, the following steps were followed:

- The free space on the log file was queried using *dbcc sqlperf(logspace)*.
- Transactions were run against the database with a full load of users.
- The free space was again queried using *dbcc sqlperf(logspace)*.
- The space used was calculated as the difference between the first and second query.
- The number of NEW-ORDERS was verified from the difference in the sum(d_next_o_id) taken from before and after the run.
- The space used was divided by the number of NEW-ORDERS giving a space used per NEW-ORDER transaction.
- The space used per transaction was multiplied by the measured tpmC rate times 480 minutes.

The same methodology was used to compute growth requirements for dynamic tables Order, Order-Line and History.

Details of both the 8-hour transaction log space requirements and the 60-day space requirements are shown in Appendix D.

Clause 5 Related Items

Throughput

Measured tpmC must be reported

Measured tpmC 140,264 tpmC
Price per tpmC USD \$3.56

Response Times

Ninetieth percentile, maximum and average response times must be reported for all transaction types as well as for the menu response time.

Table 5.2: Response Times

Type	Average	90 th %	Maximum
New-Order	0.32	0.53	9.08
Payment	0.29	0.50	9.77
Order-Status	0.30	0.52	8.48
Interactive Delivery	0.10	0.11	0.24
Deferred Delivery	0.18	0.30	4.98
Stock-Level	0.30	0.52	7.32
Menu	0.10	0.11	0.41

Keying and Think Times

The minimum, the average, and the maximum keying and think times must be reported for each transaction type.

Table 5.3: Keying Times

Type	Minimum	Average	Maximum
New-Order	18.02	18.03	18.40
Payment	3.02	3.03	3.41
Order-Status	2.02	2.03	2.39
Interactive Delivery	2.02	2.03	2.23
Stock-Level	2.02	2.03	2.41

Table 5.4: Think Times

Type	Minimum	Average	Maximum
New-Order	0.00	12.12	121.13
Payment	0.00	12.12	121.13
Order-Status	0.00	10.12	101.03
Interactive Delivery	0.00	5.09	50.72
Stock-Level	0.00	5.09	50.72

Response Time Frequency Distribution Curves and Other Graphs

Response Time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type.

The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New-Order transaction.

Think Time frequency distribution curves (see Clause 5.6.3) must be reported for each transaction type.

Keying Time frequency distribution curves (see Clause 5.6.4) must be reported for each transaction type.

A graph of throughput versus elapsed time (see Clause 5.6.5) must be reported for the New-Order transaction.

Figure 3. New Order Response Time Distribution

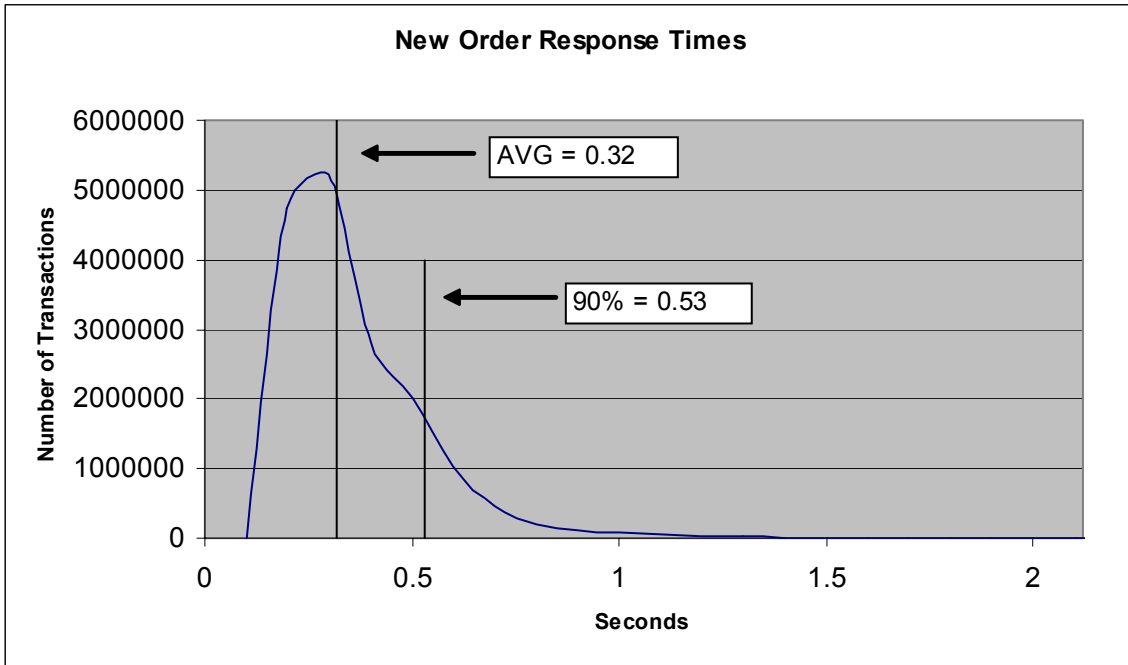


Figure 4. Payment Response Time Distribution

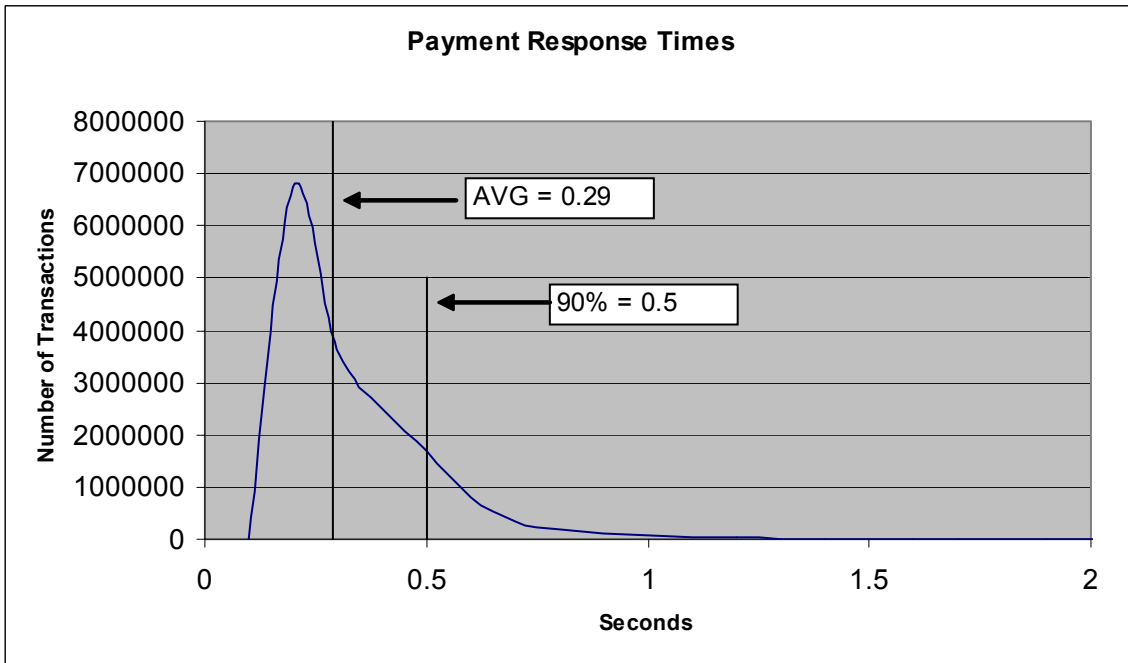


Figure 5. Order Status Response Time Distribution

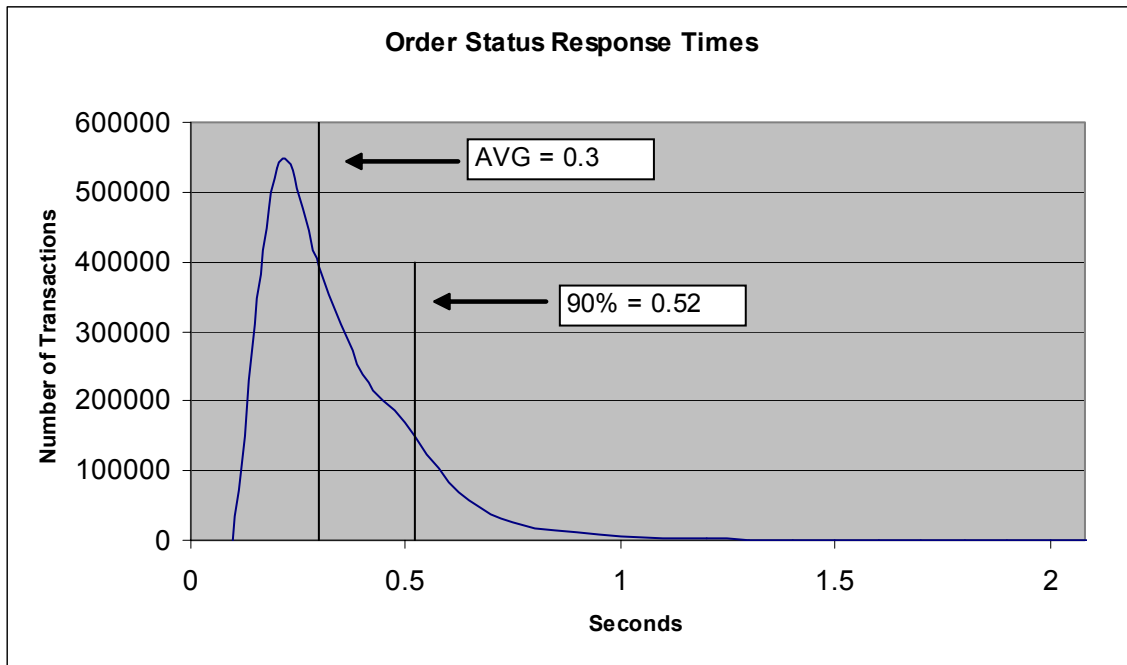


Figure 6. Delivery Response Time Distribution

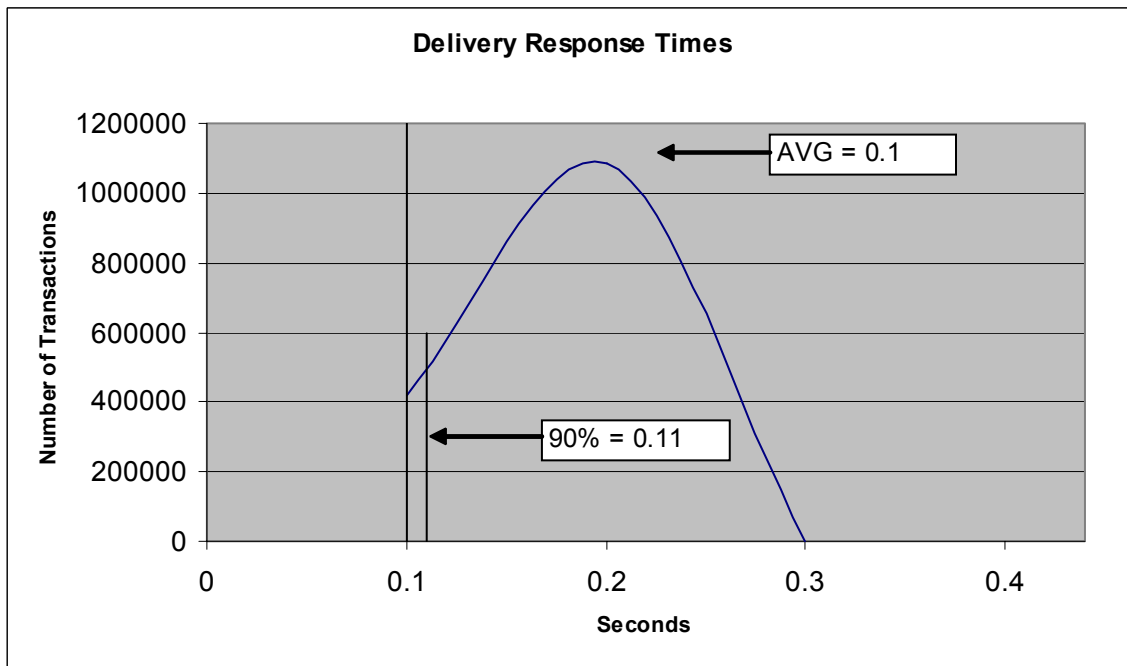


Figure 7. Stock Level Response Time Distribution

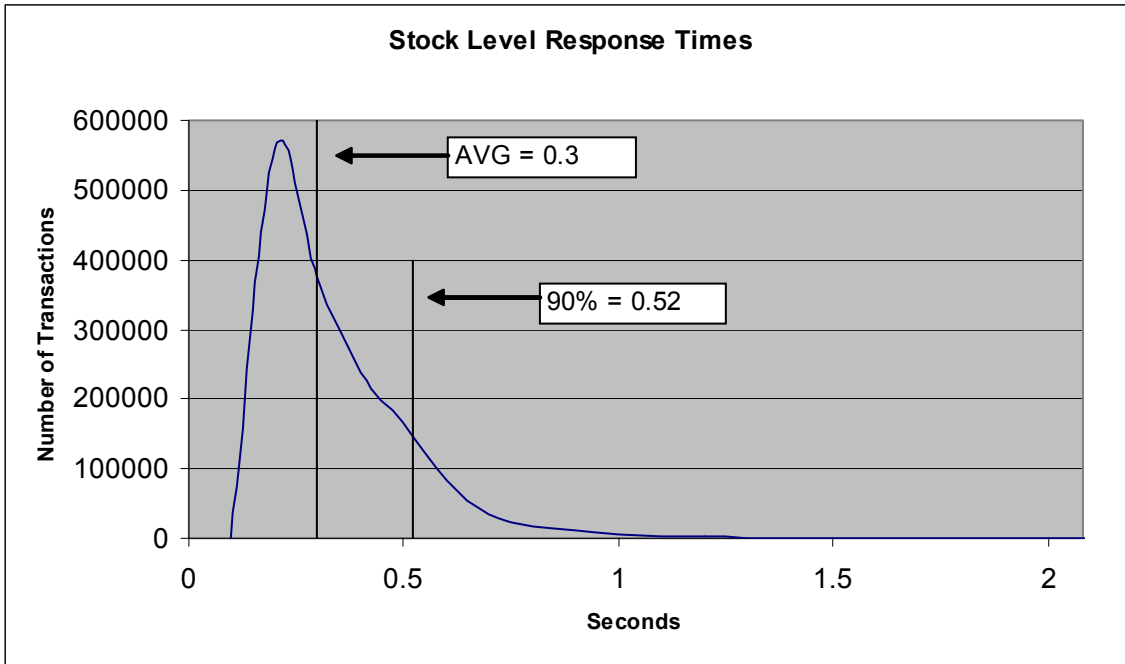


Figure 8. Response Time vs. Throughput

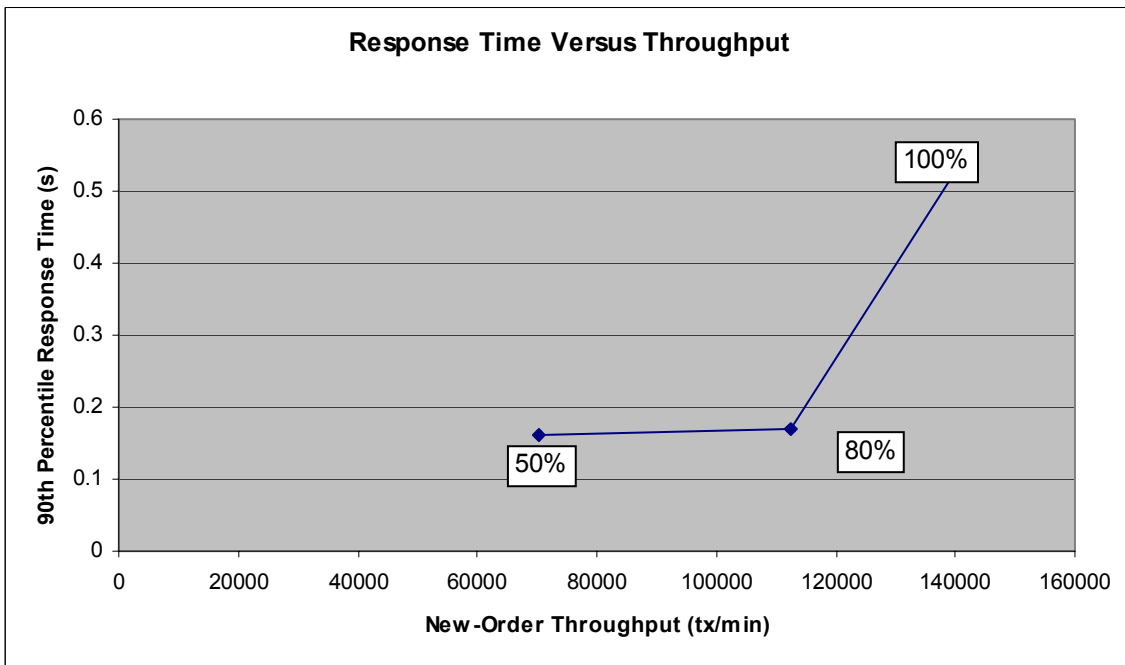


Figure 9. New Order Think Time Distribution

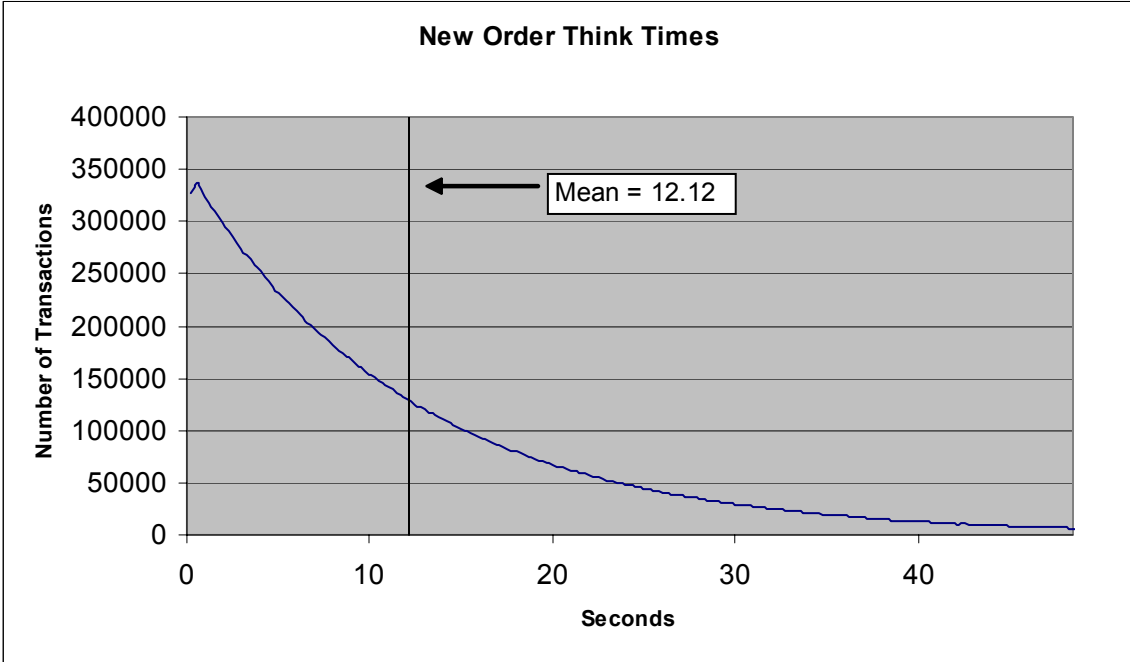
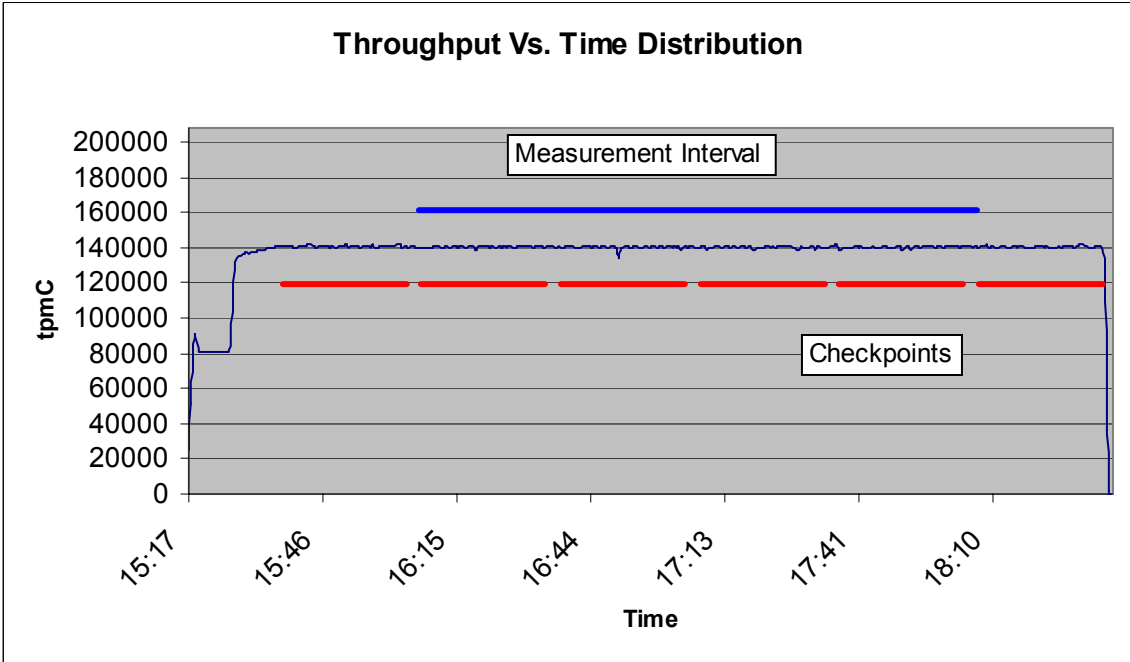


Figure 10. Throughput vs. Time Distribution



Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval must be disclosed.

Steady state was determined using real time monitor utilities from the RTE. Steady state was further confirmed by the throughput data collected during the run and graphed in Figure 10.

Work Performed During Steady State

A description of how the work normally performed during a sustained test (for example checkpointing, writing redo/undo log records, etc.), actually occurred during the measurement interval must be reported.

The RTE generated the required input data to choose a transaction from the menu. This data was timestamped. The input screen for the requested transaction was returned and timestamped. The difference between these two timestamps was the menu response time. The RTE writes to the log file once per transaction on selective fields such as order id. There is one log file per driver engine.

The RTE generated the required input data for the chosen transaction. It waited to complete the minimum required key time before transmitting the input screen. The transmission was timestamped. The return of the screen with the required response data was timestamped. The difference between these two timestamps was the response time for that transaction.

The RTE then waited the required think time interval before repeating the process starting at selecting a transaction from the menu.

The RTE transmissions were sent to application processes running on the client machines through Ethernet LANs. These client application processes handled all screen I/O as well as all requests to the database on the server. The applications communicated with the database server over gigabit Ethernet LANs using DBLIB and RPC calls.

To perform checkpoints at specific intervals, the SQL Server *recovery interval* was set to 32768 and a script was written to schedule multiple checkpoints at specific intervals. The script included a wait time between each checkpoint equal to 30 minutes. The measurement interval was 120 minutes. The checkpoint script was started manually after the RTE had all users logged in and the database had achieved steady state.

At each checkpoint, Microsoft SQL Server wrote to disk all memory pages that had been updated but not yet physically written to disk. The positioning of the measurement interval is depicted on the graph in Figure 9.

Measurement Period Duration

A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC) must be included.

The reported measured interval was exactly 120 minutes long.

Regulation of Transaction Mix

The method of regulation of the transaction mix (e.g., card decks or weighted random distribution) must be described. If weighted distribution is used and the RTE adjusts the weights associated with each transaction type, the maximum adjustments to the weight from the initial value must be disclosed.

The RTE was given a weighted random distribution, which was not adjusted during the run.

Transaction Statistics

The percentage of the total mix for each transaction type must be disclosed. The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed. The average number of order-lines entered per New-Order transaction must be disclosed. The percentage of remote order lines per New-Order transaction must be disclosed. The percentage of remote Payment transactions must be disclosed. The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed. The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed.

Table 5.5: Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse payments	85.00%
	Remote warehouse payments	15.00%
	Accessed by last name	60.02%
Delivery	Skipped transactions (interactive)	0
	Skipped transactions (deferred)	0
Order Status	Accessed by last name	60.07%
Transaction Mix	New Order	44.93%
	Payment	43.03%
	Order status	4.01%
	Delivery	4.01%
	Stock level	4.02%

Checkpoint Count and Location

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

The initial checkpoint was started 22 minutes after the start of the ramp-up. Subsequent checkpoints occurred every 30 minutes. Each checkpoint in the measurement interval lasted 27 minutes and 30 seconds. The measurement interval contains four checkpoints.

Checkpoint Duration

The start time and duration in seconds of at least the four longest checkpoints during the Measurement Interval must be disclosed.

Checkpoint Start Time	Duration
16:07:53.37	26 minutes, 40 seconds
16:37:50.35	26 minutes, 40 seconds
17:07:47.43	26 minutes, 40 seconds
17:37:44.43	26 minutes, 40 seconds

Clause 6 Related Items

RTE Descriptions

If the RTE is commercially available, then its inputs must be specified. Otherwise, a description must be supplied of what inputs (e.g., scripts) to the RTE had been used.

The RTE used was Microsoft Benchcraft RTE. Benchcraft is a proprietary tool provided by Microsoft and is not commercially available. The RTE's input is listed in Appendix A.

Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to the priced system. The results of the test described in Clause 6.6.3.4 must be disclosed.

The driver system consisted of 8 HP ProLiant servers. These driver machines emulated the users' web browsers.

Functional Diagrams

A complete functional diagram of both the benchmark configuration and the configuration of the proposed (target) system must be disclosed. A detailed list of all hardware and software functionality being performed on the Driver System and its interface to the SUT must be disclosed.

The driver system performed the data generation and input functions of the priced display device. It also captured the input and output data and timestamps for post-processing of the reported metrics. No other functionality was included on the driver system.

Section 1.4 of this report contains detailed diagrams of both the benchmark configuration and the priced configuration.

Networks

The network configuration of both the tested services and proposed (target) services that are being represented and a thorough explanation of exactly which parts of the proposed configuration are being replaced with the Driver System must be disclosed.

The bandwidth of the networks used in the tested/priced configuration must be disclosed.

In the tested configuration, 8 driver (RTE) machines were connected through a gigabit Ethernet switch to the 8 client machines at 1Gbs, thus providing the path from the RTEs to the clients. The server (SUT) was connected to the clients through the second gigabit Ethernet switch.

The priced configuration was connected in the same manner as the tested configuration.

Operator Intervention

If the configuration requires operator intervention (see Clause 6.6.6), the mechanism and the frequency of this intervention must be disclosed.

This configuration does not require any operator intervention to sustain eight hours of the reported throughput.

Clause 7 Related Items

System Pricing

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

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 used must be disclosed.

The details of the hardware and software are reported in the front of this report as part of the executive summary. All third party quotations are included at the end of this report as Appendix E.

Availability, Throughput, and Price Performance

The committed delivery date for general availability (availability date) of products used in the price calculation must be reported. When the priced system included products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available.

A statement of the measured tpmC as well as the respective calculations for the 5-year pricing, price/performance (price/tpmC), and the availability date must be included.

- **Maximum Qualified Throughput** **140,264 tpmC**
- **Price per tpmC** **USD \$3.56per tpmC**
- **Availability** **November 22, 2006**

Orderability Date

For each of the components that are not orderable on the report date of the FDR, the following information must be included in the FDR:

- Name and part number of the item that is not orderable
- The date when the component can be ordered (on or before the Availability Date)
- The method to be used to order the component (at or below the quoted price) when that date arrives
- The method for verifying the price

Some line item components used in this benchmark are not orderable at the time of this publication. These items will be orderable on or before the system Availability Date indicated as part of this submission. For specific information on these items regarding orderable dates and pricing verification, please see table in Appendix F.

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 system is being priced for the United States of America.

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.

The component pricing based on usage is shown below:

- 8 Microsoft Windows Server 2000 Standard Edition
- 1 Microsoft Windows Server 2003, Enterprise x64 Edition SP1
- 1 Microsoft SQL Server 2005 Enterprise x64 Edition SP1 (per processor)
- 1 Microsoft Visual C++
- HP Servers include 3 years of support.

Clause 9 Related Items

Auditor's Report

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 C was audited by Lorna Livingtree of Performance Metrics, Inc.

Performance Metrics, Inc.
PO Box 984
Klamath CA 95548
(phone) 707-482-0523
(fax) 707-482-0575
e-mail: lornaL@perfmetrics.com

Availability of the Full Disclosure Report

The Full Disclosure Report must be readily available to the public at a reasonable charge, similar to the charges for similar documents by the test sponsor. The report must be made available when results are made public. In order to use the phrase "TPC Benchmark™ C", the Full Disclosure Report must have been submitted to the TPC Administrator as well as written permission obtained to distribute same.

Requests for this TPC Benchmark C Full Disclosure Report should be sent to:

TPC
Presidio of San Francisco
Building 572B Ruger St. (surface)
P.O. Box 29920 (mail)
San Francisco, CA 94129-0920

or

Hewlett-Packard Company
Database Performance Engineering
P.O. Box 692000
Houston, TX 77269-2000



June 20, 2006

Mr. David Adams
Database Performance Engineer
Hewlett-Packard Company
20555 SH 249
Houston, TX 77070

I have verified by remote the TPC Benchmark™ C for the following configuration:

Platform: HP ProLiant DL380 G5 5160
Database Manager: Microsoft SQL Server 2005 Enterprise X64 Edition SP1
Operating System: Microsoft Windows Server 2003 Enterprise X64 Edition SP1
Transaction Monitor: Microsoft COM+

System Under Test:				
CPU's	Memory	Disks (total)	90% Response	TpmC
2 Intel 5160 @ 3.0 GHz	Main: 32 GB	506 @ 36 GB 16 @ 72 GB	0.53	140,264

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.
- The database files were properly sized.
- The database was properly scaled with 12,400 warehouses, 11,200 of which were active during the measured interval.
- The ACID properties were successfully demonstrated.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was present on the tested system.
- Eight hours of growth space for the dynamic tables was present on the tested system.
- The data for the 60 days space calculation was verified.
- The steady state portion of the test was 120 minutes.
- There was one complete checkpoint in steady state before the measured interval.
- There were 4 checkpoints started and completed inside the measured interval.
- The system pricing was checked for major components and maintenance.
- Third party quotes were verified for compliance.

Auditor Notes: None

Sincerely,

A handwritten signature in cursive script that reads "Lorna Livingtree".

Lorna Livingtree
Auditor

Appendix A: Source Code

The client source code is listed below.

delivery.h

```
/* Generated by IDL compiler version DEC DCE V2.0.0-6
*/
#ifndef _delivery_v1_0_included
#define _delivery_v1_0_included
#include <idlbase.h>
#include <dce\rpc.h>
#include "trpc/trpc.h"

#ifdef __cplusplus
extern "C" {
#endif

#ifndef nbase_v0_0_included
#include "dce\nbase.h"
#endif
#ifndef trpcImports_v0_0_included
#include "trpc\trpcImports.h"
#endif
#ifndef mon_handle_v1_0_included
#include "tpm\mon\mon_handle.h"
#endif
#ifndef tpcc_types_v1_0_included
#include "tpcc_type.h"
#endif
#include <dce\rpcexc.h>
extern EXCEPTION encina_x_transaction_aborted;
extern EXCEPTION encina_x_server_shutdown;
extern EXCEPTION encina_x_permission_denied;
extern EXCEPTION encina_x_object_not_found;
extern EXCEPTION encina_x_empty_slot1;
extern EXCEPTION encina_x_empty_slot2;
extern EXCEPTION encina_x_empty_slot3;
extern EXCEPTION encina_x_empty_slot4;
extern EXCEPTION encina_x_empty_slot5;
extern EXCEPTION encina_x_undefined_exception;
extern void IDL_STD_STDCALL _delivery_GetApplId(
#ifdef IDL_PROTOTYPES
/* [in] */ handle_t handle,
/* [out] */ trpc_byteData_t applString,
/* [out] */ idl_ulong_int *applStringLength,
/* [out] */ trpc_byteData_t address,
/* [out] */ idl_ulong_int *addressLength,
/* [out] */ error_status_t *c_status,
/* [out] */ error_status_t *f_status
#endif
);
#endif
```

```
/* [out] */ error_status_t *c_status,
/* [out] */ error_status_t *f_status
#endif
);
extern void IDL_STD_STDCALL _impTPCCDelivery(
#ifdef IDL_PROTOTYPES
/* [in] */ handle_t trpc_h,
/* [in] */ idl_long_int length,
/* [in, out] */ idl_char *dataP,
/* [in, out] */ data_header *headerP,
/* [in] */ trpc_byteData_t applAndAddress,
/* [in] */ idl_ulong_int applAndAddressLength,
/* [in] */ trpc_callbackData_t inCallbackData,
/* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
globalref mon_handle_t handle;
#ifdef __cplusplus
#pragma extern_model __save
#pragma extern_model __common_block __shr
#endif
typedef struct _delivery_v1_0_epv_t {
void ( IDL_STD_STDCALL * _delivery_GetApplId ) (
#ifdef IDL_PROTOTYPES
/* [in] */ handle_t handle,
/* [out] */ trpc_byteData_t applString,
/* [out] */ idl_ulong_int *applStringLength,
/* [out] */ trpc_byteData_t address,
/* [out] */ idl_ulong_int *addressLength,
/* [out] */ error_status_t *c_status,
/* [out] */ error_status_t *f_status
#endif
);
void ( IDL_STD_STDCALL * _impTPCCDelivery ) (
#ifdef IDL_PROTOTYPES
/* [in] */ handle_t trpc_h,
/* [in] */ idl_long_int length,
/* [in, out] */ idl_char *dataP,
/* [in, out] */ data_header *headerP,
/* [in] */ trpc_byteData_t applAndAddress,
/* [in] */ idl_ulong_int applAndAddressLength,
/* [in] */ trpc_callbackData_t inCallbackData,
/* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
} _delivery_v1_0_epv_t;
extern rpc_if_handle_t delivery_v1_0_c_ifspec;
extern rpc_if_handle_t delivery_v1_0_s_ifspec;
#ifdef __cplusplus
#pragma extern_model __restore
#endif
#ifdef __cplusplus
}
#endif
#endif
```

neworder.h

```
/* Generated by IDL compiler version DEC DCE V2.0.0-6
*/
#ifndef _neworder_v1_0_included
#define _neworder_v1_0_included
#include <idlbase.h>
#include <dce\rpc.h>
#include "trpc/trpc.h"

#ifdef __cplusplus
extern "C" {
#endif

#ifndef nbase_v0_0_included
#include "dce\nbase.h"
#endif
#ifndef trpcImports_v0_0_included
#include "trpc\trpcImports.h"
#endif
#ifndef mon_handle_v1_0_included
#include "tpm\mon\mon_handle.h"
#endif
#ifndef tpcc_types_v1_0_included
#include "tpcc_type.h"
#endif
#include <dce\rpcexc.h>
extern EXCEPTION encina_x_transaction_aborted;
extern EXCEPTION encina_x_server_shutdown;
extern EXCEPTION encina_x_permission_denied;
extern EXCEPTION encina_x_object_not_found;
extern EXCEPTION encina_x_empty_slot1;
extern EXCEPTION encina_x_empty_slot2;
extern EXCEPTION encina_x_empty_slot3;
extern EXCEPTION encina_x_empty_slot4;
extern EXCEPTION encina_x_empty_slot5;
extern EXCEPTION encina_x_undefined_exception;
extern void IDL_STD_STDCALL _neworder_GetApplId(
#ifdef IDL_PROTOTYPES
/* [in] */ handle_t handle,
/* [out] */ trpc_byteData_t applString,
/* [out] */ idl_ulong_int *applStringLength,
/* [out] */ trpc_byteData_t address,
/* [out] */ idl_ulong_int *addressLength,
/* [out] */ error_status_t *c_status,
/* [out] */ error_status_t *f_status
#endif
);
extern void IDL_STD_STDCALL _impTPCCNewOrder(
#ifdef IDL_PROTOTYPES
/* [in] */ handle_t trpc_h,
/* [in] */ idl_long_int length,
/* [in, out] */ idl_char *dataP,
/* [in, out] */ data_header *headerP,
/* [in] */ trpc_byteData_t applAndAddress,
/* [in] */ idl_ulong_int applAndAddressLength,
/* [in] */ trpc_callbackData_t inCallbackData,
/* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
#endif
```

```

#endif
);
extern void IDL_STD_STDCALL _impTPCCNOInfo(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [out] */ dbInfo_data_t *dataP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
globalref mon_handle_t handle;
#if defined(__VMS) && (defined(__DECC) ||
defined(__cplusplus))
#pragma extern_model __save
#pragma extern_model __common_block __shr
#endif
typedef struct _neworder_v1_0_epv_t {
void ( IDL_STD_STDCALL * _neworder_GetAppId)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
void ( IDL_STD_STDCALL * _impTPCCNewOrder)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
void ( IDL_STD_STDCALL * _impTPCCNOInfo)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [out] */ dbInfo_data_t *dataP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
} _neworder_v1_0_epv_t;
extern rpc_if_handle_t _neworder_v1_0_c_ifspec;
extern rpc_if_handle_t _neworder_v1_0_s_ifspec;
#if defined(__VMS) && (defined(__DECC) ||
defined(__cplusplus))
#pragma extern_model __restore
#endif
#endif
#endif
#endif

```

```

}
#else
#endif
#endif

```

orderstatus.h

```

/* Generated by IDL compiler version DEC DCE V2.0.0-6
*/
#ifdef _orderstatus_v1_0_included
#define _orderstatus_v1_0_included
#endif
#include <dce\idlbase.h>
#endif
#include <dce\rpc.h>
#include "trpc/trpc.h"

#ifdef __cplusplus
extern "C" {
#endif

#ifdef nbase_v0_0_included
#include "dce\nbase.h"
#endif

#ifdef trpcImports_v0_0_included
#include "trpc\trpcImports.h"
#endif

#ifdef mon_handle_v1_0_included
#include "tpm\mon\mon_handle.h"
#endif

#ifdef tpcc_types_v1_0_included
#include "tpcc_type.h"
#endif

#include <dce\rpcexc.h>
extern EXCEPTION encina_x_transaction_aborted;
extern EXCEPTION encina_x_server_shutdown;
extern EXCEPTION encina_x_permission_denied;
extern EXCEPTION encina_x_object_not_found;
extern EXCEPTION encina_x_empty_slot1;
extern EXCEPTION encina_x_empty_slot2;
extern EXCEPTION encina_x_empty_slot3;
extern EXCEPTION encina_x_empty_slot4;
extern EXCEPTION encina_x_empty_slot5;
extern EXCEPTION encina_x_undefined_exception;
extern void IDL_STD_STDCALL _orderstatus_GetAppId(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
#endif
extern void IDL_STD_STDCALL _impTPCCOrderStatus(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,

```

```

    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
globalref mon_handle_t handle;
#if defined(__VMS) && (defined(__DECC) ||
defined(__cplusplus))
#pragma extern_model __save
#pragma extern_model __common_block __shr
#endif
typedef struct _orderstatus_v1_0_epv_t {
void ( IDL_STD_STDCALL * _orderstatus_GetAppId)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
void ( IDL_STD_STDCALL * _impTPCCOrderStatus)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
} _orderstatus_v1_0_epv_t;
extern rpc_if_handle_t _orderstatus_v1_0_c_ifspec;
extern rpc_if_handle_t _orderstatus_v1_0_s_ifspec;
#if defined(__VMS) && (defined(__DECC) ||
defined(__cplusplus))
#pragma extern_model __restore
#endif
#endif
#endif

```

payment.h

```

/* Generated by IDL compiler version DEC DCE V2.0.0-6
*/
#ifdef _payment_v1_0_included
#define _payment_v1_0_included
#endif
#include IDLBASE_H

```

```

#include <dce\idlbase.h>
#endif
#include <dce\rpc.h>
#include "trpc/trpc.h"

#ifdef __cplusplus
extern "C" {
#endif

#ifndef nbase_v0_0_included
#include "dce\nbase.h"
#endif
#ifndef trpcImports_v0_0_included
#include "trpc\trpcImports.h"
#endif
#ifndef mon_handle_v1_0_included
#include "tpm\mon\mon_handle.h"
#endif
#ifndef tpcc_types_v1_0_included
#include "tpcc_type.h"
#endif
#include <dce\rpcexc.h>
extern EXCEPTION encina_x_transaction_aborted;
extern EXCEPTION encina_x_server_shutdown;
extern EXCEPTION encina_x_permission_denied;
extern EXCEPTION encina_x_object_not_found;
extern EXCEPTION encina_x_empty_slot1;
extern EXCEPTION encina_x_empty_slot2;
extern EXCEPTION encina_x_empty_slot3;
extern EXCEPTION encina_x_empty_slot4;
extern EXCEPTION encina_x_empty_slot5;
extern EXCEPTION encina_x_undefined_exception;
extern void IDL_STD_STDCALL _payment_GetApplId(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
extern void IDL_STD_STDCALL _impTPCCPayment(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
globalref mon_handle_t handle;
#if defined(__VMS) && (defined(__DECC) ||
defined(__cplusplus))
#pragma extern_model __save
#pragma extern_model __common_block __shr
#endif

```

```

typedef struct _payment_v1_0_epv_t {
void ( IDL_STD_STDCALL * _payment_GetApplId) (
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
void ( IDL_STD_STDCALL * _impTPCCPayment) (
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
};
extern rpc_if_handle_t _payment_v1_0_c_ifspec;
extern rpc_if_handle_t _payment_v1_0_s_ifspec;
#if defined(__VMS) && (defined(__DECC) ||
defined(__cplusplus))
#pragma extern_model __restore
#endif

#ifdef __cplusplus
}
#else
#endif

```

stocklevel.h

```

/* Generated by IDL compiler version DEC DCE V2.0.0-6
*/
#ifndef _stocklevel_v1_0_included
#define _stocklevel_v1_0_included
#endif
#include <dce\idlbase.h>
#include <dce\rpc.h>
#include "trpc/trpc.h"

#ifdef __cplusplus
extern "C" {
#endif

#ifndef nbase_v0_0_included
#include "dce\nbase.h"
#endif
#ifndef trpcImports_v0_0_included
#include "trpc\trpcImports.h"

```

```

#endif
#ifndef mon_handle_v1_0_included
#include "tpm\mon\mon_handle.h"
#endif
#ifndef tpcc_types_v1_0_included
#include "tpcc_type.h"
#endif
#include <dce\rpcexc.h>
extern EXCEPTION encina_x_transaction_aborted;
extern EXCEPTION encina_x_server_shutdown;
extern EXCEPTION encina_x_permission_denied;
extern EXCEPTION encina_x_object_not_found;
extern EXCEPTION encina_x_empty_slot1;
extern EXCEPTION encina_x_empty_slot2;
extern EXCEPTION encina_x_empty_slot3;
extern EXCEPTION encina_x_empty_slot4;
extern EXCEPTION encina_x_undefined_exception;
extern void IDL_STD_STDCALL _stocklevel_GetApplId(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
extern void IDL_STD_STDCALL _impTPCCStockLevel(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
globalref mon_handle_t handle;
#if defined(__VMS) && (defined(__DECC) ||
defined(__cplusplus))
#pragma extern_model __save
#pragma extern_model __common_block __shr
#endif
typedef struct _stocklevel_v1_0_epv_t {
void ( IDL_STD_STDCALL * _stocklevel_GetApplId) (
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
void ( IDL_STD_STDCALL * _impTPCCStockLevel) (
#ifdef IDL_PROTOTYPES

```

```

/* [in] */ handle_t trpc_h,
/* [in] */ idl_long_int length,
/* [in, out] */ idl_char *dataP,
/* [in, out] */ data_header *headerP,
/* [in] */ trpc_byteData_t applAndAddress,
/* [in] */ idl_ulong_int applAndAddressLength,
/* [in] */ trpc_callbackData_t inCallbackData,
/* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
}_stocklevel_v1_0_epv_t;
extern rpc_if_handle_t _stocklevel_v1_0_c_ifspec;
extern rpc_if_handle_t _stocklevel_v1_0_s_ifspec;
#if defined(__VMS) && (defined(__DECC) ||
defined(__cplusplus))
#pragma extern_model __restore
#endif

#ifdef __cplusplus
}
#else
#endif
#endif

```

client_utils.c

```

/* client_utils.c
*/

#include <stdio.h>
#include <time.h>
#include <windows.h>
#include <winperf.h>
#include <winsock.h>
#include "client_utils.h"

#define Li2Double(x) ((double)((x).HighPart) *
4.294967296E9 + (double)((x).LowPart))

static LARGE_INTEGER pFreq;
static double sFreq;
static int print_thread_id = 1;
static int user_id = 0;
static char *user_code = "C";

/*
 * get_thread_id
 * A function that returns the thread ID of the
current thread
*/
static int get_thread_id()
{
return(GetCurrentThreadId());
}

/*
 * get_prefix
 * Format the output prefix for printing:

```

```

* It contains the user_id, 'C' or 'T'
depending on whether it
* is a terminal or a client and optional a
thread identifier
* The prefix is written in the buffer passed
in by the caller.
*/
static void get_prefix(char *buffer)
{
if (print_thread_id) {
int thread_id = get_thread_id();
sprintf(buffer, "%s(%d-%s-
%d)%s",
user_id < 10 ? " " : user_id <
100 ? " " : "",
user_id,
user_code,
thread_id,
thread_id < 10 ? " " : "");
} else {
sprintf(buffer, "%s(%2d-%s)",
user_id < 10 ? " " : "", user_id,
user_code);
}
}

/*
 * err_printf
 * A var-arg function that appends the current
time and
* other data to the print request and sends it
to stderr
* if it is not a web client, to a file if it is
*/
void err_printf(char *format, ...)
{
time_t cur_timet;
char time_str[30];
char line_prefix[50];
va_list ap;

va_start(ap, format);

cur_timet = time(&cur_timet);
strftime(time_str, 29, "%X",
localtime(&cur_timet));

get_prefix(line_prefix);

fprintf(ERROROUT, "%s %s - ", line_prefix,
time_str);
vfprintf(ERROROUT, format, ap);
fflush(ERROROUT);

va_end(ap);
}

/*
 * encina_error_message
 *

```

```

* Report an encina error message by interpreting it
and writing
* it to both the logfile (if any) and to standard
error
*/
void encina_error_message(char *msg, unsigned long n)
{
char errorMsg[ENCINA_MAX_STATUS_STRING_SIZE];
encina_StatusToString(n,
errorMsg);
ENCINA_MAX_STATUS_STRING_SIZE, errorMsg);
err_printf("ERROR: %s. Error code = %s (%d 0x%x)
\n", msg, errorMsg, n, n);
}

int get_time_init()
{
QueryPerformanceFrequency(&pFreq);
sFreq=Li2Double(pFreq);
return 0;
}

int get_local_time(time_type *timeP)
{
double cur_t;
LARGE_INTEGER counter;

QueryPerformanceCounter(&counter);
cur_t = Li2Double(counter) / sFreq;
timeP->sec = (long)cur_t;
/* timeP->usec = ((long)cur_t - timeP->sec) *
1000000;*/
timeP->usec = (long)((cur_t - timeP->sec) *
1000000);
return 0;
}

/*
 * time_diff_ms
 * Return the difference in milliseconds between
two times
*/
int time_diff_ms(struct timeval *t2, struct timeval
*t1)
{
int t_diff;

t_diff = (t2->tv_usec + 1000000 - t1->tv_usec +
500) / 1000 +
(t2->tv_sec - t1->tv_sec - 1) * 1000;

return(t_diff);
}

/*
 * perfClntDataInit:
 * Initialization for the shared file mapping.
 *
 * return: pointer to the shared memory space
 *
 * This routine creates a named mapped memory section
that is used

```



```

* to communicate the TPCC performance data to the
extensible
* counter DLL for NT perfmon.
*/
total_tran_count_t *perfClntDataInit()
{
    HANDLE hMappedObject;
    total_tran_count_t *pClntInfo = NULL;
    TCHAR szMappedObjectName[] =
TEXT("TPCC_CLNT_COUNTER_BLOCK");

    /* create named section for the performance
data */
    hMappedObject =
CreateFileMapping((HANDLE)0xFFFFFFFF,
    NULL,
    PAGE_READWRITE,
    0,
    sizeof(total_tran_count_t),
    szMappedObjectName);
    if (hMappedObject == NULL) {
        err_printf("perfClntDataInit:
CreateFileMapping failed %x\n",
            GetLastError());
    } else {
        pClntInfo = NULL;
        /* map the section and assign the counter
block pointer
* to this section of memory
*/
        pClntInfo = (total_tran_count_t *)
MapViewOfFile(hMappedObject,
                FILE_MAP_ALL_ACCESS,
                0,
                0,
                0);
        if (pClntInfo == NULL) {
            err_printf("perfClntDataInit:
MapViewOfFile failed %x\n",
                GetLastError());
        } else {
            err_printf("perfClntDataInit:
MapViewOfFile success\n");
        }
    }
    return(pClntInfo);
}

```

client_utils.h

```

#ifndef TPCC_CLIENT_UTILS_H
#define TPCC_CLIENT_UTILS_H

#include <stdio.h>
#include <time.h>
#include <dce/rpc.h>
#include <dce/dce_error.h>
#include <encina/encina.h>

```

```

#include <stdlib.h>
#include <utils/trace.h>
#include <winsock.h>
#include "mon_client.h"
#include "../include/tpcc_type.h"

extern FILE * errtpcc;
extern FILE * logtpcc;
extern int debug;
extern char log_file_name[];
extern void logprintf( char *format, ...);
extern void err_printf( char *format, ...);
extern void encina_error_message(char *msg, unsigned
long n);
extern int time_diff_ms(struct timeval *t2, struct
timeval *t1);

typedef struct {
    int num;
    int errs;
    double RTtotal[2]; // 1 for server RT and 0 for
client RT
    int RTcount;
} tran_info_t;

/*
* total_tran_count_t
*
* structure that holds the total count of
transaction of each type
* as well as the reposne times.
*/
typedef struct {
    tran_info_t tran[MAX_TRAN_TYPE + 1];
    int errors;
    double time;
} total_tran_count_t;

/* enc_status_t
* structure that holds error information
*/
typedef struct {
    int status;
    int line;
    char file[268];
    unsigned long encinaError;
    char errorMsg[ENCINA_MAX_STATUS_STRING_SIZE];
} enc_status_t;

#define FALSE 0
#define TRUE 1

#define DPRINT(args) if (0) err_printf args

#define CHECK_ENVIRON(str,var) if (str == NULL) {
fprintf(ERROROUT, \
        "%s environment variable is
not defined.\n",var); }

```

```

#define CHK_STATUS(st, val, _errMsg)
\
    if(st) {
\
        enc_status.status=val;
\
        strcpy(enc_status.file, __FILE__);
\
        enc_status.line= __LINE__;
\
        enc_status.encinaError = st;
\
        if(_errMsg)strcpy(enc_status.errorMsg,
_errMsg);
\
        if(st!=1) return;
\
    }

#define UTIL_IDENT(a)          a

#if ENCINA_C_ANSI_STRING_TOKEN_SUPPORT
#define UTIL_STRING(a)        # a
#define UTIL_CONCAT(a, b)    a ## b
#else /* ENCINA_C_ANSI_STRING_TOKEN_SUPPORT */
#define UTIL_STRING(a)        "a"
#define UTIL_CONCAT(a, b)    UTIL_IDENT(a)b
#endif /* ENCINA_C_ANSI_STRING_TOKEN_SUPPORT */

/* ENCINA_CALL: Make fail-fast calls on the various
services. */
#define ENCINA_CALL(proc_name,call) \
{ \
    unsigned long _status; \
    ENCINA_CALL_RC(proc_name,call, _status); \
    if (_status) exit_program(_status); \
}

#define ENCINA_CALL_RC(proc_name,call,rc) \
{ \
    char _errorMsg[ENCINA_MAX_STATUS_STRING_SIZE]; \
    DPRINT(("ENCINA_CALL_RC: before call %s\n", \
proc_name)); \
    rc = (call); \
    DPRINT(("ENCINA_CALL_RC: after call %s\n", \
proc_name)); \
    if (rc) { \
        encina_StatusToString(rc, \
ENCINA_MAX_STATUS_STRING_SIZE, \
            _errorMsg); \
        err_printf( "%x\n", rc); \
        err_printf( "%s\n", _errorMsg); \
    } \
}

```

```

err_printf( "%s \n", proc_name);
\
}
\
}

void err_printf(char *format, ...);
void encina_error_message(char *msg, unsigned long
n);
int get_time_init();
int get_local_time(time_type *timeP);
int time_diff_ms(struct timeval *t2, struct timeval
*t1);

#endif /* TPCC_CLIENT_UTILS_H */

```

databuf.h

```

/*
 * databuf.h
 *
 * $Revision: 1.1 $
 * $Date: 1998/11/06 21:10:11 $
 * $Log: databuf.h,v $
 * Revision 4.2 95/05/16 10:55:31 10:55:31 tpcc
(TPCC Benchmark)
 * Added necessary RCS ident strings
 *
 * Revision 4.1 95/05/09 15:21:02 15:21:02 strue
(Scott Truesdale)
 * New code from Transarc - initial version
 *
 * Revision 3.2 95/04/03 17:43:09 17:43:09 strue
(Scott Truesdale)
 * Changes from Transarc - added sql error handling
in client; cleaned up debug handling with macros;
added check on db paramters via call to server.
 *
 * Revision 3.1 95/04/03 15:10:30 15:10:30 strue
(Scott Truesdale)
 * Base of rev 3 - shipped to transarc
 *
 *
 *
 * $TALog: databuf.h,v $
 * Revision 1.1 1998/11/06 21:10:11 dongfeng
 * - Move all files common to client and server to
tpcc/common
 * directory
 * [added by delta dongfeng-23677-TPCC-new-directory-
structures, r1.1]
 *
 * Revision 1.3 1998/10/22 15:33:04 wenjian
 * Make changes to Encina server code to connect with
SQL server and add
 * callsql.c and sql directory.
 *
 * Add ERR_BAD_ITEM_ID, which is returned by SLQnew
and same as INVALID_NEWO

```

```

 * [from r1.2 by delta wenjian-23529-TPCC-integrate-
with-SQL-server, r1.1]
 *
 * Revision 1.2 1998/01/23 15:07:47 oz
 * - Updated the SP TPCC directory to the latest
files used
 * during the SP tpcc audit.
 * [from r1.1 by delta oz-20774-TPCC-update-to-
latest-SP-version-11-27, r1.1]
 *
 * Revision 1.1 1997/04/20 11:57:57 oz
 * - This is the code base modified at IBM
Poughkeepsie
 * by Ofer Zajicek and Radha Sivaramakrishnan for
the
 * SP scaling test for TPCC.
 * [added by delta oz-19782-TPCC-add-ibm-sp-code,
r1.1]
 *
 * Revision 1.31 1995/10/30 19:10:54 oz
 * [merge of changes from 1.29 to 1.30 into 1.27]
 *
 * Revision 1.30 1995/10/27 15:41:30 oz
 * - Modified the tpc-c code to work with the new
informix
 * sql code that is in ex_trans.ec
 * [from r1.29 by delta oz-16761-TPCC-modify-code-to-
work-with-oracle, r1.1]
 *
 * Revision 1.27 1995/10/20 18:44:30 ctipper
 * [merge of changes from 1.17 to 1.25 into 1.22]
 *
 * Revision 1.25 1995/10/20 18:15:34 ctipper
 * Incorporate changes per code review.
 *
 * - add DISTRIBUTED_TRAN_FAILED,
TPCC_DB_INFO_PARTIAL, and
 * TPCC_DB_INFO_FAILED error codes to tpcc_rc_t
 * - got rid of MAX_NUM_SERVERS variables
 * [from r1.23 by delta ctipper-16547-TPCC-more-
distributed-trans, r1.2]
 *
 * Revision 1.23 1995/10/13 17:00:26 ctipper
 * This delta encompasses all changes necessary to do
distributed, XA
 * transactions with the TPCC benchmark. This
includes the changes
 * necessary to build with Informix version 6.
 *
 * Each client still talks to only one server,
however, if a distributed
 * transaction is necessary, the client sends the
request to a different
 * interface of that server which then forwards all
or part of the
 * request on to the appropriate remote server.
 *
 * - added new error codes to the tpcc_rc_t
enumeration.
 * - defined MAX_NUM_SERVERS to be 10

```

```

 * [from r1.19 by delta ctipper-16547-TPCC-more-
distributed-trans, r1.1]
 *
 * Revision 1.19 1995/09/20 21:02:39 oz
 * -Corrected code for the payment transaction
 * - The distributed case now no longer uses
 * stored procedures
 * [from r1.18 by delta oz-16547-TPCC-add-
distributed-transactions, r1.2]
 *
 * Revision 1.18 1995/09/20 17:51:10 oz
 * - Added distributed transactions for the new order
and
 * payment transaction
 *
 * - Added new error codes
 * [from r1.17 by delta oz-16547-TPCC-add-
distributed-transactions, r1.1]
 *
 * Revision 1.22 1995/10/02 20:31:07 oz
 * - Corrected definition of ERROR()
 * [from r1.21 by delta oz-16638-tpcc-modify-
terminal-for-RTE, r1.3]
 *
 * Revision 1.21 1995/10/02 18:51:45 oz
 * - Added definitions needed for utils.c and
liberty.c
 * [from r1.20 by delta oz-16638-tpcc-modify-
terminal-for-RTE, r1.2]
 *
 * Revision 1.20 1995/10/02 15:52:35 oz
 * - Modified the TPC-C benchmark to be compatible
with the RTE.
 * - There are now 3 terminal processes:
 * emulator: the old terminal process with a
built in
 * simple emulator
 * curses: An interactive terminal process using
curses
 * liberty: An interactive terminal process to be
used with
 * the RTE compatible with the liberty
freedom terminal.
 *
 * - Define TRUE and FALSE only if they are not
already defined.
 * (curses.h defines TRUE)
 * - Removed READ_TO_DATE and YEAR_TO_SECOND
 * - Added term_type_t
 * - Added
 * GOOD_INPUT (0)
 * WRONG_INPUT (10)
 * [from r1.17 by delta oz-16638-tpcc-modify-
terminal-for-RTE, r1.1]
 *
 * Revision 1.17 1995/07/28 15:28:23 oz
 * - Added a -null and -no_marshallng option to TPCC
 *
 * - Added INVALID_TRAN_TYPE return code
 * [from r1.16 by delta oz-16070-TPCC-add-null-and-
marshalling-test, r1.1]

```

```

*
* Revision 1.16 1995/07/18 17:02:38 oz
* - Added a DCE_ERROR error code
* [from r1.15 by delta oz-15938-TPCC-add-dce-only-
client, r1.1]
*
* Revision 1.15 1995/05/22 19:50:48 shl
* [merge of changes from 1.12 to 1.13 into 1.14]
*
* Revision 1.13 1995/05/18 15:11:27 oz
* [from r1.12 by delta oz-15290-TPCC-incorporate-hp-
drop-of-05-16-95, r1.1]
*
* Revision 1.14 1995/05/22 17:26:35 ctipper
* [merge of changes from 1.5 to 1.9 into 1.11]
*
* [*** log entries omitted ***]
*
*/

#ifndef TPCC_DATABUF_H
#define TPCC_DATABUF_H

#define I_NAME_LEN 24
#define I_DATA 50
#define W_NAME_LEN 10
#define ADDR_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9
#define DIST_INFO_LEN 24
#define S_DATA_LEN 50
#define D_NAME_LEN 10
#define H_DATA_LEN 24
#define CARRIER_LEN 2
#define C_LAST_LEN 17
#define C_MID_LEN 2
#define PHONE_LEN 16
#define CREDIT_LEN 2
#define C_DATA_LEN 500
#define BC_DTA_LEN 23

#define YEAR_TO_DATE 1
#define YEAR_TO_SECOND 2

#define ERROR(x) fprintf(stderr, "Error:
%s\n", #x), exit(11)

#define MAX_STR_LEN 255
#define MAX_OL 15

#ifndef TRUE
#define TRUE 1
#endif
#ifndef FALSE
#define FALSE 0
#endif

#define CANCEL -1

#define DATETIME_LEN 19

```

```

#define D_PER_W 10

#define COLLECTOR 1 /* ctipper
5/3/95 */

#define ERR_BAD_ITEM_ID 1 /* copied from sql/tpcc.h
*/
#define RPC_ERROR -2
#define SUCCESS_CODE 0

#define CHAR_NULL '\0' /* strue
1/23/95 */

typedef enum {
liberty_term,
curses_term,
emulator_term
} term_type_t;

typedef enum {
GOOD_INPUT = 0,

SQL_ERROR = 2,
DCE_ERROR = 4,
NO_SUCH_LAST_NAME = 5,
INVALID_TRAN_TYPE = 6,
INVALID_HANDLE = 7,

WRONG_INPUT = 10,

DISTRIBUTED_TRAN_FAILED = 15,

TPCC_DB_INFO_PARTIAL = 20,
TPCC_DB_INFO_FAILED,

TPCC_ERROR_BEGIN_NEWO = 110,

TPCC_ERROR_DECL_NEWO_SEL_ITEM,
TPCC_ERROR_OPEN_NEWO_SEL_ITEM,
TPCC_ERROR_OPEN_DIST_NEWO_SEL_ITEM,
TPCC_ERROR_FETCH_NEWO_SEL_ITEM,
TPCC_ERROR_FETCH_DIST_NEWO_SEL_ITEM,
TPCC_ERROR_PREP_NEWO_SEL_STCK,
TPCC_ERROR_DECL_NEWO_SEL_STCK,
TPCC_ERROR_OPEN_NEWO_SEL_STCK,
TPCC_ERROR_OPEN_DIST_NEWO_SEL_STCK,
TPCC_ERROR_FETCH_NEWO_SEL_STCK,
TPCC_ERROR_FETCH_DIST_NEWO_SEL_STCK,
TPCC_ERROR_NEWO_SELECT,
TPCC_ERROR_NEWO_UPD_STCK,
TPCC_ERROR_DIST_NEWO_UPD_STCK,
TPCC_ERROR_NEWO_SELECT_2,
TPCC_ERROR_DECL_NEWO_SEL_CUST,
TPCC_ERROR_OPEN_NEWO_SEL_CUST,
TPCC_ERROR_OPEN_DIST_NEWO_SEL_CUST,
TPCC_ERROR_FETCH_NEWO_SEL_CUST,
TPCC_ERROR_FETCH_DIST_NEWO_SEL_CUST,
TPCC_ERROR_DECL_NEWO_SEL_DIST,
TPCC_ERROR_OPEN_NEWO_SEL_DIST,
TPCC_ERROR_OPEN_DIST_NEWO_SEL_DIST,
TPCC_ERROR_FETCH_NEWO_SEL_DIST,

```

```

TPCC_ERROR_FETCH_DIST_NEWO_SEL_DIST,
TPCC_ERROR_PREP_NEWO_INS_OL,
TPCC_ERROR_DECL_NEWO_INS_OL,
TPCC_ERROR_OPEN_NEWO_INS_OL,
TPCC_ERROR_OPEN_DIST_NEWO_INS_OL,
TPCC_ERROR_PUT_NEWO_INS_OL,
TPCC_ERROR_PUT_DIST_NEWO_INS_OL,
TPCC_ERROR_DECL_NEWO_SEL_WARE,
TPCC_ERROR_OPEN_NEWO_SEL_WARE,
TPCC_ERROR_OPEN_DIST_NEWO_SEL_WARE,
TPCC_ERROR_FETCH_NEWO_SEL_WARE,
TPCC_ERROR_FETCH_DIST_NEWO_SEL_WARE,
TPCC_ERROR_EXECUTE_NEWO_UPD_INS,
TPCC_ERROR_UPDATE_NEWO_NEXT_OID,
TPCC_ERROR_PREP_NEWO_INS,
TPCC_ERROR_EXECUTE_DIST_NEWO_INS,
TPCC_ERROR_EXECUTE_NEWO_COMMIT,
TPCC_ERROR_ROLLBACK_NEWO,
TPCC_ERROR_REMOTE_OL_SELECT,
TPCC_ERROR_REMOTE_OL_UPDATE,

TPCC_ERROR_OPEN_ORDS_CNT_CID = 200,
TPCC_ERROR_FETCH_ORDS_CNT_CID,
TPCC_ERROR_OPEN_ORDS_SEL_CLAST,
TPCC_ERROR_FETCH_ORDS_SEL_CLAST,
TPCC_ERROR_OPEN_ORDS_SEL_CID,
TPCC_ERROR_FETCH_ORDS_SEL_CID,
TPCC_ERROR_OPEN_ORDS_SEL_OLDORD,
TPCC_ERROR_FETCH_ORDS_OLDORD,
TPCC_ERROR_OPEN_ORDS_SEL_OL,
TPCC_ERROR_FETCH_ORDS_SEL_OL,
TPCC_ERROR_EXECUTE_ORDS_COMMIT,

TPCC_ERROR_OPEN_DELIVERY_OLDEST_OID = 300,
TPCC_ERROR_FETCH_DELIVERY_OLDEST_OID,
TPCC_ERROR_EXECUTE_DELIVERY_COMMIT,
TPCC_ERROR_OPEN_DELIVERY_SEL_ORD,
TPCC_ERROR_FETCH_DELIVERY_SEL_ORD,
TPCC_ERROR_OPEN_DELIVERY_SEL_SUM_OL,
TPCC_ERROR_FETCH_DELIVERY_SEL_SUM_OL,
TPCC_ERROR_EXECUTE_DELIVERY_EXEC_DVRY,
TPCC_ERROR_SELECT_DELIVERY_ORDER_ID,
TPCC_ERROR_SELECT_DELIVERY_CARRIER_ID,
TPCC_ERROR_SELECT_DELIVERY_BALANCE,

TPCC_ERROR_OPEN_STOCKLEVEL_SEL_OID = 400,
TPCC_ERROR_FETCH_STOCKLEVEL_SEL_OID,
TPCC_ERROR_OPEN_STOCKLEVEL_CNT_SID,
TPCC_ERROR_FETCH_STOCKLEVEL_CNT_SID,
TPCC_ERROR_OPEN_STOCKLEVEL_FIND,
TPCC_ERROR_FETCH_STOCKLEVEL_FIND,
TPCC_ERROR_EXECUTE_STOCKLEVEL_COMMIT,

TPCC_ERROR_OPEN_PAYMENT_CNT_CID = 500,
TPCC_ERROR_FETCH_PAYMENT_CNT_CID,
TPCC_ERROR_OPEN_PAYMENT_SEL_CLAST,
TPCC_ERROR_FETCH_PAYMENT_SEL_CLAST,
TPCC_ERROR_OPEN_PAYMENT_SEL_CID,
TPCC_ERROR_FETCH_PAYMENT_SEL_CID,
TPCC_ERROR_DECL_PAYMENT_SEL_DIST,
TPCC_ERROR_OPEN_PAYMENT_SEL_DIST,

```

```

TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_DIST,
TPCC_ERROR_FETCH_PAYMENT_SEL_DIST,
TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_DIST,
TPCC_ERROR_DECL_PAYMENT_SEL_WARE,
TPCC_ERROR_OPEN_PAYMENT_SEL_WARE,
TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_WARE,
TPCC_ERROR_FETCH_PAYMENT_SEL_WARE,
TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_WARE,
TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_LAST,
TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_ID,
TPCC_ERROR_COMMIT_PAYMENT_UPD_CUST,
TPCC_ERROR_SELECT_PAYMENT_W_YTD,
TPCC_ERROR_SELECT_PAYMENT_D_YTD,
TPCC_ERROR_BEGIN_PAYMENT,
TPCC_ERROR_EXECUTE_PAYMENT_COMMIT,
TPCC_ERROR_PAYMENT_UPD_CUST_BY_NAME,
TPCC_ERROR_PAYMENT_UPD_CUST_BY_ID,
TPCC_ERROR_PAYMENT_UPDATE_DIST,
TPCC_ERROR_PAYMENT_UPDATE_WH,
TPCC_ERROR_PAYMENT_INSERT_HISTORY,
TPCC_ERROR_EXECUTE_PAYMENT_WH_DIST
} tpcc_rc_t;

typedef enum {
    TPCC_DEADLOCK_MSG = 10,
    TPCC_RETRY_MSG
} tpcc_msg_t;

#endif /* __TPCC_DATABUF_H__ */

```

delivery.h

```

#ifdef TRANSARC_delivery_h
#define TRANSARC_delivery_h

#include <trpc/trpc.h>
#include "delivery.h"

#include <encina/c_prologue.h>

#if defined(BUILDDLL)
#define DLLEXPORT __declspec( dllexport )
#else
#define DLLEXPORT extern
#endif

#ifdef ENCINA_STUB_CALLING
#define ENCINA_STUB_CALLING ENCINA_RPC_CALLING
#endif

#define delivery_v1_0_c_ifspec
    delivery_v1_0_c_ifspec
#define delivery_v1_0_s_ifspec
    delivery_v1_0_s_ifspec

typedef struct delivery_v1_0_epv {
void (ENCINA_STUB_CALLING *impTPCCDelivery) (
#ifdef IDL_PROTOTYPES

```

```

    idl_long_int length,
    idl_char *dataP,
    data_header *headerP,
    trpc_status_t *trpcStatus

#endif
);

} delivery_v1_0_epv_t;

DLLEXPORT void ENCINA_STUB_CALLING impTPCCDelivery (
#ifdef IDL_PROTOTYPES
    idl_long_int length,
    idl_char *dataP,
    data_header *headerP,
    trpc_status_t *trpcStatus

#endif
);

trpc_handle_t          ENCINA_CALLING
mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
    mon_handle_t          handle,
    trpc_tranInfo_t      *tranInfoP,
    trpc_ifSpec_t        *ifSpecP
#endif
);

void          ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
    mon_handle_t          handle,
    trpc_handle_t        trpcHandle,
    trpc_tranInfo_t      *tranInfoP,
    trpc_ifSpec_t        *ifSpecP
#endif
);

trpc_handle_t          ENCINA_CALLING
mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
    mon_handle_t          handle,
    trpc_tranInfo_t      *tranInfoP,
    trpc_ifSpec_t        *ifSpecP
#endif
);

void          ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
    mon_handle_t          handle,
    trpc_handle_t        trpcHandle,
    trpc_tranInfo_t      *tranInfoP,
    trpc_ifSpec_t        *ifSpecP
#endif
);

extern delivery_v1_0_epv_t
    delivery_v1_0_client_epv;
extern _delivery_v1_0_epv_t
    delivery_v1_0_manager_epv;
extern rpc_mgr_epv_t
    delivery_v1_0_mgr_epv;

```

```

#include <encina/c_epilogue.h>
#endif /* TRANSARC_delivery_h */

```

dlldata.c

```

/*****
****
DllData file -- generated by MIDL compiler

DO NOT ALTER THIS FILE

This file is regenerated by MIDL on every IDL file
compile.

To completely reconstruct this file, delete it and
rerun MIDL
on all the IDL files in this DLL, specifying this
file for the
/dlldata command line option

****/

#include <rpcproxy.h>

#ifdef __cplusplus
extern "C" {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )

PROXYFILE_LIST_START
/* Start of list */
REFERENCE_PROXY_FILE( tpcc_com_ps ),
/* End of list */
PROXYFILE_LIST_END

DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

#ifdef __cplusplus
} /*extern "C" */
#endif

/* end of generated dlldata file */

```

install.c

```

/* FILE: INSTALL.C
*
* Microsoft
TPC-C Kit Ver. 4.51.000
* Copyright
Microsoft, 2003
* All Rights Reserved
*
* not audited

```

```

*
* PURPOSE: Automated installation
application for TPC-C Web Kit
* Contact: Charles Levine
(clevine@microsoft.com)
*
* Change history:
* 4.20.000 - added COM installation
steps
* 4.50.000 - added IIS6 configuration options
* 4.51.000 - added routines to copy
Visual Studio runtime module (MSVCR70.DLL)
* to
SystemRoot\System32
*/

#include <windows.h>
#include <direct.h>
#include <io.h>
#include <stdlib.h>
#include <stdio.h>
#include <commctrl.h>
#include "..\..\common\src\ReadRegistry.h"
#include <process.h>

#include "resource.h"

#define WM_INITTEXT WM_USER+100

HICON hIcon;
HINSTANCE hInst;

DWORD versionExeMS;
DWORD versionExeLS;
DWORD versionExeMM;
DWORD versionDllMS;
DWORD versionDllLS;

// TPC-C registry settings
TPCCREGISTRYDATA Reg;

static int iPoolThreadLimit;
static int iMaxPoolThreads;
static int iThreadTimeout;
static int iListenBackLog;
static int iAcceptExOutstanding;
static int iUriEnableCache;
static int iUriScavengerPeriod;
static int iMaxConnections;

static int iIISMajorVersion;
static int iNumberOfProcessors;

static int iMaxPhysicalMemory;
//max physical memory in MB
static char szLastFileName[64]; //
last file we worked on (for error reporting)

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT
uMsg, WPARAM wParam, LPARAM lParam);

```

```

BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT
uMsg, WPARAM wParam, LPARAM lParam);
BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam);
BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam);
static void ProcessOK(HWND hwnd,
char *szDllPath, char *szWindowsPath);
static void
ReadRegistrySettings(void);
static void
WriteRegistrySettings(char *szDllPath);
static BOOL RegisterDLL(char
*szFileName);
static int
CopyFiles(HWND hDlg, char *szDllPath, char
*szWindowsPath);
static BOOL GetInstallPath(char
*szDllPath);
static BOOL
GetWindowsInstallPath(char *szWindowsPath);
static void GetVersionInfo(char
*szDLLPath, char *szExePath);
static BOOL
CheckWWWebService(void);
static BOOL
StartWWWebService(void);
static BOOL StopWWWebService(void);
static void UpdateDialog(HWND
hDlg);
static void ConfigureIIS6(HWND
hwnd, HWND hDlg);

SYSTEM_INFO siSysInfo;

BOOL install_com(char *szDllPath);

#include "..\..\common\src\ReadRegistry.cpp"

int WINAPI WinMain( HINSTANCE hInstance, HINSTANCE
hPrevInstance, LPSTR lpCmdLine, int nCmdShow )
{
    int iRc;

    hInst = hInstance;

    InitCommonControls();

    hIcon = LoadIcon(hInstance,
MAKEINTRESOURCE(IDI_ICON1));

    iRc = DialogBox(hInstance,
MAKEINTRESOURCE(IDD_DIALOG4), GetDesktopWindow(),
LicenseDlgProc);
    if ( iRc )
    {
        iRc = DialogBox(hInstance,
MAKEINTRESOURCE(IDD_DIALOG1), GetDesktopWindow(),
MainDlgProc);
        if ( iRc )
        {

```

```

        DialogBoxParam(hInstance,
MAKEINTRESOURCE(IDD_DIALOG2), GetDesktopWindow(),
UpdatedDlgProc, (LPARAM)iRc);
    }
}

DestroyIcon(hIcon);
return 0;
}

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam)
{
    HGLOBAL hRes;
    HRSRC hResInfo;
    BYTE *pSrc, *pDst;
    DWORD dwSize;
    static HFONT hFont;

    switch(uMsg)
    {
        case WM_INITDIALOG:
            hFont = CreateFont(-12,
0, 0, 0, 400, 0, 0, 0, 0, 0, 0, 0, 0, "Arial");
            SendMessage(
GetDlgItem(hwnd, IDR_LICENSE1), WM_SETFONT,
(WPARAM)hFont, MAKELPARAM(0, 0) );
            PostMessage(hwnd,
WM_INITTEXT, (WPARAM)0, (LPARAM)0);
            return TRUE;
        case WM_INITTEXT:
            hResInfo =
FindResource(hInst, MAKEINTRESOURCE(IDR_LICENSE1),
"LICENSE");
            dwSize =
SizeofResource(hInst, hResInfo);
            hRes =
LoadResource(hInst, hResInfo );
            pSrc = (BYTE
*)LockResource(hRes);
            pDst = (unsigned char
*)malloc(dwSize+1);
            if ( pDst )
            {
                memcpy(pDst,
pSrc, dwSize);
            }
            pDst[dwSize]
= 0;

            SetDlgItemText(hwnd, IDC_LICENSE, (const
char *)pDst);
            free(pDst);
            else
            SetDlgItemText(hwnd, IDC_LICENSE, (const
char *)pSrc);
            return TRUE;
        case WM_DESTROY:
            DeleteObject(hFont);

```

```

        return TRUE;
    case WM_COMMAND:
        if ( wParam == IDOK )
            EndDialog(hwnd, TRUE);
        if ( wParam == IDCANCEL
)
            EndDialog(hwnd, FALSE);
        default:
            break;
    }
    return FALSE;
}

BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam)
{
    switch(uMsg)
    {
        case WM_INITDIALOG:
            switch(lParam)
            {
                case 1:
                case 2:

                    SetDlgItemText(hwnd, IDC_RESULTS, "TPC-C
Web Client Installed");

                    break;
            }
            return TRUE;
        case WM_COMMAND:
            if ( wParam == IDOK )
                EndDialog(hwnd, TRUE);
            break;
        default:
            break;
    }
    return FALSE;
}

BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam)
{
    PAINTSTRUCT ps;
    MEMORYSTATUS memoryStatus;
    OSVERSIONINFO VI;
    char szTmp[256];
    static char szDllPath[256];
    static char szWindowsPath[256];
    static char szExePath[256];

    switch(uMsg)
    {
        case WM_INITDIALOG:

```

```

        GlobalMemoryStatus(&memoryStatus);
        iMaxPhysicalMemory =
(memoryStatus.dwTotalPhys/ 1048576);

        if (
GetWindowsInstallPath(szWindowsPath) )
        {
            MessageBox(hwnd, "Error: Cannot determine
Windows System Root.", NULL, MB_ICONSTOP | MB_OK);
            EndDialog(hwnd, FALSE);
            return TRUE;
        }
        if (
GetInstallPath(szDllPath) )
        {
            MessageBox(hwnd, "Error internet service
inetsrv is not installed.", NULL, MB_ICONSTOP |
MB_OK);
            EndDialog(hwnd, FALSE);
            return TRUE;
        }
        // set default values
        ZeroMemory( &Reg,
sizeof(Reg) );

        Reg.dwNumberOfDeliveryThreads = 4;
        Reg.dwMaxConnections =
100;

        Reg.dwMaxPendingDeliveries = 100;
        Reg.eDB_Protocol =
DBLIB;

        Reg.eTxnMon = None;
        strcpy(Reg.szDbServer,
"");
        strcpy(Reg.szDbName,
"tpcc");
        strcpy(Reg.szDbUser,
"sa");
        strcpy(Reg.szDbPassword,
"");

        iPoolThreadLimit =
iMaxPhysicalMemory * 2;

        iThreadTimeout = 86400;
        iListenBackLog = 15;
        iAcceptExOutstanding =
40;

        ReadTPCCRegistrySettings( &Reg );
        ReadRegistrySettings();

```

```

        // copy the hardware
information to the SYSTEM_INFO structure
        GetSystemInfo(&siSysInfo);
        // store the number of
processors on this system
        iNumberOfProcessors =
siSysInfo.dwNumberOfProcessors;

        GetModuleFileName(hInst, szExePath,
sizeof(szExePath));
        GetVersionInfo(szDllPath, szExePath);

        wsprintf(szTmp,
"Version %d.%2d.%3d", versionExeMS, versionExeMM,
versionExeLS);
        SetDlgItemText(hwnd,
IDC_VERSION, szTmp);

        SetDlgItemText(hwnd,
IDC_PATH, szDllPath);

        SetDlgItemText(hwnd,
ED_DB_SERVER, Reg.szDbServer);
        SetDlgItemText(hwnd,
ED_DB_USER_ID, Reg.szDbUser);
        SetDlgItemText(hwnd,
ED_DB_PASSWORD, Reg.szDbPassword);
        SetDlgItemText(hwnd,
ED_DB_NAME, Reg.szDbName);

        SetDlgItemInt(hwnd,
ED_THREADS, Reg.dwNumberOfDeliveryThreads, FALSE);
        SetDlgItemInt(hwnd,
ED_MAXCONNECTION, Reg.dwMaxConnections, FALSE);
        SetDlgItemInt(hwnd,
ED_MAXDELIVERIES, Reg.dwMaxPendingDeliveries, FALSE);
        SetDlgItemInt(hwnd,
ED_IIS_MAX_THREAD_POOL_LIMIT, iPoolThreadLimit,
FALSE);
        SetDlgItemInt(hwnd,
ED_IIS_THREAD_TIMEOUT, iThreadTimeout, FALSE);
        SetDlgItemInt(hwnd,
ED_IIS_LISTEN_BACKLOG, iListenBackLog, FALSE);
        SetDlgItemInt(hwnd,
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,
iAcceptExOutstanding, FALSE);

        CheckDlgButton(hwnd,
IDC_DBLIB, 0);
        CheckDlgButton(hwnd,
IDC_ODBC, 0);
        if ( Reg.eDB_Protocol
== DBLIB )
            CheckDlgButton(hwnd, IDC_DBLIB, 1);
        else
            CheckDlgButton(hwnd, IDC_ODBC, 1);

```

```

// check OS version
level for COM. Must be at least Windows 2000
VI.dwOSVersionInfoSize
= sizeof(VI);
GetVersionEx( &VI );
if (VI.dwMajorVersion <
5)
{
    HWND hDlg =
GetDlgItem( hwnd, IDC_TM_MTS );
    EnableWindow(
hDlg, 0 ); // disable COM option
    if
(Reg.eTxnMon == COM)
        Reg.eTxnMon = None;
}
CheckDlgButton(hwnd,
IDC_TM_NONE, 0);
CheckDlgButton(hwnd,
IDC_TM_TUXEDO, 0);
CheckDlgButton(hwnd,
IDC_TM_MTS, 0);
CheckDlgButton(hwnd,
IDC_TM_ENCINA, 0);
switch (Reg.eTxnMon)
{
case None:
    CheckDlgButton(hwnd, IDC_TM_NONE, 1);
    break;
case TUXEDO:
    CheckDlgButton(hwnd, IDC_TM_TUXEDO, 1);
    break;
case ENCINA:
    CheckDlgButton(hwnd, IDC_TM_ENCINA, 1);
    break;
case COM:
    CheckDlgButton(hwnd, IDC_TM_MTS, 1);
    break;
}
return TRUE;
case WM_PAINT:
    if ( IsIconic(hwnd) )
    {
        BeginPaint( hwnd, &ps );
        DrawIcon( ps.hdc, 0, 0, hIcon );
        EndPaint( hwnd, &ps );
    }
    break;
case WM_COMMAND:

```

```

if ( HIWORD(wParam) ==
BN_CLICKED )
{
    switch(
LOWORD(wParam) )
{
case IDC_DBLIB:
    return TRUE;
case IDC_ODBC:
    return TRUE;
case IDOK:
    ProcessOK(hwnd, szDllPath, szWindowsPath);
    return TRUE;
case IDCANCEL:
    EndDialog(hwnd, FALSE);
    return TRUE;
default:
    return FALSE;
}
}
break;
default:
    break;
}
return FALSE;
}
static void ProcessOK(HWND hwnd, char *szDllPath,
char *szWindowsPath)
{
    int d;
    HWND hDlg;
    int rc;
    BOOL bSvcRunning;
    char szFullName[256];
    char szErrTxt[128];
    // read settings from dialog
    Reg.dwNumberOfDeliveryThreads =
GetDlgItemInt( hwnd, ED_THREADS, &d, FALSE );
    Reg.dwMaxConnections = GetDlgItemInt( hwnd,
ED_MAXCONNECTION, &d, FALSE );
    Reg.dwMaxPendingDeliveries =
GetDlgItemInt( hwnd, ED_MAXDELIVERIES, &d, FALSE );
    GetDlgItemText( hwnd, ED_DB_SERVER,
Reg.szDbServer, sizeof(Reg.szDbServer));

```

```

    GetDlgItemText( hwnd, ED_DB_USER_ID,
Reg.szDbUser, sizeof(Reg.szDbUser));
    GetDlgItemText( hwnd, ED_DB_PASSWORD,
Reg.szDbPassword, sizeof(Reg.szDbPassword));
    GetDlgItemText( hwnd, ED_DB_NAME,
Reg.szDbName, sizeof(Reg.szDbName));
}
if ( IsDlgButtonChecked(hwnd, IDC_DBLIB) )
{
    Reg.eDB_Protocol = DBLIB;
    rc = 1;
}
else if ( IsDlgButtonChecked(hwnd,
IDC_ODBC) )
{
    Reg.eDB_Protocol = ODBC;
    rc = 2;
}
if ( IsDlgButtonChecked(hwnd, IDC_TM_NONE)
)
    Reg.eTxnMon = None;
else if ( IsDlgButtonChecked(hwnd,
IDC_TM_TUXEDO) )
    Reg.eTxnMon = TUXEDO;
else if ( IsDlgButtonChecked(hwnd,
IDC_TM_MTS) )
    Reg.eTxnMon = COM;
else if ( IsDlgButtonChecked(hwnd,
IDC_TM_ENCINA) )
    Reg.eTxnMon = ENCINA;
iPoolThreadLimit = GetDlgItemInt( hwnd,
ED_IIS_MAX_THREAD_POOL_LIMIT, &d, FALSE );
iThreadTimeout = GetDlgItemInt( hwnd,
ED_IIS_THREAD_TIMEOUT, &d, FALSE );
iListenBackLog = GetDlgItemInt( hwnd,
ED_IIS_LISTEN_BACKLOG, &d, FALSE );
iAcceptExOutstanding = GetDlgItemInt( hwnd,
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE, &d, FALSE );
ShowWindow(hwnd, SW_HIDE);
hDlg = CreateDialog(hInst,
MAKEINTRESOURCE(IDD_DIALOG3), hwnd, CopyDlgProc);
ShowWindow(hDlg, SW_SHOWNA);
UpdateDialog(hDlg);
// check to see if the web services are
running
bSvcRunning = CheckWWWWebService();
if ( bSvcRunning )
{
    SetDlgItemText( hDlg, IDC_STATUS,
"Stopping Web Service.");
    SendDlgItemMessage( hDlg,
IDC_PROGRESS1, PBM_STEPIT, 0, 0 );
    UpdateDialog( hDlg );
    StopWWWWebService();
}

```

```

        SendDlgItemMessage(hDlg,
IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);
    }

    // write binaries to inetpub\wwwroot
    rc = CopyFiles(hDlg, szDllPath,
szWindowsPath);
    if ( !rc )
    {
        ShowWindow(hwnd, SW_SHOWNA);
        DestroyWindow(hDlg);
        strcpy( szErrTxt, "Error(s)
occured when creating " );
        strcat( szErrTxt, szLastFileName
);
        MessageBox(hwnd, szErrTxt, NULL,
MB_ICONSTOP | MB_OK);
        EndDialog(hwnd, 0);
        return;
    }

    // while we have the web services shutdown,
check to see if this
    // is IIS6. If it is, then call
ConfigureIIS6
    if ( iIISMajorVersion == 6)
    {
        ConfigureIIS6(hwnd, hDlg);
    }

    //if we stopped service restart it.
    if ( bSvcRunning )
    {
        SetDlgItemText(hDlg, IDC_STATUS,
"Starting Web Service.");
        SendDlgItemMessage(hDlg,
IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);
        StartWWWebService();
    }

    // update registry
    SetDlgItemText(hDlg, IDC_STATUS, "Updating
Registry.");
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);
    WriteRegistrySettings(szDllPath);

    // register com proxy stub
    strcpy(szFullName, szDllPath);
    strcat(szFullName, "tpcc_com_ps.dll");
    if (!RegisterDLL(szFullName))
    {
        ShowWindow(hwnd, SW_SHOWNA);
        DestroyWindow(hDlg);
        strcpy( szErrTxt, "Error occured
when registering " );
        strcat( szErrTxt, szFullName );

```

```

        MessageBox(hwnd, szErrTxt, NULL,
MB_ICONSTOP | MB_OK);
        EndDialog(hwnd, 0);
        return;
    }

    // if using COM
    if (Reg.eTxnMon == COM)
    {
        SetDlgItemText(hDlg, IDC_STATUS,
"Configuring COM.");
        SendDlgItemMessage(hDlg,
IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        if (install_com(szDllPath))
        {
            ShowWindow(hwnd,
SW_SHOWNA);
            DestroyWindow(hDlg);
            strcpy( szErrTxt,
"Error occured when configuring COM settings." );
            MessageBox(hwnd,
szErrTxt, NULL, MB_ICONSTOP | MB_OK);
            EndDialog(hwnd, 0);
            return;
        }

        Sleep(100);

        ShowWindow(hwnd, SW_SHOWNA);
        DestroyWindow(hDlg);
        EndDialog(hwnd, rc);
        return;
    }

static void ReadRegistrySettings(void)
{
    HKEY    hKey;
    DWORD   size;
    DWORD   type;

    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\\Microsoft\\InetStp", 0, KEY_READ, &hKey)
== ERROR_SUCCESS )
    {
        size = sizeof(iIISMajorVersion);
        if ( RegQueryValueEx(hKey,
"MajorVersion", 0, &type, (char *)&iIISMajorVersion,
&size) == ERROR_SUCCESS )
            if ( !iIISMajorVersion

                iIISMajorVersion = 5;
            }

        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Param
eters", 0, KEY_READ, &hKey) == ERROR_SUCCESS )

```

```

        {
            if ( iIISMajorVersion == 6)
            {
                // since IIS6 handles
the pool thread parameters differently, we need to
fill in the dialog
                // with the
MaxPoolThreads rather than PoolThreadLimit
                // for ease of coding,
we are just going to stuff the value into
iPoolThreadLimit
                size = sizeof(iPoolThreadLimit);
                if (
RegQueryValueEx(hKey, "MaxPoolThreads", 0, &type,
(char *)&iPoolThreadLimit, &size) == ERROR_SUCCESS )
                    if ( !iPoolThreadLimit

                        iPoolThreadLimit = iMaxPhysicalMemory * 2;
                    }
                else
                {
                    size =
sizeof(iPoolThreadLimit);
                    if (
RegQueryValueEx(hKey, "MaxPoolThreads", 0, &type,
(char *)&iPoolThreadLimit, &size) == ERROR_SUCCESS )
                        if ( !iPoolThreadLimit

                            iPoolThreadLimit = iMaxPhysicalMemory * 2;
                        }

                    size = sizeof(iThreadTimeout);
                    if ( RegQueryValueEx(hKey,
"ThreadTimeout", 0, &type, (char *)&iThreadTimeout,
&size) == ERROR_SUCCESS )
                        if ( !iThreadTimeout )

                            iThreadTimeout = 86400;

                    size = sizeof(iListenBackLog);
                    if ( RegQueryValueEx(hKey,
"ListenBackLog", 0, &type, (char *)&iListenBackLog,
&size) == ERROR_SUCCESS )
                        if ( !iListenBackLog )

                            iListenBackLog = 15;

                    RegCloseKey(hKey);
                }

                if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Paramete
rs", 0, KEY_READ, &hKey) == ERROR_SUCCESS )
                {
                    size =
sizeof(iAcceptExOutstanding);
                    if ( RegQueryValueEx(hKey,
"AcceptExOutstanding", 0, &type, (char
*)&iAcceptExOutstanding, &size) == ERROR_SUCCESS )

```



```

        if (
!iAcceptExOutstanding )
            iAcceptExOutstanding = 40;
            RegCloseKey(hKey);
        }
        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\HTTP\\Parameter
s", 0, KEY_READ, &hKey) == ERROR_SUCCESS )
        {
            size = sizeof(iUriEnableCache);
            if ( RegQueryValueEx(hKey,
"UriEnableCache", 0, &type, (char *)&iUriEnableCache,
&size) == ERROR_SUCCESS )
                if ( !iUriEnableCache )
                    iUriEnableCache = 0;
            size =
sizeof(iUriScavengerPeriod);
            if ( RegQueryValueEx(hKey,
"UriScavengerPeriod", 0, &type, (char
*)&iUriScavengerPeriod, &size) == ERROR_SUCCESS )
                if (
!iUriScavengerPeriod )
                    iUriScavengerPeriod = 10800;
            size = sizeof(iMaxConnections);
            if ( RegQueryValueEx(hKey,
"MaxConnections", 0, &type, (char *)&iMaxConnections,
&size) == ERROR_SUCCESS )
                if ( !iMaxConnections )
                    iMaxConnections = 100000;
            RegCloseKey(hKey);
        }
}

static void WriteRegistrySettings(char *szDllPath)
{
    HKEY    hKey;
    DWORD   dwDisposition;
    char    szTmp[256];
    char    *ptr;
    int     iRc;

    if ( RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\\Microsoft\\TPCC", 0, NULL,
REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey,
&dwDisposition) == ERROR_SUCCESS )
    {
        strcpy(szTmp, szDllPath);
        ptr = strstr(szTmp, "tpcc");
        if ( ptr )
            *ptr = 0;
    }
}

```

```

            RegSetValueEx(hKey, "Path", 0,
REG_SZ, szTmp, strlen(szTmp)+1);

            RegSetValueEx(hKey,
"NumberOfDeliveryThreads", 0, REG_DWORD, (char
*)&Reg.dwNumberOfDeliveryThreads,
sizeof(Reg.dwNumberOfDeliveryThreads));
            RegSetValueEx(hKey,
"MaxConnections", 0, REG_DWORD, (char
*)&Reg.dwMaxConnections,
sizeof(Reg.dwMaxConnections));
            RegSetValueEx(hKey,
"MaxPendingDeliveries", 0, REG_DWORD, (char
*)&Reg.dwMaxPendingDeliveries,
sizeof(Reg.dwMaxPendingDeliveries));

            RegSetValueEx(hKey,
"DB_Protocol", 0, REG_SZ,
szDBNames[Reg.eDB_Protocol],
strlen(szDBNames[Reg.eDB_Protocol])+1);
            RegSetValueEx(hKey, "TxnMonitor",
0, REG_SZ, szTxnMonNames[Reg.eTxnMon],
strlen(szTxnMonNames[Reg.eTxnMon])+1);

            RegSetValueEx(hKey, "DbServer",
0, REG_SZ, Reg.szDbServer, strlen(Reg.szDbServer)+1);
            RegSetValueEx(hKey, "DbName", 0,
REG_SZ, Reg.szDbName, strlen(Reg.szDbName)+1);
            RegSetValueEx(hKey, "DbUser", 0,
REG_SZ, Reg.szDbUser, strlen(Reg.szDbUser)+1);
            RegSetValueEx(hKey, "DbPassword",
0, REG_SZ, Reg.szDbPassword,
strlen(Reg.szDbPassword)+1);

            strcpy(szTmp, "YES");
            RegSetValueEx(hKey,
"COM_SinglePool", 0, REG_SZ, szTmp, strlen(szTmp)+1);

            RegFlushKey(hKey);
            RegCloseKey(hKey);
        }
    }

    if (
(iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Param
eters", 0, NULL, REG_OPTION_NON_VOLATILE,
KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) ==
ERROR_SUCCESS )
    {
        // if this is IIS6, then we need
to treat the PoolThreadLimit differently
        // if IIS6, then PoolThreadLimit
is the maximum number of threads for the entire
system.
        // IIS6 added MaxPoolThreads
which controls the number of threads per processor.
For IIS6
        // we will set MaxPoolThreads to
the value the user provided in the dialog and then
set
    }
}

```

```

        // PoolThreadLimit to
MaxPoolThreads * number of processors on this system
        if ( iiISMajorVersion == 6 )
        {
            iMaxPoolThreads =
iPoolThreadLimit;
            iPoolThreadLimit =
iMaxPoolThreads * iNumberOfProcessors;
            RegSetValueEx(hKey,
"PoolThreadLimit", 0, REG_DWORD, (char
*)&iPoolThreadLimit, sizeof(iPoolThreadLimit));
            RegSetValueEx(hKey,
"MaxPoolThreads", 0, REG_DWORD, (char
*)&iMaxPoolThreads, sizeof(iMaxPoolThreads));
        }
        else
        {
            RegSetValueEx(hKey,
"PoolThreadLimit", 0, REG_DWORD, (char
*)&iPoolThreadLimit, sizeof(iPoolThreadLimit));
            RegSetValueEx(hKey,
"ThreadTimeout", 0, REG_DWORD, (char
*)&iThreadTimeout, sizeof(iThreadTimeout));
            RegSetValueEx(hKey,
"ListenBackLog", 0, REG_DWORD, (char
*)&iListenBackLog, sizeof(iListenBackLog));

            RegFlushKey(hKey);
            RegCloseKey(hKey);
        }
    }

    if (
(iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Paramet
ers", 0, NULL, REG_OPTION_NON_VOLATILE,
KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) ==
ERROR_SUCCESS )
    {
        RegSetValueEx(hKey,
"AcceptExOutstanding", 0, REG_DWORD, (char
*)&iAcceptExOutstanding,
sizeof(iAcceptExOutstanding));

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

    return;
}

BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam)
{
    if ( uMsg == WM_INITDIALOG )
    {
        SendDlgItemMessage(hwnd,
IDC_PROGRESS1, PBM_SETRANGE, 0, MAKELPARAM(0, 16));
        SendDlgItemMessage(hwnd,
IDC_PROGRESS1, PBM_SETSTEP, (WPARAM)1, 0);
    }
}

```

```

        return TRUE;
    }
    return FALSE;
}

BOOL RegisterDLL(char *szFileName)
{
    HINSTANCE hLib;
    FARPROC lpDllEntryPoint;

    hLib = LoadLibrary(szFileName);
    if ( hLib == NULL )
        return FALSE;
    // Find the entry point.
    lpDllEntryPoint = GetProcAddress(hLib,
"DllRegisterServer");
    if (lpDllEntryPoint != NULL)
    {
        return ((*lpDllEntryPoint)() ==
S_OK);
    }
    else
        return FALSE; //unable to
locate entry point
}

BOOL FileFromResource( char *szResourceName, int
iResourceId, char *szDllPath, char *szFileName )
{
    HGLOBAL hDLL;
    HRSRC hResInfo;
    HANDLE hFile;
    DWORD dwSize;
    BYTE *pSrc;
    DWORD d;
    char szFullName[256];

    hResInfo = FindResource(hInst,
MAKEINTRESOURCE(iResourceId), szResourceName);

    strcpy(szFullName, szDllPath);
    strcat(szFullName, szFileName);

    dwSize = SizeofResource(hInst, hResInfo);
    hDLL = LoadResource(hInst, hResInfo);
    pSrc = (BYTE *)LockResource(hDLL);
    remove(szFullName);

    if ( !(hFile = CreateFile(szFullName,
GENERIC_WRITE, 0, NULL, CREATE_ALWAYS,
FILE_ATTRIBUTE_NORMAL, NULL)) )
        return FALSE;

    if ( !WriteFile(hFile, pSrc, dwSize, &d,
NULL) )
        return FALSE;

    CloseHandle(hFile);

    UnlockResource(hDLL);

```

```

FreeResource(hDLL);
return TRUE;
}

static int CopyFiles(HWND hDlg, char *szDllPath, char
*szWindowsPath)
{
    SetDlgItemText(hDlg, IDC_STATUS, "Copying
Files...");
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    // install TPCC.DLL
    strcpy( szLastFileName, "tpcc.dll" );
    if (!FileFromResource( "TPCCDLL",
IDR_TPCCDLL, szDllPath, szLastFileName ))
        return 0;
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    // install MSVCR70.DLL
    strcpy( szLastFileName, "msvcr70.dll" );
    if (!FileFromResource( "MSVCR70",
IDR_MSVCRT701, szWindowsPath, szLastFileName ))
        return 0;
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    // install tpcc_dblib.dll
    strcpy( szLastFileName, "tpcc_dblib.dll" );
    if (!FileFromResource( "DBLIB_DLL",
IDR_DBLIB_DLL, szDllPath, szLastFileName ))
        return 0;
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    // install tpcc_odbc.dll
    strcpy( szLastFileName, "tpcc_odbc.dll" );
    if (!FileFromResource( "ODBC_DLL",
IDR_ODBC_DLL, szDllPath, szLastFileName ))
        return 0;
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    // install tuxapp.exe
    strcpy( szLastFileName, "tuxapp.exe" );
    if (!FileFromResource( "TUXEDO_APP",
IDR_TUXEDO_APP, szDllPath, szLastFileName ))
        return 0;
    //SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    //UpdateDialog(hDlg);

    // install tpcc_tuxedo.dll

```

```

    strcpy( szLastFileName, "tpcc_tuxedo.dll"
);
    if (!FileFromResource( "TUXEDO_DLL",
IDR_TUXEDO_DLL, szDllPath, szLastFileName ))
        return 0;
    //SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    //UpdateDialog(hDlg);

    // install tpcc_com.dll
    strcpy( szLastFileName, "tpcc_com.dll" );
    if (!FileFromResource( "COM_DLL",
IDR_COM_DLL, szDllPath, szLastFileName ))
        return 0;
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    // install tpcc_com_all.tlb
    strcpy( szLastFileName, "tpcc_com_all.tlb"
);
    if (!FileFromResource( "COM_TYPLIB",
IDR_COMTYPLIB_DLL, szDllPath, szLastFileName ))
        return 0;
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    // install tpcc_com_ps.dll
    strcpy( szLastFileName, "tpcc_com_ps.dll"
);
    if (!FileFromResource( "COM_PS_DLL",
IDR_COMPS_DLL, szDllPath, szLastFileName ))
        return 0;
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    // install tpcc_com_all.dll
    strcpy( szLastFileName, "tpcc_com_all.dll"
);
    if (!FileFromResource( "COM_ALL_DLL",
IDR_COMALL_DLL, szDllPath, szLastFileName ))
        return 0;
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    return 1;
}

static BOOL GetInstallPath(char *szDllPath)
{
    HKEY hKey;
    BYTE szData[256];
    DWORD sv;
    BOOL bRc;

```

```

int len;
int iRc;

// Registry key
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InetStp\PathWWW
Root is used to find the
// IIS default web site directory and
determine that IIS is installed.

szDllPath[0] = 0;
bRc = TRUE;
if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\Microsoft\InetStp", 0, KEY_ALL_ACCESS,
&hKey) == ERROR_SUCCESS )
{
    sv = sizeof(szData);
    iRc = RegQueryValueEx( hKey,
"PathWWWRoot", NULL, NULL, szData, &sv ); // used by
IIS 5.0 & 6.0
    if (iRc == ERROR_SUCCESS)
    {
        bRc = FALSE;
        strcpy(szDllPath,
szData);
        len =
strlen(szDllPath);
        if ( szDllPath[len-1]
!= '\\' )
        {
            szDllPath[len] = '\\';
            szDllPath[len+1] = 0;
        }
        RegCloseKey( hKey );
    }
    return bRc;
}

static BOOL GetWindowsInstallPath(char
*szWindowsPath)
{
    HKEY hKey;
    BYTE szData[256];
    DWORD sv;
    BOOL bRc;
    int len;
    int iRc;

    // Registry key
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\SystemRoot is used to find the
// system root to install the VC70 DLL.

    szWindowsPath[0] = 0;
    bRc = TRUE;

```

```

    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\Microsoft\Windows NT\CurrentVersion", 0,
KEY_ALL_ACCESS, &hKey) == ERROR_SUCCESS )
    {
        sv = sizeof(szData);
        iRc = RegQueryValueEx( hKey,
"SystemRoot", NULL, NULL, szData, &sv );
        if (iRc == ERROR_SUCCESS)
        {
            bRc = FALSE;
            strcpy(szWindowsPath,
szData);
            len =
strlen(szWindowsPath);
            if ( szWindowsPath[len-
1] != '\\' )
            {
                szWindowsPath[len] = '\\';
                szWindowsPath[len+1] = 0;
                // now append the path
                strcat(szWindowsPath,
"SYSTEM32\");
            }
            RegCloseKey( hKey );
        }
        return bRc;
    }
    static void GetVersionInfo(char *szDLLPath, char
*szExePath)
    {
        DWORD d;
        DWORD dwSize;
        DWORD dwBytes;
        char *ptr;
        VS_FIXEDFILEINFO *vs;

        versionDllMS = 0;
        versionDllLS = 0;
        if ( _access(szDLLPath, 00) == 0 )
        {
            dwSize =
GetFileVersionInfoSize(szDLLPath, &d);
            if ( dwSize )
            {
                ptr = (char
*)malloc(dwSize);
                GetFileVersionInfo(szDLLPath, 0, dwSize,
ptr);
                VerQueryValue( ptr,
"\",&vs, &dwBytes);

```

```

                versionDllMS = vs-
>dwProductVersionMS;
                versionDllLS = vs-
>dwProductVersionLS;
            }
            versionExeMS = 0x7FFF;
            versionExeLS = 0x7FFF;
            dwSize = GetFileVersionInfoSize(szExePath,
&d);
            if ( dwSize )
            {
                ptr = (char *)malloc(dwSize);
                GetFileVersionInfo(szExePath, 0,
dwSize, ptr);
                VerQueryValue( ptr, "\",&vs,
&dwBytes);
                versionExeMS = vs-
>dwProductVersionMS;
                versionExeLS = LOWORD(vs-
>dwProductVersionLS);
                versionExeMM = HIWORD(vs-
>dwProductVersionLS);
                free(ptr);
            }
            return;
        }
        static BOOL CheckWWWWebService(void)
        {
            SC_HANDLE schSCManager;
            SC_HANDLE schService;
            SERVICE_STATUS ssStatus;

            schSCManager = OpenSCManager(NULL, NULL,
SC_MANAGER_ALL_ACCESS);
            schService = OpenService(schSCManager,
TEXT("W3SVC"), SERVICE_ALL_ACCESS);
            if (schService == NULL)
                return FALSE;

            if (! QueryServiceStatus(schService,
&ssStatus) )
                goto ServiceNotRunning;

            if (! ControlService(schService,
SERVICE_CONTROL_STOP, &ssStatus) )
                goto ServiceNotRunning;
            //start Service pending, Check the status
until the service is running.
            if (! QueryServiceStatus(schService,
&ssStatus) )
                goto ServiceNotRunning;

            CloseServiceHandle(schService);
            return TRUE;
        }
        ServiceNotRunning:

```

```

        CloseServiceHandle(schService);
        return FALSE;
    }

static BOOL StartWWWebService(void)
{
    SC_HANDLE      schSCManager;
    SC_HANDLE      schService;
    SERVICE_STATUS ssStatus;
    DWORD          dwOldCheckPoint;

    schSCManager = OpenSCManager(NULL, NULL,
    SC_MANAGER_ALL_ACCESS);
    schService = OpenService(schSCManager,
    TEXT("W3SVC"), SERVICE_ALL_ACCESS);
    if (schService == NULL)
        return FALSE;

    if (! StartService(schService, 0, NULL) )
        goto StartWWWebErr;
    //start Service pending, Check the status
    until the service is running.
    if (! QueryServiceStatus(schService,
    &ssStatus) )
        goto StartWWWebErr;
    while( ssStatus.dwCurrentState !=
    SERVICE_RUNNING)
    {
        dwOldCheckPoint =
    ssStatus.dwCheckPoint;
        //Save the current checkpoint.
        Sleep(ssStatus.dwWaitHint);

        //Wait for the specified interval.
        if (
    !QueryServiceStatus(schService, &ssStatus) )
            //Check the status again.
            break;
        if (dwOldCheckPoint >=
    ssStatus.dwCheckPoint) //Break if
    the checkpoint has not been incremented.
            break;
    }

    if (ssStatus.dwCurrentState ==
    SERVICE_RUNNING)
        goto StartWWWebErr;

    CloseServiceHandle (schService);
    return TRUE;

StartWWWebErr:
    CloseServiceHandle(schService);
    return FALSE;
}

static BOOL StopWWWebService(void)
{

```

```

    SC_HANDLE      schSCManager;
    SC_HANDLE      schService;
    SERVICE_STATUS ssStatus;
    DWORD          dwOldCheckPoint;

    schSCManager = OpenSCManager(NULL, NULL,
    SC_MANAGER_ALL_ACCESS);
    //schService = OpenService(schSCManager,
    TEXT("W3SVC"), SERVICE_ALL_ACCESS);
    schService = OpenService(schSCManager,
    TEXT("IISADMIN"), SERVICE_ALL_ACCESS);
    if (schService == NULL)
        return FALSE;

    if (! QueryServiceStatus(schService,
    &ssStatus) )
        goto StopWWWebErr;

    if ( !ControlService(schService,
    SERVICE_CONTROL_STOP, &ssStatus) )
        goto StopWWWebErr;
    //start Service pending, Check the status
    until the service is running.
    if (! QueryServiceStatus(schService,
    &ssStatus) )
        goto StopWWWebErr;
    while( ssStatus.dwCurrentState ==
    SERVICE_RUNNING)
    {
        dwOldCheckPoint =
    ssStatus.dwCheckPoint;
        //Save the current checkpoint.
        Sleep(ssStatus.dwWaitHint);

        //Wait for the specified interval.
        if (
    !QueryServiceStatus(schService, &ssStatus) )
            //Check the status again.
            break;
        if (dwOldCheckPoint >=
    ssStatus.dwCheckPoint) //Break if
    the checkpoint has not been incremented.
            break;
    }

    if (ssStatus.dwCurrentState ==
    SERVICE_RUNNING)
        goto StopWWWebErr;

    CloseServiceHandle (schService);
    return TRUE;

StopWWWebErr:
    CloseServiceHandle(schService);
    return FALSE;
}

static void UpdateDialog(HWND hDlg)
{

```

```

    MSG msg;

    UpdateWindow(hDlg);
    while( PeekMessage(&msg, hDlg, 0, 0,
    PM_REMOVE) )
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    Sleep(250);
    return;
}

static void ConfigureIIS6(HWND hwnd, HWND hDlg)
{
    int      irc;
    char     szErrTxt[128];
    FILE     *fErrorFile;

    SetDlgItemText(hDlg, IDC_STATUS,
    "Configuring IIS6...");
    //SendDlgItemMessage(hDlg, IDC_PROGRESS1,
    PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    irc = system("IIS6_CONFIG.CMD");

    // since the return code from the command
    file is always 1,
    // check to see if the file iis6_config.err
    exists
    // if it does, then something hosed
    fErrorFile = fopen("IIS6_CONFIG.err","r");
    if ( fErrorFile != NULL )
    {
        ShowWindow(hwnd, SW_SHOWNA);
        DestroyWindow(hDlg);
        strcpy( szErrTxt, "IIS6
    configuration error." );
        strcat( szErrTxt, "Check
    iis6_config.err" );
        MessageBox(hwnd, szErrTxt, NULL,
    MB_ICONSTOP | MB_OK);
        EndDialog(hwnd, 0);
        return;
    }
}

install.h
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by install.rc
//
#define IDD_DIALOG1          101
#define IDI_ICON1           102
#define IDR_TPCCDLL         103
#define IDD_DIALOG2        105
#define IDI_ICON2           106

```

```

#define IDR_DELIVERY 107
#define IDD_DIALOG3 108

#define BN_LOG
1001
#define ED_KEEP
1002
#define ED_THREADS
1003
#define ED_THREADS2
1004
#define IDC_PATH
1007
#define IDC_VERSION
1009
#define IDC_RESULTS
1010
#define IDC_PROGRESS1
1011
#define IDC_STATUS
1012
#define IDC_BUTTON1
1013
#define ED_MAXCONNECTION
1014
#define ED_IIS_MAX_THREAD_POOL_LIMIT
1015
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE
1017
#define ED_IIS_THREAD_TIMEOUT
1018
#define ED_IIS_LISTEN_BACKLOG
1019
#define IDC_DBLIB
1021
#define IDC_ODBC
1022
#define IDC_CONNECT_POOL
1023
#define ED_USER_CONNECT_DELAY_TIME
1024

// Next default values for new objects
//

install.rc
// Microsoft Visual C++ generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "afxres.h"

```

```

////////////////////////////////////
////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

////////////////////////////////////
////////////////////////////////////
// English (U.S.) resources

#ifdef AFX_RESOURCE_DLL
#ifdef AFX_TARG_ENU
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

////////////////////////////////////
////////////////////////////////////
//
// Dialog
//

IDD_DIALOG1 DIALOGEX 0, 0, 219, 351
STYLE DS_SETFONT | DS_MODALFRAME | DS_CENTER |
WS_MINIMIZEBOX | WS_POPUP |
WS_CAPTION | WS_SYSMENU
CAPTION "TPC-C Web Client Installation Utility"
FONT 8, "MS Sans Serif", 0, 0, 0x1
BEGIN
    EDITTEXT        ED_THREADS,164,45,34,12,ES_RIGHT
    | ES_NUMBER,
    WS_EX_RTLREADING

    EDITTEXT        ED_MAXDELIVERIES,164,59,34,12,ES_RIGHT | ES_NUMBER,
    WS_EX_RTLREADING

    EDITTEXT        ED_MAXCONNECTION,164,73,34,12,ES_RIGHT | ES_NUMBER,
    WS_EX_RTLREADING

    CONTROL
    "None",IDC_TM_NONE,"Button",BS_AUTORADIOBUTTON |
    WS_GROUP |
    WS_TABSTOP,43,100,33,10

    CONTROL
    "COM",IDC_TM_MTS,"Button",BS_AUTORADIOBUTTON |
    WS_TABSTOP,43,113,32,10

    CONTROL
    "TUXEDO",IDC_TM_TUXEDO,"Button",BS_AUTORADIOBUTTON |
    WS_TABSTOP,106,100,46,10

    CONTROL
    "ENCINA",IDC_TM_ENCINA,"Button",BS_AUTORADIOBUTTON |
    WS_DISABLED |
    WS_TABSTOP,106,113,43,10

    EDITTEXT
    ED_DB_SERVER,131,152,67,12,ES_AUTOHSCROLL
    EDITTEXT
    ED_DB_USER_ID,131,165,67,12,ES_AUTOHSCROLL
    EDITTEXT
    ED_DB_PASSWORD,131,178,67,12,ES_AUTOHSCROLL
    EDITTEXT
    ED_DB_NAME,131,191,67,12,ES_AUTOHSCROLL

```

```

CONTROL
"DBLIB",IDC_DBLIB,"Button",BS_AUTORADIOBUTTON |
WS_GROUP |
WS_TABSTOP,45,219,39,12

CONTROL
"ODBC",IDC_ODBC,"Button",BS_AUTORADIOBUTTON |
WS_TABSTOP,
91,219,39,12

EDITTEXT
ED_IIS_MAX_THREAD_POOL_LIMIT,164,263,34,12,ES_RIGHT |
ES_NUMBER,WS_EX_RTLREADING

EDITTEXT
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,164,277,34,12,ES_RI
GHT |
ES_NUMBER,WS_EX_RTLREADING

EDITTEXT
ED_IIS_THREAD_TIMEOUT,164,291,34,12,ES_RIGHT |
ES_NUMBER,
WS_EX_RTLREADING

EDITTEXT
ED_IIS_LISTEN_BACKLOG,164,305,34,12,ES_RIGHT |
ES_NUMBER,
WS_EX_RTLREADING

DEFPUSHBUTTON "OK",IDOK,53,331,50,14
PUSHBUTTON "Cancel",IDCANCEL,119,331,50,14
EDITTEXT
IDC_PATH,106,26,91,13,ES_AUTOHSCROLL | ES_READONLY
LTEXT "Number of Delivery
Threads:",IDC_STATIC,35,45,115,12
LTEXT "Max Number of
Connections:",IDC_STATIC,35,73,115,12
RTEXT "Version
4.11",IDC_VERSION,120,4,89,9
LTEXT "IIS Max Thread Pool
Limit:",IDC_STATIC,36,263,115,12
LTEXT "Web Service Backlog Queue
Size:",IDC_STATIC,36,277,115,
12
LTEXT "IIS Thread Timeout
(seconds):",IDC_STATIC,36,291,115,12
LTEXT "IIS Listen
Backlog:",IDC_STATIC,36,307,115,10
GROUPBOX "Database
Interface",IDC_STATIC,35,208,163,27,WS_GROUP
LTEXT "Installation
directory:",IDC_STATIC,35,29,71,10
GROUPBOX "Transaction
Monitor",IDC_STATIC,33,90,165,37
LTEXT "Server
Name:",IDC_STATIC,35,155,56,8
LTEXT "User ID:",IDC_STATIC,35,168,60,8
LTEXT "User
Password:",IDC_STATIC,35,181,83,8
LTEXT "Database
Name:",IDC_STATIC,35,194,54,8
GROUPBOX "SQL Server Connection
Properties",IDC_STATIC,22,139,187,
102
GROUPBOX "Web Client
Properties",IDC_STATIC,22,15,187,118

```

```

GROUPBOX "IIS
Settings", IDC_STATIC, 22, 247, 187, 79
LTEXT "Max Pending
Deliveries:", IDC_STATIC, 35, 59, 115, 12
END

IDD_DIALOG2 DIALOGEX 0, 0, 117, 62
STYLE DS_SETFONT | DS_SETFOREGROUND | DS_3DLOOK |
DS_CENTER | WS_POPUP |
WS_BORDER
EXSTYLE WS_EX_STATICEDGE
FONT 12, "MS Sans Serif", 0, 0, 0x1
BEGIN
DEFPUSHBUTTON "OK", IDOK, 33, 45, 50, 9
CTEXT "HTML TPC-C Installation
Successful", IDC_RESULTS, 7, 22,
102, 18, 0, WS_EX_CLIENTEDGE
ICON
IDI_ICON2, IDC_STATIC, 50, 7, 18, 20, SS_REALSIZEIMAGE,
WS_EX_TRANSPARENT
END

IDD_DIALOG3 DIALOG 0, 0, 91, 40
STYLE DS_SYSMODAL | DS_SETFONT | DS_MODALFRAME |
DS_3DLOOK | DS_CENTER |
WS_CAPTION
CAPTION "Installing TPC-C Web Client"
FONT 12, "Arial Black"
BEGIN
CONTROL
"Progress1", IDC_PROGRESS1, "msctls_progress32", WS_BORD
ER,
7, 20, 77, 13
CTEXT
"Static", IDC_STATUS, 7, 7, 77, 12, SS_SUNKEN
END

IDD_DIALOG4 DIALOG 0, 0, 291, 202
STYLE DS_SETFONT | DS_MODALFRAME | DS_CENTER |
WS_POPUP | WS_CAPTION |
WS_SYSMENU
CAPTION "Client End User License"
FONT 8, "MS Sans Serif"
BEGIN
EDITTEXT
IDC_LICENSE, 7, 7, 271, 167, ES_MULTILINE | ES_AUTOVSCROLL
|
ES_AUTOHSCROLL | ES_READONLY |
WS_VSCROLL | WS_HSCROLL
DEFPUSHBUTTON "I &Agree", IDOK, 87, 181, 50, 14
PUSHBUTTON "&Cancel", IDCANCEL, 153, 181, 50, 14
END

#####
//
// DESIGNINFO
//
#ifdef APSTUDIO_INVOKED

```

```

GUIDELINES DESIGNINFO
BEGIN
IDD_DIALOG1, DIALOG
BEGIN
LEFTMARGIN, 22
RIGHTMARGIN, 209
VERTGUIDE, 35
VERTGUIDE, 198
TOPMARGIN, 4
BOTTOMMARGIN, 345
END

IDD_DIALOG2, DIALOG
BEGIN
LEFTMARGIN, 7
RIGHTMARGIN, 109
TOPMARGIN, 7
BOTTOMMARGIN, 54
END

IDD_DIALOG3, DIALOG
BEGIN
LEFTMARGIN, 7
RIGHTMARGIN, 84
TOPMARGIN, 7
BOTTOMMARGIN, 33
END

IDD_DIALOG4, DIALOG
BEGIN
LEFTMARGIN, 7
RIGHTMARGIN, 278
TOPMARGIN, 7
BOTTOMMARGIN, 195
END
END
#endif // APSTUDIO_INVOKED

#ifdef APSTUDIO_INVOKED
#####
//
// TEXTINCLUDE
//
1 TEXTINCLUDE
BEGIN
"resource.h\0"
END

2 TEXTINCLUDE
BEGIN
#include "afxres.h"\r\n"
"\0"
END

3 TEXTINCLUDE
BEGIN
"\r\n"
"\0"

```

```

END
#endif // APSTUDIO_INVOKED

#####
//
// Icon
//
// Icon with lowest ID value placed first to ensure
application icon
// remains consistent on all systems.
IDI_ICON1 ICON
"icon1.ico"
IDI_ICON2 ICON
"icon2.ico"

#####
//
// TPCCDLL
//
IDR_TPCCDLL TPCCDLL
"..\\..\\sapi_dll\\bin\\tpcc.dll"

#####
//
// Version
//
VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,4,50,0
PRODUCTVERSION 0,4,50,0
FILEFLAGSMASK 0x3fL
#ifdef DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
#endef
FILEOS 0x40004L
FILETYPE 0x1L
FILESUBTYPE 0x0L
BEGIN
BLOCK "StringFileInfo"
BEGIN
BLOCK "040904b0"
BEGIN
VALUE "Comments", "TPC-C Web Client
Installer"
VALUE "CompanyName", "Microsoft"
VALUE "FileDescription", "install"
VALUE "FileVersion", "0,4,20,0"
VALUE "InternalName", "install"
VALUE "LegalCopyright", "Copyright ©
1999"
VALUE "OriginalFilename", "install.exe"
VALUE "ProductName", "Microsoft install"

```

```

        VALUE "ProductVersion", "0, 4, 20, 0"
    END
END
BLOCK "VarFileInfo"
BEGIN
    VALUE "Translation", 0x409, 1200
END
END

////////////////////////////////////
//
// LICENSE
//
IDR_LICENSE1          LICENSE
"license.txt"

////////////////////////////////////
//
// DBLIB_DLL
//
IDR_DBLIB_DLL         DBLIB_DLL
"..\\..\\db_dblib_dll\\bin\\Release\\tpcc_dblib.dll"

////////////////////////////////////
//
// ODBC_DLL
//
IDR_ODBC_DLL         ODBC_DLL
"..\\..\\db_odbc_dll\\bin\\Release\\tpcc_odbc.dll"

////////////////////////////////////
//
// TUXEDO_APP
//
IDR_TUXEDO_APP        TUXEDO_APP
"..\\..\\tuxapp\\bin\\tuxapp.exe"

////////////////////////////////////
//
// TUXEDO_DLL
//
IDR_TUXEDO_DLL        TUXEDO_DLL
"..\\..\\tm_tuxedo_dll\\bin\\tpcc_tuxedo.dll"

////////////////////////////////////
//
// COM_DLL
//

```

```

IDR_COM_DLL           COM_DLL
"..\\..\\tm_com_dll\\bin\\tpcc_com.dll"

////////////////////////////////////
//
// COM_PS_DLL
//
IDR_COMPS_DLL        COM_PS_DLL
"..\\..\\tpcc_com_ps\\bin\\tpcc_com_ps.dll"

////////////////////////////////////
//
// COM_ALL_DLL
//
IDR_COMALL_DLL       COM_ALL_DLL
"..\\..\\tpcc_com_all\\bin\\tpcc_com_all.dll"

////////////////////////////////////
//
// COM_TYPLIB
//
IDR_COMTYPLIB_DLL   COM_TYPLIB
"..\\..\\tpcc_com_all\\src\\tpcc_com_all.tlb"

////////////////////////////////////
//
// MSVCRT70
//
IDR_MSVCRT70        MSVCRT70
"C:\\WINDOWS\\system32\\msvcr70.dll"
#endif // English (U.S.) resources

////////////////////////////////////
//
//
// Generated from the TEXTINCLUDE 3 resource.
//

////////////////////////////////////
//
// not APSTUDIO_INVOKED
//

install_com.cpp
/* FILE: INSTALL_COM.CPP

```

```

*
* Microsoft
* TPC-C Kit Ver. 4.51.000
* Copyright
* Microsoft, 1999
* All Rights Reserved
*
* not audited
*
* PURPOSE: installation code for COM
application for TPC-C Web Kit
* Contact: Charles Levine
(clevine@microsoft.com)
*
* Change history:
* 4.20.000 - first version
*/

#define WIN32_WINNT 0x0500

#include <comdef.h>
#include <comadmin.h>
#include <stdio.h>
#include <tchar.h>

extern "C"
{
    BOOL install_com(char *szDllPath);
}

BOOL install_com(char *szDllPath)
{
    ICOMAdminCatalog* pCOMAdminCat = NULL;
    ICatalogCollection* pCatalogCollectionApp
= NULL;
    ICatalogCollection* pCatalogCollectionCo
= NULL;
    ICatalogCollection* pCatalogCollectionItf
= NULL;
    ICatalogCollection*
pCatalogCollectionMethod = NULL;

    ICatalogObject*
pCatalogObjectApp = NULL;
    ICatalogObject*
pCatalogObjectCo = NULL;
    ICatalogObject*
pCatalogObjectItf = NULL;
    ICatalogObject*
pCatalogObjectMethod = NULL;

    _bstr_t
bstrTemp, bstrTemp2, bstrTemp3, bstrTemp4;
    _bstr_t
bstrDllPath = szDllPath;
    _variant_t
vTmp, vKey;
    long
lActProp, lCount, lCountCo, lCountItf,
lCountMethod;
    bool
bTmp;

```

```

CoInitializeEx(NULL, COINIT_MULTITHREADED);

HRESULT hr =
CoCreateInstance(CLSID_COMAdminCatalog,
                NULL,
                CLSCTX_INPROC_SERVER,
                IID_ICOMAdminCatalog,
                (void**)
                &pCOMAdminCat);

if (!SUCCEEDED(hr)) goto Error;

bstrTemp = "Applications";

// Attempt to connect to "Applications" in
the Catalog
hr = pCOMAdminCat->GetCollection(bstrTemp,
                                (IDispatch**)
                                &pCatalogCollectionApp);
if (!SUCCEEDED(hr)) goto Error;

// Attempt to load the "Applications"
collection
hr = pCatalogCollectionApp->Populate();
if (!SUCCEEDED(hr)) goto Error;

hr = pCatalogCollectionApp-
>get_Count(&lCount);
if (!SUCCEEDED(hr)) goto Error;

// iterate through applications to delete
existing "TPC-C" application (if any)
while (lCount > 0)
{
    hr = pCatalogCollectionApp-
>get_Item(lCount - 1, (IDispatch**)
    &pCatalogObjectApp);
    if (!SUCCEEDED(hr)) goto Error;

    hr = pCatalogObjectApp-
>get_Name(&vTmp);
    if (!SUCCEEDED(hr)) goto Error;

    if (wcsncmp(vTmp.bstrVal, L"TPC-
C"))
    {
        lCount--;
        continue;
    }
    else
    {

```

```

        hr =
pCatalogCollectionApp->Remove(lCount - 1);
        if (!SUCCEEDED(hr))
        goto Error;
    }
}

hr = pCatalogCollectionApp-
>SaveChanges(&lActProp);
if (!SUCCEEDED(hr)) goto Error;

// add the new application
hr = pCatalogCollectionApp-
>Add((IDispatch**) &pCatalogObjectApp);
if (!SUCCEEDED(hr)) goto Error;

// set properties
bstrTemp = "Name";
vTmp = "TPC-C";
hr = pCatalogObjectApp->put_Value(bstrTemp,
vTmp);
if (!SUCCEEDED(hr)) goto Error;

// set as a library (in process)
application
bstrTemp = "Activation";
lActProp = COMAdminActivationInproc;
vTmp = lActProp;
hr = pCatalogObjectApp->put_Value(bstrTemp,
vTmp);
if (!SUCCEEDED(hr)) goto Error;

// set security level to process
bstrTemp = "AccessChecksLevel";
lActProp =
COMAdminAccessChecksApplicationLevel;
vTmp = lActProp;
hr = pCatalogObjectApp->put_Value(bstrTemp,
vTmp);
if (!SUCCEEDED(hr)) goto Error;

// save key to get the Components
collection later
hr = pCatalogObjectApp->get_Key(&vKey);
if (!SUCCEEDED(hr)) goto Error;

// save changes (app creation) so component
installation will work
hr = pCatalogCollectionApp-
>SaveChanges(&lActProp);
if (!SUCCEEDED(hr)) goto Error;

pCatalogObjectApp->Release();
pCatalogObjectApp = NULL;

bstrTemp = "TPC-C";
// app name
bstrTemp2 = bstrDllPath +
"tpcc_com_all.dll"; // DLL

```

```

        bstrTemp3 = bstrDllPath +
"tpcc_com_all.tlb"; // type library (TLB)
        bstrTemp4 = bstrDllPath +
"tpcc_com_ps.dll"; // proxy/stub dll

        hr = pCOMAdminCat-
>InstallComponent(bstrTemp,

        bstrTemp2,

        bstrTemp3,

        bstrTemp4);
        if (!SUCCEEDED(hr)) goto Error;

        bstrTemp = "Components";
        hr = pCatalogCollectionApp-
>GetCollection(bstrTemp, vKey, (IDispatch**)
        &pCatalogCollectionCo);
        if (!SUCCEEDED(hr)) goto Error;

        hr = pCatalogCollectionCo->Populate();
        if (!SUCCEEDED(hr)) goto Error;

        hr = pCatalogCollectionCo-
>get_Count(&lCountCo);
        if (!SUCCEEDED(hr)) goto Error;

        // iterate through components in
application and set the properties
        while (lCountCo > 0)
        {
            hr = pCatalogCollectionCo-
>get_Item(lCountCo - 1, (IDispatch**)
            &pCatalogObjectCo);
            if (!SUCCEEDED(hr)) goto Error;

            // used for debugging (view the
name)
            hr = pCatalogObjectCo-
>get_Name(&vTmp);
            if (!SUCCEEDED(hr)) goto Error;

            bstrTemp = "ConstructionEnabled";
            bTmp = TRUE;
            vTmp = bTmp;
            hr = pCatalogObjectCo-
>put_Value(bstrTemp, vTmp);
            if (!SUCCEEDED(hr)) goto Error;

            bstrTemp = "ConstructorString";
            bstrTemp2 = "dummy string (do not
remove)";
            vTmp = bstrTemp2;
            hr = pCatalogObjectCo-
>put_Value(bstrTemp, vTmp);
            if (!SUCCEEDED(hr)) goto Error;

```


<pre> bstrTemp = "JustInTimeActivation"; bTmp = TRUE; vTmp = bTmp; hr = pCatalogObjectCo- >put_Value(bstrTemp, vTmp); if (!SUCCEEDED(hr)) goto Error; bstrTemp = "MaxPoolSize"; vTmp.Clear(); // clear variant so it isn't stored as a bool (_variant_t feature) vTmp = (long)30; hr = pCatalogObjectCo- >put_Value(bstrTemp, vTmp); if (!SUCCEEDED(hr)) goto Error; bstrTemp = "ObjectPoolingEnabled"; bTmp = TRUE; vTmp = bTmp; hr = pCatalogObjectCo- >put_Value(bstrTemp, vTmp); if (!SUCCEEDED(hr)) goto Error; // save key to get the InterfacesForComponent collection hr = pCatalogObjectCo- >get_Key(&vKey); if (!SUCCEEDED(hr)) goto Error; bstrTemp = "InterfacesForComponent"; hr = pCatalogCollectionCo- >GetCollection(bstrTemp, vKey, (IDispatch**) &pCatalogCollectionItf); if (!SUCCEEDED(hr)) goto Error; hr = pCatalogCollectionItf- >Populate(); if (!SUCCEEDED(hr)) goto Error; hr = pCatalogCollectionItf- >get_Count(&lCountItf); if (!SUCCEEDED(hr)) goto Error; // iterate through interfaces in component while (lCountItf > 0) { hr = pCatalogCollectionItf->get_Item(lCountItf - 1, (IDispatch**) &pCatalogObjectItf); if (!SUCCEEDED(hr)) goto Error; // save key to get the MethodsForInterface collection </pre>	<pre> hr = pCatalogObjectItf- >get_Key(&vKey); if (!SUCCEEDED(hr)) goto Error; bstrTemp = "MethodsForInterface"; hr = pCatalogCollectionItf->GetCollection(bstrTemp, vKey, (IDispatch**) &pCatalogCollectionMethod); if (!SUCCEEDED(hr)) goto Error; hr = pCatalogCollectionMethod->Populate(); if (!SUCCEEDED(hr)) goto Error; hr = pCatalogCollectionMethod->get_Count(&lCountMethod); if (!SUCCEEDED(hr)) goto Error; // iterate through methods of interface while (lCountMethod > 0) { hr = pCatalogCollectionMethod->get_Item(lCountMethod - 1, (IDispatch**) &pCatalogObjectMethod); if (!SUCCEEDED(hr)) goto Error; bstrTemp = "AutoComplete"; bTmp = TRUE; vTmp = bTmp; hr = pCatalogObjectMethod->put_Value(bstrTemp, vTmp); if (!SUCCEEDED(hr)) goto Error; pCatalogObjectMethod->Release(); pCatalogObjectMethod = NULL; lCountMethod- -; } // save changes hr = pCatalogCollectionMethod->SaveChanges(&lActProp); if (!SUCCEEDED(hr)) goto Error; pCatalogObjectItf- >Release(); </pre>	<pre> pCatalogObjectItf = NULL; lCountItf--; } pCatalogObjectCo->Release(); pCatalogObjectCo = NULL; lCountCo--; } // save changes hr = pCatalogCollectionCo- >SaveChanges(&lActProp); if (!SUCCEEDED(hr)) goto Error; pCatalogCollectionApp->Release(); pCatalogCollectionApp = NULL; pCatalogCollectionCo->Release(); pCatalogCollectionCo = NULL; pCatalogCollectionItf->Release(); pCatalogCollectionItf = NULL; pCatalogCollectionMethod->Release(); pCatalogCollectionMethod = NULL; Error: CoUninitialize(); if (!SUCCEEDED(hr)) { LPTSTR lpBuf; DWORD dwRes = FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER FORMAT_MESSAGE_FROM_SYSTEM, NULL, hr, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT), (LPTSTR) &lpBuf, 0, NULL); // _tprintf(_T("Error adding components. HRESULT: 0x%x\n%s"), hr, lpBuf); return TRUE; } else </pre>
--	---	---

```
}  
    return FALSE;  
}
```

license.txt

END-USER LICENSE AGREEMENT FOR
MICROSOFT TPC-C BENCHMARK KIT

IMPORTANT READ CAREFULLY: This Microsoft End-User License Agreement (EULA) is a legal agreement between you (either an individual or a single entity) and Microsoft Corporation for the Microsoft software product identified above, which includes computer software and may include associated media, printed materials, and online or electronic documentation (SOFTWARE PRODUCT). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this Agreement, you are not authorized to use the SOFTWARE PRODUCT.

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1. GRANT OF LICENSE. This EULA grants you the following rights:

Use. Microsoft grants to you the right to install and use copies of the SOFTWARE PRODUCT only in conjunction with validly licensed copies of Microsoft SQL Server and/or Microsoft Windows NT Server software. You may also make copies of the SOFTWARE PRODUCT for backup and archival purposes.

2. RESTRICTIONS.

--You must maintain all copyright notices on all copies of the SOFTWARE PRODUCT.

--You may not distribute copies of the SOFTWARE PRODUCT to third parties.

--You may not rent, lease or lend the SOFTWARE PRODUCT.

--You may not use the SOFTWARE PRODUCT or any derivative works thereof to internally test database management system software other than Microsoft SQL Server and/or operating system software other than Microsoft Windows NT.

-- You may not disclose the results of any benchmark tests

using the SOFTWARE PRODUCT to any third party without Microsoft's prior written approval.

-- You may not disclose or provide the SOFTWARE PRODUCT or any derivative works thereof, or any information relating to the SOFTWARE PRODUCT (including the existence of the SOFTWARE PRODUCT or the results of use and testing or benchmark testing), to any third party without Microsoft's written permission.

3. TERMINATION. Without prejudice to any other rights, Microsoft may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PRODUCT.

4. COPYRIGHT. All title and copyrights in and to the SOFTWARE PRODUCT and any copies thereof are owned by Microsoft or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of the SOFTWARE PRODUCT is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content.

5. UPGRADES. If the SOFTWARE PRODUCT is labeled as an upgrade, you must be properly licensed to use a product identified by Microsoft as being eligible for the upgrade in order to use the SOFTWARE PRODUCT. A SOFTWARE PRODUCT labeled as an upgrade replaces and/or supplements the product that formed the basis for your eligibility for the upgrade. You may use the resulting upgraded product only in accordance with the terms of this EULA.

6. U.S. GOVERNMENT RESTRICTED RIGHTS.

The SOFTWARE PRODUCT is provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software Restricted Rights at 48 CFR

52.227-19, as applicable. Manufacturer is Microsoft Corporation/One Microsoft Way/Redmond, WA 98052-6399.

7. EXPORT RESTRICTIONS.

You agree that you will not export or re-export the SOFTWARE PRODUCT to any country, person, entity or end user subject to U.S.A. export restrictions. Restricted countries currently include, but are not necessarily limited to Cuba, Iran, Iraq, Libya, North Korea, Syria, and the Federal Republic of Yugoslavia (Serbia and Montenegro, U.N. Protected Areas and areas of Republic of Bosnia and Herzegovina under the control of Bosnian Serb forces). You warrant and represent that neither the U.S.A. Bureau of Export Administration nor any other federal agency has suspended, revoked or denied your export privileges.

8. NO WARRANTY. ANY USE OF THE SOFTWARE PRODUCT IS AT YOUR OWN RISK. THE SOFTWARE PRODUCT IS PROVIDED FOR USE ONLY WITH MICROSOFT SQL SERVER AND/OR MICROSOFT WINDOWS NT SERVER SOFTWARE. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, MICROSOFT AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT.

9. NO LIABILITY FOR CONSEQUENTIAL DAMAGES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL MICROSOFT OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE PRODUCT, EVEN IF MICROSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

10. LIMITATION OF LIABILITY. MICROSOFT'S ENTIRE LIABILITY AND YOUR EXCLUSIVE REMEDY UNDER THIS EULA SHALL NOT EXCEED FIVE DOLLARS (US\$5.00).

11. MISCELLANEOUS

This EULA is governed by the laws of the State of Washington, U.S.A.

Should you have any questions concerning this EULA, or if you desire to contact Microsoft for any reason, please contact the Microsoft subsidiary serving your country, or write: Microsoft Sales Information Center/One Microsoft Way/Redmond, WA 98052-6399.

Si vous avez acquis votre produit Microsoft au CANADA, la garantie limitée suivante vous concerne:

EXCLUSION DE GARANTIES. Microsoft renonce entièrement toute garantie pour le LOGICIEL. Le LOGICIEL et toute autre documentation s'y rapportant sont fournis ® comme tels sans aucune garantie quelle qu'elle soit, expresse ou implicite, y compris, mais ne se limitant pas aux garanties implicites de la qualité, marchande ou un usage particulier. Le risque total découlant de l'utilisation ou de la performance du LOGICIEL est entre vos mains.

RESPONSABILITÉ LIMITÉE. La seule obligation de Microsoft et votre recours exclusif concernant ce contrat n'excéderont pas cinq dollars (US\$5.00).

ABSENCE DE RESPONSABILITÉ POUR LES DOMMAGES INDIRECTS. Microsoft ou ses fournisseurs ne pourront être tenus responsables en aucune circonstance de tout dommage quel qu'il soit (y compris mais non de façon limitative les dommages directs ou indirects causés par la perte de bénéfices commerciaux, l'interruption des affaires, la perte d'information commerciale ou toute autre perte pécuniaire) résultant de l'utilisation ou de l'impossibilité d'utilisation de ce produit, et ce, même si la société, Microsoft a, à l'avance, avisé de l'éventualité de tels dommages. Certains États/juridictions ne permettent pas l'exclusion ou la limitation de responsabilité relative aux dommages indirects ou consécutifs, et la limitation ci-dessus peut ne pas s'appliquer dans votre pays. La présente Convention est régie par les lois de la province d'Ontario, Canada. Chacune des parties reconnaît irrévocablement la

compétence des tribunaux de la province d'Ontario et consent à instituer tout litige qui pourrait découler de la présente auprès des tribunaux situés dans le district judiciaire de York, province d'Ontario. Au cas où vous auriez des questions concernant cette licence ou que vous désiriez vous mettre en rapport avec Microsoft pour quelque raison que ce soit, veuillez contacter la succursale Microsoft desservant votre pays, dont l'adresse est fournie dans ce produit, ou écrire à: Microsoft Customer Sales and Service, One Microsoft Way, Redmond, Washington 98052 6399.

Methods.h

```

/* FILE: METHODS.H Microsoft
 * TPC-C Kit Ver. 4.20.000 Copyright
 * Microsoft, 1999
 * All Rights Reserved
 * not yet
 * audited
 * PURPOSE: Header file for COM components.
 * Change history:
 * 4.20.000 - first version
 */

enum COMPONENT_ERROR
{
    ERR_MISSING_REGISTRY_ENTRIES = 1,
    ERR_LOADDLL_FAILED,
    ERR_GETPROCADDR_FAILED,
    ERR_UNKNOWN_DB_PROTOCOL
};

class CCOMPONENT_ERR : public CBaseErr
{
public:
    CCOMPONENT_ERR(COMPONENT_ERROR
Err)
    {
        m_Error = Err;
        m_szTextDetail = NULL;
        m_SystemErr = 0;
        m_szErrorText = NULL;
    };
};

```

```

        CCOMPONENT_ERR(COMPONENT_ERROR
Err, char *szTextDetail, DWORD dwSystemErr)
    {
        m_Error = Err;
        m_szTextDetail = new
char[strlen(szTextDetail)+1];
        strcpy(m_szTextDetail,
szTextDetail);
        m_SystemErr =
dwSystemErr;
        m_szErrorText = NULL;
    };
~CCOMPONENT_ERR()
{
    if (m_szTextDetail !=
NULL)
        delete []
m_szTextDetail;
    if (m_szErrorText !=
NULL)
        delete []
m_szErrorText;
};

    COMPONENT_ERROR m_Error;
    char
    *m_szTextDetail;
    char
    *m_szErrorText;
    DWORD
    m_SystemErr;

    int ErrorType() {return
ERR_TYPE_COMPONENT;};
    char *ErrorTypeStr() { return
"COMPONENT"; }
    int ErrorNum() {return m_Error;};
    char *ErrorText();

    static void WriteMessageToEventLog(LPTSTR lpszMsg);

////////////////////////////////////
////////////////////////////////////
// CTPCC_Common
class CTPCC_Common :
    public ITPCC,
    public IObjectControl,
    public IObjectConstruct,
    public
CComObjectRootEx<CComSingleThreadModel>
{
public:
    BEGIN_COM_MAP(CTPCC_Common)
        COM_INTERFACE_ENTRY(ITPCC)
        COM_INTERFACE_ENTRY(IObjectControl)
        COM_INTERFACE_ENTRY(IObjectConstruct)
    END_COM_MAP()
};

```

```

CTPCC_Common();
~CTPCC_Common();

// ITPCC
public:
    HRESULT __stdcall NewOrder(
        VARIANT txn_in, VARIANT* txn_out);
    HRESULT __stdcall Payment(
        VARIANT txn_in, VARIANT* txn_out);
    HRESULT __stdcall Delivery(
        VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall StockLevel( VARIANT
txn_in, VARIANT* txn_out);
    HRESULT __stdcall OrderStatus(
        VARIANT txn_in, VARIANT* txn_out);

    HRESULT __stdcall CallSetComplete();

// IObjectControl
STDMETHODIMP_(BOOL) CanBePooled() { return
m_bCanBePooled; }
STDMETHODIMP Activate() { return S_OK; }
// we don't support COM Services
transactions (no enlistment)
STDMETHODIMP_(void) Deactivate() { /*
nothing to do */ }

// IObjectConstruct
STDMETHODIMP Construct(IDispatch * pUnk);

// helper methods
private:
    BOOL m_bCanBePooled;
    CTPCC_BASE *m_pTxn;

    struct COM_DATA
    {
        int retval;
        int error;
        union
        {
            NEW_ORDER_DATA
            Payment;
            DELIVERY_DATA
            StockLevel;
            ORDER_STATUS_DATA
            OrderStatus;
        } u;
    };

};

////////////////////////////////////
// CTPCC

```

```

class CTPCC :
    public CTPCC_Common,
    public CComCoClass<CTPCC, &CLSID_TPCC>
{
public:
    DECLARE_REGISTRY_RESOURCEID(IDR_TPCC)

    BEGIN_COM_MAP(CTPCC)
        //COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx<CComSingleThreadModel>)
        COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
    END_COM_MAP()

};

////////////////////////////////////
// CNewOrder
class CNewOrder :
    public CTPCC_Common,
    public CComCoClass<CNewOrder,
&CLSID_NewOrder>
{
public:
    DECLARE_REGISTRY_RESOURCEID(IDR_NEWORDER)

    BEGIN_COM_MAP(CNewOrder)
        // COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
        COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
    END_COM_MAP()

// ITPCC
public:
    // HRESULT __stdcall NewOrder(
    VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall Payment(
    VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall StockLevel( VARIANT
txn_in, VARIANT* txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall OrderStatus(
    VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
};

////////////////////////////////////
// COrderStatus
class COrderStatus :
    public CTPCC_Common,
    public CComCoClass<COrderStatus,
&CLSID_OrderStatus>
{
public:
    DECLARE_REGISTRY_RESOURCEID(IDR_ORDERSTATUS)

```

```

    BEGIN_COM_MAP(COrderStatus)
        // COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
        COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
    END_COM_MAP()

// ITPCC
public:
    HRESULT __stdcall NewOrder(
        VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall Payment(
        VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall StockLevel( VARIANT
txn_in, VARIANT* txn_out) {return E_NOTIMPL;}
    // HRESULT __stdcall OrderStatus(
    VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
};

////////////////////////////////////
// CPayment
class CPayment :
    public CTPCC_Common,
    public CComCoClass<CPayment,
&CLSID_Payment>
{
public:
    DECLARE_REGISTRY_RESOURCEID(IDR_PAYMENT)

    BEGIN_COM_MAP(CPayment)
        // COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
        COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
    END_COM_MAP()

// ITPCC
public:
    HRESULT __stdcall NewOrder(
        VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    // HRESULT __stdcall Payment(
    VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall StockLevel( VARIANT
txn_in, VARIANT* txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall OrderStatus(
        VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
};

////////////////////////////////////
// CStockLevel

```

```

class CStockLevel :
    public CTPCC_Common,
    public CComCoClass<CStockLevel,
&CLSID_StockLevel>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_STOCKLEVEL)

BEGIN_COM_MAP(CStockLevel)
//      COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
        COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
        HRESULT __stdcall NewOrder(
            VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
        HRESULT __stdcall Payment(
            VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
//      HRESULT __stdcall StockLevel( VARIANT
txn_in, VARIANT* txn_out) {return E_NOTIMPL;}
        HRESULT __stdcall OrderStatus(
            VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
};

```

mon_client.c

```

/*
 *      mon_client.c
 *
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <time.h>
#include <pthread.h>
#include <tpm/mon/mon.h>
#include <utils/trace.h>
#include <../include/delivery.h>
#include <../include/neworder.h>
#include <../include/payment.h>
#include <../include/stocklevel.h>
#include <../include/orderstatus.h>
#include <../include/tpcc_type.h>
#include "mon_client.h"
#include "client_utils.h"

extern total_tran_count_t *perfClntDataInit();
static void read_mon_environment(void);

static char *cellName;
static int envRetrieval = 0;
static int useSecurity = FALSE;

```

```

static CRITICAL_SECTION  init_lock;
static total_tran_count_t *pClientInfo=NULL; /*
keep stats for the client process */
static num_active_threads = 0;
static int iStatsFrequency = 1;
FILE *errtpcc;
char *errFile = "C:/temp/tpcc_encina.out";
enc_status_t enc_status;

#define NewOrder_code      NEWO_TRANS
#define Payment_code      PAYMENT_TRANS
#define OrderStatus_code  ORDER_STAT_TRANS
#define Delivery_code     DELIVERY_TRANS
#define StockLevel_code   STOCK_TRANS

#define INT_ENV_VALUE(var, default) \
    (var = getenv(#var) ? atoi(getenv(#var)) : default)

#define PRE_RPC_WORK(headerP, tran, sub_tran) \
    if (iStatsFrequency > 0) \
        pre_rpc(headerP, tran, sub_tran);
\
    else
\
        (headerP->stats = 0;
#define POST_RPC_WORK(headerP, tran) \
    if (iStatsFrequency > 0) \
        post_rpc(headerP, tran)

/* CALTPCC
 * Macro to sends 1 RPC and then handles any errors.
 *
 * The macro takes the name of the RPC (e.g.,
NewOrder)
 * and makes the RPC by calling the appropriate
function
 * (e.g., impTPCCNewOrder).
 */
#define
CALLTPCC(name, length, dataP, header, trpcStatusP)
\
{
\
UTIL_CONCAT(impTPCC, name) (length, dataP, &header, trpcSt
atusP);
\
    if (*(trpcStatusP)) {
\
        char msg[100];
\
        sprintf(msg, "TRPC error during impTPCC%s",
UTIL_STRING(name)); \
        header.returncode = TRPC_ERROR;
\
        encina_error_message(msg, *(trpcStatusP));
\
    } else if ((header.returncode != TPCC_SUCCESS) &&
\
        (header.returncode != INVALID_NEWO)) {
\

```

```

        char msg[100];
\
        sprintf(msg, "App error during impTPCC%s: ",
UTIL_STRING(name)); \
        encina_error_message(msg, header.returncode);
\
    }
\
}
\
}

/*
 * pre_rpc -- For debug purposes
 *
 * Called before an RPC is made.
 * Set the state of the thread and keep track of the
time the RPC is sent.
 * This is used by the Background thread to report
the state of the client.
 */
static void pre_rpc(data_header *headerP,
                    int tran_type,
                    int sub_tran_type)
{
    if (iStatsFrequency < 1) {
        headerP->stats = 0;
    } else {
        int num;
        num = ++ (pClientInfo->tran[tran_type].num);
        headerP->stats = (num % iStatsFrequency==0) ?
1 : 0;
        if (headerP->stats)
            { /* measure the time for RT */
                get_local_time(&headerP->clnt_start);
                headerP->srv_start.sec = 0; /*
initialize the server time */
                headerP->srv_start.usec = 0;
                headerP->srv_end.sec = 0;
                headerP->srv_end.usec = 0;
            }
    }
}

/*
 * post_rpc
 *
 * Called when the RPC returns from the server
 *
 * Keeps track of the client response time and the
server response time
 * as well as the state of the thread. This is used
by the background
 * debug thread to report the state of the client
 */
static void post_rpc(data_header *headerP,
                    int tran_type)
{
    double time_diff;
    int tran_failed;
    struct timeval start_time, end_time;

```

```

        if (headerP->stats)
            get_local_time(&headerP-
>clnt_end);
        else
            return;

        /* Store the info for each client.
        * Note: Since we don't use mutex for performance
        reason, pClientInfo
        * may not be accurate if more than one
        thread work on the same
        * data at a same time. But this should
        give us reasonable info.
        */
        if ((headerP->returncode == TPCC_SUCCESS) ||
            (headerP->returncode == INVALID_NEWO)) {
            tran_failed = 0;
        } else {
            pClientInfo->tran[tran_type].errs ++;
            pClientInfo->errors ++;
            tran_failed = 1;
        }
        if (headerP->stats && tran_type <= MAX_TRAN_TYPE
            && tran_type > 0
            && !tran_failed) {
            /* update total server round trip response
            time */
            start_time.tv_sec = headerP-
            >srv_start.sec;
            start_time.tv_usec = headerP-
            >srv_start.usec;
            end_time.tv_sec = headerP->srv_end.sec;
            end_time.tv_usec = headerP->srv_end.usec;
            time_diff = time_diff_ms(&end_time,
            &start_time);
            pClientInfo->tran[tran_type].RTtotal[1] +=
            time_diff;
            DPRINT(("srv start_time %d.%d, end_time
            %d.%d, time_diff %f\n",
            start_time.tv_sec,
            start_time.tv_usec,
            end_time.tv_sec,
            end_time.tv_usec,
            time_diff));

            /* update total client round trip response
            time */
            start_time.tv_sec = headerP-
            >clnt_start.sec;
            start_time.tv_usec = headerP-
            >clnt_start.usec;
            end_time.tv_sec = headerP->clnt_end.sec;
            end_time.tv_usec = headerP-
            >clnt_end.usec;
            time_diff = time_diff_ms(&end_time,
            &start_time);
            pClientInfo->tran[tran_type].RTtotal[0] +=
            time_diff;
            DPRINT(("clnt start_time %d.%d, end_time
            %d.%d, time_diff %f\n",

```

```

            start_time.tv_sec,
            start_time.tv_usec,
            end_time.tv_sec,
            end_time.tv_usec,
            time_diff));

            /* update num for the number of trans
            which have RT measured */
            pClientInfo->tran[tran_type].RTcount ++;
        }
    }
}

/*
* The following send_*** functions are called from
CTPCC_ENCINA class.
*/
/*
* send_new_order
* Send a new order request to the server
*/
int send_new_order(long length, unsigned char *dataP)
{
    trpc_status_t trpcStatus;
    data_header header;

    PRE_RPC_WORK(&header, NEWO_TRANS, 0);

    CALLTPCC(NewOrder,length,dataP,header,&trpcStatus)
    POST_RPC_WORK(&header, NEWO_TRANS);
    if (header.returncode == INVALID_NEWO)
        return TPCC_SUCCESS;
    else
        return header.returncode;
}

/*
* send_payment
* Send a payment request to the server
*/
int send_payment(long length, unsigned char *dataP)
{
    trpc_status_t trpcStatus;
    data_header header;

    PRE_RPC_WORK(&header, PAYMENT_TRANS, 0);

    CALLTPCC(Payment,length,dataP,header,&trpcStatus);
    POST_RPC_WORK(&header, PAYMENT_TRANS);
    return header.returncode;
}

/*
* send_order_status
* Send a order status request to the server
*/
int send_order_status(long length, unsigned char
*dataP)
{

```

```

    trpc_status_t trpcStatus;
    data_header header;

    PRE_RPC_WORK(&header, ORDER_STAT_TRANS, 0);

    CALLTPCC(OrderStatus,length,dataP,header,&trpcStatus)
    ;
    POST_RPC_WORK(&header, ORDER_STAT_TRANS);
    return header.returncode;
}

/*
* send_delivery
* Send a delivery request to the server
*/
int send_delivery(long length, unsigned char *dataP)
{
    trpc_status_t trpcStatus;
    data_header header;

    PRE_RPC_WORK(&header, DELIVERY_TRANS, 0);

    CALLTPCC(Delivery,length,dataP,header,&trpcStatus);
    POST_RPC_WORK(&header, DELIVERY_TRANS);
    return header.returncode;
}

/*
* send_stock_level
* Send a stock level request to the server
*/
int send_stock_level(long length, unsigned char
*dataP)
{
    trpc_status_t trpcStatus;
    data_header header;

    PRE_RPC_WORK(&header, STOCK_TRANS, 0);

    CALLTPCC(StockLevel,length,dataP,header,&trpcStatus);
    POST_RPC_WORK(&header, STOCK_TRANS);
    return header.returncode;
}

/*
* Enroll the client:
* get the necessary handles.
* This function should be called only once. Use
static var client_enrolled to control it.
*/
void enroll_client()
{
    static char *clientName="tpcc_client";
    unsigned long status ;
    static int client_enrolled = 0;
    unsigned32 client_authnLevel;
    unsigned32 client_authzSvc;
    time_type a_time;
    char err_msg[100];

```

```

MUTEX_INIT(&init_lock);
get_local_time(&a_time);
srand(a_time.sec ^ a_time.usec);

MUTEX_LOCK(&init_lock);
if (client_enrolled) {
    MUTEX_UNLOCK(&init_lock);
    return;
}

/* open output file for tracing */
errtpcc = fopen(errFile, "w");
if (!errtpcc)
{
    sprintf(err_msg, "Cannot open
file %s", errFile);
    CHK_STATUS(1,
ERRROUT_FILE_NOT_FOUND, err_msg);
}

get_time_init();
// initialize the space for perfmon
pClientInfo = perfCntDataInit();
if (pClientInfo == NULL) // in case something
wrong
    pClientInfo =
malloc(sizeof(total_tran_count_t));
memset(pClientInfo, 0,
sizeof(total_tran_count_t));

read_mon_environment();

if (!cellName)
    CHK_STATUS(30, CELL_NAME_UNAVAILABLE,
"ENCINA_TPM_CELL is not set!");

if (useSecurity) {
    client_authnLevel =
rpc_c_protect_level_connect;
    client_authzSvc =
rpc_c_authz_dce;
} else {
    client_authnLevel =
rpc_c_protect_level_none;
    client_authzSvc =
rpc_c_authz_none;
}

if (envRetrieval == 0) {
ENCINA_CALL_RC("mon_RetrieveEnable", mon_RetrieveEnabl
e(FALSE), status);
    CHK_STATUS(status, MON_RETRIEVEENABLE_FAILED,
"mon_RetrieveEnable failed");
}

err_printf("enroll_client: calling mon_InitClient
\n");

```

```

ENCINA_CALL_RC("mon_InitClient", mon_InitClient(client
Name, cellName), status);
    CHK_STATUS(status, MON_INITCLIENT_FAILED,
"mon_InitClient failed");

DPRINT(("mon_SecuritySetDefaults-> authn %d,
authz %d\n",
    client_authnLevel, client_authzSvc));
ENCINA_CALL_RC("mon_SecuritySetDefaults",
    mon_SecuritySetDefaults(client_authnLevel, c
lient_authzSvc),
    status);
    CHK_STATUS(status, MON_SECURITYSET_FAILED,
"mon_SecuritySetDefaults failed");

ENCINA_CALL_RC("mon_SetHandleCacheRefreshInterval",
    mon_SetHandleCacheRefreshInterval(300),
status);
    CHK_STATUS(status, MON_SETREFRESHINTERVAL_FAILED,
"mon_SetHandleCacheRefreshInterval
failed");

{
    dbInfo_data_t data;
    trpc_status_t trpcStatus;
    /* Get DB Info -- currently id does not do
anything
    but it will tell us if there is a server
out there.
    Better to know instead of when all the
terminals
are up and ready
*/
    impTPCCNOInfo(&data, &trpcStatus);
    if (trpcStatus) {
        char msg[100];
        sprintf(msg, "TRPC error during db info
at init.");
        encina_error_message(msg, trpcStatus);
        CHK_STATUS(33, NOINFO_TRPC_ERROR,
"TRPC error during db info at
init");
    }
}

client_enrolled = 1;
MUTEX_UNLOCK(&init_lock);
err_printf("end of enroll_client\n");
}

/*-----*/
/* Read environment paramaters and registry
entries */
/*-----*/

```

```

static void read_mon_environment()
{
    char *env_str;
    char *registryKey =
"SOFTWARE\\TransarcCorporation\\TxTpc";
    HKEY hKey;
    DWORD size;
    DWORD type;
    char szTmp[256];

    cellName = getenv("ENCINA_TPM_CELL");
    CHECK_ENVIRON(cellName, "ENCINA_TPM_CELL");

    if (env_str = getenv("TPCC_ENV_RETRIEVE")) {
        envRetrieval = atoi(env_str);
    }

    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
registryKey, 0, KEY_READ, &hKey) != ERROR_SUCCESS )
        return;

    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "StatsFrequency", 0,
&type, szTmp, &size) == ERROR_SUCCESS)
        iStatsFrequency = atoi(szTmp);

    RegCloseKey(hKey);
}

```

mon_client.h

```

/*
 * mon_client.h
 */

#ifndef MON_CLIENT_H
#define MON_CLIENT_H

#define MUTEX_T CRITICAL_SECTION
#define MUTEX_LOCK(a) EnterCriticalSection(a)
#define MUTEX_UNLOCK(a) LeaveCriticalSection(a)
#define MUTEX_INIT(mut)
InitializeCriticalSection(mut)
#define MUTEX_DESTROY(mut) DeleteCriticalSection(mut)
#define ERRROUT errtpcc

/*initialization status */
#define INIT_SUCCESS 0
#define INIT_FAILED 1
#define CELL_NAME_UNAVAILABLE 2
#define MON_RETRIEVEENABLE_FAILED 3
#define MON_INITCLIENT_FAILED 4
#define MON_SECURITYSET_FAILED 5
#define MON_SETREFRESHINTERVAL_FAILED 6
#define NOINFO_TRPC_ERROR 7
#define ENROLL_CLIENT_EXCEPTION 8
#define ERRROUT_FILE_NOT_FOUND 9
#define LOG_FILE_NOT_FOUND 10
#define TPCC_KEY_NOT_FOUND 11

```

```

#define TERM_ALLOC_FAILED 12

/*
 * Routines and declarations that are common to all
 * clients
 */
#if defined(__cplusplus)
extern "C" {
#endif
int send_new_order(long, unsigned char *);
int send_payment(long, unsigned char *);
int send_order_status(long, unsigned char *);
int send_delivery(long, unsigned char *);
int send_stock_level(long, unsigned char *);
void enroll_client();
#if defined(__cplusplus)
}
#endif
#endif /* MON_CLIENT_H */

```

neworder.h

```

#ifndef TRANSARC_neworder_h
#define TRANSARC_neworder_h

#include <trpc/trpc.h>
#include "_neworder.h"

#include <encina/c_prologue.h>

#if defined(BUILDDLL)
#define DLLEXPORT __declspec( dlllexport )
#else
#define DLLEXPORT extern
#endif

#ifndef ENCINA_STUB_CALLING
#define ENCINA_STUB_CALLING ENCINA_RPC_CALLING
#endif

#define neworder_v1_0_c_ifspec
neworder_v1_0_c_ifspec
#define neworder_v1_0_s_ifspec
neworder_v1_0_s_ifspec

typedef struct neworder_v1_0_epv {
void (ENCINA_STUB_CALLING *impTPCCNewOrder) (
#ifdef IDL_PROTOTYPES

        idl_long_int length,
        idl_char *dataP,
        data_header *headerP,
        trpc_status_t *trpcStatus
#endif
);

void (ENCINA_STUB_CALLING *impTPCCNOInfo) (
#ifdef IDL_PROTOTYPES

```

```

        dbInfo_data_t *dataP,
        trpc_status_t *trpcStatus
#endif
);
} neworder_v1_0_epv_t;

DLLEXPORT void ENCINA_STUB_CALLING impTPCCNewOrder (
#ifdef IDL_PROTOTYPES

        idl_long_int length,
        idl_char *dataP,
        data_header *headerP,
        trpc_status_t *trpcStatus
#endif
);

DLLEXPORT void ENCINA_STUB_CALLING impTPCCNOInfo (
#ifdef IDL_PROTOTYPES

        dbInfo_data_t *dataP,
        trpc_status_t *trpcStatus
#endif
);

trpc_handle_t          ENCINA_CALLING
mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t          handle,
        trpc_tranInfo_t      *tranInfoP,
        trpc_ifSpec_t        *ifSpecP
#endif
);

void          ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t          handle,
        trpc_handle_t        trpcHandle,
        trpc_tranInfo_t      *tranInfoP,
        trpc_ifSpec_t        *ifSpecP
#endif
);

trpc_handle_t          ENCINA_CALLING
mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t          handle,
        trpc_tranInfo_t      *tranInfoP,
        trpc_ifSpec_t        *ifSpecP
#endif
);

void          ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t          handle,
        trpc_handle_t        trpcHandle,
        trpc_tranInfo_t      *tranInfoP,
        trpc_ifSpec_t        *ifSpecP
#endif
);

```

```

extern neworder_v1_0_epv_t
neworder_v1_0_client_epv;
extern _neworder_v1_0_epv_t
neworder_v1_0_manager_epv;
extern rpc_mgr_epv_t
neworder_v1_0_mgr_epv;

#include <encina/c_epilogue.h>
#endif /* TRANSARC_neworder_h */

```

orderstatus.h

```

#ifndef TRANSARC_orderstatus_h
#define TRANSARC_orderstatus_h

#include <trpc/trpc.h>
#include "_orderstatus.h"

#include <encina/c_prologue.h>

#if defined(BUILDDLL)
#define DLLEXPORT __declspec( dlllexport )
#else
#define DLLEXPORT extern
#endif

#ifndef ENCINA_STUB_CALLING
#define ENCINA_STUB_CALLING ENCINA_RPC_CALLING
#endif

#define orderstatus_v1_0_c_ifspec
_orderstatus_v1_0_c_ifspec
#define orderstatus_v1_0_s_ifspec
_orderstatus_v1_0_s_ifspec

typedef struct orderstatus_v1_0_epv {
void (ENCINA_STUB_CALLING *impTPCCOrderStatus) (
#ifdef IDL_PROTOTYPES

        idl_long_int length,
        idl_char *dataP,
        data_header *headerP,
        trpc_status_t *trpcStatus
#endif
);
} orderstatus_v1_0_epv_t;

DLLEXPORT void ENCINA_STUB_CALLING impTPCCOrderStatus (
#ifdef IDL_PROTOTYPES

        idl_long_int length,
        idl_char *dataP,
        data_header *headerP,
        trpc_status_t *trpcStatus
#endif
);

```



```

trpc_handle_t          ENCINA_CALLING
mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
    mon_handle_t      handle,
    trpc_tranInfo_t   *tranInfoP,
    trpc_ifSpec_t     *ifSpecP
#endif
);

void          ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
    mon_handle_t      handle,
    trpc_handle_t     trpcHandle,
    trpc_tranInfo_t   *tranInfoP,
    trpc_ifSpec_t     *ifSpecP
#endif
);

trpc_handle_t          ENCINA_CALLING
mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
    mon_handle_t      handle,
    trpc_tranInfo_t   *tranInfoP,
    trpc_ifSpec_t     *ifSpecP
#endif
);

void          ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
    mon_handle_t      handle,
    trpc_handle_t     trpcHandle,
    trpc_tranInfo_t   *tranInfoP,
    trpc_ifSpec_t     *ifSpecP
#endif
);

extern orderstatus_v1_0_epv_t
orderstatus_v1_0_client_epv;
extern _orderstatus_v1_0_epv_t
orderstatus_v1_0_manager_epv;
extern rpc_mgr_epv_t
orderstatus_v1_0_mgr_epv;

#include <encina/c_epilogue.h>
#endif /* TRANSARC_orderstatus_h */

```

payment.h

```

#ifdef TRANSARC_payment_h
#define TRANSARC_payment_h

#include <trpc/trpc.h>
#include "payment.h"

#include <encina/c_prologue.h>

#if defined(BUILDDLL)
#define DLLEXPORT __declspec( dllexport )
#else
#define DLLEXPORT extern

```

```

#endif

#ifdef ENCINA_STUB_CALLING
#define ENCINA_STUB_CALLING ENCINA_RPC_CALLING
#endif

#define payment_v1_0_c_ifspec_payment_v1_0_c_ifspec
#define payment_v1_0_s_ifspec_payment_v1_0_s_ifspec

typedef struct payment_v1_0_epv {
void (ENCINA_STUB_CALLING *impTPCCPayment) (
#ifdef IDL_PROTOTYPES
    idl_long_int length,
    idl_char *dataP,
    data_header *headerP,
    trpc_status_t *trpcStatus
#endif
);
} payment_v1_0_epv_t;

DLLEXPORT void ENCINA_STUB_CALLING impTPCCPayment (
#ifdef IDL_PROTOTYPES
    idl_long_int length,
    idl_char *dataP,
    data_header *headerP,
    trpc_status_t *trpcStatus
#endif
);

trpc_handle_t          ENCINA_CALLING
mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
    mon_handle_t      handle,
    trpc_tranInfo_t   *tranInfoP,
    trpc_ifSpec_t     *ifSpecP
#endif
);

void          ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
    mon_handle_t      handle,
    trpc_handle_t     trpcHandle,
    trpc_tranInfo_t   *tranInfoP,
    trpc_ifSpec_t     *ifSpecP
#endif
);

trpc_handle_t          ENCINA_CALLING
mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
    mon_handle_t      handle,
    trpc_tranInfo_t   *tranInfoP,
    trpc_ifSpec_t     *ifSpecP
#endif
);

void          ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES

```

```

    mon_handle_t      handle,
    trpc_handle_t     trpcHandle,
    trpc_tranInfo_t   *tranInfoP,
    trpc_ifSpec_t     *ifSpecP
#endif
);

extern payment_v1_0_epv_t
    payment_v1_0_client_epv;
extern _payment_v1_0_epv_t
    payment_v1_0_manager_epv;
extern rpc_mgr_epv_t
    payment_v1_0_mgr_epv;

#include <encina/c_epilogue.h>
#endif /* TRANSARC_payment_h */

```

ReadRegistry.cpp

```

/* FILE: READREGISTRY.CPP
 * Microsoft
 * TPC-C Kit Ver. 4.20.000
 * Copyright
 * Microsoft, 1999
 * All Rights Reserved
 * not yet
 * audited
 * PURPOSE: Implementation for TPC-C Tuxedo
 * class.
 * Contact: Charles Levine
 * (clevine@microsoft.com)
 * Change history:
 * 4.20.000 - first version
 */

/* FUNCTION: ReadTPCCRegistrySettings
 * PURPOSE: This function reads the NT
 * registry for startup parameters. There parameters are
 * under the TPCC key.
 * RETURNS FALSE = no errors
 * TRUE = error reading
 * registry
 */
BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg
)
{
    HKEY hKey;
    DWORD size;
    DWORD type;
    DWORD dwTmp;
    char szTmp[256];

```

```

        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\\Microsoft\\TPCC", 0, KEY_READ, &hKey) !=
ERROR_SUCCESS )
            return TRUE;

        // determine database protocol to use; may
be either ODBC or DBLIB
        pReg->eDB_Protocol = Unspecified;
        size = sizeof(szTmp);
        if ( RegQueryValueEx(hKey, "DB_Protocol",
0, &type, (BYTE *)&szTmp, &size) == ERROR_SUCCESS )
        {
            if ( !strcmp(szTmp,
szDBNames[ODBC] ) )
                pReg->eDB_Protocol =
ODBC;
            else if ( !strcmp(szTmp,
szDBNames[DBLIB] ) )
                pReg->eDB_Protocol =
DBLIB;
        }

        pReg->eTxnMon = None;
        // determine txn monitor to use; may be
either TUXEDO, or blank
        size = sizeof(szTmp);
        if ( RegQueryValueEx(hKey, "TxnMonitor", 0,
&type, (BYTE *)&szTmp, &size) == ERROR_SUCCESS )
        {
            if ( !strcmp(szTmp,
szTxnMonNames[TUXEDO] ) )
                pReg->eTxnMon = TUXEDO;
            else if ( !strcmp(szTmp,
szTxnMonNames[ENCINA] ) )
                pReg->eTxnMon = ENCINA;
            else if ( !strcmp(szTmp,
szTxnMonNames[COM] ) )
                pReg->eTxnMon = COM;
        }

        pReg->bCOM_SinglePool = FALSE;
        size = sizeof(szTmp);
        if ( RegQueryValueEx(hKey,
"COM_SinglePool", 0, &type, (BYTE *)&szTmp, &size) ==
ERROR_SUCCESS )
        {
            if ( !strcmp(szTmp, "YES" ) )
                pReg->bCOM_SinglePool =
TRUE;
        }

        pReg->dwMaxConnections = 0;
        size = sizeof(dwTmp);
        if ( ( RegQueryValueEx(hKey,
"MaxConnections", 0, &type, (LPBYTE)&dwTmp, &size) ==
ERROR_SUCCESS )
            && (type == REG_DWORD) )
            pReg->dwMaxConnections = dwTmp;

        pReg->dwMaxPendingDeliveries = 0;
        size = sizeof(dwTmp);

```

```

        if ( ( RegQueryValueEx(hKey,
"MaxPendingDeliveries", 0, &type, (LPBYTE)&dwTmp,
&size) == ERROR_SUCCESS )
            && (type == REG_DWORD) )
            pReg->dwMaxPendingDeliveries =
dwTmp;

        pReg->dwNumberOfDeliveryThreads = 0;
        size = sizeof(dwTmp);
        if ( ( RegQueryValueEx(hKey,
"NumberOfDeliveryThreads", 0, &type, (LPBYTE)&dwTmp,
&size) == ERROR_SUCCESS )
            && (type == REG_DWORD) )
            pReg->dwNumberOfDeliveryThreads =
dwTmp;

        size = sizeof( pReg->szPath );
        if ( RegQueryValueEx(hKey, "Path", 0,
&type, (BYTE *)&pReg->szPath, &size) != ERROR_SUCCESS
)
            pReg->szPath[0] = 0;

        size = sizeof( pReg->szDbServer );
        if ( RegQueryValueEx(hKey, "DbServer", 0,
&type, (BYTE *)&pReg->szDbServer, &size) !=
ERROR_SUCCESS )
            pReg->szDbServer[0] = 0;

        size = sizeof( pReg->szDbName );
        if ( RegQueryValueEx(hKey, "DbName", 0,
&type, (BYTE *)&pReg->szDbName, &size) !=
ERROR_SUCCESS )
            pReg->szDbName[0] = 0;

        size = sizeof( pReg->szDbUser );
        if ( RegQueryValueEx(hKey, "DbUser", 0,
&type, (BYTE *)&pReg->szDbUser, &size) !=
ERROR_SUCCESS )
            pReg->szDbUser[0] = 0;

        size = sizeof( pReg->szDbPassword );
        if ( RegQueryValueEx(hKey, "DbPassword", 0,
&type, (BYTE *)&pReg->szDbPassword, &size) !=
ERROR_SUCCESS )
            pReg->szDbPassword[0] = 0;

        size = sizeof( pReg->szSPPrefix );
        if ( RegQueryValueEx(hKey, L"SPPrefix", 0,
&type, (BYTE *)&pReg->szSPPrefix, &size) !=
ERROR_SUCCESS )
            pReg->szSPPrefix[0] = L'\0';

        pReg->dwConnectDelay = 0;
        size = sizeof(dwTmp);
        if ( ( RegQueryValueEx(hKey,
"ConnectDelay", 0, &type, (LPBYTE)&dwTmp, &size) ==
ERROR_SUCCESS )
            && (type == REG_DWORD) )
            pReg->dwConnectDelay = dwTmp;

        pReg->bCallNoDuplicatesNewOrder = FALSE;

```

```

        size = sizeof(dwTmp);
        if ( ( RegQueryValueEx(hKey,
"CallNoDuplicatesNewOrder", 0, &type, (LPBYTE)&dwTmp,
&size) == ERROR_SUCCESS )
            && (type == REG_DWORD) )
            pReg->bCallNoDuplicatesNewOrder =
dwTmp;

        RegCloseKey(hKey);

        return FALSE;
}

```

ReadRegistry.h

```

/* FILE: ReadRegistry.h
* Microsoft
TPC-C Kit Ver. 4.20.000
* Copyright
Microsoft, 1999
* All Rights Reserved
* not audited
* PURPOSE: Header for registry related code.
* Change history:
* 4.20.000 - first version
*/

enum DBPROTOCOL { Unspecified, ODBC, DBLIB };
const char *szDBNames[] = { "Unspecified", "ODBC",
"DBLIB" };

enum TXNMON { None, TUXEDO, ENCINA, COM };
const char *szTxnMonNames[] = { "NONE", "TUXEDO",
"ENCINA", "COM" };

//This structure defines the data necessary to keep
distinct for each terminal or client connection.
typedef struct _TPCCREGISTRYDATA
{
    enum DBPROTOCOL eDB_Protocol;
    enum TXNMON eTxnMon;
    BOOL bCOM_SinglePool;
    DWORD dwMaxConnections;
    DWORD dwMaxPendingDeliveries;
    DWORD dwNumberOfDeliveryThreads;
    char szPath[128];
    char szDbServer[32];
    char szDbName[32];
    char szDbUser[32];
    char szDbPassword[32];
    wchar_t szSPPrefix[32];
    //tpcc_odbc.dll stored procedures prefix
    DWORD dwConnectDelay; // delay in
ms to use in pacing connection open and close
    BOOL bCallNoDuplicatesNewOrder; //
whether to check for non-duplicate item ids and call
a different New Order SP

```

```

} TPCCREGISTRYDATA, *PTPCCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg
);

```

RESOURCE.H

```

//{{NO DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by install.rc
//
#define IDD_DIALOG1 101
#define IDI_ICON1 102
#define IDR_TPCCDLL 103
#define IDD_DIALOG2 105
#define IDI_ICON2 106
#define IDR_DELIVERY 107
#define IDD_DIALOG3 108
#define IDR_LICENSE1 112
#define IDD_DIALOG4 113
#define IDR_TPCCOBJ1 117
#define IDR_TPCCSTUB1 118
#define IDR_DBLIB_DLL 122
#define IDR_ODBC_DLL 123
#define IDR_TUXEDO_APP 124
#define IDR_TUXEDO_DLL 125
#define IDR_COM_DLL 126
#define IDR_COMPS_DLL 127
#define IDR_COMALL_DLL 128
#define IDR_COMTYPLIB_DLL 129
#define IDR_MSVCRT701 130
#define BN_LOG 1001
#define ED_KEEP 1002
#define ED_THREADS 1003
#define ED_THREADS2 1004
#define IDC_PATH 1007
#define IDC_VERSION 1009
#define IDC_RESULTS 1010
#define IDC_PROGRESS1 1011
#define IDC_STATUS 1012
#define IDC_BUTTON1 1013
#define ED_MAXCONNECTION 1014
#define ED_IIS_MAX_THREAD_POOL_LIMIT 1015
#define ED_MAXDELIVERIES 1016
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT 1018
#define ED_IIS_LISTEN_BACKLOG 1019
#define IDC_DBLIB 1021
#define IDC_LICENSE 1022
#define IDC_ODBC 1022
#define IDC_CONNECT_POOL 1023
#define ED_DB_SERVER 1023
#define ED_USER_CONNECT_DELAY_TIME 1024
#define ED_DB_USER_ID 1024
#define IDC_MTS 1025
#define IDC_TM_MTS 1025
#define IDC_TM_TUXEDO 1026
#define IDC_TM_NONE 1027
#define ED_DB_PASSWORD 1028
#define ED_DB_NAME 1029

```

```

#define IDC_TM_ENCINA 1030

// Next default values for new objects
//
#ifndef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 131
#define _APS_NEXT_COMMAND_VALUE 40001
#define _APS_NEXT_CONTROL_VALUE 1031
#define _APS_NEXT_SYMED_VALUE 101
#endif
#endif

```

stocklevel.h

```

#ifndef TRANSARC_stocklevel_h
#define TRANSARC_stocklevel_h

#include <trpc/trpc.h>
#include "_stocklevel.h"

#include <encina/c_prologue.h>

#if defined(BUILDDLL)
#define DLLEXPORT __declspec( dllexport )
#else
#define DLLEXPORT extern
#endif

#ifndef ENCINA_STUB_CALLING
#define ENCINA_STUB_CALLING ENCINA_RPC_CALLING
#endif

#define stocklevel_v1_0_c_ifspec
        _stocklevel_v1_0_c_ifspec
#define stocklevel_v1_0_s_ifspec
        _stocklevel_v1_0_s_ifspec

typedef struct stocklevel_v1_0_epv {
void (ENCINA_STUB_CALLING *impTPCCstockLevel) (

        idl_long_int length,
        idl_char *dataP,
        data_header *headerP,
        trpc_status_t *trpcStatus

#endif
);

} stocklevel_v1_0_epv_t;

DLLEXPORT void ENCINA_STUB_CALLING impTPCCstockLevel
(
#ifndef IDL_PROTOTYPES

        idl_long_int length,
        idl_char *dataP,
        data_header *headerP,
        trpc_status_t *trpcStatus

#endif
)

```

```

);

trpc_handle_t ENCINA_CALLING
mon_handle_t_tranBind(
#ifndef IDL_PROTOTYPES
        mon_handle_t handle,
        trpc_tranInfo_t *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
);

void ENCINA_CALLING mon_handle_t_tranUnBind(
#ifndef IDL_PROTOTYPES
        mon_handle_t handle,
        trpc_handle_t trpcHandle,
        trpc_tranInfo_t *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
);

trpc_handle_t ENCINA_CALLING
mon_handle_t_tranBind(
#ifndef IDL_PROTOTYPES
        mon_handle_t handle,
        trpc_tranInfo_t *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
);

void ENCINA_CALLING mon_handle_t_tranUnBind(
#ifndef IDL_PROTOTYPES
        mon_handle_t handle,
        trpc_handle_t trpcHandle,
        trpc_tranInfo_t *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
);

extern stocklevel_v1_0_epv_t
        stocklevel_v1_0_client_epv;
extern _stocklevel_v1_0_epv_t
        stocklevel_v1_0_manager_epv;
extern rpc_mgr_epv_t
        stocklevel_v1_0_mgr_epv;

#include <encina/c_epilogue.h>
#endif /* TRANSARC_stocklevel_h */

```

tpcc.cpp

```

/* FILE: TPCC.C
 * Microsoft
 * TPC-C Kit Ver. 4.20.000
 * Copyright
 * Microsoft, 1999
 * All Rights Reserved
 * Version
 * 4.10.000 audited by Richard Gimarc, Performance
 * Metrics, 3/17/99

```

```

*
* PURPOSE: Main module for TPCC.DLL which is
an ISAPI service dll.
* Contact: Charles Levine
(clevine@microsoft.com)
*
* Change history:
* 4.20.000 - reworked error
handling; added options for COM and Encina txn
monitors
*/

#include <windows.h>
#include <process.h>
#include <tchar.h>
#include <stdio.h>
#include <stdarg.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <io.h>
#include <assert.h>

#include <sqltypes.h>

#ifdef ICECAP
#include <icapexp.h>
#endif

#include "..\..\common\src\trans.h"
//tpckit transaction header contains
definitions of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\ReadRegistry.h"

#include "..\..\common\txnlog\include\rtetime.h"
#include "..\..\common\txnlog\include\spinlock.h"
#include "..\..\common\txnlog\include\txnlog.h"

// Database layer includes
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"
// DBLIB implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"
// ODBC implementation of TPC-C txns

// Txn monitor layer includes
#include "..\..\tm_com_dll\src\tpcc_com.h"
// COM Services implementation on
TPC-C txns
#include "..\..\tm_tuxedo_dll\src\tpcc_tux.h"
// interface to Tuxedo libraries
#include "..\..\tm_encina_dll\src\tpcc_enc.h"
// interface to Encina libraries

#include "httpext.h"
//ISAPI DLL information header

```

```

#include "tpcc.h"
//this dlls specific structure, value e.t.
header.

#define LEN_ERR_STRING 256

// defines for Make<Txn>Form calls to distinguish
input and output flavors
#define OUTPUT_FORM 0
#define INPUT_FORM 1

char
    szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

//Terminal client id structure
TERM Term = { 0, 0, 0, NULL };

// The WEBCLIENT_VERSION string specifies the version
level of this web client interface.
// The RTE must be synchronized with the interface
level on login, otherwise the login
// will fail. This is a sanity check to catch
problems resulting from mismatched versions
// of the RTE and web client.
#define WEBCLIENT_VERSION "420"

static CRITICAL_SECTION
    TermCriticalSection;

static HINSTANCE hLibInstanceTm = NULL;
static HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC *pCTPCC_ODBC_new;
TYPE_CTPCC_TUXEDO *pCTPCC_TUXEDO_new;
TYPE_CTPCC_ENCINA *pCTPCC_ENCINA_new;
TYPE_CTPCC_ENCINA *pCTPCC_ENCINA_post_init;
TYPE_CTPCC_COM *pCTPCC_COM_new;

// For deferred Delivery txns:

CTxnLog
    *txnDelilog = NULL;
//used to log delivery transaction
information

HANDLE hWorkerSemaphore = INVALID_HANDLE_VALUE;
HANDLE hDoneEvent =
INVALID_HANDLE_VALUE;
HANDLE *pDeliHandles = NULL;

// configuration settings from registry
TPCCREGISTRYDATA Reg;

```

```

DWORD
    dwNumDeliveryThreads = 4;
CRITICAL_SECTION DelBuffCriticalSection;
//critical section for delivery
transactions cache
DELIVERY_TRANSACTION *pDelBuff
    = NULL;

DWORD
    dwDelBuffSize = 100;
// size of circular buffer for delivery

txns
DWORD
    dwDelBuffFreeCount;
// number of buffers free

DWORD
    dwDelBuffBusyIndex = 0;
// index position of entry waiting to be delivered
DWORD
    dwDelBuffFreeIndex = 0;
// index position of unused entry

// Critical section to synchronize connection open
and close.
//
CRITICAL_SECTION hConnectCriticalSection;

#include "..\..\common\src\ReadRegistry.cpp"

/* FUNCTION: DllMain
*
* PURPOSE: This function is the entry point
for the DLL. This implementation is based on the
* fact that
DLL_PROCESS_ATTACH is only called from the inet
service once.
*
* ARGUMENTS: HANDLE hModule
module handle
*
* ul_reason_for_call reason for call
* LPVOID LPVOID
reserved for future use
*
* RETURNS: BOOL FALSE
errors occurred in
initialization
*
TRUE DLL
successfully initialized
*/

BOOL APIENTRY DllMain(HANDLE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
    DWORD i;
    char szEvent[LEN_ERR_STRING] = "\0";
    char szLogFile[128];
    char szDllName[128];

// debugging...

```

```

// DebugBreak();

try
{
    switch( ul_reason_for_call )
    {
        case
            DLL_PROCESS_ATTACH:
                {
                    DWORD dwSize = MAX_COMPUTERNAME_LENGTH+1;
                    GetComputerName( szMyComputerName, &dwSize);
                    szMyComputerName[ dwSize ] = 0;
                }

                DisableThreadLibraryCalls( (HMODULE)hModule );

                InitializeCriticalSection( &TermCriticalSection );

                if (
                    ReadTPCCRegistrySettings( &Reg ) )
                    throw new CWBCLNT_ERR(
                        ERR_MISSING_REGISTRY_ENTRIES );

                dwDelBuffSize
                = min( Reg.dwMaxPendingDeliveries, 10000 ); // min
                with 10000 as a sanity constraint

                dwNumDeliveryThreads = min(
                    Reg.dwNumberOfDeliveryThreads, 100 ); // min with
                100 as a sanity constraint

                TermInit();

                // load DLL
                for txn monitor
                if
                (Reg.eTxnMon == TUXEDO)
                {
                    strcpy( szDllName, Reg.szPath );

                    strcat( szDllName, "tpcc_tuxedo.dll" );

                    hLibInstanceTm = LoadLibrary( szDllName );
                    if
                    (hLibInstanceTm == NULL)

                        throw new CWBCLNT_ERR( ERR_LOADDLL_FAILED,
                            szDllName, GetLastError() );

                    //
                    get function pointer to wrapper for class constructor
                }
            }
    }
}

```

```

        pCTPCC_TUXEDO_new = (TYPE_CTPCC_TUXEDO*)
        GetProcAddress( hLibInstanceTm, "CTPCC_TUXEDO_new" );
        if
        (pCTPCC_TUXEDO_new == NULL)

            throw new CWBCLNT_ERR(
                ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
        else if
        (Reg.eTxnMon == ENCINA)
        {
            strcpy( szDllName, Reg.szPath );

            strcat( szDllName, "tpcc_encina.dll" );

            hLibInstanceTm = LoadLibrary( szDllName );
            if
            (hLibInstanceTm == NULL)

                throw new CWBCLNT_ERR( ERR_LOADDLL_FAILED,
                    szDllName, GetLastError() );

            //
            get function pointer to wrapper for class constructor

            pCTPCC_ENCINA_new = (TYPE_CTPCC_ENCINA*)
            GetProcAddress( hLibInstanceTm, "CTPCC_ENCINA_new" );

            pCTPCC_ENCINA_post_init =
            (TYPE_CTPCC_ENCINA*)
            GetProcAddress( hLibInstanceTm, "CTPCC_ENCINA_post_init
            " );
            if
            (pCTPCC_ENCINA_new == NULL)

                throw new CWBCLNT_ERR(
                    ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
            else if
            (Reg.eTxnMon == COM)
            {
                strcpy( szDllName, Reg.szPath );

                strcat( szDllName, "tpcc_com.dll" );

                hLibInstanceTm = LoadLibrary( szDllName );
                if
                (hLibInstanceTm == NULL)

                    throw new CWBCLNT_ERR( ERR_LOADDLL_FAILED,
                        szDllName, GetLastError() );

                //
                get function pointer to wrapper for class constructor

                pCTPCC_COM_new = (TYPE_CTPCC_COM*)
                GetProcAddress( hLibInstanceTm, "CTPCC_COM_new" );
                if
                (pCTPCC_COM_new == NULL)

```

```

                throw new CWBCLNT_ERR(
                    ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
            }
        }

        // load DLL
        for database connection
        if
        ((Reg.eTxnMon == None) || (dwNumDeliveryThreads > 0))
        {
            if
            (Reg.eDB_Protocol == DBLIB)
            {
                strcpy( szDllName, Reg.szPath );

                strcat( szDllName, "tpcc_dblib.dll" );

                hLibInstanceDb = LoadLibrary( szDllName );
                if (hLibInstanceDb == NULL)

                    throw new CWBCLNT_ERR(
                        ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                // get function pointer to wrapper for
                class constructor

                pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
                GetProcAddress( hLibInstanceDb, "CTPCC_DBLIB_new" );

                if (pCTPCC_DBLIB_new == NULL)

                    throw new CWBCLNT_ERR(
                        ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
            }
            else if (Reg.eDB_Protocol == ODBC)
            {
                strcpy( szDllName, Reg.szPath );

                strcat( szDllName, "tpcc_odbc.dll" );

                hLibInstanceDb = LoadLibrary( szDllName );
                if (hLibInstanceDb == NULL)

                    throw new CWBCLNT_ERR(
                        ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                // get function pointer to wrapper for
                class constructor

                pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
                GetProcAddress( hLibInstanceDb, "CTPCC_ODBC_new" );
                if (pCTPCC_ODBC_new == NULL)

```

```

        throw new CWBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
    }
}
    if
(dwNumDeliveryThreads)
    {
Initialize delivery delay critical section
//
//
        InitializeCriticalSection(&hConnectCritical
Section);
//
for deferred delivery txns:
//
        hDoneEvent = CreateEvent( NULL, TRUE /*
manual reset */, FALSE /* initially not signalled */,
NULL );
        InitializeCriticalSection(&DelBuffCriticalS
ection);
        hWorkerSemaphore = CreateSemaphore( NULL,
0, dwDelBuffSize, NULL );
        dwDelBuffFreeCount = dwDelBuffSize;
        InitJulianTime(NULL);
//
create unique log file name based on delilog-yyymmdd-
hhmm.log
        SYSTEMTIME Time;
        GetLocalTime( &Time );
        wsprintf( szLogFile, "%sdelivery-
%2.2d%2.2d%2.2d-%2.2d%2.2d-%2.2ds%2.2dms.log",
Reg.szPath, Time.wYear % 100, Time.wMonth,
Time.wDay, Time.wHour, Time.wMinute, Time.wSecond,
Time.wMilliseconds );
        txnDelilog = new CTxnLog(szLogFile,
TXN_LOG_WRITE);
//write event into txn log for START
        txnDelilog-
>WriteCtrlRecToLog(TXN_EVENT_START, szMyComputerName,
sizeof(szMyComputerName));

```

```

//
allocate structures for delivery buffers and thread
mgmt
        pDeliHandles = new
HANDLE[dwNumDeliveryThreads];
        pDelBuff = new
DELIVERY_TRANSACTION[dwDelBuffSize];
//
launch DeliveryWorkerThread to perform actual
delivery txns
        for(i=0; i<dwNumDeliveryThreads; i++)
        {
                pDeliHandles[i] = (HANDLE) _beginthread(
DeliveryWorkerThread, 0, NULL );
                if (pDeliHandles[i] ==
INVALID_HANDLE_VALUE)
                {
                        throw new CWBCLNT_ERR(
ERR_DELIVERY_THREAD_FAILED );
                }
                break;
        case
DLL_PROCESS_DETACH:
                if
(dwNumDeliveryThreads)
                {
                        if
(txnDelilog != NULL)
                        {
                                //write event into txn log for STOP
                                txnDelilog-
>WriteCtrlRecToLog(TXN_EVENT_STOP, szMyComputerName,
sizeof(szMyComputerName));
                                // This will do a clean shutdown of the
delivery log file
                                CTxnLog *txnDelilogLocal = txnDelilog;
                                txnDelilog= NULL;
                                delete txnDelilogLocal;
                                delete [] pDeliHandles;
                                delete [] pDelBuff;

```

```

        CloseHandle( hWorkerSemaphore );
        CloseHandle( hDoneEvent );
        DeleteCriticalSection(&DelBuffCriticalSecti
on);
//
Delete delivery delay critical section
//
        DeleteCriticalSection(&hConnectCriticalSect
ion);
        DeleteCriticalSection(&TermCriticalSection)
;
        if
(hLibInstanceTm != NULL)
        {
                FreeLibrary( hLibInstanceTm );
                hLibInstanceTm = NULL;
        }
        if
(hLibInstanceDb != NULL)
        {
                FreeLibrary( hLibInstanceDb );
                hLibInstanceDb = NULL;
        }
        Sleep(500);
        break;
        default:
                /* nothing
*/;
        }
    }
    catch (CBaseErr *e)
    {
        TCHAR szMsg[256];
        _sntprintf(szMsg, sizeof(szMsg),
"%s error, code %d: %s",
e-
>ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
        WriteMessageToEventLog( szMsg );
        delete e;
        TerminateExtension(0);
        return FALSE;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception. DLL could not load."));
        TerminateExtension(0);

```

```

        return FALSE;
    }
    return TRUE;
}

/* FUNCTION: GetExtensionVersion
 *
 * PURPOSE:      This function is called by the
inet service when the DLL is first loaded.
 *
 * ARGUMENTS:    HSE_VERSION_INFO  *pVer
                passed in structure in which to place
                expected version number.
 *
 * RETURNS:      TRUE      inet service
                expected return value.
 */
BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO
 *pVer)
{
    pVer->dwExtensionVersion =
    MAKELONG(HSE_VERSION_MINOR, HSE_VERSION_MAJOR);
    lstrcpyn(pVer->lpszExtensionDesc, "TPC-C
Server.", HSE_MAX_EXT_DLL_NAME_LEN);

    // TODO: why do we need this here instead
of in the DLL attach?
    if (Reg.eTxnMon == ENCINA)
        pCTPCC_ENCINA_post_init();

    return TRUE;
}

/* FUNCTION: TerminateExtension
 *
 * PURPOSE:      This function is called by the
inet service when the DLL is about to be unloaded.
 *
 * Release all resources
in anticipation of being unloaded.
 *
 * RETURNS:      TRUE      inet service
                expected return value.
 */
BOOL WINAPI TerminateExtension( DWORD dwFlags )
{
    if (pDeliHandles)
    {
        SetEvent( hDoneEvent );
        for(DWORD i=0;
i<dwNumDeliveryThreads; i++)
            WaitPorSingleObject(
pDeliHandles[i], INFINITE );
    }

    TermDeleteAll();
    return TRUE;
}

```

```

/* FUNCTION: HttpExtensionProc
 *
 * PURPOSE:      This function is the main entry
point for the TPCC DLL. The internet service
 *
 * calls this function
passing in the http string.
 *
 * ARGUMENTS:    EXTENSION_CONTROL_BLOCK
                *pECB      structure pointer to passed in
internet
 *
                service information.
 *
 * RETURNS:      DWORD
                HSE_STATUS_SUCCESS
                Connection can be dropped if
error
 *
                HSE_STATUS_SUCCESS_AND_KEEP_CONN
                keep connect valid comment sent
 *
 * COMMENTS:     None
 */
DWORD WINAPI
HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB)
{
    int iCmd, FormId,
TermId, iSyncId;
    char szBuffer[4096];

    int lpbSize;
    static char szHeader[] = "200 Ok";
    DWORD dwSize = 6;
    // initial value is strlen(szHeader)
    char szHeader1[4096];

#ifdef ICECAP
    StartCAP();
#endif

    try
    {
        //process http query
        ProcessQueryString(pECB, &iCmd,
&FormId, &TermId, &iSyncId);

        if (TermId != 0)
        {
            if ( TermId < 0 ||
TermId >= Term.iNumEntries ||
Term.pClientData[TermId].iNextFree != -1 )
            {
                //
                debugging...
            }

```

```

                char
                szTmp[128];
                wsprintf(
                szTmp, "Invalid term ID; TermId = %d", TermId );
                WriteMessageToEventLog( szTmp );

                throw new
                CWBCLNT_ERR( ERR_INVALID_TERMID );
                //must have a valid
                syncid here since termid is valid
                if (iSyncId !=
                Term.pClientData[TermId].iSyncId)
                throw new
                CWBCLNT_ERR( ERR_INVALID_SYNC_CONNECTION );
                //set use time
                Term.pClientData[TermId].iTickCount =
                GetTickCount();
            }

            switch(iCmd)
            {
            case 0:
                WelcomeForm(pECB,
                szBuffer);
                break;
            case 1:
                switch( FormId )
                {
                case
                WELCOME_FORM:
                case
                MAIN_MENU_FORM:
                break;
                case
                NEW_ORDER_FORM:
                ProcessNewOrderForm(pECB, TermId,
                szBuffer);
                break;
                case
                PAYMENT_FORM:
                ProcessPaymentForm(pECB, TermId, szBuffer);
                break;
                case
                DELIVERY_FORM:
                ProcessDeliveryForm(pECB, TermId,
                szBuffer);
                break;

```

```

                                case
ORDER_STATUS_FORM:
    ProcessOrderStatusForm(pECB, TermId,
szBuffer);
    break;
                                case
STOCK_LEVEL_FORM:
    ProcessStockLevelForm(pECB, TermId,
szBuffer);
    break;
                                }
                                break;
                                case 2:
// new-order selected
from menu; display new-order input form
    MakeNewOrderForm(TermId, NULL, INPUT_FORM,
szBuffer);
    break;
                                case 3:
// payment selected
from menu; display payment input form
    MakePaymentForm(TermId,
NULL, INPUT_FORM, szBuffer);
    break;
                                case 4:
// delivery selected
from menu; display delivery input form
    MakeDeliveryForm(TermId, NULL, INPUT_FORM,
szBuffer);
    break;
                                case 5:
// order-status
selected from menu; display order-status input form
    MakeOrderStatusForm(TermId, NULL,
INPUT_FORM, szBuffer);
    break;
                                case 6:
// stock-level selected
from menu; display stock-level input form
    MakeStockLevelForm(TermId, NULL,
INPUT_FORM, szBuffer);
    break;
                                case 7:
// ExitCmd
TermDelete(TermId);
WelcomeForm(pECB,
szBuffer);
    break;
                                case 8:
SubmitCmd(pECB,
szBuffer);
    break;
                                case 9:

```

```

                                // menu
                                MakeMainMenuForm(TermId,
Term.pClientData[TermId].iSyncId, szBuffer);
                                break;
                                case 10:
// CMD=Clear
// resets all
connections; should only be used when no other
connections are active
    TermDeleteAll();
TermInit();
WelcomeForm(pECB,
szBuffer);
    break;
                                case 11:
// CMD=Stats
StatsCmd(pECB,
szBuffer);
    break;
                                }
                                catch (CBaseErr *e)
                                {
ErrorForm( pECB, e->ErrorType(),
e->ErrorNum(), TermId, iSyncId, e->ErrorText(),
szBuffer );
delete e;
                                }
                                catch (...)
                                {
ErrorForm( pECB, ERR_TYPE WEBDLL,
0, TermId, iSyncId, "Error: Unhandled exception in
Web Client.", szBuffer );
                                }
#ifdef ICECAP
StopCAP();
#endif
    lpbSize = strlen(szBuffer);
    wsprintf(szHeader1,
"Content-Type:
text/html\r\n"
"Content-Length:
%d\r\n"
"Connection: Keep-
Alive\r\n\r\n", lpbSize);
    strcat( szHeader1, szBuffer );
    (*pECB->ServerSupportFunction)(pECB-
>ConnID, HSE_REQ_SEND_RESPONSE_HEADER, szHeader,
(LPDWORD) &dwSize, (LPDWORD)szHeader1);
//finish up and keep connection
pECB->dwHttpStatusCode = 200;
return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}
void WriteMessageToEventLog(LPTSTR lpszMsg)

```

```

{
TCHAR szMsg[256];
HANDLE hEventSource;
LPTSTR lpszStrings[2];
// Use event logging to log the error.
//
hEventSource = RegisterEventSource(NULL,
TEXT("TPCC.DLL"));
    _stprintf(szMsg, TEXT("Error in TPCC.DLL: "));
    lpszStrings[0] = szMsg;
    lpszStrings[1] = lpszMsg;
    if (hEventSource != NULL)
    {
ReportEvent(hEventSource, // handle of event
source
EVENTLOG_ERROR_TYPE, // event type
0, // event category
0, // event ID
NULL, // current user's
SID
2, // strings in
lpszStrings
0, // no bytes of raw
data
(LPCTSTR *)lpszStrings, // array of
error strings
NULL); // no raw data
        (VOID) DeregisterEventSource(hEventSource);
    }
}
/* FUNCTION: DeliveryWorkerThread
*
* PURPOSE: This function processes deferred
delivery txns. There are typically several
threads running this
routine. The number of threads is determined by an
entry
read from the registry.
The thread waits for work by waiting on semaphore.
*
* When a delivery txn is
posted, the semaphore is released. After processing
the delivery txn,
information is logged to record the txn status and
execution
time.
*/
/*static*/ void DeliveryWorkerThread(void *ptr)
{
CTPCC_BASE *pTxn = NULL;
DELIVERY_TRANSACTION
delivery;
PDELIVERY_DATA
pDeliveryData;

```



```

TXN_RECORD_TPCC_DELIV_DEF    txnDeliRec;

DWORD
index;
HANDLE
handles[2];

SYSTEMTIME                    trans_end;
//delivery transaction finished
time
SYSTEMTIME                    trans_start;
//delivery transaction start time

assert(txnDeliLog != NULL);

try
{
    if (Reg.eDB_Protocol == ODBC)
    {
        if (Reg.dwConnectDelay
> 0)
        {
            Synchronize connect (for VIA)
            //
            //
            EnterCriticalSection(&hConnectCriticalSection);

            Sleep(Reg.dwConnectDelay);

            pTxn =
pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword,

                szMyComputerName, Reg.szDbName,

                Reg.szSPPrefix,
Reg.bCallNoDuplicatesNewOrder );

            LeaveCriticalSection(&hConnectCriticalSection);
        }
        else
        {
            if (Reg.eDB_Protocol ==
DBLIB)
                pTxn =
pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
            pDeliveryData = pTxn-
>BuffAddr_Delivery();
        }
        catch (CBaseErr *e)
        {

```

```

char szTmp[1024];
wsprintf( szTmp, "Error in
Delivery Txn thread. Could not connect to database.
"
"%s.
Server=%s, User=%s, Password=%s, Database=%s",
e-
>ErrorText(), Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, Reg.szDbName );
WriteMessageToEventLog( szTmp );
delete e;
goto ErrorExit;
}
catch (...)
{
    WriteMessageToEventLog(TEXT("Unhandled
exception caught in DeliveryWorkerThread."));
    goto ErrorExit;
}

while (TRUE)
{
    try
    {
        //while delivery thread
        running, i.e. user has not requested termination
        while (TRUE)
        {
            // need to
            wait for multiple objects: program exit or worker
            semaphore;
            handles[0] =
hDoneEvent;
            handles[1] =
hWorkerSemaphore;
            index =
WaitForMultipleObjects( 2, &handles[0], FALSE,
INFINITE );
            if (index ==
WAIT_OBJECT_0)
                goto ErrorExit;

            ZeroMemory(&txnDeliRec,
sizeof(txnDeliRec));

            txnDeliRec.TxnType =
TXN_REC_TYPE_TPCC_DELIV_DEF;

            // make a
            local copy of current entry from delivery buffer and
            increment buffer index

            EnterCriticalSection(&DelBuffCriticalSection);
            delivery =
*(pDelBuff+dwDelBuffBusyIndex);
            dwDelBuffFreeCount++;

```

```

dwDelBuffBusyIndex++;
if
(dwDelBuffBusyIndex == dwDelBuffSize) // wrap-
around if at end of buffer
    dwDelBuffBusyIndex = 0;

LeaveCriticalSection(&DelBuffCriticalSection);

pDeliveryData->w_id = delivery.w_id;
pDeliveryData->o_carrier_id =
delivery.o_carrier_id;

txnDeliRec.w_id = pDeliveryData->w_id;
txnDeliRec.o_carrier_id = pDeliveryData-
>o_carrier_id;

txnDeliRec.TxnStartT0 =
Get64BitTime(&delivery.queue);

GetLocalTime(
&trans_start );
pTxn-
>Delivery();
GetLocalTime(
&trans_end );

//log txn
txnDeliRec.TxnStatus = ERR_SUCCESS;
for (int i=0;
i<10; i++)
    txnDeliRec.o_id[i] = pDeliveryData-
>o_id[i];

txnDeliRec.DeltaT4 =
(int)(Get64BitTime(&trans_end) -
txnDeliRec.TxnStartT0);

txnDeliRec.DeltaTxnExec =
(int)(Get64BitTime(&trans_end) -
Get64BitTime(&trans_start));

if
(txnDeliLog != NULL)
    txnDeliLog->WriteToLog(&txnDeliRec);
}
catch (CBaseErr *e)
{
    char szTmp[1024];

```

```

        wsprintf( szTmp, "%s
Error (code %d) in Delivery Txn thread. %s",
e->ErrorTypeStr(), e->ErrorNum(), e->ErrorText() );
WriteMessageToEventLog(
szTmp );

// log the error txn
txnDeliRec.TxnStatus =
e->ErrorType();
if (txnDeliLog != NULL)
    txnDeliLog-
>WriteToLog(&txnDeliRec);

delete e;
}
catch (...)
{
    // unhandled exception;
shouldn't happen; not much we can do...
    WriteMessageToEventLog(TEXT("Unhandled
exception caught in DeliveryWorkerThread.));
}
}
ErrorExit:
if (Reg.dwConnectDelay > 0)
{
    // Synchronize disconnect (for
VIA)
    //
    EnterCriticalSection(&hConnectCriticalSecti
on);

    Sleep(Reg.dwConnectDelay);

    delete pTxn;

    LeaveCriticalSection(&hConnectCriticalSecti
on);
}
_endthread();
}
/* FUNCTION: PostDeliveryInfo
*
* PURPOSE: This function enters the delivery
txn into the deferred delivery buffer.
*
* RETURNS: BOOL FALSE
delivery information posted successfully
TRUE error cannot post delivery info
*/
BOOL PostDeliveryInfo(long w_id, short o_carrier_id)
{

```

```

    BOOL bError;
    EnterCriticalSection(&DelBuffCriticalSectio
n);
    if (dwDelBuffFreeCount > 0)
    {
        bError = FALSE;
        (pDelBuff+dwDelBuffFreeIndex)-
>w_id = w_id;
        (pDelBuff+dwDelBuffFreeIndex)-
>o_carrier_id = o_carrier_id;

        GetLocalTime(&(pDelBuff+dwDelBuffFreeIndex)
->queue);

        dwDelBuffFreeCount--;
        dwDelBuffFreeIndex++;
        if (dwDelBuffFreeIndex ==
dwDelBuffSize)
            dwDelBuffFreeIndex = 0;
        // wrap-around if at end of
buffer
    }
    else
        // No free buffers. Return an
error, which indicates that the delivery buffer is
full.
        // Most likely, the number of
delivery worker threads needs to be increased to keep
up
        // with the txn rate.
        bError = TRUE;
    LeaveCriticalSection(&DelBuffCriticalSectio
n);

    if (!bError)
        // increment worker semaphore to
wake up a worker thread
        ReleaseSemaphore(
hWorkerSemaphore, 1, NULL );
    return bError;
}
/* FUNCTION: ProcessQueryString
*
* PURPOSE: This function extracts the
relevant information out of the http command passed
in from
the browser.
*
* COMMENTS: If this is the initial connection
i.e. client is at welcome screen then
there will
not be a terminal id or current form id. If this is
the case
then the
pTermid and pFormid return values are undefined.
*/

```

```

void ProcessQueryString(EXTENSION_CONTROL_BLOCK
*pECB, int *pCmd, int *pFormId, int *pTermId, int
*pSyncId)
{
    char *ptr = pECB->lpszQueryString;
    char szBuffer[25];
    int i;

    //allowable client command strings i.e.
CMD=command
    static char *szCmds[] =
    {
        "Process", "..NewOrder..",
        "..Payment..", "..Delivery..", "..Order-Status..",
        "..Stock-Level..",
        "..Exit..", "Submit", "Menu",
        "Clear", "Stats", ""
    };

    *pCmd = 0; // default is
the login screen
    *pTermId = 0;

    // if no params (i.e., empty query string),
then return login screen
    if (strlen(pECB->lpszQueryString) == 0)
        return;

    // parse FORMID, TERMID, and SYNCID
    *pFormId = GetIntKeyValue(&ptr, "FORMID",
NO_ERR, NO_ERR);
    *pTermId = GetIntKeyValue(&ptr, "TERMID",
NO_ERR, NO_ERR);
    *pSyncId = GetIntKeyValue(&ptr, "SYNCID",
NO_ERR, NO_ERR);

    // parse CMD
    GetKeyValue(&ptr, "CMD", szBuffer,
sizeof(szBuffer), ERR_COMMAND_UNDEFINED);

    // see which command it matches
for(i=0; ; i++)
    {
        if (szCmds[i][0] == 0)
            // no more; no match;
return error
            throw new CWBCLNT_ERR(
ERR_COMMAND_UNDEFINED );
        if ( !strcmp(szCmds[i], szBuffer) )
        {
            *pCmd = i+1;
            break;
        }
    }
}
/* FUNCTION: void WelcomeForm
*
*/

```

```

void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer)
{
    char szTmp[1024];

    //welcome to tpc-c html form buffer, this
is first form client sees.
    strcpy( szBuffer,
" <HTML><HEAD><TITLE>TPC-C Web
Client</TITLE></HEAD><BODY>"

" <B><BIG>Microsoft TPC-C Web Client (ver
4.20)</BIG></B> <BR> <BR>"

" <font face=\"Courier New\"><PRE>"

"Compiled: " __DATE__ ", " __TIME__ <BR>"

"Source: " __FILE__ " (" __TIMESTAMP__ )"

<BR>"

" </PRE></font>"

" <FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"

" <INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">"

" <INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">"

" <INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"1\">"

" <INPUT TYPE=\"hidden\" NAME=\"TERMINID\"
VALUE=\"0\">"

" <INPUT TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"0\">"

" <INPUT TYPE=\"hidden\" NAME=\"VERSION\"
VALUE=\" " WEBCLIENT_VERSION " \">"
);

    sprintf( szTmp,
"Configuration
Settings: <BR><font face=\"Courier New\"
color=\"blue\"><PRE>"

"Txn Monitor          = <B>%s</B><BR>"

"Database protocol    = <B>%s</B><BR>"

"Max Connections     = <B>%d</B><BR>"
"#
of Delivery Threads = <B>%d</B><BR>"

"Max Pending Deliveries = <B>%d</B><BR>"

szTxnMonNames[Reg.eTxnMon],
szDBNames[Reg.eDB_Protocol],

```

```

    Reg.dwMaxConnections,
dwNumDeliveryThreads, dwDelBuffSize );
    strcat( szBuffer, szTmp);

    if (Reg.eTxnMon == COM)
    {
        sprintf( szTmp,
"COM Single
Pool = <B>%s</B><BR>",
Reg.bCOM_SinglePool ?
"YES" : "NO" );
        strcat( szBuffer, szTmp);
    }
    strcat( szBuffer, "</PRE></font>");

    if (Reg.eTxnMon == None)
        // connection options may be
specified when not using a txn monitor
        sprintf( szTmp,
"Please enter
your database options for this connection:<BR>"

" <font face=\"Courier New\"
color=\"blue\"><PRE>"

"DB Server = <INPUT NAME=\"db_server\"
SIZE=20 VALUE=\"%s\"><BR>"

"DB User ID = <INPUT NAME=\"db_user\"
SIZE=20 VALUE=\"%s\"><BR>"

"DB Password = <INPUT NAME=\"db_passwd\"
SIZE=20 VALUE=\"%s\"><BR>"

"DB Name = <INPUT NAME=\"db_name\"
SIZE=20 VALUE=\"%s\"><BR>"

" </PRE></font>"

Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
Reg.szDbName );
    else
        // if using a txn monitor,
connection options are determined from registry;
can't
        // set per user. show options
fyi
        sprintf( szTmp,
"Database
options which will be used by the transaction
monitor:<BR>"

" <font face=\"Courier New\"
color=\"blue\"><PRE>"

"DB Server          = <B>%s</B><BR>"

"DB User ID        = <B>%s</B><BR>"

"DB Password       = <B>%s</B><BR>"

"DB Name           = <B>%s</B><BR>"

" </PRE></font>"

```

```

    Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
    Reg.szDbName );
        strcat( szBuffer, szTmp);

        sprintf( szTmp,
"Please enter your
Warehouse and District for this session:<BR>"

" <font face=\"Courier New\"
color=\"blue\"><PRE>" );
        strcat( szBuffer, szTmp);
        strcat( szBuffer,
"Warehouse ID = <INPUT
NAME=\"w_id\" SIZE=6><BR>"

"District ID = <INPUT NAME=\"d_id\"
SIZE=2><BR>"

" </PRE></font><HR>"

" <INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Submit\">"

" </FORM></BODY></HTML>");
    }

/* FUNCTION: SubmitCmd
 *
 * PURPOSE: This function allocated a new
terminal id in the Term structure array.
 *
 */

void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer)
{
    int          iNewTerm;
    char        *ptr = pECB->lpszQueryString;

    char        szVersion[32] = { 0 };
    char        szServer[32] = { 0 };
    char        szUser[32] = "sa";

    char        szPassword[32] = { 0 };
    char        szDatabase[32] = "tpcc";

    // validate version field; the version
field ensures that the RTE is synchronized with the
web client
    GetKeyValue(&ptr, "VERSION", szVersion,
sizeof(szVersion), ERR_VERSION_MISMATCH);
    if ( strcmp( szVersion, WEBCLIENT_VERSION )
)
        throw new CWBCLNT_ERR(
ERR_VERSION_MISMATCH );

    if (Reg.eTxnMon == None)
    {
        // parse Server name
        GetKeyValue(&ptr, "db_server",
szServer, sizeof(szServer), ERR_NO_SERVER_SPECIFIED);
        // parse User name

```

```

        GetKeyValue(&ptr, "db_user",
szUser, sizeof(szUser), NO_ERR);
        // parse Password
        GetKeyValue(&ptr, "db_passwd",
szPassword, sizeof(szPassword), NO_ERR);
        // parse Database name
        GetKeyValue(&ptr, "db_name",
szDatabase, sizeof(szDatabase), NO_ERR);
    }

    // parse warehouse ID
    int w_id = GetIntKeyValue(&ptr, "w_id",
ERR_HTML_ILL_FORMED, ERR_W_ID_INVALID);
    if ( w_id < 1 )
        throw new CWEBCLNT_ERR(
ERR_W_ID_INVALID );

    // parse district ID
    int d_id = GetIntKeyValue(&ptr, "d_id",
ERR_HTML_ILL_FORMED, ERR_D_ID_INVALID);
    if ( d_id < 1 || d_id > 10 )
        throw new CWEBCLNT_ERR(
ERR_D_ID_INVALID );

    iNewTerm = TermAdd();

    Term.pClientData[iNewTerm].w_id = w_id;
    Term.pClientData[iNewTerm].d_id = d_id;

    try
    {
        if (Reg.eTxnMon == TUXEDO)

            Term.pClientData[iNewTerm].pTxn =
pCTPCC_TUXEDO_new();
        else if (Reg.eTxnMon == ENCINA)

            Term.pClientData[iNewTerm].pTxn =
pCTPCC_ENCINA_new();
        else if (Reg.eTxnMon == COM)

            Term.pClientData[iNewTerm].pTxn =
pCTPCC_COM_new( Reg.bCOM_SinglePool );
        else if (Reg.eDB_Protocol ==
ODBC)

            Term.pClientData[iNewTerm].pTxn =
pCTPCC_ODBC_new( szServer, szUser, szPassword,
szMyComputerName,

szDatabase, Reg.szSPPrefix,

Reg.bCallNoDuplicatesNewOrder );
        else if (Reg.eDB_Protocol ==
DBLIB)

```

```

        Term.pClientData[iNewTerm].pTxn =
pCTPCC_DBLIB_new( szServer, szUser, szPassword,
szMyComputerName, szDatabase );
    }
    catch (...)
    {
        TermDelete(iNewTerm);
        throw; // pass
exception upward
    }

    MakeMainMenuForm(iNewTerm,
Term.pClientData[iNewTerm].iSyncId, szBuffer);
}

/* FUNCTION: StatsCmd
 *
 * PURPOSE: This function returns to the
browser the total number of active terminal ids.
 * This routine is for
development/debugging purposes.
 */

void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer)
{
    int i;
    int iTotals;

    EnterCriticalSection(&TermCriticalSection);

    iTotals = 0;
    for(i=0; i<Term.iNumEntries; i++)
    {
        if (Term.pClientData[i].iNextFree
== -1)

            iTotals++;

        LeaveCriticalSection(&TermCriticalSection);

        wsprintf( szBuffer,

" <HTML><HEAD><TITLE>TPC-C Web Client
Stats</TITLE></HEAD>"

" <BODY><B><BIG> Total
Active Connections: %d </BIG></B><BR></BODY></HTML>"
, iTotals );
    }

    char *CWEBCLNT_ERR::ErrorText()
    {
        static SERRORMSG errorMsgs[] =
        {
            { ERR_COMMAND_UNDEFINED,

"Command undefined."
},

```

```

        { ERR_D_ID_INVALID,

"Invalid District ID Must be 1 to 10."
},
        {
ERR_DELIVERY_CARRIER_ID_RANGE,

"Delivery Carrier ID out of range
must be 1 - 10."
},
        {
ERR_DELIVERY_CARRIER_INVALID,

"Delivery Carrier ID invalid must be
numeric 1 - 10."
},
        {
ERR_DELIVERY_MISSING_OCD_KEY,

"Delivery missing Carrier ID key \"OCD*\"."
},
        {
ERR_DELIVERY_THREAD_FAILED,

"Could not start delivery worker
thread."
},
        {
ERR_GETPROCADDR_FAILED,

"Could not map proc in DLL. GetProcAddr
error. DLL="
},
        {
ERR_HTML_ILL_FORMED,

"Required key field is missing from HTML
string."
},
        {
ERR_INVALID_SYNC_CONNECTION,

"Invalid Terminal Sync ID."
},
        {
ERR_INVALID_TERMID,

"Invalid Terminal ID."
},
        {
ERR_LOADDLL_FAILED,

"Load of DLL failed. DLL="
},
        {
ERR_MAX_CONNECTIONS_EXCEEDED,

"No connections available. Max Connections
is probably too low."
},
        {
ERR_MISSING_REGISTRY_ENTRIES,

"Required registry entries are missing.
Rerun INSTALL to correct."
},
        {
ERR_NEWORDER_CUSTOMER_INVALID,

"New Order customer id invalid
data type, range = 1 to 3000."
},
        {
ERR_NEWORDER_CUSTOMER_KEY,

```

```

        "New Order missing Customer key
\CID*\."
    },
    {
        ERR_NEWORDER_DISTRICT_INVALID,
        "New Order District ID Invalid
range 1 - 10."
    },
    {
        ERR_NEWORDER_FORM_MISSING_DID,
        "New Order missing District key
\DID*\."
    },
    {
        ERR_NEWORDER_ITEMID_INVALID,
        "New Order Item Id is wrong data type, must
be numeric."
    },
    {
        ERR_NEWORDER_ITEMID_RANGE,
        "New Order Item Id is out of
range. Range = 1 to 999999."
    },
    {
        ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
        "New Order Item Id field entered without a
corresponding Supp_W."
    },
    {
        ERR_NEWORDER_MISSING_IID_KEY,
        "New Order missing Item Id key \IID*\."
    },
    {
        ERR_NEWORDER_MISSING_QTY_KEY,
        "New Order Missing Qty key \Qty##*\."
    },
    {
        ERR_NEWORDER_MISSING_SUPPW_KEY,
        "New Order missing Supp_W key
\SP##*\."
    },
    {
        ERR_NEWORDER_NOITEMS_ENTERED,
        "New Order No order lines entered."
    },
    {
        ERR_NEWORDER_QTY_INVALID,
        "New Order Qty invalid must be
numeric range 1 - 99."
    },
    {
        ERR_NEWORDER_QTY_RANGE,
        "New Order Qty is out of range. Range = 1
to 99."
    },
    {
        ERR_NEWORDER_QTY_WITHOUT_SUPPW,
        "New Order Qty field entered
without a corresponding Supp_W."
    },
    {
        ERR_NEWORDER_SUPPW_INVALID,
        "New Order Supp_W invalid data

```

```

type must be numeric."
    },
    {
        ERR_NO_SERVER_SPECIFIED,
        "No Server name specified."
    },
    {
        ERR_ORDERSTATUS_CID_AND_CLT,
        "Order Status Only Customer ID or Last Name
may be entered, not both."
    },
    {
        ERR_ORDERSTATUS_CID_INVALID,
        "Order Status Customer ID invalid, range
must be numeric 1 - 3000."
    },
    {
        ERR_ORDERSTATUS_CLT_RANGE,
        "Order Status Customer last name
longer than 16 characters."
    },
    {
        ERR_ORDERSTATUS_DID_INVALID,
        "Order Status District invalid, value must
be numeric 1 - 10."
    },
    {
        ERR_ORDERSTATUS_MISSING_CID_CLT,
        "Order Status Either Customer ID or Last
Name must be entered."
    },
    {
        ERR_ORDERSTATUS_MISSING_CID_KEY,
        "Order Status missing Customer key
\CID*\."
    },
    {
        ERR_ORDERSTATUS_MISSING_CLT_KEY,
        "Order Status missing Customer Last Name
key \CLT*\."
    },
    {
        ERR_ORDERSTATUS_MISSING_DID_KEY,
        "Order Status missing District key
\DID*\."
    },
    {
        ERR_PAYMENT_CDI_INVALID,
        "Payment Customer district
invalid must be numeric."
    },
    {
        ERR_PAYMENT_CID_AND_CLT,
        "Payment Only Customer ID or Last
Name may be entered, not both."
    },
    {
        ERR_PAYMENT_CUSTOMER_INVALID,
        "Payment Customer data type invalid, must
be numeric."
    },
    {
        ERR_PAYMENT_CWI_INVALID,
        "Payment Customer Warehouse
invalid, must be numeric."
    },
    {
        ERR_PAYMENT_DISTRICT_INVALID,

```

```

        "Payment District ID is invalid, must be 1
- 10."
    },
    {
        ERR_PAYMENT_HAM_INVALID,
        "Payment Amount invalid data type
must be numeric."
    },
    {
        ERR_PAYMENT_HAM_RANGE,
        "Payment Amount out of range, 0 - 9999.99."
    },
    {
        ERR_PAYMENT_LAST_NAME_TO_LONG,
        "Payment Customer last name
longer than 16 characters."
    },
    {
        ERR_PAYMENT_MISSING_CDI_KEY,
        "Payment missing Customer district key
\CDI*\."
    },
    {
        ERR_PAYMENT_MISSING_CID_CLT,
        "Payment Either Customer ID or Last Name
must be entered."
    },
    {
        ERR_PAYMENT_MISSING_CID_KEY,
        "Payment missing Customer Key \CID*\."
    },
    {
        ERR_PAYMENT_MISSING_CLT_KEY,
        "Payment missing Customer Last Name key
\CLT*\."
    },
    {
        ERR_PAYMENT_MISSING_CWI_KEY,
        "Payment missing Customer Warehouse key
\CWI*\."
    },
    {
        ERR_PAYMENT_MISSING_DID_KEY,
        "Payment missing District Key \DID*\."
    },
    {
        ERR_PAYMENT_MISSING_HAM_KEY,
        "Payment missing Amount key \HAM*\."
    },
    {
        ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
        "Stock Level; missing Threshold key
\TT*\."
    },
    {
        ERR_STOCKLEVEL_THRESHOLD_INVALID,
        "Stock Level; Threshold value must be in
the range = 1 - 99."
    },

```

```

        {
            ERR_STOCKLEVEL_THRESHOLD_RANGE,
            "Stock Level Threshold out of
range, range must be 1 - 99." },
        {
            ERR_VERSION_MISMATCH,
            "Invalid version field. RTE and Web Client
are probably out of sync." },
        {
            ERR_W_ID_INVALID,
            "Invalid Warehouse ID."
        }
    },
    {
        0,
        ""
    }
};

char szTmp[256];
int i = 0;
while (TRUE)
{
    if (errorMsgs[i].szMsg[0] == 0)
    {
        strcpy( szTmp, "Unknown
error number." );
        break;
    }
    if (m_Error ==
errorMsgs[i].iError)
    {
        strcpy( szTmp,
errorMsgs[i].szMsg );
        break;
    }
    i++;
}

if (m_szTextDetail)
    strcat( szTmp, m_szTextDetail );
if (m_SystemErr)
    wsprintf( szTmp+strlen(szTmp), "
Error=%d", m_SystemErr );

m_szErrorText = new char[strlen(szTmp)+1];
strcpy( m_szErrorText, szTmp );
return m_szErrorText;
}

/* FUNCTION: GetKeyValue
*
* PURPOSE: This function parses a http
formatted string for specific key values.
*
* ARGUMENTS: char
*pQueryString http string from client
browser

```

```

*
* char key
value to look for *pKey
*
* char
* *pValue
character array into which to place key's
value
*
* int
* iMax
maximum length of key value array.
*
* WEBERROR
err
error value to throw
*
* RETURNS: nothing.
*
* ERROR: if (the pKey value is not found)
then
*
* if
(err == 0)
*
* return (empty string)
*
* else
*
* throw CWEBCLNT_ERR(err)
*
* COMMENTS: http keys are formatted either
KEY=value& or KEY=value0. This DLL formats
*
* TPC-C input
fields in such a manner that the keys can be
extracted in the
*
* above manner.
*/

void GetKeyValue(char **pQueryString, char *pKey,
char *pValue, int iMax, WEBERROR err)
{
    char *ptr;

    if ( !(ptr=strstr(*pQueryString, pKey)) )
        goto ErrorExit;
    ptr += strlen(pKey);
    if ( *ptr != '=' )
        goto ErrorExit;
    ptr++;

    iMax--; // one position is for terminating
null
    while( *ptr && *ptr != '&' && iMax)
    {
        *pValue++ = *ptr++;
        iMax--;
    }
    *pValue = 0; // terminating null

    *pQueryString = ptr;
    return;

ErrorExit:
    if (err != NO_ERR)

```

```

        throw new CWEBCLNT_ERR( err );
        *pValue = 0; // return empty result string
    }

/* FUNCTION: GetIntKeyValue
*
* PURPOSE: This function parses a http
formatted string for a specific key value.
*
* ARGUMENTS: char
*pQueryString http string from client
browser
*
* char key
value to look for *pKey
*
* WEBERROR
* NoKeyErr error value to throw if
key not found
*
* WEBERROR
* NotIntErr error value to throw if
value not numeric
*
* RETURNS: integer
*
* ERROR: if (the pKey value is not found)
then
*
* if
(NoKeyErr != NO_ERR)
*
* throw CWEBCLNT_ERR(err)
*
* else
*
* return 0
*
* else if (non-
numeric char found) then
*
* if
(NotIntErr != NO_ERR) then
*
* throw CWEBCLNT_ERR(err)
*
* else
*
* return 0
*
* COMMENTS: http keys are formatted either
KEY=value& or KEY=value0. This DLL formats
*
* TPC-C input
fields in such a manner that the keys can be
extracted in the
*
* above manner.
*/

int GetIntKeyValue(char **pQueryString, char *pKey,
WEBERROR NoKeyErr, WEBERROR NotIntErr)
{
    char *ptr0;
    char *ptr;

    if ( !(ptr=strstr(*pQueryString, pKey)) )
        goto ErrorNoKey;

```

```

ptr += strlen(pKey);
if ( *ptr != '=' )
    goto ErrorNoKey;
ptr++;

ptr0 = ptr;          // remember
starting point
// scan string until a terminator (null or
&) or a non-digit
while( *ptr && *ptr != '&' && isdigit(*ptr)
)
    ptr++;

// make sure we stopped scanning for the
right reason
if ((ptr0 == ptr) || (*ptr && *ptr != '&'))
{
    if (NotIntErr != NO_ERR)
        throw new CWBCLNT_ERR(
NoKeyErr );
    return 0;
}

*pQueryString = ptr;
return atoi(ptr0);

ErrorNoKey:
if (NoKeyErr != NO_ERR)
    throw new CWBCLNT_ERR( NoKeyErr
);
return 0;
}

/* FUNCTION: TermInit
*
* PURPOSE:      This function initializes the
client terminal structure; it is called when the
TPCC.DLL
*
*               is first loaded by the
inet service.
*
*/

void TermInit(void)
{
    EnterCriticalSection(&TermCriticalSection);

    Term.iMasterSyncId = 1;
    Term.iNumEntries =
Reg.dwMaxConnections+1;

    Term.pClientData = NULL;
    Term.pClientData =
(PCLIENTDATA)malloc(Term.iNumEntries *
sizeof(CLIENTDATA));
    if (Term.pClientData == NULL)
    {
        LeaveCriticalSection(&TermCriticalSection);
        throw new CWBCLNT_ERR(
ERR_MEM_ALLOC_FAILED );
    }
}

```

```

}

ZeroMemory( Term.pClientData,
Term.iNumEntries * sizeof(CLIENTDATA) );

Term.iFreeList =
Term.iNumEntries-1;
// build free list
// note: Term.pClientData[0].iNextFree gets
set to -1, which marks it as "in use".
// This is intentional, as the zero
entry is used as an anchor and never
// allocated as an actual
terminal.
for(int i=0; i<Term.iNumEntries; i++)
    Term.pClientData[i].iNextFree =
i-1;

LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermDeleteAll
*
* PURPOSE:      This function frees allocated
resources associated with the terminal structure.
*
* ARGUMENTS:   none
*
* RETURNS:     None
*
* COMMENTS:    This function is called only when
the inet service unloads the TPCC.DLL
*
*/

void TermDeleteAll(void)
{
    EnterCriticalSection(&TermCriticalSection);

    for(int i=1; i<Term.iNumEntries; i++)
    {
        if (Term.pClientData[i].iNextFree
== -1)
            delete
Term.pClientData[i].pTxn;

        Term.iFreeList = 0;
        Term.iNumEntries = 0;
        if ( Term.pClientData )
            free(Term.pClientData);
        Term.pClientData = NULL;

        LeaveCriticalSection(&TermCriticalSection);
    }

/* FUNCTION: TermAdd
*
* PURPOSE:      This function assigns a terminal
id which is used to identify a client browser.
*

```

```

* RETURNS:     int
               assigned terminal id
*/

int TermAdd(void)
{
    DWORD i;
    int iNewTerm, iTickCount;

    if (Term.iNumEntries == 0)
        return -1;

    EnterCriticalSection(&TermCriticalSection);
    if (Term.iFreeList != 0)
    {
        // position is available
        iNewTerm = Term.iFreeList;
        Term.iFreeList =
Term.pClientData[iNewTerm].iNextFree;

        Term.pClientData[iNewTerm].iNextFree = -1;
        // indicates this position is in use
    }
    else
    {
        // no open slots, so find the
slot that hasn't been used in the longest time and
reuse it
        for(iNewTerm=1, i=1,
iTickCount=0x7FFFFFFF; i<Reg.dwMaxConnections; i++)
        {
            if (iTickCount >
Term.pClientData[i].iTickCount)
            {
                iTickCount =
Term.pClientData[i].iTickCount;
                iNewTerm = i;
            }
        }
        // if oldest term is less than
one minute old, it probably means that more
connections
        // are being attempted than were
specified as "Max Connections" at install. In this
case,
        // do not bump existing
connection; instead, return error to requestor.
        if ((GetTickCount() - iTickCount)
< 60000)
        {
            LeaveCriticalSection(&TermCriticalSection);
            throw new CWBCLNT_ERR(
ERR_MAX_CONNECTIONS_EXCEEDED );
        }
    }

    Term.pClientData[iNewTerm].iTickCount =
GetTickCount();
    Term.pClientData[iNewTerm].iSyncId =
Term.iMasterSyncId++;
    Term.pClientData[iNewTerm].pTxn = NULL;
}

```

```

        LeaveCriticalSection(&TermCriticalSection);
        return iNewTerm;
    }

/* FUNCTION: TermDelete
 *
 * PURPOSE:      This function makes a terminal
entry in the Term array available for reuse.
 *
 * ARGUMENTS:   int          id
                Terminal id of client exiting
 *
 */
void TermDelete(int id)
{
    if ( id > 0 && id < Term.iNumEntries )
    {
        delete Term.pClientData[id].pTxn;

        // put onto free list

        EnterCriticalSection(&TermCriticalSection);

        Term.pClientData[id].iNextFree =
Term.iFreeList;
        Term.iFreeList = id;

        LeaveCriticalSection(&TermCriticalSection);
    }

/* FUNCTION: MakeErrorForm
 */
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int
iType, int iErrorNum, int iTermId, int iSyncId, char
*szErrorText, char *szBuffer )
{
    wsprintf(szBuffer,
"HTML<>HEAD<>TITLE>TPC-C
Error</TITLE></HEAD><BODY>
"FORM ACTION="tpcc.dll"
METHOD="GET">
"INPUT TYPE="hidden"
NAME="STATUSID" VALUE="%d">
"INPUT TYPE="hidden"
NAME="ERROR" VALUE="%d">
"INPUT TYPE="hidden"
NAME="FORMID" VALUE="%d">
"INPUT TYPE="hidden"
NAME="TERMIN" VALUE="%d">
"INPUT TYPE="hidden"
NAME="SYNCID" VALUE="%d">
" BOLD>An Error
Occurred</BOLD><BR><BR>
" %s"

```

```

"BR><BR><HR>
"INPUT TYPE="submit"
NAME="CMD" VALUE="..NewOrder..">
"INPUT TYPE="submit"
NAME="CMD" VALUE="..Payment..">
"INPUT TYPE="submit"
NAME="CMD" VALUE="..Delivery..">
"INPUT TYPE="submit"
NAME="CMD" VALUE="..Order-Status..">
"INPUT TYPE="submit"
NAME="CMD" VALUE="..Stock-Level..">
"INPUT TYPE="submit"
NAME="CMD" VALUE="..Exit..">
</FORM></BODY></HTML>
, iType, iErrorNum,
MAIN_MENU_FORM, iTermId, iSyncId, szErrorText );
}

/* FUNCTION: MakeMainMenuForm
 */
void MakeMainMenuForm(int iTermId, int iSyncId, char
*szForm)
{
    wsprintf(szForm,
"HTML<>HEAD<>TITLE>TPC-C Main
Menu</TITLE></HEAD><BODY>
"Select Desired
Transaction.<BR><HR>
"FORM ACTION="tpcc.dll"
METHOD="GET">
"INPUT TYPE="hidden"
NAME="STATUSID" VALUE="0">
"INPUT TYPE="hidden"
NAME="ERROR" VALUE="0">
"INPUT TYPE="hidden"
NAME="FORMID" VALUE="%d">
"INPUT TYPE="hidden"
NAME="TERMIN" VALUE="%d">
"INPUT TYPE="hidden"
NAME="SYNCID" VALUE="%d">
"INPUT TYPE="submit"
NAME="CMD" VALUE="..NewOrder..">
"INPUT TYPE="submit"
NAME="CMD" VALUE="..Payment..">
"INPUT TYPE="submit"
NAME="CMD" VALUE="..Delivery..">
"INPUT TYPE="submit"
NAME="CMD" VALUE="..Order-Status..">
"INPUT TYPE="submit"
NAME="CMD" VALUE="..Stock-Level..">
"INPUT TYPE="submit"
NAME="CMD" VALUE="..Exit..">
</FORM></BODY></HTML>
, MAIN_MENU_FORM, iTermId,

iSyncId);
}

/* FUNCTION: MakeStockLevelForm
 *

```

```

 * PURPOSE:      This function constructs the
Stock Level HTML page.
 *
 * COMMENTS:     The internal client buffer is
created when the terminal id is assigned and should
not
                be freed
                except when the client terminal id is no longer
needed.
 */
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA
*pStockLevelData, BOOL bInput, char *szForm)
{
    int          c;

    c = wsprintf(szForm,
"HTML<>HEAD<>TITLE>TPC-C Stock
Level</TITLE></HEAD><FORM ACTION="tpcc.dll"
METHOD="GET">
"INPUT TYPE="hidden"
NAME="STATUSID" VALUE="0">
"INPUT TYPE="hidden"
NAME="ERROR" VALUE="0">
"INPUT TYPE="hidden"
NAME="FORMID" VALUE="%d">
"INPUT TYPE="hidden"
NAME="TERMIN" VALUE="%d">
"INPUT TYPE="hidden"
NAME="SYNCID" VALUE="%d">
<PRE><font face="Courier">
Stock-Level<BR>
"Warehouse: %6.6d District:
%2.2d<BR> <BR>",
STOCK_LEVEL_FORM, iTermId,
Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id,
Term.pClientData[iTermId].d_id);

    if ( bInput )
    {
        strcpy(szForm+c,
"Stock Level Threshold:
<INPUT NAME="TT*" SIZE=2><BR> <BR>
"low stock:
</font><BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR></PRE><HR>
"INPUT TYPE="submit"
NAME="CMD" VALUE="Process">
"INPUT TYPE="submit"
NAME="CMD" VALUE="Menu">
" </FORM></HTML>" );
    }
    else
    {
        wsprintf(szForm+c,
"Stock Level Threshold:
%2.2d<BR> <BR>"

```



```

"low stock:
%3.3d</font> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR>"
" <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR></PRE><HR>"
"<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..NewOrder..\">"
"<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
"<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
"<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
"<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
"<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
"</FORM></HTML>"
pStockLevelData-
>threshold, pStockLevelData->low_stock);
}
}
/* FUNCTION: MakeNewOrderForm
*
* COMMENTS: The internal client buffer is
created when the terminal id is assigned and should
not
* be freed
except when the client terminal id is no longer
needed.
*/

void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA
*pNewOrderData, BOOL bInput, char *szForm)
{
int i, c;
BOOL bValid;
static char szBR[] = " <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR>";

if (!bInput)
assert( pNewOrderData-
>exec_status_code == eOK || pNewOrderData-
>exec_status_code == eInvalidItem );

bValid = (bInput || (pNewOrderData-
>exec_status_code == eOK));

c = sprintf(szForm,
"<HTML><HEAD><TITLE>TPC-C New
Order</TITLE></HEAD><BODY>"
"<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
"<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"%d\">"
"<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
"<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"

```

```

"<INPUT TYPE=\"hidden\"
NAME=\"TERMINID\" VALUE=\"%d\">"
"<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
"<PRE><font face=\"Courier\">
New Order<BR>"
, bValid ? 0 : ERR_BAD_ITEM_ID,
NEW_ORDER_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);

if ( bInput )
{
c += sprintf(szForm+c,
"Warehouse: %6.6d ", Term.pClientData[iTermId].w_id
);

strcpy( szForm+c,
"District: <INPUT
NAME=\"DID*\" SIZE=1>
Date:<BR>"
"Customer: <INPUT
NAME=\"CID*\" SIZE=4> Name:
Credit: %Disc:<BR>"
"Order Number:
Number of Lines: W_tax: D_tax:<BR>
<BR>"
" Supp_W Item_Id Item
Name Qty Stock B/G Price
Amount<BR>"
"<INPUT
NAME=\"SP00*\" SIZE=4> <INPUT NAME=\"IID00*\"
SIZE=6> <INPUT
NAME=\"Qty00*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP01*\" SIZE=4> <INPUT NAME=\"IID01*\"
SIZE=6> <INPUT
NAME=\"Qty01*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP02*\" SIZE=4> <INPUT NAME=\"IID02*\"
SIZE=6> <INPUT
NAME=\"Qty02*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP03*\" SIZE=4> <INPUT NAME=\"IID03*\"
SIZE=6> <INPUT
NAME=\"Qty03*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP04*\" SIZE=4> <INPUT NAME=\"IID04*\"
SIZE=6> <INPUT
NAME=\"Qty04*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP05*\" SIZE=4> <INPUT NAME=\"IID05*\"
SIZE=6> <INPUT
NAME=\"Qty05*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP06*\" SIZE=4> <INPUT NAME=\"IID06*\"
SIZE=6> <INPUT
NAME=\"Qty06*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP07*\" SIZE=4> <INPUT NAME=\"IID07*\"
SIZE=6> <INPUT
NAME=\"Qty07*\" SIZE=1><BR>"

```

```

" <INPUT
NAME=\"SP08*\" SIZE=4> <INPUT NAME=\"IID08*\"
SIZE=6> <INPUT
NAME=\"Qty08*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP09*\" SIZE=4> <INPUT NAME=\"IID09*\"
SIZE=6> <INPUT
NAME=\"Qty09*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP10*\" SIZE=4> <INPUT NAME=\"IID10*\"
SIZE=6> <INPUT
NAME=\"Qty10*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP11*\" SIZE=4> <INPUT NAME=\"IID11*\"
SIZE=6> <INPUT
NAME=\"Qty11*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP12*\" SIZE=4> <INPUT NAME=\"IID12*\"
SIZE=6> <INPUT
NAME=\"Qty12*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP13*\" SIZE=4> <INPUT NAME=\"IID13*\"
SIZE=6> <INPUT
NAME=\"Qty13*\" SIZE=1><BR>"
"<INPUT
NAME=\"SP14*\" SIZE=4> <INPUT NAME=\"IID14*\"
SIZE=6> <INPUT
NAME=\"Qty14*\" SIZE=1><BR>"
"Execution Status:
Total:<BR>"
"</font></PRE><HR>"
"<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
"<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
"</FORM></HTML>"
);
}
else
{
c += sprintf(szForm+c,
"Warehouse: %6.6d District: %2.2d
Date: ",
pNewOrderData->w_id,
pNewOrderData->d_id);

if ( bValid )
{
c += sprintf(szForm+c,
"%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
pNewOrderData->o_entry_d.day,
pNewOrderData->o_entry_d.month,
pNewOrderData->o_entry_d.year,
pNewOrderData->o_entry_d.hour,
pNewOrderData->o_entry_d.minute,

```

```

    pNewOrderData->o_entry_d.second);
    }

    c += wsprintf(szForm+c,
"<BR>Customer:  %4.4d  Name: %-16s  Credit: %-2s
",
                pNewOrderData->c_id,
pNewOrderData->c_last, pNewOrderData->c_credit);

    if ( bValid )
    {
        c += sprintf(szForm+c,

"%%Disc: %5.2f          <BR>"

"Order Number: %8.8d  Number of Lines:
%2.2d          W_tax: %5.2f  D_tax: %5.2f <BR> <BR>"

" Supp_W  Item_Id  Item Name
Qty  Stock  B/G  Price      Amount<BR>",

100.0*pNewOrderData->c_discount,

pNewOrderData->o_id,

pNewOrderData->o_ol_cnt,

pNewOrderData->w_tax,          100.0 *
pNewOrderData->d_tax);          100.0 *

        for(i=0;
i<pNewOrderData->o_ol_cnt; i++)
        {
            c +=
sprintf(szForm+c, "%6.6d  %6.6d  %-24s  %2.2d
%3.3d  %1.1s  $%6.2f  $%7.2f <BR>",

pNewOrderData->OL[i].ol_supply_w_id,
pNewOrderData->OL[i].ol_i_id,
pNewOrderData->OL[i].ol_i_name,
pNewOrderData->OL[i].ol_quantity,
pNewOrderData->OL[i].ol_stock,
pNewOrderData->OL[i].ol_brand_generic,
pNewOrderData->OL[i].ol_i_price,
pNewOrderData->OL[i].ol_amount );
        }
    }
    else
    {
        c += wsprintf(szForm+c,
"%Disc:<BR>"

```

```

"Order
Number: %8.8d  Number of Lines:          W_tax:
D_tax:<BR> <BR>"

    Item_Id  Item Name          Qty  Stock  B/G
Price      Amount<BR>"

pNewOrderData->o_id);

        i = 0;
    }
    strncpy( szForm+c, szBR, (15-i)*5
);
    c += (15-i)*5;
    if ( bValid )
        c += sprintf(szForm+c,
"Execution Status: Transaction committed.
Total:  $%8.2f  ",

pNewOrderData->total_amount);
    else
        c += wsprintf(szForm+c,
"Execution Status: Item number is not valid.
Total:");

    strcpy(szForm+c,
"
<BR></font></PRE><HR>"

" <INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..NewOrder..\">"
" <INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
" <INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
" <INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
" <INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
" <INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
" </FORM></HTML>"

);
}

/* FUNCTION: MakePaymentForm
 *
 * COMMENTS:      The internal client buffer is
created when the terminal id is assigned and should
not
 *                be freed
 *                except when the client terminal id is no longer
needed.
 */

void MakePaymentForm(int iTermId, PAYMENT_DATA
*pPaymentData, BOOL bInput, char *szForm)
{
    int c;

```

```

    c = wsprintf(szForm,
" <HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>"
" <FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
" <INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
" <INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
" <INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"\">"
" <INPUT TYPE=\"hidden\"
NAME=\"TERMINID\" VALUE=\"\">"
" <INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"\">"
" <PRE><font face=\"Courier\">
Payment<BR>"
" Date: "
" , PAYMENT_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);

    if ( !bInput )
    {
        c += wsprintf(szForm+c, "%2.2d-
%2.2d-%4.4d %2.2d:%2.2d:%2.2d",

pPaymentData-
>h_date.day,          pPaymentData-
>h_date.month,        pPaymentData-
>h_date.year,         pPaymentData-
>h_date.hour,         pPaymentData-
>h_date.minute,       pPaymentData-
>h_date.second);
    }
    if ( bInput )
    {
        c += wsprintf(szForm+c,
"%6.6d"
" <BR> <BR>Warehouse:
"
District: <INPUT NAME=\"DID*\" SIZE=1><BR> <BR> <BR>
<BR> <BR>"
"Customer: <INPUT
NAME=\"CID*\" SIZE=4>"
"Cust-Warehouse: <INPUT
NAME=\"CWI*\" SIZE=4> "
"Cust-District: <INPUT
NAME=\"CDI*\" SIZE=1><BR>"
"Name:
<INPUT NAME=\"CLT*\" SIZE=16>
Since:<BR>"
"
Credit:<BR>"
"
Disc:<BR>"

```

```

Phone:<BR> <BR>"
"
"Amount Paid:
$<INPUT NAME="HAM*" SIZE=7> New Cust-
Balance:<BR>"
"Credit Limit:<BR>
<BR>Cust-Data: <BR> <BR> <BR> <BR>
<BR></font></PRE><HR>"
" <INPUT TYPE="submit\"
NAME="CMD\" VALUE="Process\"><INPUT TYPE="submit\"
NAME="CMD\" VALUE="Menu\">"
" </BODY></FORM></HTML>"
Term.pClientData[iTermId].w_id;
}
else
{
c += sprintf(szForm+c,
" <BR> <BR>Warehouse:
%6.6d District: %2.2d<BR>"
"%-20s
"%-20s
"%-20s %-2s %5.5s-%4.4s<BR> <BR>"
"Customer: %4.4d Cust-
Warehouse: %6.6d Cust-District: %2.2d<BR>"
"Name: %-16s %-2s %-
16s Since: %2.2d-%2.2d-%4.4d<BR>"
"%-20s
"Credit: %-2s<BR>"
Term.pClientData[iTermId].w_id, pPaymentData->d_id
pPaymentData->w_street_1, pPaymentData->d_street_1
pPaymentData->w_street_2, pPaymentData->d_street_2
pPaymentData->w_state, pPaymentData->w_zip,
pPaymentData->w_zip+5
pPaymentData->d_state, pPaymentData->d_city,
pPaymentData->d_state, pPaymentData->d_zip,
pPaymentData->d_zip+5
pPaymentData->c_w_id, pPaymentData->c_id,
pPaymentData->c_d_id, pPaymentData-
>c_first, pPaymentData->c_middle, pPaymentData-
>c_last
pPaymentData-
>c_since.day, pPaymentData->c_since.month,
pPaymentData->c_since.year
pPaymentData-
>c_street_1, pPaymentData->c_credit
);
c += sprintf(szForm+c,
"%-20s
%%Disc: %5.2f<BR>",

```

```

pPaymentData-
>c_street_2, 100.0*pPaymentData->c_discount);
c += sprintf(szForm+c,
"%-20s %-2s
%5.5s-%4.4s Phone: %6.6s-%3.3s-%3.3s-%4.4s<BR>
<BR>",
pPaymentData->c_city,
pPaymentData->c_state, pPaymentData->c_zip,
pPaymentData->c_zip+5,
pPaymentData->c_phone,
pPaymentData->c_phone+6, pPaymentData->c_phone+9,
pPaymentData->c_phone+12 );
c += sprintf(szForm+c,
"Amount Paid:
$%7.2f New Cust-Balance: $%14.2f<BR>"
"Credit Limit:
$%13.2f<BR> <BR>"
, pPaymentData-
>h_amount, pPaymentData->c_balance
, pPaymentData-
>c_credit_lim
);
if ( pPaymentData->c_credit[0] ==
'B' && pPaymentData->c_credit[1] == 'C' )
c += sprintf(szForm+c,
"%-50.50s<BR> %-
50.50s<BR> %-50.50s<BR>
pPaymentData->c_data, pPaymentData-
>c_data+50, pPaymentData->c_data+100, pPaymentData-
>c_data+150 );
else
strcpy(szForm+c, "Cust-
Data: <BR> <BR> <BR> <BR>");
strcat(szForm,
" <INPUT TYPE="submit\" NAME="CMD\"
VALUE="..NewOrder..\">"
" <INPUT TYPE="submit\" NAME="CMD\"
VALUE="..Payment..\">"
" <INPUT TYPE="submit\" NAME="CMD\"
VALUE="..Delivery..\">"
" <INPUT TYPE="submit\" NAME="CMD\"
VALUE="..Order-Status..\">"
" <INPUT TYPE="submit\" NAME="CMD\"
VALUE="..Stock-Level..\">"
" <INPUT TYPE="submit\" NAME="CMD\"
VALUE="..Exit..\">"

```

```

" </BODY></FORM></HTML>");
}
}
/* FUNCTION: MakeOrderStatusForm
*
* COMMENTS: The internal client buffer is
created when the terminal id is assigned and should
not
* be freed
* except when the client terminal id is no longer
needed.
*/
void MakeOrderStatusForm(int iTermId,
ORDER_STATUS_DATA *pOrderStatusData, BOOL bInput,
char *szForm)
{
int i, c;
static char szBR[] = " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR>";
c = sprintf(szForm,
"<HTML><HEAD><TITLE>TPC-C Order-
Status</TITLE></HEAD><BODY>"
" <FORM ACTION="tpcc.dll"
METHOD="GET\">"
" <INPUT TYPE="hidden\"
NAME="STATUSID\" VALUE="0\">"
" <INPUT TYPE="hidden\"
NAME="ERROR\" VALUE="0\">"
" <INPUT TYPE="hidden\"
NAME="FORMID\" VALUE="\"%d\">"
" <INPUT TYPE="hidden\"
NAME="TERMINID\" VALUE="\"%d\">"
" <INPUT TYPE="hidden\"
NAME="SYNCID\" VALUE="\"%d\">"
" <PRE><font face="Courier\">
Order-Status<BR>"
" Warehouse: %6.6d ",
ORDER_STATUS_FORM, iTermId,
Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id);
if ( bInput )
{
strcpy(szForm+c,
"District: <INPUT
NAME="DID*" SIZE=1><BR>"
"Customer: <INPUT
NAME="CID*" SIZE=4> Name:
<INPUT NAME="CLT*" SIZE=23><BR>"
" Cust-Balance:<BR>
<BR>"
"Order-Number:
Entry-Date: Carrier-
Number:<BR>"

```

```

                "Supply-W    Item-Id
Qty    Amount    Delivery-Date<BR> <BR> <BR> <BR>
<BR>"
                " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR></font></PRE>"
                "<HR><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\"><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                "</BODY></FORM></HTML>"
);
    }
    else
    {
        c += sprintf(szForm+c,
            "District: %2.2d<BR>"
            "Customer: %4.4d
Name: %-16s %-2s %-16s<BR>",
            pOrderStatusData->d_id,
            pOrderStatusData->c_id,
            pOrderStatusData->c_first, pOrderStatusData->c_middle,
            pOrderStatusData->c_last);
        c += sprintf(szForm+c, "Cust-
Balance: $%9.2f<BR> <BR>",
            pOrderStatusData->c_balance);
        c += sprintf(szForm+c,
            "Order-Number: %8.8d
Entry-Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d
Carrier-Number: %2.2d<BR>"
            "Supply-W    Item-Id
Qty    Amount    Delivery-Date<BR>",
            pOrderStatusData->o_id,
            pOrderStatusData->o_entry_d.day,
            pOrderStatusData->o_entry_d.month,
            pOrderStatusData->o_entry_d.year,
            pOrderStatusData->o_entry_d.hour,
            pOrderStatusData->o_entry_d.minute,
            pOrderStatusData->o_entry_d.second,
            pOrderStatusData->o_carrier_id);
        for(i=0; i< pOrderStatusData->o_ol_cnt; i++)
        {
            c += sprintf(szForm+c,
                " %6.6d    %6.6d    %2.2d    $%8.2f    %2.2d-
                %2.2d-%4.4d<BR>",
                pOrderStatusData->ol[i].ol_supply_w_id,
                pOrderStatusData->ol[i].ol_i_id,

```

```

            pOrderStatusData->ol[i].ol_quantity,
            pOrderStatusData->ol[i].ol_amount,
            pOrderStatusData->ol[i].ol_delivery_d.day,
            pOrderStatusData->ol[i].ol_delivery_d.month,
            pOrderStatusData->ol[i].ol_delivery_d.year);
        }
        strncpy( szForm+c, szBR, (15-i)*5
);
        c += (15-i)*5;
        strcpy(szForm+c,
            "</font></PRE><HR><INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..NewOrder..\">"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
            "</BODY></FORM></HTML>"
);
    }
}
/* FUNCTION: MakeDeliveryForm
 *
 * COMMENTS:    The internal client buffer is
created when the terminal id is assigned and should
not
 *              be freed
 *              except when the client terminal id is no longer
needed.
 */
void MakeDeliveryForm(int iTermId, DELIVERY_DATA
 *pDeliveryData, BOOL bInput, char *szForm)
{
    int    c;
    c = sprintf(szForm,
        "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD><BODY>"
        "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
        "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"

```

```

        "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\"
NAME=\"TERMINID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">
Delivery<BR>"
        "Warehouse: %6.6d<BR> <BR>",
        (bInput && (pDeliveryData->exec_status_code != eOK)) ? ERR_TYPE_DELIVERY_POST :
        0,
        DELIVERY_FORM, iTermId,
        Term.pClientData[iTermId].iSyncId,
        Term.pClientData[iTermId].w_id);
        if ( bInput )
        {
            strcpy( szForm+c,
                "Carrier Number: <INPUT
NAME=\"OCD*\" SIZE=1<BR> <BR>"
                "Execution Status: <BR>
<BR> <BR> <BR> <BR> <BR> <BR>
" <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> </font></PRE><HR>"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
            );
        }
        else
        {
            sprintf( szForm+c,
                "Carrier Number:
%2.2d<BR> <BR>"
                "Execution Status: %s
<BR> <BR> <BR> <BR> <BR> <BR>
" <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> </font></PRE>"
                "<HR><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
                "</BODY></FORM></HTML>"
            );
            pDeliveryData->o_carrier_id,
            (pDeliveryData->exec_status_code == eOK) ? "Delivery has been
queued." : "Delivery Post Failed "
        );

```

```

    }
}
/* FUNCTION: ProcessNewOrderForm
 * PURPOSE:      This function gets and validates
the input data from the new order form
 *              filling in the required
input variables. it then calls the SQLNewOrder
 *              transaction, constructs
the output form and writes it back to client
 *              browser.
 */
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK
 *pECB, int iTermId, char *szBuffer)
{
    PNEW_ORDER_DATA      pNewOrder;

    pNewOrder = Term.pClientData[iTermId].pTxn-
->BuffAddr_NewOrder();

    ZeroMemory(pNewOrder,
sizeof(NEW_ORDER_DATA));
    pNewOrder->w_id =
Term.pClientData[iTermId].w_id;
    GetNewOrderData(pECB->lpszQueryString,
pNewOrder);

    Term.pClientData[iTermId].pTxn->NewOrder();

    pNewOrder = Term.pClientData[iTermId].pTxn-
->BuffAddr_NewOrder();
    MakeNewOrderForm(iTermId, pNewOrder,
OUTPUT_FORM, szBuffer );
}

/* FUNCTION: void ProcessPaymentForm
 * PURPOSE:      This function gets and validates
the input data from the payment form
 *              filling in the required
input variables. It then calls the SQLPayment
 *              transaction, constructs
the output form and writes it back to client
 *              browser.
 * ARGUMENTS:    EXTENSION_CONTROL_BLOCK
 *pECB          passed in structure pointer from
inetsrv.
 *              int
 *              iTermId  client browser terminal id
 */
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK
 *pECB, int iTermId, char *szBuffer)
{
    PPAYMENT_DATA      pPayment;

```

```

    pPayment = Term.pClientData[iTermId].pTxn-
->BuffAddr_Payment();
    ZeroMemory(pPayment, sizeof(PAYMENT_DATA));
    pPayment->w_id =
Term.pClientData[iTermId].w_id;
    GetPaymentData(pECB->lpszQueryString,
pPayment);

    Term.pClientData[iTermId].pTxn->Payment();

    pPayment = Term.pClientData[iTermId].pTxn-
->BuffAddr_Payment();
    MakePaymentForm(iTermId, pPayment,
OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessOrderStatusForm
 * PURPOSE:      This function gets and validates
the input data from the Order Status
 *              form filling in the
required input variables. It then calls the
 *              SQLOrderStatus
transaction, constructs the output form and writes it
 *              back to client browser.
 * ARGUMENTS:    EXTENSION_CONTROL_BLOCK
 *pECB          passed in structure pointer from
inetsrv.
 *              int
 *              iTermId  client browser terminal id
 */
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK
 *pECB, int iTermId, char *szBuffer)
{
    PORDER_STATUS_DATA pOrderStatus;

    pOrderStatus =
Term.pClientData[iTermId].pTxn-
->BuffAddr_OrderStatus();
    ZeroMemory(pOrderStatus,
sizeof(ORDER_STATUS_DATA));
    pOrderStatus->w_id =
Term.pClientData[iTermId].w_id;
    GetOrderStatusData(pECB->lpszQueryString,
pOrderStatus);

    Term.pClientData[iTermId].pTxn-
->OrderStatus();

    pOrderStatus =
Term.pClientData[iTermId].pTxn-
->BuffAddr_OrderStatus();
    MakeOrderStatusForm(iTermId, pOrderStatus,
OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessDeliveryForm

```

```

 *
 * PURPOSE:      This function gets and validates
the input data from the delivery form
 *              filling in the required
input variables. It then calls the PostDeliveryInfo
 *              Api, The client is then
informed that the transaction has been posted.
 *
 * ARGUMENTS:    EXTENSION_CONTROL_BLOCK
 *pECB          passed in structure pointer from
inetsrv.
 *              int
 *              iTermId  client browser terminal id
 */
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK
 *pECB, int iTermId, char *szBuffer)
{
    char *ptr = pECB->lpszQueryString;

    PDELIVERY_DATA      pDelivery;

    pDelivery = Term.pClientData[iTermId].pTxn-
->BuffAddr_Delivery();
    ZeroMemory(pDelivery,
sizeof(DELIVERY_DATA));
    pDelivery->w_id =
Term.pClientData[iTermId].w_id;

    pDelivery->o_carrier_id =
GetIntKeyValue(&ptr, "OCD*");
    ERR_DELIVERY_MISSING_OCD_KEY,
    ERR_DELIVERY_CARRIER_INVALID);
    if ( pDelivery->o_carrier_id > 10 ||
pDelivery->o_carrier_id < 1 )
        throw new CWEBCLNT_ERR(
ERR_DELIVERY_CARRIER_ID_RANGE );

    if (dwNumDeliveryThreads)
    {
        //post delivery info
        if ( PostDeliveryInfo(pDelivery-
>w_id, pDelivery->o_carrier_id) )
            pDelivery-
>exec_status_code = eDeliveryFailed;
        else
            pDelivery-
>exec_status_code = eOK;
    }
    else // delivery is done synchronously if
no delivery threads configured
        Term.pClientData[iTermId].pTxn-
>Delivery();

    pDelivery = Term.pClientData[iTermId].pTxn-
->BuffAddr_Delivery();
    MakeDeliveryForm(iTermId, pDelivery,
OUTPUT_FORM, szBuffer);
}

```

```

/* FUNCTION: ProcessStockLevelForm
 *
 * PURPOSE:      This function gets and validates
the input data from the Stock Level
 *
 *              form filling in the
required input variables. It then calls the
 *              SQLStockLevel
transaction, constructs the output form and writes it
 *              back to client browser.
 *
 * ARGUMENTS:    EXTENSION_CONTROL_BLOCK
 *              *pECB      passed in structure pointer from
inetsrv.
 *
 *              int
 *
 *              iTermId   client browser terminal id
 *
 */
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK
 *pECB, int iTermId, char *szBuffer)
{
    char *ptr = pECB-
>lpszQueryString;

    PSTOCK_LEVEL_DATA pStockLevel;

    pStockLevel =
Term.pClientData[iTermId].pTxn-
>BuffAddr_StockLevel();
ZeroMemory( pStockLevel,
sizeof(STOCK_LEVEL_DATA) );

    pStockLevel->w_id =
Term.pClientData[iTermId].w_id;
    pStockLevel->d_id =
Term.pClientData[iTermId].d_id;

    pStockLevel->threshold =
GetIntKeyValue(&ptr, "TT*",
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_INVALID);
    if ( pStockLevel->threshold >= 100 ||
pStockLevel->threshold < 0 )
        throw new CWBCLNT_ERR(
ERR_STOCKLEVEL_THRESHOLD_RANGE );

    Term.pClientData[iTermId].pTxn-
>StockLevel();

    pStockLevel =
Term.pClientData[iTermId].pTxn-
>BuffAddr_StockLevel();
    MakeStockLevelForm(iTermId, pStockLevel,
OUTPUT_FORM, szBuffer);
}

/* FUNCTION: GetNewOrderData
 *

```

```

 * PURPOSE:      This function extracts and
validates the new order form data from an http
command string.
 *
 * ARGUMENTS:    LPSTR      client
 *              lpszQueryString
 *              browser http command string
 *
 *              NEW_ORDER_DATA *pNewOrderData
 *              pointer to new order data structure
 *
 */
void GetNewOrderData(LPSTR lpszQueryString,
NEW_ORDER_DATA *pNewOrderData)
{
    char szTmp[26];
    int i;
    short items;
    int ol_i_id, ol_quantity;
    char *ptr = lpszQueryString;

    static char szSP[MAX_OL_NEW_ORDER_ITEMS][6]
=
"SP03*", "SP04*", {"SP00*", "SP01*", "SP02*",
"SP03*", "SP04*", "SP05*", "SP06*", "SP07*",
"SP08*", "SP09*", "SP10*", "SP11*", "SP12*",
"SP13*", "SP14*"};
    static char
szIID[MAX_OL_NEW_ORDER_ITEMS][7] =
"IID03*", "IID04*", {"IID00*", "IID01*", "IID02*",
"IID03*", "IID04*", "IID05*", "IID06*", "IID07*",
"IID08*", "IID09*", "IID10*", "IID11*", "IID12*",
"IID13*", "IID14*"};
    static char
szQty[MAX_OL_NEW_ORDER_ITEMS][7] =
"Qty03*", "Qty04*", {"Qty00*", "Qty01*", "Qty02*",
"Qty03*", "Qty04*", "Qty05*", "Qty06*", "Qty07*",
"Qty08*", "Qty09*", "Qty10*", "Qty11*", "Qty12*",
"Qty13*", "Qty14*"};

    pNewOrderData->d_id = GetIntKeyValue(&ptr,
"DID*", ERR_NEWORDER_FORM_MISSING_DID,
ERR_NEWORDER_DISTRICT_INVALID);
    pNewOrderData->c_id = GetIntKeyValue(&ptr,
"CID*", ERR_NEWORDER_CUSTOMER_KEY,
ERR_NEWORDER_CUSTOMER_INVALID);

    for(i=0, items=0; i<MAX_OL_NEW_ORDER_ITEMS;
i++)
    {
        GetKeyValue(&ptr, szSP[i], szTmp,
sizeof(szTmp), ERR_NEWORDER_MISSING_SUPPW_KEY);
        if ( szTmp[0] )
            items++;
    }
}

```

```

    if ( !IsNumeric(szTmp)
)
        throw new
CWBCLNT_ERR( ERR_NEWORDER_SUPPW_INVALID );
    pNewOrderData-
>OL[items].ol_supply_w_id = atoi(szTmp);

    ol_i_id =
pNewOrderData->OL[items].ol_i_id =
GetIntKeyValue(&ptr, szIID[i],
ERR_NEWORDER_MISSING_IID_KEY,
ERR_NEWORDER_ITEMID_INVALID);
    if ( ol_i_id > 999999
|| ol_i_id < 1 )
        throw new
CWBCLNT_ERR( ERR_NEWORDER_ITEMID_RANGE );

    ol_quantity =
pNewOrderData->OL[items].ol_quantity =
GetIntKeyValue(&ptr, szQty[i],
ERR_NEWORDER_MISSING_QTY_KEY,
ERR_NEWORDER_QTY_INVALID);
    if ( ol_quantity > 99
|| ol_quantity < 1 )
        throw new
CWBCLNT_ERR( ERR_NEWORDER_QTY_RANGE );

    items++;
}
else
    // nothing entered for
supply warehouse, so item id and qty must also be
blank
    GetKeyValue(&ptr,
szIID[i], szTmp, sizeof(szTmp),
ERR_NEWORDER_MISSING_IID_KEY);
    if ( szTmp[0] )
        throw new
CWBCLNT_ERR( ERR_NEWORDER_ITEMID_WITHOUT_SUPPW );

    GetKeyValue(&ptr,
szQty[i], szTmp, sizeof(szTmp),
ERR_NEWORDER_MISSING_QTY_KEY);
    if ( szTmp[0] )
        throw new
CWBCLNT_ERR( ERR_NEWORDER_QTY_WITHOUT_SUPPW );

}
if ( items == 0 )
    throw new CWBCLNT_ERR(
ERR_NEWORDER_NOITEMS_ENTERED );

    pNewOrderData->o_ol_cnt = items;
}

/* FUNCTION: GetPaymentData
 *

```

```

* PURPOSE:      This function extracts and
validates the payment form data from an http command
string.
*
* ARGUMENTS:   LPSTR          client
               lpszQueryString
browser http command string
*
               PAYMENT_DATA
               pointer to
*pPaymentData
payment data structure
*/

void GetPaymentData(LPSTR lpszQueryString,
PAYMENT_DATA *pPaymentData)
{
    char        szTmp[26];
    char        *ptr = lpszQueryString;
    BOOL        bCustIdBlank;
    int         iLen;

    pPaymentData->d_id = GetIntKeyValue(&ptr,
"DID*", ERR_PAYMENT_MISSING_DID_KEY,
ERR_PAYMENT_DISTRICT_INVALID);

    GetKeyValue(&ptr, "CID*", szTmp,
sizeof(szTmp), ERR_PAYMENT_MISSING_CID_KEY);
    if ( szTmp[0] == 0 )
    {
        bCustIdBlank = TRUE;
        pPaymentData->c_id = 0;
    }
    else
    {
        // parse customer id and verify
        that last name was NOT entered
        bCustIdBlank = FALSE;
        if ( !IsNumeric(szTmp) )
            throw new CWBCLNT_ERR(
ERR_PAYMENT_CUSTOMER_INVALID );
        pPaymentData->c_id = atoi(szTmp);
    }

    pPaymentData->c_w_id = GetIntKeyValue(&ptr,
"CWI*", ERR_PAYMENT_MISSING_CWI_KEY,
ERR_PAYMENT_CWI_INVALID);
    pPaymentData->c_d_id = GetIntKeyValue(&ptr,
"CDI*", ERR_PAYMENT_MISSING_CDI_KEY,
ERR_PAYMENT_CDI_INVALID);

    if ( bCustIdBlank )
    {
        // customer id is blank, so last
        name must be entered
        GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_PAYMENT_MISSING_CLT_KEY);
        if ( szTmp[0] == 0 )
            throw new CWBCLNT_ERR(
ERR_PAYMENT_MISSING_CID_CLT );

        _strupr( szTmp );
        if ( strlen(szTmp) >
LAST_NAME_LEN )

```

```

            throw new CWBCLNT_ERR(
ERR_PAYMENT_LAST_NAME_TO_LONG );

        strcpy(pPaymentData->c_last,
szTmp);
        // pad with spaces so that the
        client layer doesn't have to do it
        // before passing parameters to
        stored procedure
        iLen = strlen(pPaymentData-
>c_last);
        memset(pPaymentData->c_last +
iLen, ' ', LAST_NAME_LEN - iLen);
        pPaymentData-
>c_last[LAST_NAME_LEN] = 0;
    }
    else
    {
        // parse customer id and verify
        that last name was NOT entered
        GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_PAYMENT_MISSING_CLT_KEY);
        if ( szTmp[0] != 0 )
            throw new CWBCLNT_ERR(
ERR_PAYMENT_CID_AND_CLT );
    }

    GetKeyValue(&ptr, "HAM*", szTmp,
sizeof(szTmp), ERR_PAYMENT_MISSING_HAM_KEY);
    if (!IsDecimal(szTmp))
        throw new CWBCLNT_ERR(
ERR_PAYMENT_HAM_INVALID );
    pPaymentData->h_amount = atof(szTmp);
    if ( pPaymentData->h_amount >= 10000.00 ||
pPaymentData->h_amount < 0 )
        throw new CWBCLNT_ERR(
ERR_PAYMENT_HAM_RANGE );
}

/* FUNCTION: GetOrderStatusData
*
* PURPOSE:      This function extracts and
validates the payment form data from an http command
string.
*
*/
void GetOrderStatusData(LPSTR lpszQueryString,
ORDER_STATUS_DATA *pOrderStatusData)
{
    char        szTmp[26];
    char        *ptr = lpszQueryString;
    int         iLen;

    pOrderStatusData->d_id =
GetIntKeyValue(&ptr, "DID*",
ERR_ORDERSTATUS_MISSING_DID_KEY,
ERR_ORDERSTATUS_DID_INVALID);

    GetKeyValue(&ptr, "CID*", szTmp,
sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CID_KEY);
    if ( szTmp[0] == 0 )

```

```

    {
        // customer id is blank, so last
        name must be entered
        pOrderStatusData->c_id = 0;
        GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CLT_KEY);
        if ( szTmp[0] == 0 )
            throw new CWBCLNT_ERR(
ERR_ORDERSTATUS_MISSING_CID_CLT );

        _strupr( szTmp );
        if ( strlen(szTmp) >
LAST_NAME_LEN )
            throw new CWBCLNT_ERR(
ERR_ORDERSTATUS_CLT_RANGE );

        strcpy(pOrderStatusData->c_last,
szTmp);
        // pad with spaces so that the
        client layer doesn't have to do it
        // before passing parameters to
        stored procedure
        iLen = strlen(pOrderStatusData-
>c_last);
        memset(pOrderStatusData->c_last +
iLen, ' ', LAST_NAME_LEN - iLen);
        pOrderStatusData-
>c_last[LAST_NAME_LEN] = 0;
    }
    else
    {
        // parse customer id and verify
        that last name was NOT entered
        if ( !IsNumeric(szTmp) )
            throw new CWBCLNT_ERR(
ERR_ORDERSTATUS_CID_INVALID );
        pOrderStatusData->c_id =
atoi(szTmp);
        GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CLT_KEY);
        if ( szTmp[0] != 0 )
            throw new CWBCLNT_ERR(
ERR_ORDERSTATUS_CID_AND_CLT );
    }
}

/* FUNCTION: BOOL IsNumeric(char *ptr)
*
* PURPOSE:      This function determines if a
string is numeric. It fails if any characters other
than numeric and null
terminator are present.
*
* ARGUMENTS:   char
               *ptr          pointer to string to check.
*
* RETURNS:     BOOL         FALSE    if
string is not all numeric
               TRUE        if string contains only numeric
characters i.e. '0' - '9'
*/

```

```

BOOL IsNumeric(char *ptr)
{
    if ( *ptr == 0 )
        return FALSE;

    while( *ptr && isdigit(*ptr) )
        ptr++;
    return ( !*ptr );
}

/* FUNCTION: BOOL IsDecimal(char *ptr)
 * PURPOSE: This function determines if a
 * string is a non-negative decimal value.
 * It fails if any characters other than a
 * series of numbers followed by
 * a decimal point,
 * another series of numbers, and a null terminator are
 * present.
 * ARGUMENTS: char
 * ptr pointer to string to check.
 * RETURNS: BOOL FALSE if
 * string is not a valid non-negative decimal value
 * TRUE if string is OK
 */
BOOL IsDecimal(char *ptr)
{
    char *dotptr;
    BOOL bValid;

    if ( *ptr == 0 )
        return FALSE;

    // find decimal point
    dotptr = strchr( ptr, '.' );
    if (dotptr == NULL)
        // no decimal point, so just
        check for numeric
        return IsNumeric(ptr);
    *dotptr = 0; // temporarily replace
    decimal with a terminator

    if ( *ptr != 0 )
        bValid = IsNumeric(ptr);
    // string starts with decimal point
    else if (*(dotptr+1) == 0)
        return FALSE; // nothing but a
    decimal point is bad
    else
        bValid = TRUE;

    if (*(dotptr+1) != 0)
        // check text after decimal point
        bValid &= IsNumeric(dotptr+1);

    *dotptr = '.'; // replace decimal point

```

```

        return bValid;
    }

tpcc.h


---


/* FILE: TPCC.H Microsoft
 * TPC-C Kit Ver. 4.20.000 Copyright
 * Microsoft, 1999 All Rights Reserved
 * Version
 * 4.10.000 audited by Richard Gimarc, Performance
 * Metrics, 3/17/99
 * PURPOSE: Header file for ISAPI TPCC.DLL,
 * defines structures and functions used in the isapi
 * tpcc.dll.
 */

//VERSION RESOURCE DEFINES
#define _APS_NEXT_RESOURCE_VALUE 101
#define _APS_NEXT_COMMAND_VALUE 40001
#define _APS_NEXT_CONTROL_VALUE 1000
#define _APS_NEXT_SYMED_VALUE 101

#define TP_MAX_RETRIES 50

//note that the welcome form must be processed first
//as terminal ids assigned here, once the
//terminal id is assigned then the forms can be
//processed in any order.
#define WELCOME_FORM 1
//beginning form no term id assigned, form
id
#define MAIN_MENU_FORM 2
//term id assigned main menu form id
#define NEW_ORDER_FORM 3
//new order form id
#define PAYMENT_FORM 4
//payment form id
#define DELIVERY_FORM 5
//delivery form id
#define ORDER_STATUS_FORM 6 //order
status id

```

```

#define STOCK_LEVEL_FORM 7 //stock level
form id

//This macro is used to prevent the compiler error
unused formal parameter
#define UNUSEDPARAM(x) (x = x)

//This structure defines the data necessary to keep
distinct for each terminal or client connection.
typedef struct _CLIENTDATA
{
    int iNextFree; //index of
//index of
next free element or -1 if this entry in use.
    int w_id; //warehouse
id assigned at welcome form
    int d_id;
//district id
assigned at welcome form

    int iSyncId; //synchronization id
    int iTickCount; //time of
last access;

    CTPCC_BASE *pTxn;
} CLIENTDATA, *PCLIENTDATA;

//This structure is used to define the operational
interface for terminal id support
typedef struct _TERM
{
    int iNumEntries;
//total allocated terminal array entries
    int iFreeList;
//next available terminal array element or
-1 if none
    int iMasterSyncId; //synchronization id
    CLIENTDATA *pClientData; //pointer to
allocated client data
} TERM;

typedef TERM *PTERM; //pointer to
terminal structure type

enum WEBERROR
{
    NO_ERR,

```



```

ERR_COMMAND_UNDEFINED,
ERR_D_ID_INVALID,
ERR_DELIVERY_CARRIER_ID_RANGE,
ERR_DELIVERY_CARRIER_INVALID,
ERR_DELIVERY_MISSING_OCD_KEY,
ERR_DELIVERY_THREAD_FAILED,
ERR_GETPROCADDR_FAILED,
ERR_HTML_ILL_FORMED,
ERR_INVALID_SYNC_CONNECTION,
ERR_INVALID_TERMID,
ERR_LOADDLL_FAILED,
ERR_MAX_CONNECTIONS_EXCEEDED,
ERR_MEM_ALLOC_FAILED,
ERR_MISSING_REGISTRY_ENTRIES,
ERR_NEWORDER_CUSTOMER_INVALID,
ERR_NEWORDER_CUSTOMER_KEY,
ERR_NEWORDER_DISTRICT_INVALID,
ERR_NEWORDER_FORM_MISSING_DID,
ERR_NEWORDER_ITEMID_INVALID,
ERR_NEWORDER_ITEMID_RANGE,

ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
ERR_NEWORDER_MISSING_IID_KEY,
ERR_NEWORDER_MISSING_QTY_KEY,
ERR_NEWORDER_MISSING_SUPPW_KEY,
ERR_NEWORDER_NOITEMS_ENTERED,
ERR_NEWORDER_QTY_INVALID,
ERR_NEWORDER_QTY_RANGE,
ERR_NEWORDER_QTY_WITHOUT_SUPPW,
ERR_NEWORDER_SUPPW_INVALID,
ERR_NO_SERVER_SPECIFIED,
ERR_ORDERSTATUS_CID_AND_CLT,
ERR_ORDERSTATUS_CID_INVALID,
ERR_ORDERSTATUS_CLT_RANGE,
ERR_ORDERSTATUS_DID_INVALID,
ERR_ORDERSTATUS_MISSING_CID_CLT,
ERR_ORDERSTATUS_MISSING_CID_KEY,
ERR_ORDERSTATUS_MISSING_CLT_KEY,
ERR_ORDERSTATUS_MISSING_DID_KEY,
ERR_PAYMENT_CDI_INVALID,
ERR_PAYMENT_CID_AND_CLT,
ERR_PAYMENT_CUSTOMER_INVALID,
ERR_PAYMENT_CWI_INVALID,
ERR_PAYMENT_DISTRICT_INVALID,
ERR_PAYMENT_HAM_INVALID,
ERR_PAYMENT_HAM_RANGE,
ERR_PAYMENT_LAST_NAME_TOO_LONG,
ERR_PAYMENT_MISSING_CDI_KEY,
ERR_PAYMENT_MISSING_CID_CLT,
ERR_PAYMENT_MISSING_CID_KEY,
ERR_PAYMENT_MISSING_CLT,
ERR_PAYMENT_MISSING_CLT_KEY,
ERR_PAYMENT_MISSING_CWI_KEY,
ERR_PAYMENT_MISSING_DID_KEY,
ERR_PAYMENT_MISSING_HAM_KEY,

ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_INVALID,
ERR_STOCKLEVEL_THRESHOLD_RANGE,
ERR_VERSION_MISMATCH,
ERR_W_ID_INVALID

```

```

};

class CWEBCLNT_ERR : public CBaseErr
{
public:
    CWEBCLNT_ERR(WEBERROR Err)
    {
        m_Error = Err;
        m_szTextDetail = NULL;
        m_SystemErr = 0;
        m_szErrorText = NULL;
    };

    CWEBCLNT_ERR(WEBERROR Err, char
    *szTextDetail, DWORD dwSystemErr)
    {
        m_Error = Err;
        m_szTextDetail = new
        char[strlen(szTextDetail)+1];
        strcpy( m_szTextDetail,
        szTextDetail );
        m_SystemErr =
        dwSystemErr;
        m_szErrorText = NULL;
    };

    ~CWEBCLNT_ERR()
    {
        if (m_szTextDetail !=
        NULL) delete []
        m_szTextDetail;
        if (m_szErrorText !=
        NULL) delete []
        m_szErrorText;
    };

    WEBERROR m_Error;
    char
    *m_szTextDetail; //
    char
    *m_szErrorText;
    DWORD m_SystemErr;

    int ErrorType() {return
    ERR_TYPE_WEBDLL};
    char *ErrorTypeStr() { return
    "WEBCLIENT"; }
    int ErrorNum() {return m_Error};
    char *ErrorText();

    //These constants have already been defined in
    engstut.h, but since we do
    //not want to include it in the delisrv executable
    #define TXN_EVENT_START 2
    #define TXN_EVENT_STOP 4

```

```

#define TXN_EVENT_WARNING 6
//used to record a warning into the log

//function prototypes

BOOL APIENTRY DllMain(HANDLE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved);
void WriteMessageToEventLog(LPTSTR lpszMsg);
void ProcessQueryString(EXTENSION_CONTROL_BLOCK
*pECB, int *pCmd, int *pFormId, int *pTermId, int
*pSyncId);
void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int iTermId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int iTermId);
void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int
iError, int iErrorType, char *szMsg, int iTermId);
void GetKeyValue(char **pQueryString, char *pKey,
char *pValue, int iMax, WEBERROR err);
int GetIntKeyValue(char **pQueryString, char *pKey,
WEBERROR NoKeyErr, WEBERROR NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
int TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int
iType, int iErrorNum, int iTermId, int iSyncId, char
*szErrorText, char *szBuffer );
void MakeMainMenuForm(int iTermId, int iSyncId, char
*szForm);
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA
*pStockLevelData, BOOL bInput, char *szForm);
void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA
*pNewOrderData, BOOL bInput, char *szForm);
void MakePaymentForm(int iTermId, PAYMENT_DATA
*pPaymentData, BOOL bInput, char *szForm);
void MakeOrderStatusForm(int iTermId,
ORDER_STATUS_DATA *pOrderStatusData, BOOL bInput,
char *szForm);
void MakeDeliveryForm(int iTermId, DELIVERY_DATA
*pDeliveryData, BOOL bInput, char *szForm);
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer);
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer);
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer);
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer);
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer);
void GetNewOrderData(LPSTR lpszQueryString,
NEW_ORDER_DATA *pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString,
PAYMENT_DATA *pPaymentData);

```

```

void GetOrderStatusData(LPSTR lpszQueryString,
ORDER_STATUS_DATA *pOrderStatusData);
BOOL PostDeliveryInfo(long w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);

```

tpcc.rc

```

//Microsoft Developer Studio generated resource
script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "afxres.h"

//
// English (U.S.) resources
//
//
// English (U.S.) resources
//
//
//
// English (U.S.) resources
//
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,4,0,0
PRODUCTVERSION 0,4,0,0
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"

```

```

        BEGIN
            VALUE "Comments", "TPC-C HTML DLL Server
(DBLIB)\0"
            VALUE "CompanyName", "Microsoft\0"
            VALUE "FileDescription", "TPC-C HTML DLL
Server (DBLIB)\0"
            VALUE "FileVersion", "0,4,0,0\0"
            VALUE "InternalName", "tpcc\0"
            VALUE "LegalCopyright", "Copyright ©
1997\0"
            VALUE "OriginalFilename", "tpcc.dll\0"
            VALUE "ProductName", "Microsoft tpcc\0"
            VALUE "ProductVersion", "0,4,0,0\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

#ifdef ! _MAC

#ifdef APSTUDIO_INVOKED
//
// TEXTINCLUDE
//
1 TEXTINCLUDE DISCARDABLE
BEGIN
    "resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
    "#include \"afxres.h\"\r\n"
    "\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
    "\r\n"
    "\0"
END

#endif // APSTUDIO_INVOKED

//
// Dialog
//
IDD_DIALOG1 DIALOG DISCARDABLE 0, 0, 186, 95
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION |
WS_SYSMENU
CAPTION "Dialog"

```

```

FONT 8, "MS Sans Serif"
BEGIN
    DEFPUSHBUTTON "OK", IDOK, 129, 7, 50, 14
    PUSHBUTTON "Cancel", IDCANCEL, 129, 24, 50, 14
END

//
// DESIGNINFO
//
#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 179
        TOPMARGIN, 7
        BOTTOMMARGIN, 88
    END
END
#endif // APSTUDIO_INVOKED

//
// English (U.S.) resources
//
//
//
//
//
// not APSTUDIO_INVOKED

```

tpcc_com.cpp

```

/* FILE: TPCC_COM.CPP
* Microsoft
TPC-C Kit Ver. 4.20.000 Copyright
*
Microsoft, 1999
* All Rights Reserved
*
* not yet
audited
*
PURPOSE: Source file for TPC-C COM+ class
implementation.

```

```

*       Contact: Charles Levine
(clevine@microsoft.com)
*
*   Change history:
*       4.20.000 - first version
*/

// needed for CoInitializeEx
#define _WIN32_WINNT 0x0400

#include <windows.h>

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\trans.h"
//tpckit transaction header contains
definitions of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_com.h"

#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\tpcc_com_all\src\tpcc_com_all_i.c"

// wrapper routine for class constructor
_declspec( dllexport ) CTPCC_COM* CTPCC_COM_new(BOOL
bSinglePool)
{
    return new CTPCC_COM( bSinglePool );
}

CTPCC_COM::CTPCC_COM(BOOL bSinglePool)
{
    HRESULT hr = NULL;
    long lRet = 0;
    ULONG ulTmpSize = 0;

    m_pTxn          = NULL;
    m_pNewOrder     = NULL;
    m_pPayment      = NULL;
    m_pStockLevel   = NULL;
    m_pOrderStatus  = NULL;

    m_bSinglePool   = bSinglePool;

    ulTmpSize = (ULONG) sizeof(COM_DATA);
    VariantInit(&m_vTxn);
    m_vTxn.vt = VT_SAFEARRAY;

    m_vTxn.parray =
SafeArrayCreateVector(VT_UI1, ulTmpSize, ulTmpSize);
    if (!m_vTxn.parray)
        throw new CCOMERR( E_FAIL );

    memset( (void*)m_vTxn.parray-
>pvData, 0, ulTmpSize);
    m_pTxn = (COM_DATA*)m_vTxn.parray->pvData;

    hr = CoInitializeEx(NULL,
COINIT_MULTITHREADED);

```

```

    if (FAILED(hr))
    {
        throw new CCOMERR( hr );
    }

    // create components
    if (m_bSinglePool)
    {
        hr = CoCreateInstance(CLSID_TPCC,
NULL, CLSCTX_SERVER, IID_ITPCC, (void
**) &m_pNewOrder);
        if (FAILED(hr))
            throw new CCOMERR(hr);

        // all txns will use same
        component
            m_pPayment = m_pNewOrder;
            m_pStockLevel = m_pNewOrder;
            m_pOrderStatus = m_pNewOrder;
        }
        else
        {
            // use different components for
            each txn

            hr =
CoCreateInstance(CLSID_NewOrder, NULL, CLSCTX_SERVER,
IID_ITPCC, (void **) &m_pNewOrder);
            if (FAILED(hr))
                throw new CCOMERR(hr);

            hr =
CoCreateInstance(CLSID_Payment, NULL, CLSCTX_SERVER,
IID_ITPCC, (void **) &m_pPayment);
            if (FAILED(hr))
                throw new CCOMERR(hr);

            hr =
CoCreateInstance(CLSID_StockLevel, NULL,
CLSCTX_SERVER, IID_ITPCC, (void **) &m_pStockLevel);
            if (FAILED(hr))
                throw new CCOMERR(hr);

            hr =
CoCreateInstance(CLSID_OrderStatus, NULL,
CLSCTX_SERVER, IID_ITPCC, (void **) &m_pOrderStatus);
            if (FAILED(hr))
                throw new CCOMERR(hr);
        }

        // call setcomplete to release each
        component
        back into pool
        hr = m_pNewOrder->CallSetComplete();
        if (FAILED(hr))
            throw new CCOMERR(hr);

        if (!m_bSinglePool)
        {
            hr = m_pPayment-
>CallSetComplete();
            if (FAILED(hr))

```

```

                throw new CCOMERR(hr);

            hr = m_pStockLevel-
>CallSetComplete();
            if (FAILED(hr))
                throw new CCOMERR(hr);

            hr = m_pOrderStatus-
>CallSetComplete();
            if (FAILED(hr))
                throw new CCOMERR(hr);
        }
    }

CTPCC_COM::~CTPCC_COM()
{
    if (m_pTxn)
        SafeArrayDestroy(m_vTxn.parray);

    ReleaseInterface(m_pNewOrder);
    if (!m_bSinglePool)
    {
        ReleaseInterface(m_pPayment);
        ReleaseInterface(m_pStockLevel);
        ReleaseInterface(m_pOrderStatus);
    }
    CoUninitialize();
}

void CTPCC_COM::NewOrder()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pNewOrder->NewOrder(m_vTxn,
&vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR( hr );
    memcpy(m_pTxn, (void *)vTxn_out.parray-
>pvData, vTxn_out.parray->rgsabound[0].cElements);
    SafeArrayDestroy(vTxn_out.parray);

    if ( m_pTxn->ErrorType != ERR_SUCCESS )
        throw new CCOMERR( m_pTxn-
>ErrorType, m_pTxn->error );
}

void CTPCC_COM::Payment()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pPayment->Payment(m_vTxn,
&vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR( hr );
    memcpy(m_pTxn, (void *)vTxn_out.parray-
>pvData, vTxn_out.parray->rgsabound[0].cElements);
    SafeArrayDestroy(vTxn_out.parray);

    if ( m_pTxn->ErrorType != ERR_SUCCESS )
        throw new CCOMERR( m_pTxn-
>ErrorType, m_pTxn->error );
}

```

```

}

void CTPCC_COM::StockLevel()
{
    VARIANT    vTxn_out;

    HRESULT hr = m_pStockLevel->StockLevel(m_vTxn, &vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR( hr );
    memcpy(m_pTxn, (void *)vTxn_out.parray->pvData, vTxn_out.parray->rgsabound[0].cElements);
    SafeArrayDestroy(vTxn_out.parray);

    if ( m_pTxn->ErrorType != ERR_SUCCESS )
        throw new CCOMERR( m_pTxn->ErrorType, m_pTxn->error );
}

void CTPCC_COM::OrderStatus()
{
    VARIANT    vTxn_out;

    HRESULT hr = m_pOrderStatus->OrderStatus(m_vTxn, &vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR( hr );
    memcpy(m_pTxn, (void *)vTxn_out.parray->pvData, vTxn_out.parray->rgsabound[0].cElements);
    SafeArrayDestroy(vTxn_out.parray);

    if ( m_pTxn->ErrorType != ERR_SUCCESS )
        throw new CCOMERR( m_pTxn->ErrorType, m_pTxn->error );
}

```

tpcc_com.h

```

/* FILE: TPCC_COM.H
 * Microsoft
 * TPC-C Kit Ver. 4.20.000
 * Copyright
 * Microsoft, 1999
 * All Rights Reserved
 *
 * not yet
 * audited
 *
 * PURPOSE: Header file for TPC-C COM+ class
 * implementation.
 *
 * Change history:
 * 4.20.000 - first version
 */

#pragma once

#include <stdio.h>
#include "..\..\tpcc_com_ps\src\tpcc_com_ps.h"

```

```

// need to declare functions for import, unless
// define has already been created
// by the DLL's .cpp module for export.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

class CCOMERR : public CBaseErr
{
private:
    char m_szErrorText[64];

public:
    // use this interface for genuine
    COM errors
    CCOMERR( HRESULT hr )
    {
        m_hr = hr;
        m_iErrorType = 0;
        m_iError = 0;
    }

    // use this interface to
    impersonate a non-COM error type
    CCOMERR( int iErrorType, int
    iError )
    {
        m_iErrorType =
        m_iError = iError;
        m_hr = S_OK;
    }

    int m_hr;
    int m_iErrorType;
    int m_iError;

    // A CCOMERR class can
    impersonate another class, which happens if the error
    // was not actually a COM
    Services error, but was simply transmitted back via
    COM.
    int ErrorType()
    {
        if (m_iErrorType == 0)
            return
            ERR_TYPE_COM;
        else
            return
            m_iErrorType;
    }

    char *ErrorTypeStr() { return
    "COM"; }

    int ErrorNum() {return m_hr;}

    char *ErrorText()
    {
        if (m_hr == S_OK)

```

```

        sprintf(
        m_szErrorText, "Error: Class %d, error # %d",
        m_iErrorType, m_iError );
        else
            sprintf(
            m_szErrorText, "Error: COM HRESULT %x", m_hr );
            return m_szErrorText;
    }
};

class DllDecl CTPCC_COM : public CTPCC_BASE
{
private:
    BOOL m_bSinglePool;

    // COM Interface pointers
    ITPCC*
    m_pNewOrder;
    ITPCC*
    m_pPayment;
    ITPCC*
    m_pStockLevel;
    ITPCC*
    m_pOrderStatus;

    struct COM_DATA
    {
        int ErrorType;
        int error;
        union
        {
            NEW_ORDER_DATA NewOrder;
            PAYMENT_DATA Payment;
            DELIVERY_DATA Delivery;
            STOCK_LEVEL_DATA StockLevel;
            ORDER_STATUS_DATA OrderStatus;
        } u;
    } *m_pTxn;

public:
    CTPCC_COM(BOOL bSinglePool);
    ~CTPCC_COM(void);

    inline PNEW_ORDER_DATA
    BuffAddr_NewOrder() { return
    &m_pTxn->u.NewOrder; };
    inline PPAYMENT_DATA
    BuffAddr_Payment() { return
    &m_pTxn->u.Payment; };
    inline PDELIVERY_DATA
    BuffAddr_Delivery() { return
    &m_pTxn->u.Delivery; };
    inline PSTOCK_LEVEL_DATA
    BuffAddr_StockLevel() { return
    &m_pTxn->u.StockLevel; };

```

```

        inline PORDER_STATUS_DATA
        BuffAddr_OrderStatus() { return
&m_pTxn->u.OrderStatus; };

        void NewOrder          ();
        void Payment           ();
        void StockLevel        ();
        void OrderStatus       ();
        void Delivery          ();
    { throw new CCOMERR(E_NOTIMPL); } // not supported
};

inline void ReleaseInterface(IUnknown *pUnk)
{
    if (pUnk)
    {
        pUnk->Release();
        pUnk = NULL;
    }
}

// wrapper routine for class constructor
extern "C" __declspec(dllexport) CTPCC_COM*
CTPCC_COM_new(BOOL);

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);

```

tpcc_com_all.cpp

```

/*      FILE:          TPCC_COM_ALL.CPP
 *      Microsoft
 *      TPC-C Kit Ver. 4.20.000
 *      Copyright
 *      Microsoft, 1999
 *      All Rights Reserved
 *      Version
 *      4.10.000 audited by Richard Gimarc, Performance
 *      Metrics, 3/17/99
 *      PURPOSE:      Implementation for TPC-C Tuxedo
 *      class.
 *      Contact:      Charles Levine
 *      (clevine@microsoft.com)
 *      Change history:
 *      4.20.000 - updated rev number to
 *      match kit
 */

#define STRICT
#define WIN32_WINNT 0x0400
#define ATL_APARTMENT_THREADED

#include <stdio.h>
#include <atlbase.h>
//You may derive a class from CComModule and use it
if you want to override

```

```

//something, but do not change the name of _Module
extern CComModule _Module;

#include <atlcom.h>
#include <initguid.h>
#include <transact.h>
//#include <atlimpl.cpp>
#include <comsvcs.h>

#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>

#include "tpcc_com_ps.h"
#include "..\..\common\src\trans.h"
//tpckit transaction
header contains definations of structures specific to
TPC-C
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\error.h"
#include "..\..\common\src\ReadRegistry.h"
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"
// DBLIB implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"
// ODBC implementation of TPC-C txns

#include "resource.h"
#include "tpcc_com_all.h"
#include "tpcc_com_all_i.c"
#include "Methods.h"
#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\common\src\ReadRegistry.cpp"

CComModule _Module;

BEGIN_OBJECT_MAP(ObjectMap)
    OBJECT_ENTRY(CLSID_TPCC, CTPCC)
    OBJECT_ENTRY(CLSID_NewOrder, CNewOrder)
    OBJECT_ENTRY(CLSID_OrderStatus,
COrderStatus)
    OBJECT_ENTRY(CLSID_Payment, CPayment)
    OBJECT_ENTRY(CLSID_StockLevel, CStockLevel)
END_OBJECT_MAP()

// configuration settings from registry
TPCCREGISTRYDATA Reg;
char
    szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

static HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC *pCTPCC_ODBC_new;

// Critical section to synchronize connection open
and close.
//
CRITICAL_SECTION hConnectCriticalSection;

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// DLL Entry Point

extern "C"
BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD
dwReason, LPVOID /*lpReserved*/)
{
    char szDllName[128];

    try
    {
        if (dwReason ==
DLL_PROCESS_ATTACH)
        {
            _Module.Init(ObjectMap,
hInstance);

            DisableThreadLibraryCalls(hInstance);

            DWORD dwSize =
MAX_COMPUTERNAME_LENGTH+1;

            GetComputerName(szMyComputerName, &dwSize);

            szMyComputerName[dwSize] = 0;

            if (
ReadTPCCRegistrySettings( &Reg ))
            throw new
CCOMPONENT_ERR( ERR_MISSING_REGISTRY_ENTRIES );

            if (Reg.eDB_Protocol ==
DBLIB)
            {
                strcpy(
szDllName, Reg.szPath );
                strcat(
szDllName, "tpcc_dblib.dll");

                hLibInstanceDb = LoadLibrary( szDllName );
                if
(hLibInstanceDb == NULL)
                throw new CCOMPONENT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                // get
                function pointer to wrapper for class constructor

                pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb, "CTPCC_DBLIB_new");
                if
(pCTPCC_DBLIB_new == NULL)
                throw new CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
            }
            else if
(Reg.eDB_Protocol == ODBC)

```

```

        {
            strcpy(
szDllName, Reg.szPath );
            strcat(
szDllName, "tpcc_odbc.dll");

            hLibInstanceDb = LoadLibrary( szDllName );
            if
(hLibInstanceDb == NULL)

                throw new CCOMPONENT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );

            // get
            function pointer to wrapper for class constructor

            pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb, "CTPCC_ODBC_new");
            if
(pCTPCC_ODBC_new == NULL)

                throw new CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
            else
                throw new
CCOMPONENT_ERR( ERR_UNKNOWN_DB_PROTOCOL );

            if (Reg.dwConnectDelay
> 0)
                {
                    InitializeCriticalSection(&hConnectCritical
Section);
                }
            else if (dwReason ==
DLL_PROCESS_DETACH)
                _Module.Term();

        }
        catch (CBaseErr *e)
        {
            TCHAR szMsg[256];
            _sntprintf(szMsg, sizeof(szMsg),
"%s error, code %d: %s",
e-
->ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
            WriteMessageToEventLog( szMsg );

            delete e;
            return FALSE;
        }
        catch (...)
        {
            WriteMessageToEventLog(TEXT("Unhandled
exception in object DllMain"));
            return FALSE;
        }
    }
}

```

```

    }
    return TRUE; // OK
}

////////////////////////////////////
////////////////////////////////////
// Used to determine whether the DLL can be unloaded
by OLE

STDAPI DllCanUnloadNow(void)
{
    return (_Module.GetLockCount()==0) ? S_OK :
S_FALSE;
}

////////////////////////////////////
////////////////////////////////////
// Returns a class factory to create an object of the
requested type

STDAPI DllGetClassObject(REFCLSID rclsid, REFIID
riid, LPVOID* ppv)
{
    return _Module.GetClassObject(rclsid, riid,
ppv);
}

////////////////////////////////////
////////////////////////////////////
// DllRegisterServer - Adds entries to the system
registry

STDAPI DllRegisterServer(void)
{
    // registers object, typelib and all
interfaces in typelib
    return _Module.RegisterServer(TRUE);
}

////////////////////////////////////
////////////////////////////////////
// DllUnregisterServer - Removes entries from the
system registry

STDAPI DllUnregisterServer(void)
{
    _Module.UnregisterServer();
    return S_OK;
}

static void WriteMessageToEventLog(LPTSTR lpszMsg)
{
    TCHAR szMsg[256];
    HANDLE hEventSource;
    LPTSTR lpszStrings[2];

    // Use event logging to log the error.
    //

```

```

    hEventSource = RegisterEventSource(NULL,
TEXT("tpcc_com_all.dll"));

    _stprintf(szMsg, TEXT("Error in COM+ TPC-C
Component: "));
    lpszStrings[0] = szMsg;
    lpszStrings[1] = lpszMsg;

    if (hEventSource != NULL)
    {
        ReportEvent(hEventSource, // handle of event
source
            EVENTLOG_ERROR_TYPE, // event type
            0, // event category
            0, // event ID
            NULL, // current user's
SID
            2, // strings in
lpszStrings
            0, // no bytes of raw
data
            (LPCTSTR *)lpszStrings, // array of
error strings
            NULL); // no raw data

        (VOID) DeregisterEventSource(hEventSource);
    }
}

inline void ReleaseInterface(IUnknown *pUnk)
{
    if (pUnk)
    {
        pUnk->Release();
        pUnk = NULL;
    }
}

/* FUNCTION: CCOMPONENT_ERR::ErrorText
*
*/

char* CCOMPONENT_ERR::ErrorText(void)
{
    static SERRORMSG errorMsgs[] =
    {
        { ERR_MISSING_REGISTRY_ENTRIES,
"Required entries missing from registry."
},
        { ERR_LOADDLL_FAILED,
"Load of DLL failed. DLL="
},
        { ERR_GETPROCADDR_FAILED,
"Could not map proc in DLL. GetProcAddress
error. DLL="
},
        { ERR_UNKNOWN_DB_PROTOCOL,
"Unknown database protocol specified in
registry."
},
        { 0, ""
}
}

```

```

    }
};

char szTmp[256];
int i = 0;
while (TRUE)
{
    if (errorMsgs[i].szMsg[0] == 0)
    {
        strcpy( szTmp, "Unknown
error number." );
        break;
    }
    if (m_Error ==
errorMsgs[i].iError)
    {
        strcpy( szTmp,
errorMsgs[i].szMsg );
        break;
    }
    i++;
}

if (m_szTextDetail)
    strcat( szTmp, m_szTextDetail );
if (m_SystemErr)
    sprintf( szTmp+strlen(szTmp), "
Error=%d", m_SystemErr );

m_szErrorText = new char[strlen(szTmp)+1];
strcpy( m_szErrorText, szTmp );
return m_szErrorText;
}

CTPCC_Common::CTPCC_Common()
{
    m_pTxn = NULL;
    m_bCanBePooled = TRUE;
}

CTPCC_Common::~CTPCC_Common()
{
    // Pace connection close for VIA.
    //
    if (Reg.dwConnectDelay > 0)
    {
        EnterCriticalSection(&hConnectCriticalSection);

        Sleep(Reg.dwConnectDelay);
    }

    if (m_pTxn)
    {
        delete m_pTxn;
    }
}

```

```

    if (Reg.dwConnectDelay > 0)
    {
        LeaveCriticalSection(&hConnectCriticalSection);
    }
}

HRESULT CTPCC_Common::CallSetComplete()
{
    IObjectContext* pObjectContext = NULL;

    // get our object context
    HRESULT hr = CoGetObjectContext(
IID_IObjectContext, (void **)&pObjectContext );
pObjectContext->SetComplete();
ReleaseInterface(pObjectContext);
return hr;
}

//
// called by the ctor activator
//
STDMETHODIMP CTPCC_Common::Construct(IDispatch *
pUnk)
{
    // Code to access construction string, if
needed later...
    //
    if (!pUnk)
        return E_UNEXPECTED;
    //
    IObjectConstructString * pString
= NULL;
    //
    HRESULT hr = pUnk-
>QueryInterface(IID_IObjectConstructString, (void
**)&pString);
    //
    pString->Release();

    try
    {
        // Pace connection creation for
VIA.
        //
        if (Reg.dwConnectDelay > 0)
        {
            EnterCriticalSection(&hConnectCriticalSection);

            Sleep(Reg.dwConnectDelay);
        }

        if (Reg.eDB_Protocol == ODBC)
            m_pTxn =
pCTPCC_ODBC_new(
Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword,

szMyComputerName, Reg.szDbName,

```

```

Reg.szSPPrefix,
Reg.bCallNoDuplicatesNewOrder );
        else if (Reg.eDB_Protocol ==
DBLIB)
            m_pTxn =
pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
    }

    if (Reg.dwConnectDelay > 0)
    {
        LeaveCriticalSection(&hConnectCriticalSection);
    }
}

catch (CBaseErr *e)
{
    TCHAR szMsg[256];

    _sntprintf(szMsg, sizeof(szMsg),
"%s error in CTPCC_Common::Construct, code %d: %s",
e-
>ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
    WriteMessageToEventLog( szMsg );
    delete e;
    return E_FAIL;
}

catch (...)
{
    WriteMessageToEventLog(TEXT("Unhandled
exception in object :Construct"));
    return E_FAIL;
}

return S_OK;
}

HRESULT CTPCC_Common::NewOrder(VARIANT txn_in,
VARIANT* txn_out)
{
    PNEW_ORDER_DATA pNewOrder;
    COM_DATA *pData;
    try
    {
        pData = (COM_DATA*)txn_in.parray-
>pvData;
        pNewOrder = m_pTxn-
>BuffAddr_NewOrder();
        memcpy(pNewOrder, &pData-
>u.NewOrder, sizeof(NEW_ORDER_DATA));
        m_pTxn->NewOrder(); //
do the actual txn

        VariantInit(txn_out);
        txn_out->vt = VT_SAFEARRAY;
        txn_out->parray =
SafeArrayCreateVector(VT_UI1,

```

```

        txn_in.parray->rgsabound-
>cElements,
        txn_in.parray->rgsabound-
>cElements);
        pData = (COM_DATA*) txn_out-
>parray->pvData;
        memcpy( &pData->u.NewOrder,
pNewOrder, sizeof(NEW_ORDER_DATA));
        pData->retval = ERR_SUCCESS;
        pData->error = 0;
        return S_OK;
    }
    catch (CBaseErr *e)
    {
        // check for lost database
        connection; if yes, component is toast
        if ( ((e->ErrorType() ==
ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005) ||
            ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054) ) )
            m_bCanBePooled = FALSE;

        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception."));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        m_bCanBePooled = FALSE;
        return E_FAIL;
    }
}

HRESULT CTPCC_Common::Payment(VARIANT txn_in,
VARIANT* txn_out)
{
    PPAYMENT_DATA    pPayment;
    COM_DATA          *pData;
    try
    {
        pData = (COM_DATA*)txn_in.parray-
>pvData;
        pPayment = m_pTxn-
>BuffAddr_Payment();

        memcpy(pPayment, &pData-
>u.Payment, sizeof(PAYMENT_DATA));

        m_pTxn->Payment();    //
do the actual txn

```

```

        VariantInit(txn_out);
        txn_out->vt = VT_SAFEARRAY;
        txn_out->parray =
SafeArrayCreateVector( VT_UI1,
        txn_in.parray->rgsabound-
>cElements,
        txn_in.parray->rgsabound-
>cElements);
        pData = (COM_DATA*) txn_out-
>parray->pvData;
        memcpy( &pData->u.Payment,
pPayment, sizeof(PAYMENT_DATA));
        pData->retval = ERR_SUCCESS;
        pData->error = 0;
        return S_OK;
    }
    catch (CBaseErr *e)
    {
        // check for lost database
        connection; if yes, component is toast
        if ( ((e->ErrorType() ==
ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005) ||
            ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054) ) )
            m_bCanBePooled = FALSE;

        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception."));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        m_bCanBePooled = FALSE;
        return E_FAIL;
    }
}

HRESULT CTPCC_Common::StockLevel(VARIANT txn_in,
VARIANT* txn_out)
{
    PSTOCK_LEVEL_DATA pStockLevel;
    COM_DATA          *pData;
    try
    {
        pData = (COM_DATA*)txn_in.parray-
>pvData;
        pStockLevel = m_pTxn-
>BuffAddr_StockLevel();

```

```

        memcpy(pStockLevel, &pData-
>u.StockLevel, sizeof(STOCK_LEVEL_DATA));
        m_pTxn->StockLevel();
        VariantInit(txn_out);
        txn_out->vt = VT_SAFEARRAY;
        txn_out->parray =
SafeArrayCreateVector( VT_UI1,
        txn_in.parray->rgsabound-
>cElements,
        txn_in.parray->rgsabound-
>cElements);
        pData = (COM_DATA*)txn_out-
>parray->pvData;
        memcpy( &pData->u.StockLevel,
pStockLevel, sizeof(STOCK_LEVEL_DATA));
        pData->retval = ERR_SUCCESS;
        pData->error = 0;
        return S_OK;
    }
    catch (CBaseErr *e)
    {
        // check for lost database
        connection; if yes, component is toast
        if ( ((e->ErrorType() ==
ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005) ||
            ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054) ) )
            m_bCanBePooled = FALSE;

        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception."));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        m_bCanBePooled = FALSE;
        return E_FAIL;
    }
}

HRESULT CTPCC_Common::OrderStatus(VARIANT txn_in,
VARIANT* txn_out)
{
    PORDER_STATUS_DATA pOrderStatus;
    COM_DATA          *pData;
    try
    {
        pData = (COM_DATA*)txn_in.parray-
>pvData;

```



```

        pOrderStatus = m_pTxn-
>BuffAddr_OrderStatus();

        memcpy(pOrderStatus, &pData-
>u.OrderStatus, sizeof(ORDER_STATUS_DATA));

        m_pTxn->OrderStatus();

        VariantInit(txn_out);
        txn_out->vt = VT_SAFEARRAY;
        txn_out->parray =
SafeArrayCreateVector( VT_UI1,
>cElements,
        txn_in.parray->rgsabound-
>cElements);
        txn_in.parray->rgsabound-
>cElements);
        pData = (COM_DATA*)txn_out-
>parray->pvData;

        memcpy( &pData->u.OrderStatus,
pOrderStatus, sizeof(ORDER_STATUS_DATA));

        pData->retval = ERR_SUCCESS;
        pData->error = 0;
        return S_OK;
    }
    catch (CBaseErr *e)
    {
        // check for lost database
        connection; if yes,
        component is toast
        if ( ((e->ErrorType() ==
ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
            ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
            m_bCanBePooled = FALSE;

        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception."));

        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        m_bCanBePooled = FALSE;
        return E_FAIL;
    }
}

```

tpcc_com_all.h

```

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

```

```

/* this ALWAYS GENERATED file contains the
definitions for the interfaces */

/* File created by MIDL compiler version 6.00.0347
*/
/* at Fri Apr 15 14:48:53 2005
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf, Wl, Zp8, env-Win32 (32b run)
protocol : dce , ms_ext, c_ext
error checks: allocation ref bounds_check enum
stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@MIDL_FILE_HEADERING( )

/* verify that the <rpcndr.h> version is high enough
to compile this file*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

#include "rpc.h"
#include "rpcndr.h"

#ifndef __tpcc_com_all_h__
#define __tpcc_com_all_h__

#if defined(_MSC_VER) && (_MSC_VER >= 1020)
#pragma once
#endif

/* Forward Declarations */

#ifndef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__

#ifdef __cplusplus
typedef class TPCC TPCC;
#else
typedef struct TPCC TPCC;
#endif /* __cplusplus */

#endif /* __TPCC_FWD_DEFINED__ */

#ifndef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

#ifdef __cplusplus
typedef class NewOrder NewOrder;
#else
typedef struct NewOrder NewOrder;
#endif /* __cplusplus */

```

```

#endif /* __NewOrder_FWD_DEFINED__ */

#ifndef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

#ifdef __cplusplus
typedef class OrderStatus OrderStatus;
#else
typedef struct OrderStatus OrderStatus;
#endif /* __cplusplus */

#endif /* __OrderStatus_FWD_DEFINED__ */

#ifndef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

#ifdef __cplusplus
typedef class Payment Payment;
#else
typedef struct Payment Payment;
#endif /* __cplusplus */

#endif /* __Payment_FWD_DEFINED__ */

#ifndef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

#ifdef __cplusplus
typedef class StockLevel StockLevel;
#else
typedef struct StockLevel StockLevel;
#endif /* __cplusplus */

#endif /* __StockLevel_FWD_DEFINED__ */

/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"
#include "tpcc_com_ps.h"

#ifdef __cplusplus
extern "C"{
#endif

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

/* interface __MIDL_itf_tpcc_com_all_0000 */
/* [local] */

```

```

extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;

#ifdef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

/* library TPCCLib */
/* [helpstring] [version] [uuid] */

EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus
class DECLSPEC_UUID("122A3128-2520-11D3-BA71-00C04FBFE08B")
TPCC;
#endif

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus
class DECLSPEC_UUID("975BAABF-84A7-11D2-BA47-00C04FBFE08B")
NewOrder;
#endif

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus
class DECLSPEC_UUID("266836AD-A50D-11D2-BA4E-00C04FBFE08B")
OrderStatus;
#endif

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus
class DECLSPEC_UUID("CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B")
Payment;
#endif

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus
class DECLSPEC_UUID("2668369E-A50D-11D2-BA4E-00C04FBFE08B")
StockLevel;
#endif
#endif /* __TPCCLib_LIBRARY_DEFINED__ */

```

```

/* Additional Prototypes for ALL interfaces */
/* end of Additional Prototypes */

```

```

#ifdef __cplusplus
}
#endif
#endif

```

tpcc_com_all.rc

```

//Microsoft Developer Studio generated resource
script.
//
#include "resource.h"

```

```

#define APSTUDIO_READONLY_SYMBOLS
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "winres.h"

```

```

//
// English (U.S.) resources
//

```

```

#ifdef !defined(AFX_RESOURCE_DLL) ||
defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

```

```

#ifdef APSTUDIO_INVOKED
//
// TEXTINCLUDE
//

```

```

1 TEXTINCLUDE DISCARDABLE
BEGIN
"resource.h\0"
END

```

```

2 TEXTINCLUDE DISCARDABLE
BEGIN
#include "winres.h"\r\n"
"\0"
END

```

```

3 TEXTINCLUDE DISCARDABLE
BEGIN
"1 TYPELIB "tpcc_com_all.tlb"\r\n"
"\0"
END
#endif // APSTUDIO_INVOKED

```

```

#ifdef MAC
//
// Version
//

```

```

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x4L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN

```

```

BLOCK "StringFileInfo"
BEGIN
BLOCK "040904B0"
BEGIN
VALUE "CompanyName", "\0"
VALUE "FileDescription", "tpcc_com_all

```

```

Module\0"
VALUE "FileVersion", "1, 0, 0, 1\0"
VALUE "InternalName", "TPCCNEWORDER\0"
VALUE "LegalCopyright", "Copyright

```

```

1997\0"
VALUE "OriginalFilename",
"tpcc_com_all.DLL\0"
VALUE "ProductName", "tpcc_com_all

```

```

Module\0"
VALUE "ProductVersion", "1, 0, 0, 1\0"
VALUE "OLESelfRegister", "\0"
END

```

```

END
BLOCK "VarFileInfo"
BEGIN
VALUE "Translation", 0x409, 1200
END

```

```

END
#endif // !_MAC

```

```

//
// REGISTRY

```

```

//
IDR_TPCC          REGISTRY DISCARDABLE
"tpcc_com_all.rgs"
IDR_NEWORDER     REGISTRY DISCARDABLE
"tpcc_com_no.rgs"
IDR_ORDERSTATUS  REGISTRY DISCARDABLE
"tpcc_com_os.rgs"
IDR_PAYMENT      REGISTRY DISCARDABLE
"tpcc_com_pay.rgs"
IDR_STOCKLEVEL   REGISTRY DISCARDABLE
"tpcc_com_sl.rgs"

////////////////////////////////////
//
// String Table
//

STRINGTABLE DISCARDABLE
BEGIN
    IDS_PROJNAME        "tpcc_com_all"
END

#ifdef // English (U.S.) resources
////////////////////////////////////
////////////////////////////////////

#ifndef APSTUDIO_INVOKED
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
1 TYPELIB "tpcc_com_all.tlb"

////////////////////////////////////
//
// not APSTUDIO_INVOKED
//


```

```

tpcc_com_all.rgs
HKCR
{
    TPCC.AllTxns.1 = s 'All Txns Class'
    {
        CLSID = s '{122A3128-2520-11D3-BA71-00C04FBFE08B}'
    }
    TPCC.AllTxns = s 'TPCC Class'
    {
        CurVer = s 'TPCC.AllTxns.1'
    }
    NoRemove CLSID
    {
        ForceRemove {122A3128-2520-11D3-BA71-00C04FBFE08B} = s 'TPCC Class'
    }
}

```

```

{
    ProgID = s
    'TPCC.AllTxns.1'

    VersionIndependentProgID = s 'TPCC.AllTxns'
    InprocServer32 = s
    '%MODULE%'
    {
        val
        ThreadingModel = s 'Both'
    }
}


```

```

tpcc_com_all.i.c

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

/* this ALWAYS GENERATED file contains the IIDs and
CLSIDs */

/* link this file in with the server and any clients
*/

/* File created by MIDL compiler version 6.00.0347
*/
/* at Fri Apr 15 14:48:53 2005
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext
error checks: allocation ref bounds_check enum
stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID

```

```

#else
#include <guiddef.h>
#endif

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \
DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \
const type name =
{1,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,
0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
0,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,
0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00
,0xC0,0x4F,0xBF,0xE0,0x8B);

```

```

MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel, 0x2668369E, 0xA50D, 0x11D2, 0xBA, 0x4E, 0
x00, 0xC0, 0x4F, 0xBF, 0xE0, 0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* !defined(_M_IA64) && !defined(_M_AMD64) */

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

/* this ALWAYS GENERATED file contains the IIDs and
CLSIDs */

/* link this file in with the server and any clients
*/

/* File created by MIDL compiler version 6.00.0347
*/
/* at Fri Apr 15 14:48:53 2005
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf, Wl, Zp8, env=Win64 (32b run, appending)
protocol : dce , ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum
stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#ifdef _M_IA64 || defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif


```

```

#endif

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \

DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifdef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;

#endif // __IID_DEFINED__

#ifdef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \
    const type name =
{1,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib, 0x122A3117, 0x2520, 0x11D3, 0xBA, 0x71, 0x00,
0xC0, 0x4F, 0xBF, 0xE0, 0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC, 0x122A3128, 0x2520, 0x11D3, 0xBA, 0x71, 0x00, 0x
C0, 0x4F, 0xBF, 0xE0, 0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder, 0x975BAABF, 0x84A7, 0x11D2, 0xBA, 0x47, 0x0
0, 0xC0, 0x4F, 0xBF, 0xE0, 0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus, 0x266836AD, 0xA50D, 0x11D2, 0xBA, 0x4E,
0x00, 0xC0, 0x4F, 0xBF, 0xE0, 0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_Payment, 0xCD02F7EF, 0xA4FA, 0x11D2, 0xBA, 0x4E, 0x00
, 0xC0, 0x4F, 0xBF, 0xE0, 0x8B);

```

```

MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel, 0x2668369E, 0xA50D, 0x11D2, 0xBA, 0x4E, 0
x00, 0xC0, 0x4F, 0xBF, 0xE0, 0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* defined(_M_IA64) || defined(_M_AMD64) */

```

tpcc_com_no.rgs

```

HKCR
{
    TPCC.NewOrder.1 = s 'NewOrder Class'
    {
        CLSID = s '{975BAABF-84A7-11D2-
BA47-00C04FBFE08B}'
    }
    TPCC.NewOrder = s 'NewOrder Class'
    {
        CurVer = s 'TPCC.NewOrder.1'
        NoRemove CLSID
        {
            ForceRemove {975BAABF-84A7-11D2-
BA47-00C04FBFE08B} = s 'NewOrder Class'
            {
                ProgID = s
                'TPCC.NewOrder.1'
                VersionIndependentProgID = s
                'TPCC.NewOrder'
                InprocServer32 = s
                '%MODULE%'
                {
                    val
                    ThreadingModel = s 'Both'
                }
            }
        }
    }
}

```

tpcc_com_os.rgs

```

HKCR
{
    TPCC.OrderStatus.1 = s 'OrderStatus Class'
    {
        CLSID = s '{266836AD-A50D-11D2-
BA4E-00C04FBFE08B}'
    }
    TPCC.OrderStatus = s 'OrderStatus Class'
    {

```

```

        CurVer = s 'TPCC.OrderStatus.1'
    }
    NoRemove CLSID
    {
        ForceRemove {266836AD-A50D-11D2-
BA4E-00C04FBFE08B} = s 'OrderStatus Class'
        {
            ProgID = s
'TPCC.OrderStatus.1'

            VersionIndependentProgID = s
'TPCC.OrderStatus'

            InprocServer32 = s
'%MODULE%'
        {
            val
ThreadingModel = s 'Both'
        }
    }
}

```

tpcc_com_pay.rgs

```

HKCR
{
    TPCC.Payment.1 = s 'Payment Class'
    {
        CLSID = s '{CD02F7EF-A4FA-11D2-
BA4E-00C04FBFE08B}'
    }
    TPCC.Payment = s 'Payment Class'
    {
        CurVer = s 'TPCC.Payment.1'
    }
    NoRemove CLSID
    {
        ForceRemove {CD02F7EF-A4FA-11D2-
BA4E-00C04FBFE08B} = s 'Payment Class'
        {
            ProgID = s
'TPCC.Payment.1'

            VersionIndependentProgID = s 'TPCC.Payment'
            InprocServer32 = s
'%MODULE%'
        {
            val
ThreadingModel = s 'Both'
        }
    }
}

```

tpcc_com_ps.h

```

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

```

```

/* this ALWAYS GENERATED file contains the
definitions for the interfaces */

```

```

/* File created by MIDL compiler version 6.00.0347
*/
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, W1, Zp8, env-Win32 (32b run)
protocol : dce , ms_ext, c_ext
error checks: allocation ref bounds_check enum
stub_data
VC __declspec() decoration level:
___declspec(uuid()), ___declspec(selectany),
___declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

```

```

/* verify that the <rpcndr.h> version is high enough
to compile this file*/
#ifdef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

```

```

#include "rpc.h"
#include "rpcndr.h"

#ifdef __RPCNDR_H_VERSION__
#error this stub requires an updated version of
<rpcndr.h>
#endif // __RPCNDR_H_VERSION__

```

```

#ifdef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/

```

```

#ifdef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

#ifdef _MSC_VER) && (_MSC_VER >= 1020)
#pragma once
#endif

```

```

/* Forward Declarations */

```

```

#ifdef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#endif /* __ITPCC_FWD_DEFINED__ */

```

```

/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"

```

```

#ifdef __cplusplus

```

```

extern "C"{
#endif

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

```

```

/* interface __MIDL_itf_tpcc_com_ps_0000 */
/* [local] */

```

```

extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

```

```

#ifdef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

```

```

/* interface ITPCC */
/* [unique][helpstring][uuid][oleautomation][object]
*/

```

```

EXTERN_C const IID IID_ITPCC;

```

```

#ifdef __cplusplus) && !defined(CINTERFACE)

```

```

MIDL_INTERFACE("FEEB6AA2-84B1-11d2-BA47-
00C04FBFE08B")
ITPCC : public IUnknown
{
public:
    virtual HRESULT __stdcall NewOrder(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall Payment(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall Delivery(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall StockLevel(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall OrderStatus(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall CallSetComplete(
        void) = 0;
};

```

```

#else /* C style interface */

```

```

typedef struct ITPCCVtbl
{
    BEGIN_INTERFACE

    HRESULT ( STDMETHODCALLTYPE *QueryInterface )
    {
        ITPCC * This,
        /* [in] */ REFIID riid,
        /* [iid_is] [out] */ void **ppvObject);

    ULONG ( STDMETHODCALLTYPE *AddRef ) (
        ITPCC * This);

    ULONG ( STDMETHODCALLTYPE *Release ) (
        ITPCC * This);

    HRESULT ( STDMETHODCALLTYPE *NewOrder ) (
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( STDMETHODCALLTYPE *Payment ) (
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( STDMETHODCALLTYPE *Delivery ) (
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( STDMETHODCALLTYPE *StockLevel ) (
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( STDMETHODCALLTYPE *OrderStatus ) (
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( STDMETHODCALLTYPE *CallSetComplete ) (
        ITPCC * This);

    END_INTERFACE
} ITPCCVtbl;

interface ITPCC
{
    CONST_VTBL struct ITPCCVtbl *lpVtbl;
};

#ifndef COBJMACROS

#define ITPCC_QueryInterface(This,riid,ppvObject) \
(This)->lpVtbl->QueryInterface(This,riid,ppvObject)

```

```

#define ITPCC_AddRef(This) \
(This)->lpVtbl->AddRef(This)

#define ITPCC_Release(This) \
(This)->lpVtbl->Release(This)

#define ITPCC_NewOrder(This,txn_in,txn_out) \
(This)->lpVtbl->NewOrder(This,txn_in,txn_out)

#define ITPCC_Payment(This,txn_in,txn_out) \
(This)->lpVtbl->Payment(This,txn_in,txn_out)

#define ITPCC_Delivery(This,txn_in,txn_out) \
(This)->lpVtbl->Delivery(This,txn_in,txn_out)

#define ITPCC_StockLevel(This,txn_in,txn_out) \
(This)->lpVtbl->StockLevel(This,txn_in,txn_out)

#define ITPCC_OrderStatus(This,txn_in,txn_out) \
(This)->lpVtbl->OrderStatus(This,txn_in,txn_out)

#define ITPCC_CallSetComplete(This) \
(This)->lpVtbl->CallSetComplete(This)

#endif /* COBJMACROS */

#endif /* C style interface */

HRESULT STDMETHODCALLTYPE ITPCC_NewOrder_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_Payment_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_Delivery_Proxy(
    ITPCC * This,

```

```

/* [in] */ VARIANT txn_in,
/* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_StockLevel_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_OrderStatus_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_CallSetComplete_Proxy(
    ITPCC * This);

void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

#endif /* __ITPCC_INTERFACE_DEFINED__ */

/* Additional Prototypes for ALL interfaces */
unsigned long __RPC_USER
VARIANT_UserSize( unsigned long *, unsigned long
, VARIANT * );
unsigned char * __RPC_USER
VARIANT_UserMarshal( unsigned long *, unsigned char *, VARIANT * );

```

```

unsigned char * __RPC_USER
VARIANT_UserUnmarshal(unsigned long *, unsigned char
*, VARIANT * );
void
__RPC_USER
VARIANT_UserFree( unsigned long *, VARIANT * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

tpcc_com_ps_i.c

```

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

/* this ALWAYS GENERATED file contains the IIDs and
CLSIDs */

/* link this file in with the server and any clients
*/

/* File created by MIDL compiler version 6.00.0347
*/
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext
error checks: allocation ref bounds_check enum
stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \

#define __IID_DEFINED__

#endif /* !defined(_M_IA64) && !defined(_M_AMD64) */

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

/* this ALWAYS GENERATED file contains the IIDs and
CLSIDs */

```

```

#undef INITGUID
#else
#include <guiddef.h>
#endif

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \

DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifdef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \
    const type name =
{1,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEB6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0x0,
0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* !defined(_M_IA64) && !defined(_M_AMD64) */

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

/* this ALWAYS GENERATED file contains the IIDs and
CLSIDs */

```

```

/* link this file in with the server and any clients
*/

/* File created by MIDL compiler version 6.00.0347
*/
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, W1, Zp8, env=Win64 (32b run,appending)
protocol : dce , ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum
stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if defined(_M_IA64) || defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \

DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifdef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;

#endif // __IID_DEFINED__

```

```

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \
    const type name =
    {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}
#endif !MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC
0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* defined(M_IA64) || defined(M_AMD64)*/



---



## tpcc_com_ps_p.c



---


#pragma warning( disable: 4049 ) /* more than 64k
source lines */

/* this ALWAYS GENERATED file contains the proxy stub
code */

/* File created by MIDL compiler version 6.00.0347
*/
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, Wl, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext
error checks: allocation ref bounds_check enum
stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADER( )

#if !defined(M_IA64) && !defined(M_AMD64)
#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high
enough to compile this file*/

```

```

#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 440
#endif

#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of
<rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 1023
#define PROC_FORMAT_STRING_SIZE 193
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

typedef struct _MIDL_TYPE_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

static RPC_SYNTAX_IDENTIFIER RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,
0x10,0x48,0x60}},{2,0}};

extern const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString;

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
ITPCC_ProxyInfo;

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

#if !defined(_RPC_WIN32_)
#error Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this
stub because it uses these features:

```

```

#error -Oif or -Oicf, [wire_marshall] or
[user_marshall] attribute.
#error However, your C/C++ compilation flags indicate
you intend to run this app on earlier systems.
#error This app will die there with the
RPC_X_WRONG_STUB_VERSION error.
#endif

static const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString =
{
    0,
    {
        /* Procedure NewOrder */

        FC_AUTO_HANDLE /*
                                0x33,
                                */
        Old Flags: object, Oi2 /*
                                0x6c,
                                */
        /* 2 */ NdrFcLong( 0x0 ), /* 0 */
        /* 6 */ NdrFcShort( 0x3 ), /* 3 */
        /* 8 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
        /* 10 */ NdrFcShort( 0x0 ), /* 0 */
        /* 12 */ NdrFcShort( 0x8 ), /* 8 */
        /* 14 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
                                0x3,
                                */
        3 /*
                                */

        /* Parameter txn_in */

        /* 16 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
        /* 18 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
        /* 20 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

        /* Parameter txn_out */

        /* 22 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
        /* 24 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
        /* 26 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

        /* Return value */

        /* 28 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
        /* 30 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
        /* 32 */ 0x8, /* FC_LONG */
                                0x0,
                                */
        0 /*
                                */
    }
}

```



```

/* Procedure Payment */
/* 34 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 36 */ NdrFcLong( 0x0 ), /* 0 */
/* 40 */ NdrFcShort( 0x4 ), /* 4 */
/* 42 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
/* 44 */ NdrFcShort( 0x0 ), /* 0 */
/* 46 */ NdrFcShort( 0x8 ), /* 8 */
/* 48 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 50 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 52 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
/* 54 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 56 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 58 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
/* 60 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 62 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 64 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
/* 66 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure Delivery */
/* 68 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 70 */ NdrFcLong( 0x0 ), /* 0 */
/* 74 */ NdrFcShort( 0x5 ), /* 5 */
/* 76 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
/* 78 */ NdrFcShort( 0x0 ), /* 0 */
/* 80 */ NdrFcShort( 0x8 ), /* 8 */
/* 82 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

```

```

/* Parameter txn_in */
/* 84 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 86 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
/* 88 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 90 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 92 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
/* 94 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 96 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 98 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
/* 100 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure StockLevel */

/* 102 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 104 */ NdrFcLong( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x6 ), /* 6 */
/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* 114 */ NdrFcShort( 0x8 ), /* 8 */
/* 116 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 118 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
/* 122 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */

```

```

/* 128 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 130 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 132 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
/* 134 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure OrderStatus */

/* 136 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 138 */ NdrFcLong( 0x0 ), /* 0 */
/* 142 */ NdrFcShort( 0x7 ), /* 7 */
/* 144 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
/* 146 */ NdrFcShort( 0x0 ), /* 0 */
/* 148 */ NdrFcShort( 0x8 ), /* 8 */
/* 150 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 152 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 154 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
/* 156 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
/* 162 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
/* 168 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure CallSetComplete */

/* 170 */ 0x33, /* FC_AUTO_HANDLE */

```

```

0x6c, /*
Old Flags: object, Oi2 */
/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x8 ), /* 8 */
/* 178 */ NdrFcShort( 0x8 ), /* x86 Stack
size/offset = 8 */
/* 180 */ NdrFcShort( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x8 ), /* 8 */
/* 184 */ 0x4, /* Oi2 Flags: has
return, */
0x1, /*
1 */

/* Return value */

/* 186 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 188 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
/* 190 */ 0x8, /* FC_LONG */
0x0, /*
0 */

0x0

};

static const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString =
{
    0,
    {
        0 /* NdrFcShort( 0x0 ), */
/* 2 */
0x12, 0x0, /*
FC_UP */
/* 4 */ NdrFcShort( 0x3ca ), /* Offset=
970 (974) */
/* 6 */
0x2b, /*
FC_NON_ENCAPSULATED_UNION */
0x9, /*
FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT
*/
0x0, /*
*/
/* 10 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 12 */ NdrFcShort( 0x2 ), /* Offset= 2 (14) */
/* 14 */ NdrFcShort( 0x10 ), /* 16 */
/* 16 */ NdrFcShort( 0x2f ), /* 47 */
/* 18 */ NdrFcLong( 0x14 ), /* 20 */
/* 22 */ NdrFcShort( 0x8000b ), /* Simple arm
type: FC_HYPER */
/* 24 */ NdrFcLong( 0x3 ), /* 3 */
/* 28 */ NdrFcShort( 0x80008 ), /* Simple arm
type: FC_LONG */
/* 30 */ NdrFcLong( 0x11 ), /* 17 */
/* 34 */ NdrFcShort( 0x80001 ), /* Simple arm
type: FC_BYTE */

```

```

/* 36 */ NdrFcLong( 0x2 ), /* 2 */
/* 40 */ NdrFcShort( 0x80006 ), /* Simple arm
type: FC_SHORT */
/* 42 */ NdrFcLong( 0x4 ), /* 4 */
/* 46 */ NdrFcShort( 0x8000a ), /* Simple arm
type: FC_FLOAT */
/* 48 */ NdrFcLong( 0x5 ), /* 5 */
/* 52 */ NdrFcShort( 0x8000c ), /* Simple arm
type: FC_DOUBLE */
/* 54 */ NdrFcLong( 0xb ), /* 11 */
/* 58 */ NdrFcShort( 0x80006 ), /* Simple arm
type: FC_SHORT */
/* 60 */ NdrFcLong( 0xa ), /* 10 */
/* 64 */ NdrFcShort( 0x80008 ), /* Simple arm
type: FC_LONG */
/* 66 */ NdrFcLong( 0x6 ), /* 6 */
/* 70 */ NdrFcShort( 0xe8 ), /* Offset= 232 (302) */
/* 72 */ NdrFcLong( 0x7 ), /* 7 */
/* 76 */ NdrFcShort( 0x8000c ), /* Simple arm
type: FC_DOUBLE */
/* 78 */ NdrFcLong( 0x8 ), /* 8 */
/* 82 */ NdrFcShort( 0xe2 ), /* Offset= 226 (308) */
/* 84 */ NdrFcLong( 0xd ), /* 13 */
/* 88 */ NdrFcShort( 0xf4 ), /* Offset= 244 (332) */
/* 90 */ NdrFcLong( 0x9 ), /* 9 */
/* 94 */ NdrFcShort( 0x100 ), /* Offset=
256 (350) */
/* 96 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 100 */ NdrFcShort( 0x10c ), /* Offset=
268 (368) */
/* 102 */ NdrFcLong( 0x24 ), /* 36 */
/* 106 */ NdrFcShort( 0x31a ), /* Offset=
794 (900) */
/* 108 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 112 */ NdrFcShort( 0x314 ), /* Offset=
788 (900) */
/* 114 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 118 */ NdrFcShort( 0x312 ), /* Offset=
786 (904) */
/* 120 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 124 */ NdrFcShort( 0x310 ), /* Offset=
784 (908) */
/* 126 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 130 */ NdrFcShort( 0x30e ), /* Offset=
782 (912) */
/* 132 */ NdrFcLong( 0x4014 ), /* 16404 */
/* 136 */ NdrFcShort( 0x30c ), /* Offset=
780 (916) */
/* 138 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 142 */ NdrFcShort( 0x30a ), /* Offset=
778 (920) */
/* 144 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 148 */ NdrFcShort( 0x308 ), /* Offset=
776 (924) */
/* 150 */ NdrFcLong( 0x400b ), /* 16395 */
/* 154 */ NdrFcShort( 0x2f2 ), /* Offset=
754 (908) */
/* 156 */ NdrFcLong( 0x400a ), /* 16394 */
/* 160 */ NdrFcShort( 0x2f0 ), /* Offset=
752 (912) */
/* 162 */ NdrFcLong( 0x4006 ), /* 16390 */

```

```

/* 166 */ NdrFcShort( 0x2fa ), /* Offset=
762 (928) */
/* 168 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 172 */ NdrFcShort( 0x2f0 ), /* Offset=
752 (924) */
/* 174 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 178 */ NdrFcShort( 0x2f2 ), /* Offset=
754 (932) */
/* 180 */ NdrFcLong( 0x400d ), /* 16397 */
/* 184 */ NdrFcShort( 0x2f0 ), /* Offset=
752 (936) */
/* 186 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 190 */ NdrFcShort( 0x2ee ), /* Offset=
750 (940) */
/* 192 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 196 */ NdrFcShort( 0x2ec ), /* Offset=
748 (944) */
/* 198 */ NdrFcLong( 0x400c ), /* 16396 */
/* 202 */ NdrFcShort( 0x2ea ), /* Offset=
746 (948) */
/* 204 */ NdrFcLong( 0x10 ), /* 16 */
/* 208 */ NdrFcShort( 0x8002 ), /* Simple arm
type: FC_CHAR */
/* 210 */ NdrFcLong( 0x12 ), /* 18 */
/* 214 */ NdrFcShort( 0x8006 ), /* Simple arm
type: FC_SHORT */
/* 216 */ NdrFcLong( 0x13 ), /* 19 */
/* 220 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 222 */ NdrFcLong( 0x15 ), /* 21 */
/* 226 */ NdrFcShort( 0x800b ), /* Simple arm
type: FC_HYPER */
/* 228 */ NdrFcLong( 0x16 ), /* 22 */
/* 232 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 234 */ NdrFcLong( 0x17 ), /* 23 */
/* 238 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 240 */ NdrFcLong( 0xe ), /* 14 */
/* 244 */ NdrFcShort( 0x2c8 ), /* Offset=
712 (956) */
/* 246 */ NdrFcLong( 0x400e ), /* 16398 */
/* 250 */ NdrFcShort( 0x2cc ), /* Offset=
716 (966) */
/* 252 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 256 */ NdrFcShort( 0x2ca ), /* Offset=
714 (970) */
/* 258 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 262 */ NdrFcShort( 0x286 ), /* Offset=
646 (908) */
/* 264 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 268 */ NdrFcShort( 0x284 ), /* Offset=
644 (912) */
/* 270 */ NdrFcLong( 0x4015 ), /* 16405 */
/* 274 */ NdrFcShort( 0x282 ), /* Offset=
642 (916) */
/* 276 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 280 */ NdrFcShort( 0x278 ), /* Offset=
632 (912) */
/* 282 */ NdrFcLong( 0x4017 ), /* 16407 */

```

```

/* 286 */ NdrFcShort( 0x272 ), /* Offset=
626 (912) */
/* 288 */ NdrFcLong( 0x0 ), /* 0 */
/* 292 */ NdrFcShort( 0x0 ), /* Offset= 0 (292) */
/* 294 */ NdrFcLong( 0x1 ), /* 1 */
/* 298 */ NdrFcShort( 0x0 ), /* Offset= 0 (298) */
/* 300 */ NdrFcShort( 0xffffffff ), /* Offset= -1
(299) */
/* 302 */
0x15, /*
FC_STRUCT */
0x7, /*
7 */
/* 304 */ NdrFcShort( 0x8 ), /* 8 */
/* 306 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 308 */
0x12, 0x0, /*
FC_UP */
/* 310 */ NdrFcShort( 0xc ), /* Offset= 12 (322) */
/* 312 */
0x1b, /*
FC_CARRAY */
0x1, /*
1 */
/* 314 */ NdrFcShort( 0x2 ), /* 2 */
/* 316 */ 0x9, /* Corr desc: FC_ULONG
*/
0x0, /*
*/
/* 318 */ NdrFcShort( 0xfffc ), /* -4 */
/* 320 */ 0x6, /* FC_SHORT */
0x5b, /*
FC_END */
/* 322 */
0x17, /*
FC_CSTRUCT */
0x3, /*
3 */
/* 324 */ NdrFcShort( 0x8 ), /* 8 */
/* 326 */ NdrFcShort( 0xffffffff2 ), /* Offset= -
14 (312) */
/* 328 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 330 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 332 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 334 */ NdrFcLong( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ NdrFcShort( 0x0 ), /* 0 */
/* 342 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 344 */ 0x0, /* 0 */

```

```

0x0, /*
0 */
/* 346 */ 0x0, /* 0 */
0x0, /*
0 */
/* 348 */ 0x0, /* 0 */
0x46, /*
70 */
/* 350 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 352 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 356 */ NdrFcShort( 0x0 ), /* 0 */
/* 358 */ NdrFcShort( 0x0 ), /* 0 */
/* 360 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 362 */ 0x0, /* 0 */
0x0, /*
0 */
/* 364 */ 0x0, /* 0 */
0x0, /*
0 */
/* 366 */ 0x0, /* 0 */
0x46, /*
70 */
/* 368 */
0x12, 0x10, /*
FC_UP [pointer deref] */
/* 370 */ NdrFcShort( 0x2 ), /* Offset= 2 (372) */
/* 372 */
0x12, 0x0, /*
FC_UP */
/* 374 */ NdrFcShort( 0x1fc ), /* Offset=
508 (882) */
/* 376 */
0x2a, /*
FC_ENCAPSULATED_UNION */
0x49, /*
73 */
/* 378 */ NdrFcShort( 0x18 ), /* 24 */
/* 380 */ NdrFcShort( 0xa ), /* 10 */
/* 382 */ NdrFcLong( 0x8 ), /* 8 */
/* 386 */ NdrFcShort( 0x58 ), /* Offset= 88 (474) */
/* 388 */ NdrFcLong( 0xd ), /* 13 */
/* 392 */ NdrFcShort( 0x78 ), /* Offset= 120 (512) */
/* 394 */ NdrFcLong( 0x9 ), /* 9 */
/* 398 */ NdrFcShort( 0x94 ), /* Offset= 148 (546) */
/* 400 */ NdrFcLong( 0xc ), /* 12 */
/* 404 */ NdrFcShort( 0xbc ), /* Offset= 188 (592) */
/* 406 */ NdrFcLong( 0x24 ), /* 36 */
/* 410 */ NdrFcShort( 0x114 ), /* Offset=
276 (686) */
/* 412 */ NdrFcLong( 0x800d ), /* 32781 */
/* 416 */ NdrFcShort( 0x130 ), /* Offset=
304 (720) */
/* 418 */ NdrFcLong( 0x10 ), /* 16 */
/* 422 */ NdrFcShort( 0x148 ), /* Offset=
328 (750) */

```

```

/* 424 */ NdrFcLong( 0x2 ), /* 2 */
/* 428 */ NdrFcShort( 0x160 ), /* Offset=
352 (780) */
/* 430 */ NdrFcLong( 0x3 ), /* 3 */
/* 434 */ NdrFcShort( 0x178 ), /* Offset=
376 (810) */
/* 436 */ NdrFcLong( 0x14 ), /* 20 */
/* 440 */ NdrFcShort( 0x190 ), /* Offset=
400 (840) */
/* 442 */ NdrFcShort( 0xffffffff ), /* Offset= -1
(441) */
/* 444 */
0x1b, /*
FC_CARRAY */
0x3, /*
3 */
/* 446 */ NdrFcShort( 0x4 ), /* 4 */
/* 448 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 454 */
0x48, /*
FC_VARIABLE_REPEAT */
0x49, /*
FC_FIXED_OFFSET */
/* 456 */ NdrFcShort( 0x4 ), /* 4 */
/* 458 */ NdrFcShort( 0x0 ), /* 0 */
/* 460 */ NdrFcShort( 0x1 ), /* 1 */
/* 462 */ NdrFcShort( 0x0 ), /* 0 */
/* 464 */ NdrFcShort( 0x0 ), /* 0 */
/* 466 */ 0x12, 0x0, /* FC_UP */
/* 468 */ NdrFcShort( 0xffffffff6e ), /* Offset= -
146 (322) */
/* 470 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 472 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 474 */
0x16, /*
FC_PSTRUCT */
0x3, /*
3 */
/* 476 */ NdrFcShort( 0x8 ), /* 8 */
/* 478 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 480 */

```

```

0x46, /*
FC_NO_REPEAT */
0x5c, /*
FC_PAD */
/* 482 */ NdrFcShort( 0x4 ), /* 4 */
/* 484 */ NdrFcShort( 0x4 ), /* 4 */
/* 486 */ 0x11, 0x0, /* FC_RP */
/* 488 */ NdrFcShort( 0xfffffd4 ), /* Offset= -
44 (444) */
/* 490 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 492 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 494 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 496 */ NdrFcShort( 0x0 ), /* 0 */
/* 498 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 500 */ NdrFcShort( 0x0 ), /* 0 */
/* 502 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 506 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 508 */ NdrFcShort( 0xfffff50 ), /* Offset= -
176 (332) */
/* 510 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 512 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 514 */ NdrFcShort( 0x8 ), /* 8 */
/* 516 */ NdrFcShort( 0x0 ), /* 0 */
/* 518 */ NdrFcShort( 0x6 ), /* Offset= 6 (524) */
/* 520 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER */
/* 522 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 524 */
0x11, 0x0, /*
FC_RP */
/* 526 */ NdrFcShort( 0xfffffe0 ), /* Offset= -
32 (494) */
/* 528 */
0x21, /*
FC_BOGUS_ARRAY */

```

```

0x3, /*
3 */
/* 530 */ NdrFcShort( 0x0 ), /* 0 */
/* 532 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 534 */ NdrFcShort( 0x0 ), /* 0 */
/* 536 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 540 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 542 */ NdrFcShort( 0xfffff40 ), /* Offset= -
192 (350) */
/* 544 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 546 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 548 */ NdrFcShort( 0x8 ), /* 8 */
/* 550 */ NdrFcShort( 0x0 ), /* 0 */
/* 552 */ NdrFcShort( 0x6 ), /* Offset= 6 (558) */
/* 554 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER */
/* 556 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 558 */
0x11, 0x0, /*
FC_RP */
/* 560 */ NdrFcShort( 0xfffffe0 ), /* Offset= -
32 (528) */
/* 562 */
0x1b, /*
FC_CARRAY */
0x3, /*
3 */
/* 564 */ NdrFcShort( 0x4 ), /* 4 */
/* 566 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 568 */ NdrFcShort( 0x0 ), /* 0 */
/* 570 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 572 */
0x48, /*
FC_VARIABLE_REPEAT */
0x49, /*
FC_FIXED_OFFSET */
/* 574 */ NdrFcShort( 0x4 ), /* 4 */
/* 576 */ NdrFcShort( 0x0 ), /* 0 */
/* 578 */ NdrFcShort( 0x1 ), /* 1 */

```

```

/* 580 */ NdrFcShort( 0x0 ), /* 0 */
/* 582 */ NdrFcShort( 0x0 ), /* 0 */
/* 584 */ 0x12, 0x0, /* FC_UP */
/* 586 */ NdrFcShort( 0x184 ), /* Offset=
388 (974) */
/* 588 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 590 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 592 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 594 */ NdrFcShort( 0x8 ), /* 8 */
/* 596 */ NdrFcShort( 0x0 ), /* 0 */
/* 598 */ NdrFcShort( 0x6 ), /* Offset= 6 (604) */
/* 600 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER */
/* 602 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 604 */
0x11, 0x0, /*
FC_RP */
/* 606 */ NdrFcShort( 0xfffffd4 ), /* Offset= -
44 (562) */
/* 608 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 610 */ NdrFcLong( 0x2f ), /* 47 */
/* 614 */ NdrFcShort( 0x0 ), /* 0 */
/* 616 */ NdrFcShort( 0x0 ), /* 0 */
/* 618 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 620 */ 0x0, /* 0 */
0x0, /*
0 */
/* 622 */ 0x0, /* 0 */
0x0, /*
0 */
/* 624 */ 0x0, /* 0 */
0x46, /*
70 */
/* 626 */
0x1b, /*
FC_CARRAY */
0x0, /*
0 */
/* 628 */ NdrFcShort( 0x1 ), /* 1 */
/* 630 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */

```

```

0x0, /*
/* 632 */ NdrFcShort( 0x4 ), /* 4 */
/* 634 */ 0x1, /* FC_BYTE */
FC_END /*
/* 636 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 638 */ NdrFcShort( 0x10 ), /* 16 */
/* 640 */ NdrFcShort( 0x0 ), /* 0 */
/* 642 */ NdrFcShort( 0xa ), /* Offset= 10 (652) */
/* 644 */ 0x8, /* FC_LONG */
FC_LONG /*
/* 646 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /*
0 */
/* 648 */ NdrFcShort( 0xfffffd8 ), /* Offset= -
40 (608) */
/* 650 */ 0x36, /* FC_POINTER */
FC_END /*
/* 652 */
0x12, 0x0, /*
FC_UP /*
/* 654 */ NdrFcShort( 0xfffffe4 ), /* Offset= -
28 (626) */
/* 656 */
0x1b, /*
FC_CARRAY /*
0x3, /*
3 */
/* 658 */ NdrFcShort( 0x4 ), /* 4 */
/* 660 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 662 */ NdrFcShort( 0x0 ), /* 0 */
/* 664 */
0x4b, /*
FC_PP /*
0x5c, /*
FC_PAD /*
/* 666 */
0x48, /*
FC_VARIABLE_REPEAT */
0x49, /*
FC_FIXED_OFFSET */
/* 668 */ NdrFcShort( 0x4 ), /* 4 */
/* 670 */ NdrFcShort( 0x0 ), /* 0 */
/* 672 */ NdrFcShort( 0x1 ), /* 1 */
/* 674 */ NdrFcShort( 0x0 ), /* 0 */
/* 676 */ NdrFcShort( 0x0 ), /* 0 */
/* 678 */ 0x12, 0x0, /* FC_UP */
/* 680 */ NdrFcShort( 0xfffffd4 ), /* Offset= -
44 (636) */
/* 682 */

```

```

FC_END /*
0x5b, /*
FC_LONG /*
/* 684 */ 0x5c, /* FC_PAD */
FC_END /*
/* 686 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 688 */ NdrFcShort( 0x8 ), /* 8 */
/* 690 */ NdrFcShort( 0x0 ), /* 0 */
/* 692 */ NdrFcShort( 0x6 ), /* Offset= 6 (698) */
/* 694 */ 0x8, /* FC_LONG */
FC_POINTER /*
/* 696 */ 0x5c, /* FC_PAD */
FC_END /*
/* 698 */
0x11, 0x0, /*
FC_RP /*
/* 700 */ NdrFcShort( 0xfffffd4 ), /* Offset= -
44 (656) */
/* 702 */
0x1d, /*
FC_SMFARRAY /*
0x0, /*
0 */
/* 704 */ NdrFcShort( 0x8 ), /* 8 */
/* 706 */ 0x1, /* FC_BYTE */
FC_END /*
/* 708 */
0x15, /*
FC_STRUCT */
0x3, /*
3 */
/* 710 */ NdrFcShort( 0x10 ), /* 16 */
/* 712 */ 0x8, /* FC_LONG */
FC_SHORT /*
/* 714 */ 0x6, /* FC_SHORT */
FC_EMBEDDED_COMPLEX /*
/* 716 */ 0x0, /* 0 */
NdrFcShort( 0xfffff1
), /* Offset= -15 (702) */
FC_END /*
/* 720 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 722 */ NdrFcShort( 0x18 ), /* 24 */
/* 724 */ NdrFcShort( 0x0 ), /* 0 */
/* 726 */ NdrFcShort( 0xa ), /* Offset= 10 (736) */

```

```

/* 728 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER /*
/* 730 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /*
0 */
/* 732 */ NdrFcShort( 0xfffffe8 ), /* Offset= -
24 (708) */
/* 734 */ 0x5c, /* FC_PAD */
FC_END /*
/* 736 */
0x11, 0x0, /*
FC_RP /*
/* 738 */ NdrFcShort( 0xfffff0c ), /* Offset= -
244 (494) */
/* 740 */
0x1b, /*
FC_CARRAY /*
0x0, /*
0 */
/* 742 */ NdrFcShort( 0x1 ), /* 1 */
/* 744 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 746 */ NdrFcShort( 0x0 ), /* 0 */
/* 748 */ 0x1, /* FC_BYTE */
FC_END /*
/* 750 */
0x16, /*
FC_PSTRUCT /*
0x3, /*
3 */
/* 752 */ NdrFcShort( 0x8 ), /* 8 */
/* 754 */
0x4b, /*
FC_PP /*
0x5c, /*
FC_PAD /*
/* 756 */
0x46, /*
FC_NO_REPEAT */
0x5c, /*
FC_PAD /*
/* 758 */ NdrFcShort( 0x4 ), /* 4 */
/* 760 */ NdrFcShort( 0x4 ), /* 4 */
/* 762 */ 0x12, 0x0, /* FC_UP */
/* 764 */ NdrFcShort( 0xfffffe8 ), /* Offset= -
24 (740) */
/* 766 */
0x5b, /*
FC_END /*
0x8, /*
FC_LONG /*
/* 768 */ 0x8, /* FC_LONG */
FC_END /*
0x5b, /*

```

/* 770 */				0x3,	/*	/* 854 */ NdrFcShort(0xffffffe8),	/* Offset= -
FC_CARRAY */	0x1b,	/*	3 */			24 (830) */	
	0x1,	/*	/* 812 */ NdrFcShort(0x8),	/* 8 */		/* 856 */	
1 */			/* 814 */				0x5b,
/* 772 */ NdrFcShort(0x2),	/* 2 */		FC_PP */	0x4b,	/*	FC_END */	/*
/* 774 */ 0x19,	/* Corr desc: field			0x5c,	/*		0x8,
pointer, FC_ULONG */			FC_PAD */	/* 816 */		FC_LONG */	/*
	0x0,	/*	/* 816 */			/* 858 */ 0x8,	/* FC_LONG */
*/			FC_NO_REPEAT */	0x46,	/*	0x5b,	/*
/* 776 */ NdrFcShort(0x0),	/* 0 */			0x5c,	/*	FC_END */	/*
/* 778 */ 0x6,	/* FC_SHORT */		FC_PAD */			/* 860 */	
	0x5b,	/*	/* 818 */ NdrFcShort(0x4),	/* 4 */			0x15,
FC_END */			/* 820 */ NdrFcShort(0x4),	/* 4 */		FC_STRUCT */	/*
/* 780 */			/* 822 */ 0x12, 0x0,	/* FC_UP */			0x3,
	0x16,	/*	/* 824 */ NdrFcShort(0xffffffe8),	/* Offset= -		3 */	
FC_PSTRUCT */			24 (800) */			/* 862 */ NdrFcShort(0x8),	/* 8 */
	0x3,	/*	/* 826 */			/* 864 */ 0x8,	/* FC_LONG */
3 */			FC_END */	0x5b,	/*	0x8,	/*
/* 782 */ NdrFcShort(0x8),	/* 8 */					FC_LONG */	
/* 784 */			FC_LONG */	0x8,	/*	/* 866 */ 0x5c,	/* FC_PAD */
	0x4b,	/*	/* 828 */ 0x8,	/* FC_LONG */		0x5b,	/*
FC_PP */			FC_END */	0x5b,	/*	FC_END */	
	0x5c,	/*	/* 830 */			/* 868 */	
FC_PAD */			FC_CARRAY */	0x1b,	/*		0x1b,
/* 786 */				0x7,	/*	FC_CARRAY */	/*
	0x46,	/*	7 */				0x3,
FC_NO_REPEAT */			/* 832 */ NdrFcShort(0x8),	/* 8 */		3 */	
	0x5c,	/*	/* 834 */ 0x19,	/* Corr desc: field		/* 870 */ NdrFcShort(0x8),	/* 8 */
FC_PAD */			pointer, FC_ULONG */			/* 872 */ 0x7,	/* Corr desc: FC_USHORT
/* 788 */ NdrFcShort(0x4),	/* 4 */					*/	/*
/* 790 */ NdrFcShort(0x4),	/* 4 */		/* 836 */ NdrFcShort(0x0),	/* 0 */		0x0,	/*
/* 792 */ 0x12, 0x0,	/* FC_UP */		/* 838 */ 0xb,	/* FC_HYPER */		/* 874 */ NdrFcShort(0xfffd8),	/* -40 */
/* 794 */ NdrFcShort(0xffffffe8),	/* Offset= -					/* 876 */ 0x4c,	/* FC_EMBEDDED_COMPLEX
24 (770) */			FC_END */	0x5b,	/*	0x0,	/*
/* 796 */			/* 836 */ NdrFcShort(0x0),	/* 0 */		/* 878 */ NdrFcShort(0xfffffee),	/* Offset= -
FC_END */			/* 838 */ 0xb,	/* FC_HYPER */		18 (860) */	
	0x8,	/*	FC_END */	0x5b,	/*	/* 880 */ 0x5c,	/* FC_PAD */
			/* 840 */			0x5b,	/*
FC_LONG */			FC_PSTRUCT */	0x16,	/*	FC_END */	
/* 798 */ 0x8,	/* FC_LONG */			0x3,	/*	/* 882 */	
	0x5b,	/*	3 */				0x1a,
FC_END */			/* 842 */ NdrFcShort(0x8),	/* 8 */		FC_BOGUS_STRUCT */	/*
/* 800 */			/* 844 */				0x3,
	0x1b,	/*	FC_PP */	0x4b,	/*	3 */	
FC_CARRAY */				0x5c,	/*	/* 884 */ NdrFcShort(0x28),	/* 40 */
	0x3,	/*	FC_PAD */			/* 886 */ NdrFcShort(0xfffffee),	/* Offset= -
3 */			/* 846 */			18 (868) */	
/* 802 */ NdrFcShort(0x4),	/* 4 */		FC_NO_REPEAT */	0x46,	/*	/* 888 */ NdrFcShort(0x0),	/* Offset= 0 (888) */
/* 804 */ 0x19,	/* Corr desc: field			0x5c,	/*	/* 890 */ 0x6,	/* FC_SHORT */
pointer, FC_ULONG */			FC_PAD */			0x6,	/*
	0x0,	/*	/* 846 */			FC_SHORT */	
*/			FC_NO_REPEAT */	0x46,	/*	/* 892 */ 0x8,	/* FC_LONG */
/* 806 */ NdrFcShort(0x0),	/* 0 */			0x5c,	/*	0x8,	/*
/* 808 */ 0x8,	/* FC_LONG */		FC_PAD */			FC_LONG */	
	0x5b,	/*	/* 848 */ NdrFcShort(0x4),	/* 4 */		/* 894 */ 0x4c,	/* FC_EMBEDDED_COMPLEX
FC_END */			/* 850 */ NdrFcShort(0x4),	/* 4 */		*/	
/* 810 */			/* 852 */ 0x12, 0x0,	/* FC_UP */			
	0x16,	/*					
FC_PSTRUCT */							

```

0 */
/* 896 */ NdrPcShort( 0xfffffd8 ), /* Offset= -
520 (376) */
/* 898 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 900 */
0x12, 0x0, /*
FC_UP */
/* 902 */ NdrPcShort( 0xfffffef6 ), /* Offset= -
266 (636) */
/* 904 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 906 */ 0x1, /* FC_BYTE */
0x5c, /*
FC_PAD */
/* 908 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 910 */ 0x6, /* FC_SHORT */
0x5c, /*
FC_PAD */
/* 912 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 914 */ 0x8, /* FC_LONG */
0x5c, /*
FC_PAD */
/* 916 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 918 */ 0xb, /* FC_HYPER */
0x5c, /*
FC_PAD */
/* 920 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 922 */ 0xa, /* FC_FLOAT */
0x5c, /*
FC_PAD */
/* 924 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 926 */ 0xc, /* FC_DOUBLE */
0x5c, /*
FC_PAD */
/* 928 */
0x12, 0x0, /*
FC_UP */
/* 930 */ NdrPcShort( 0xfffffd8c ), /* Offset= -
628 (302) */
/* 932 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 934 */ NdrPcShort( 0xffffd8e ), /* Offset= -
626 (308) */
/* 936 */
0x12, 0x10, /*
FC_UP [pointer_deref] */

```

```

/* 938 */ NdrPcShort( 0xffffda2 ), /* Offset= -
606 (332) */
/* 940 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 942 */ NdrPcShort( 0xffffdb0 ), /* Offset= -
592 (350) */
/* 944 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 946 */ NdrPcShort( 0xffffd8e ), /* Offset= -
578 (368) */
/* 948 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 950 */ NdrPcShort( 0x2 ), /* Offset= 2 (952) */
/* 952 */
0x12, 0x0, /*
FC_UP */
/* 954 */ NdrPcShort( 0x14 ), /* Offset= 20 (974) */
/* 956 */
0x15, /*
FC_STRUCT */
0x7, /*
7 */
/* 958 */ NdrPcShort( 0x10 ), /* 16 */
/* 960 */ 0x6, /* FC_SHORT */
0x1, /*
FC_BYTE */
/* 962 */ 0x1, /* FC_BYTE */
0x8, /*
FC_LONG */
/* 964 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 966 */
0x12, 0x0, /*
FC_UP */
/* 968 */ NdrPcShort( 0xfffffff4 ), /* Offset= -
12 (956) */
/* 970 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 972 */ 0x2, /* FC_CHAR */
0x5c, /*
FC_PAD */
/* 974 */
0x1a, /*
FC_BOGUS_STRUCT */
0x7, /*
7 */
/* 976 */ NdrPcShort( 0x20 ), /* 32 */
/* 978 */ NdrPcShort( 0x0 ), /* 0 */
/* 980 */ NdrPcShort( 0x0 ), /* Offset= 0 (980) */
/* 982 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 984 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 986 */ 0x6, /* FC_SHORT */

```

```

0x6, /*
FC_SHORT */
/* 988 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 990 */ NdrPcShort( 0xffffc28 ), /* Offset= -
984 (6) */
/* 992 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 994 */ 0xb4, /* FC_USER_MARSHAL */
0x83, /*
131 */
/* 996 */ NdrPcShort( 0x0 ), /* 0 */
/* 998 */ NdrPcShort( 0x10 ), /* 16 */
/* 1000 */ NdrPcShort( 0x0 ), /* 0 */
/* 1002 */ NdrPcShort( 0xfffffc18 ), /*
Offset= -1000 (2) */
/* 1004 */
0x11, 0x4, /*
FC_RP [allocated_on_stack] */
/* 1006 */ NdrPcShort( 0x6 ), /* Offset= 6
(1012) */
/* 1008 */
0x13, 0x0, /*
FC_OP */
/* 1010 */ NdrPcShort( 0xfffffcdc ), /*
Offset= -36 (974) */
/* 1012 */ 0xb4, /*
FC_USER_MARSHAL */
0x83, /*
131 */
/* 1014 */ NdrPcShort( 0x0 ), /* 0 */
/* 1016 */ NdrPcShort( 0x10 ), /* 16 */
/* 1018 */ NdrPcShort( 0x0 ), /* 0 */
/* 1020 */ NdrPcShort( 0xfffffff4 ), /*
Offset= -12 (1008) */
0x0
};
static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
{
{
VARIANT_UserSize
,VARIANT_UserMarshal
,VARIANT_UserUnmarshal
,VARIANT_UserFree
}
};
/* Standard interface: __MIDL_itf_tpc_c_com_ps_0000,
ver. 0.0,

```

```

GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,

GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,

GUID={0xFEED6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,0xBF,0xE0,0x8B}} */

#pragma code_seg(".orpc")
static const unsigned short
ITPCC_FormatStringOffsetTable =
{
    0,
    34,
    68,
    102,
    136,
    170
};

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo
=
{
    &Object_StubDesc,
    _MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
};

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
{
    &Object_StubDesc,
    0,
    _MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0
};

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *) (INT_PTR) -1 /* ITPCC::NewOrder */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Payment */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Delivery */ ,
    (void *) (INT_PTR) -1 /* ITPCC::StockLevel */ ,

```

```

    (void *) (INT_PTR) -1 /* ITPCC::OrderStatus */ ,
    (void *) (INT_PTR) -1 /* ITPCC::CallSetComplete
*/
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    _MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x600015b, /* MIDL Version 6.0.0347 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0, /* proxy/server info */
    0 /* Reserved5 */
};

const CInterfaceProxyVtbl *
_tpsc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl *
_tpsc_com_ps_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const
_tpsc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};

```

```

#define _tpcc_com_ps_CHECK_IID(n)
    IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID,
n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID *
pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo
=
{
    (PCInterfaceProxyVtblList *) &
_tpsc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) &
_tpsc_com_ps_StubVtblList,
    (const PCInterfaceName * ) &
_tpsc_com_ps_InterfaceNamesList,
    0, /* no delegation
& _tpcc_com_ps_IID_Lookup,
1,
2,
0, /* table of [async_uuid] interfaces */
0, /* Filler1 */
0, /* Filler2 */
0 /* Filler3 */
};

#endif /* !defined(_M_IA64) && !defined(_M_AMD64)*/

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

/* this ALWAYS GENERATED file contains the proxy stub
code */

/* File created by MIDL compiler version 6.00.0347
*/
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, W1, Zp8, env=Win64 (32b run,appending)
protocol : dce , ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum
stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/

```



```

//@@MIDL_FILE_HEADING( )

#if defined( M_IA64 ) || defined( M_AMD64 )
#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high
enough to compile this file*/
#ifndef REDO_RPCPROXY_H_VERSION
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of
<rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 1003
#define PROC_FORMAT_STRING_SIZE 253
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

typedef struct _MIDL_TYPE_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

static RPC_SYNTAX_IDENTIFIER RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,
0x10,0x48,0x60}},{2,0}};

extern const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString;

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
ITPCC_ProxyInfo;

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

```

```

#if !defined( __RPC_WIN64__ )
#error Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString =
{
    0,
    {
        /* Procedure NewOrder */

        FC_AUTO_HANDLE /*
                                0x33,
                                */
        /*
                                0x6c,
                                */
        Old Flags: object, Oi2 /*
/* 2 */ NdrFcLong( 0x0 ), /* 0 */
/* 6 */ NdrFcShort( 0x3 ), /* 3 */
/* 8 */ NdrFcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
3 /*
/* 16 */ 0xa, /* 10 */
0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr
check, */
/* 18 */ NdrFcShort( 0x20 ), /* 32 */
/* 20 */ NdrFcShort( 0x20 ), /* 32 */
/* 22 */ NdrFcShort( 0x0 ), /* 0 */
/* 24 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 26 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 28 */ NdrFcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
/* 30 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */

/* Parameter txn_out */

/* 32 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 34 */ NdrFcShort( 0x20 ), /* ia64 Stack
size/offset = 32 */
/* 36 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */

/* Return value */

/* 38 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 40 */ NdrFcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */

```

```

/* 42 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure Payment */

/* 44 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 46 */ NdrFcLong( 0x0 ), /* 0 */
/* 50 */ NdrFcShort( 0x4 ), /* 4 */
/* 52 */ NdrFcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
/* 54 */ NdrFcShort( 0x0 ), /* 0 */
/* 56 */ NdrFcShort( 0x8 ), /* 8 */
/* 58 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
0x3, /*
3 */
/* 60 */ 0xa, /* 10 */
0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr
check, */
/* 62 */ NdrFcShort( 0x20 ), /* 32 */
/* 64 */ NdrFcShort( 0x20 ), /* 32 */
/* 66 */ NdrFcShort( 0x0 ), /* 0 */
/* 68 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 70 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 72 */ NdrFcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
/* 74 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */

/* Parameter txn_out */

/* 76 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 78 */ NdrFcShort( 0x20 ), /* ia64 Stack
size/offset = 32 */
/* 80 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */

/* Return value */

/* 82 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 84 */ NdrFcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
/* 86 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure Delivery */

/* 88 */ 0x33, /* FC_AUTO_HANDLE */

```

```

                                0x6c,          /*
Old Flags: object, Oi2 */
/* 90 */ NdrPcLong( 0x0 ), /* 0 */
/* 94 */ NdrPcShort( 0x5 ), /* 5 */
/* 96 */ NdrPcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
/* 98 */ NdrPcShort( 0x0 ), /* 0 */
/* 100 */ NdrPcShort( 0x8 ), /* 8 */
/* 102 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
/* 104 */ 0xa, /* 10 */
/* 106 */ 0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr
check, */
/* 106 */ NdrPcShort( 0x20 ), /* 32 */
/* 108 */ NdrPcShort( 0x20 ), /* 32 */
/* 110 */ NdrPcShort( 0x0 ), /* 0 */
/* 112 */ NdrPcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 114 */ NdrPcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 116 */ NdrPcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
/* 118 */ NdrPcShort( 0x3ce ), /* Type
Offset=974 */

/* Parameter txn_out */

/* 120 */ NdrPcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 122 */ NdrPcShort( 0x20 ), /* ia64 Stack
size/offset = 32 */
/* 124 */ NdrPcShort( 0x3e0 ), /* Type
Offset=992 */

/* Return value */

/* 126 */ NdrPcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 128 */ NdrPcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
/* 130 */ 0x8, /* FC_LONG */
/* 132 */ 0x0, /*
0 */

/* Procedure StockLevel */

/* 132 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 134 */ NdrPcLong( 0x0 ), /* 0 */
/* 138 */ NdrPcShort( 0x6 ), /* 6 */
/* 140 */ NdrPcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
/* 142 */ NdrPcShort( 0x0 ), /* 0 */
/* 144 */ NdrPcShort( 0x8 ), /* 8 */

```

```

/* 146 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
/* 148 */ 0xa, /* 10 */
/* 150 */ 0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr
check, */
/* 150 */ NdrPcShort( 0x20 ), /* 32 */
/* 152 */ NdrPcShort( 0x20 ), /* 32 */
/* 154 */ NdrPcShort( 0x0 ), /* 0 */
/* 156 */ NdrPcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 158 */ NdrPcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 160 */ NdrPcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
/* 162 */ NdrPcShort( 0x3ce ), /* Type
Offset=974 */

/* Parameter txn_out */

/* 164 */ NdrPcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 166 */ NdrPcShort( 0x20 ), /* ia64 Stack
size/offset = 32 */
/* 168 */ NdrPcShort( 0x3e0 ), /* Type
Offset=992 */

/* Return value */

/* 170 */ NdrPcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 172 */ NdrPcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
/* 174 */ 0x8, /* FC_LONG */
/* 176 */ 0x0, /*
0 */

/* Procedure OrderStatus */

/* 176 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 178 */ NdrPcLong( 0x0 ), /* 0 */
/* 182 */ NdrPcShort( 0x7 ), /* 7 */
/* 184 */ NdrPcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
/* 186 */ NdrPcShort( 0x0 ), /* 0 */
/* 188 */ NdrPcShort( 0x8 ), /* 8 */
/* 190 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
/* 192 */ 0xa, /* 10 */
/* 194 */ 0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr
check, */

```

```

/* 194 */ NdrPcShort( 0x20 ), /* 32 */
/* 196 */ NdrPcShort( 0x20 ), /* 32 */
/* 198 */ NdrPcShort( 0x0 ), /* 0 */
/* 200 */ NdrPcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 202 */ NdrPcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 204 */ NdrPcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
/* 206 */ NdrPcShort( 0x3ce ), /* Type
Offset=974 */

/* Parameter txn_out */

/* 208 */ NdrPcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 210 */ NdrPcShort( 0x20 ), /* ia64 Stack
size/offset = 32 */
/* 212 */ NdrPcShort( 0x3e0 ), /* Type
Offset=992 */

/* Return value */

/* 214 */ NdrPcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 216 */ NdrPcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
/* 218 */ 0x8, /* FC_LONG */
/* 220 */ 0x0, /*
0 */

/* Procedure CallSetComplete */

/* 220 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 222 */ NdrPcLong( 0x0 ), /* 0 */
/* 226 */ NdrPcShort( 0x8 ), /* 8 */
/* 228 */ NdrPcShort( 0x10 ), /* ia64 Stack
size/offset = 16 */
/* 230 */ NdrPcShort( 0x0 ), /* 0 */
/* 232 */ NdrPcShort( 0x8 ), /* 8 */
/* 234 */ 0x44, /* Oi2 Flags: has
return, has ext, */
/* 236 */ 0xa, /* 10 */
/* 238 */ 0x1, /*
1 */
/* 236 */ 0xa, /* 10 */
/* 238 */ 0x1, /*
Ext Flags: new corr desc, */
/* 238 */ NdrPcShort( 0x0 ), /* 0 */
/* 240 */ NdrPcShort( 0x0 ), /* 0 */
/* 242 */ NdrPcShort( 0x0 ), /* 0 */
/* 244 */ NdrPcShort( 0x0 ), /* 0 */

/* Return value */

/* 246 */ NdrPcShort( 0x70 ), /* Flags: out, return,
base type, */

```

```

/* 248 */ NdrFcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
/* 250 */ 0x8, /* FC_LONG */
0x0, /*
0 */
}
};
static const MIDL_TYPE_FORMAT_STRING
_MIDL_TypeFormatString =
{
0,
{
0 */
/* 2 */
FC_UP */
/* 4 */ NdrFcShort( 0x3b6 ), /* Offset=
950 (954) */
/* 6 */
FC_NON_ENCAPSULATED_UNION */
0x2b, /*
0x9, /*
FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT
*/
0x0, /*
*/
/* 10 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 12 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 14 */ NdrFcShort( 0x2 ), /* Offset= 2 (16) */
/* 16 */ NdrFcShort( 0x10 ), /* 16 */
/* 18 */ NdrFcShort( 0x2f ), /* 47 */
/* 20 */ NdrFcLong( 0x14 ), /* 20 */
/* 24 */ NdrFcShort( 0x800b ), /* Simple arm
type: FC_HYPER */
/* 26 */ NdrFcLong( 0x3 ), /* 3 */
/* 30 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 32 */ NdrFcLong( 0x11 ), /* 17 */
/* 36 */ NdrFcShort( 0x8001 ), /* Simple arm
type: FC_BYTE */
/* 38 */ NdrFcLong( 0x2 ), /* 2 */
/* 42 */ NdrFcShort( 0x8006 ), /* Simple arm
type: FC_SHORT */
/* 44 */ NdrFcLong( 0x4 ), /* 4 */
/* 48 */ NdrFcShort( 0x800a ), /* Simple arm
type: FC_FLOAT */
/* 50 */ NdrFcLong( 0x5 ), /* 5 */
/* 54 */ NdrFcShort( 0x800c ), /* Simple arm
type: FC_DOUBLE */
/* 56 */ NdrFcLong( 0xb ), /* 11 */
/* 60 */ NdrFcShort( 0x8006 ), /* Simple arm
type: FC_SHORT */
/* 62 */ NdrFcLong( 0xa ), /* 10 */
/* 66 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */

```

```

/* 68 */ NdrFcLong( 0x6 ), /* 6 */
/* 72 */ NdrFcShort( 0xe8 ), /* Offset= 232 (304) */
/* 74 */ NdrFcLong( 0x7 ), /* 7 */
/* 78 */ NdrFcShort( 0x800c ), /* Simple arm
type: FC_DOUBLE */
/* 80 */ NdrFcLong( 0x8 ), /* 8 */
/* 84 */ NdrFcShort( 0xe2 ), /* Offset= 226 (310) */
/* 86 */ NdrFcLong( 0xd ), /* 13 */
/* 90 */ NdrFcShort( 0xf6 ), /* Offset= 246 (336) */
/* 92 */ NdrFcLong( 0x9 ), /* 9 */
/* 96 */ NdrFcShort( 0x102 ), /* Offset=
258 (354) */
/* 98 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 102 */ NdrFcShort( 0x10e ), /* Offset=
270 (372) */
/* 104 */ NdrFcLong( 0x24 ), /* 36 */
/* 108 */ NdrFcShort( 0x304 ), /* Offset=
772 (880) */
/* 110 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 114 */ NdrFcShort( 0x2fe ), /* Offset=
766 (880) */
/* 116 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 120 */ NdrFcShort( 0x2fc ), /* Offset=
764 (884) */
/* 122 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 126 */ NdrFcShort( 0x2fa ), /* Offset=
762 (888) */
/* 128 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 132 */ NdrFcShort( 0x2f8 ), /* Offset=
760 (892) */
/* 134 */ NdrFcLong( 0x4014 ), /* 16404 */
/* 138 */ NdrFcShort( 0x2f6 ), /* Offset=
758 (896) */
/* 140 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 144 */ NdrFcShort( 0x2f4 ), /* Offset=
756 (900) */
/* 146 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 150 */ NdrFcShort( 0x2f2 ), /* Offset=
754 (904) */
/* 152 */ NdrFcLong( 0x400b ), /* 16395 */
/* 156 */ NdrFcShort( 0x2dc ), /* Offset=
732 (888) */
/* 158 */ NdrFcLong( 0x400a ), /* 16394 */
/* 162 */ NdrFcShort( 0x2da ), /* Offset=
730 (892) */
/* 164 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 168 */ NdrFcShort( 0x2e4 ), /* Offset=
740 (908) */
/* 170 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 174 */ NdrFcShort( 0x2da ), /* Offset=
730 (904) */
/* 176 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 180 */ NdrFcShort( 0x2dc ), /* Offset=
732 (912) */
/* 182 */ NdrFcLong( 0x400d ), /* 16397 */
/* 186 */ NdrFcShort( 0x2da ), /* Offset=
730 (916) */
/* 188 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 192 */ NdrFcShort( 0x2d8 ), /* Offset=
728 (920) */
/* 194 */ NdrFcLong( 0x6000 ), /* 24576 */

```

```

/* 198 */ NdrFcShort( 0x2d6 ), /* Offset=
726 (924) */
/* 200 */ NdrFcLong( 0x400c ), /* 16396 */
/* 204 */ NdrFcShort( 0x2d4 ), /* Offset=
724 (928) */
/* 206 */ NdrFcLong( 0x10 ), /* 16 */
/* 210 */ NdrFcShort( 0x8002 ), /* Simple arm
type: FC_CHAR */
/* 212 */ NdrFcLong( 0x12 ), /* 18 */
/* 216 */ NdrFcShort( 0x8006 ), /* Simple arm
type: FC_SHORT */
/* 218 */ NdrFcLong( 0x13 ), /* 19 */
/* 222 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 224 */ NdrFcLong( 0x15 ), /* 21 */
/* 228 */ NdrFcShort( 0x800b ), /* Simple arm
type: FC_HYPER */
/* 230 */ NdrFcLong( 0x16 ), /* 22 */
/* 234 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 236 */ NdrFcLong( 0x17 ), /* 23 */
/* 240 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 242 */ NdrFcLong( 0xe ), /* 14 */
/* 246 */ NdrFcShort( 0x2b2 ), /* Offset=
690 (936) */
/* 248 */ NdrFcLong( 0x400e ), /* 16398 */
/* 252 */ NdrFcShort( 0x2b6 ), /* Offset=
694 (946) */
/* 254 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 258 */ NdrFcShort( 0x2b4 ), /* Offset=
692 (950) */
/* 260 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 264 */ NdrFcShort( 0x270 ), /* Offset=
624 (888) */
/* 266 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 270 */ NdrFcShort( 0x26e ), /* Offset=
622 (892) */
/* 272 */ NdrFcLong( 0x4015 ), /* 16405 */
/* 276 */ NdrFcShort( 0x26c ), /* Offset=
620 (896) */
/* 278 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 282 */ NdrFcShort( 0x262 ), /* Offset=
610 (892) */
/* 284 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 288 */ NdrFcShort( 0x25c ), /* Offset=
604 (892) */
/* 290 */ NdrFcLong( 0x0 ), /* 0 */
/* 294 */ NdrFcShort( 0x0 ), /* Offset= 0 (294) */
/* 296 */ NdrFcLong( 0x1 ), /* 1 */
/* 300 */ NdrFcShort( 0x0 ), /* Offset= 0 (300) */
/* 302 */ NdrFcShort( 0xfffffff ), /* Offset= -1
(301) */
/* 304 */
FC_STRUCT */
0x15, /*
7 */
0x7, /*
*/
/* 306 */ NdrFcShort( 0x8 ), /* 8 */
/* 308 */ 0xb, /* FC_HYPER */

```

```

0x5b, /*
FC_END */
/* 310 */
0x12, 0x0, /*
FC_UP */
/* 312 */ NdrFcShort( 0xe ), /* Offset= 14 (326) */
/* 314 */
0x1b, /*
FC_CARRAY */
0x1, /*
1 */
/* 316 */ NdrFcShort( 0x2 ), /* 2 */
/* 318 */ 0x9, /* Corr desc: FC_ULONG */
0x0, /*
*/
/* 320 */ NdrFcShort( 0xffffc ), /* -4 */
/* 322 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 324 */ 0x6, /* FC_SHORT */
0x5b, /*
FC_END */
/* 326 */
0x17, /*
FC_CSTRUCT */
0x3, /*
3 */
/* 328 */ NdrFcShort( 0x8 ), /* 8 */
/* 330 */ NdrFcShort( 0xffffffff0 ), /* Offset= -
16 (314) */
/* 332 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 334 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 336 */
0x2E, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 338 */ NdrFcLong( 0x0 ), /* 0 */
/* 342 */ NdrFcShort( 0x0 ), /* 0 */
/* 344 */ NdrFcShort( 0x0 ), /* 0 */
/* 346 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 348 */ 0x0, /* 0 */
0x0, /*
0 */
/* 350 */ 0x0, /* 0 */
0x0, /*
0 */
/* 352 */ 0x0, /* 0 */
0x46, /*
70 */
/* 354 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */

```

```

/* 356 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 360 */ NdrFcShort( 0x0 ), /* 0 */
/* 362 */ NdrFcShort( 0x0 ), /* 0 */
/* 364 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 366 */ 0x0, /* 0 */
0x0, /*
0 */
/* 368 */ 0x0, /* 0 */
0x0, /*
0 */
/* 370 */ 0x0, /* 0 */
0x46, /*
70 */
/* 372 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 374 */ NdrFcShort( 0x2 ), /* Offset= 2 (376) */
/* 376 */
0x12, 0x0, /*
FC_UP */
/* 378 */ NdrFcShort( 0x1e4 ), /* Offset=
484 (862) */
/* 380 */
0x2a, /*
FC_ENCAPSULATED_UNION */
0x89, /*
137 */
/* 382 */ NdrFcShort( 0x20 ), /* 32 */
/* 384 */ NdrFcShort( 0xa ), /* 10 */
/* 386 */ NdrFcLong( 0x8 ), /* 8 */
/* 390 */ NdrFcShort( 0x50 ), /* Offset= 80 (470) */
/* 392 */ NdrFcLong( 0xd ), /* 13 */
/* 396 */ NdrFcShort( 0x70 ), /* Offset= 112 (508) */
/* 398 */ NdrFcLong( 0x9 ), /* 9 */
/* 402 */ NdrFcShort( 0x90 ), /* Offset= 144 (546) */
/* 404 */ NdrFcLong( 0xc ), /* 12 */
/* 408 */ NdrFcShort( 0xb0 ), /* Offset= 176 (584) */
/* 410 */ NdrFcLong( 0x24 ), /* 36 */
/* 414 */ NdrFcShort( 0x102 ), /* Offset=
258 (672) */
/* 416 */ NdrFcLong( 0x800d ), /* 32781 */
/* 420 */ NdrFcShort( 0x11e ), /* Offset=
286 (706) */
/* 422 */ NdrFcLong( 0x10 ), /* 16 */
/* 426 */ NdrFcShort( 0x138 ), /* Offset=
312 (738) */
/* 428 */ NdrFcLong( 0x2 ), /* 2 */
/* 432 */ NdrFcShort( 0x14e ), /* Offset=
334 (766) */
/* 434 */ NdrFcLong( 0x3 ), /* 3 */
/* 438 */ NdrFcShort( 0x164 ), /* Offset=
356 (794) */
/* 440 */ NdrFcLong( 0x14 ), /* 20 */
/* 444 */ NdrFcShort( 0x17a ), /* Offset=
378 (822) */
/* 446 */ NdrFcShort( 0xffffffff ), /* Offset= -1
(445) */
/* 448 */

```

```

0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 454 */ NdrFcShort( 0x0 ), /* 0 */
/* 456 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 458 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 462 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 464 */
0x12, 0x0, /*
FC_UP */
/* 466 */ NdrFcShort( 0xffffffff74 ), /* Offset= -
140 (326) */
/* 468 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 470 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 472 */ NdrFcShort( 0x10 ), /* 16 */
/* 474 */ NdrFcShort( 0x0 ), /* 0 */
/* 476 */ NdrFcShort( 0x6 ), /* Offset= 6 (482) */
/* 478 */ 0x8, /* FC_LONG */
0x40, /*
FC_STRUCTPAD4 */
/* 480 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 482 */
0x11, 0x0, /*
FC_RP */
/* 484 */ NdrFcShort( 0xffffffffdc ), /* Offset= -
36 (448) */
/* 486 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 488 */ NdrFcShort( 0x0 ), /* 0 */
/* 490 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 492 */ NdrFcShort( 0x0 ), /* 0 */
/* 494 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 496 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 500 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 502 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /*
0 */

```

```

/* 504 */ NdrFcShort( 0xffffffff58 ), /* Offset= -
168 (336) */
/* 506 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 508 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 510 */ NdrFcShort( 0x10 ), /* 16 */
/* 512 */ NdrFcShort( 0x0 ), /* 0 */
/* 514 */ NdrFcShort( 0x6 ), /* Offset= 6 (520) */
/* 516 */ 0x8, /* FC_LONG */
0x40, /*
FC_STRUCTPAD4 */
/* 518 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 520 */
0x11, 0x0, /*
FC_RP */
/* 522 */ NdrFcShort( 0xffffffffdc ), /* Offset= -
36 (486) */
/* 524 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 530 */ NdrFcShort( 0x0 ), /* 0 */
/* 532 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 534 */ NdrFcLong( 0xffffffffff ), /* -1 */
/* 538 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 540 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 542 */ NdrFcShort( 0xffffffff44 ), /* Offset= -
188 (354) */
/* 544 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 546 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 548 */ NdrFcShort( 0x10 ), /* 16 */
/* 550 */ NdrFcShort( 0x0 ), /* 0 */
/* 552 */ NdrFcShort( 0x6 ), /* Offset= 6 (558) */
/* 554 */ 0x8, /* FC_LONG */
0x40, /*
FC_STRUCTPAD4 */
/* 556 */ 0x36, /* FC_POINTER */

```

```

0x5b, /*
FC_END */
/* 558 */
0x11, 0x0, /*
FC_RP */
/* 560 */ NdrFcShort( 0xffffffffdc ), /* Offset= -
36 (524) */
/* 562 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 564 */ NdrFcShort( 0x0 ), /* 0 */
/* 566 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 568 */ NdrFcShort( 0x0 ), /* 0 */
/* 570 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 572 */ NdrFcLong( 0xffffffffff ), /* -1 */
/* 576 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 578 */
0x12, 0x0, /*
FC_UP */
/* 580 */ NdrFcShort( 0x176 ), /* Offset=
374 (954) */
/* 582 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 584 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 586 */ NdrFcShort( 0x10 ), /* 16 */
/* 588 */ NdrFcShort( 0x0 ), /* 0 */
/* 590 */ NdrFcShort( 0x6 ), /* Offset= 6 (596) */
/* 592 */ 0x8, /* FC_LONG */
0x40, /*
FC_STRUCTPAD4 */
/* 594 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 596 */
0x11, 0x0, /*
FC_RP */
/* 598 */ NdrFcShort( 0xffffffffdc ), /* Offset= -
36 (562) */
/* 600 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 602 */ NdrFcLong( 0x2f ), /* 47 */
/* 606 */ NdrFcShort( 0x0 ), /* 0 */
/* 608 */ NdrFcShort( 0x0 ), /* 0 */
/* 610 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 612 */ 0x0, /* 0 */

```

```

0x0, /*
0 */
/* 614 */ 0x0, /* 0 */
0x0, /*
0 */
/* 616 */ 0x0, /* 0 */
0x46, /*
70 */
/* 618 */
0x1b, /*
FC_CARRAY */
0x0, /*
0 */
/* 620 */ NdrFcShort( 0x1 ), /* 1 */
/* 622 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 624 */ NdrFcShort( 0x4 ), /* 4 */
/* 626 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 628 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END */
/* 630 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 632 */ NdrFcShort( 0x18 ), /* 24 */
/* 634 */ NdrFcShort( 0x0 ), /* 0 */
/* 636 */ NdrFcShort( 0xa ), /* Offset= 10 (646) */
/* 638 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 640 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 642 */ NdrFcShort( 0xffffffffd6 ), /* Offset= -
42 (600) */
/* 644 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 646 */
0x12, 0x0, /*
FC_UP */
/* 648 */ NdrFcShort( 0xffffffffe2 ), /* Offset= -
30 (618) */
/* 650 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 652 */ NdrFcShort( 0x0 ), /* 0 */
/* 654 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 656 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 658 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 660 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 664 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 666 */
                                0x12, 0x0, /*
FC_UP */
/* 668 */ NdrFcShort( 0xffffffffda ), /* Offset= -
38 (630) */
/* 670 */ 0x5c, /* FC_PAD */
                                0x5b, /*
FC_END */
/* 672 */
                                0x1a, /*
FC_BOGUS_STRUCT */
                                0x3, /*
3 */
/* 674 */ NdrFcShort( 0x10 ), /* 16 */
/* 676 */ NdrFcShort( 0x0 ), /* 0 */
/* 678 */ NdrFcShort( 0x6 ), /* Offset= 6 (684) */
/* 680 */ 0x8, /* FC_LONG */
                                0x40, /*
FC_STRUCTPAD4 */
/* 682 */ 0x36, /* FC_POINTER */
                                0x5b, /*
FC_END */
/* 684 */
                                0x11, 0x0, /*
FC_RP */
/* 686 */ NdrFcShort( 0xffffffffdc ), /* Offset= -
36 (650) */
/* 688 */
                                0x1d, /*
FC_SMFARRAY */
                                0x0, /*
0 */
/* 690 */ NdrFcShort( 0x8 ), /* 8 */
/* 692 */ 0x1, /* FC_BYTE */
                                0x5b, /*
FC_END */
/* 694 */
                                0x15, /*
FC_STRUCT */
                                0x3, /*
3 */
/* 696 */ NdrFcShort( 0x10 ), /* 16 */
/* 698 */ 0x8, /* FC_LONG */
                                0x6, /*
FC_SHORT */
/* 700 */ 0x6, /* FC_SHORT */
                                0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 702 */ 0x0, /* 0 */
                                NdrFcShort( 0xfffffffff1
), /* Offset= -15 (688) */
                                0x5b, /*
FC_END */
/* 706 */
                                0x1a, /*
FC_BOGUS_STRUCT */

```

```

                                0x3, /*
3 */
/* 708 */ NdrFcShort( 0x20 ), /* 32 */
/* 710 */ NdrFcShort( 0x0 ), /* 0 */
/* 712 */ NdrFcShort( 0xa ), /* Offset= 10 (722) */
/* 714 */ 0x8, /* FC_LONG */
                                0x40, /*
FC_STRUCTPAD4 */
/* 716 */ 0x36, /* FC_POINTER */
                                0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 718 */ 0x0, /* 0 */
                                NdrFcShort( 0xfffffffff7
), /* Offset= -25 (694) */
                                0x5b, /*
FC_END */
/* 722 */
                                0x11, 0x0, /*
FC_RP */
/* 724 */ NdrFcShort( 0xffffffff12 ), /* Offset= -
238 (486) */
/* 726 */
                                0x1b, /*
FC_CARRAY */
                                0x0, /*
0 */
/* 728 */ NdrFcShort( 0x1 ), /* 1 */
/* 730 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
                                0x0, /*
*/
/* 732 */ NdrFcShort( 0x0 ), /* 0 */
/* 734 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 736 */ 0x1, /* FC_BYTE */
                                0x5b, /*
FC_END */
/* 738 */
                                0x1a, /*
FC_BOGUS_STRUCT */
                                0x3, /*
3 */
/* 740 */ NdrFcShort( 0x10 ), /* 16 */
/* 742 */ NdrFcShort( 0x0 ), /* 0 */
/* 744 */ NdrFcShort( 0x6 ), /* Offset= 6 (750) */
/* 746 */ 0x8, /* FC_LONG */
                                0x40, /*
FC_STRUCTPAD4 */
/* 748 */ 0x36, /* FC_POINTER */
                                0x5b, /*
FC_END */
/* 750 */
                                0x12, 0x0, /*
FC_UP */
/* 752 */ NdrFcShort( 0xffffffffe6 ), /* Offset= -
26 (726) */
/* 754 */
                                0x1b, /*
FC_CARRAY */
                                0x1, /*
1 */

```

```

/* 756 */ NdrFcShort( 0x2 ), /* 2 */
/* 758 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
                                0x0, /*
*/
/* 760 */ NdrFcShort( 0x0 ), /* 0 */
/* 762 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 764 */ 0x6, /* FC_SHORT */
                                0x5b, /*
FC_END */
/* 766 */
                                0x1a, /*
FC_BOGUS_STRUCT */
                                0x3, /*
3 */
/* 768 */ NdrFcShort( 0x10 ), /* 16 */
/* 770 */ NdrFcShort( 0x0 ), /* 0 */
/* 772 */ NdrFcShort( 0x6 ), /* Offset= 6 (778) */
/* 774 */ 0x8, /* FC_LONG */
                                0x40, /*
FC_STRUCTPAD4 */
/* 776 */ 0x36, /* FC_POINTER */
                                0x5b, /*
FC_END */
/* 778 */
                                0x12, 0x0, /*
FC_UP */
/* 780 */ NdrFcShort( 0xffffffffe6 ), /* Offset= -
26 (754) */
/* 782 */
                                0x1b, /*
FC_CARRAY */
                                0x3, /*
3 */
/* 784 */ NdrFcShort( 0x4 ), /* 4 */
/* 786 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
                                0x0, /*
*/
/* 788 */ NdrFcShort( 0x0 ), /* 0 */
/* 790 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 792 */ 0x8, /* FC_LONG */
                                0x5b, /*
FC_END */
/* 794 */
                                0x1a, /*
FC_BOGUS_STRUCT */
                                0x3, /*
3 */
/* 796 */ NdrFcShort( 0x10 ), /* 16 */
/* 798 */ NdrFcShort( 0x0 ), /* 0 */
/* 800 */ NdrFcShort( 0x6 ), /* Offset= 6 (806) */
/* 802 */ 0x8, /* FC_LONG */
                                0x40, /*
FC_STRUCTPAD4 */
/* 804 */ 0x36, /* FC_POINTER */
                                0x5b, /*
FC_END */
/* 806 */

```

```

0x12, 0x0, /*
FC_UP */
/* 808 */ NdrFcShort( 0xffffffe6 ), /* Offset= -
26 (782) */
/* 810 */
0x1b, /*
FC_CARRAY */
0x7, /*
7 */
/* 812 */ NdrFcShort( 0x8 ), /* 8 */
/* 814 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 816 */ NdrFcShort( 0x0 ), /* 0 */
/* 818 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 820 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 822 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 824 */ NdrFcShort( 0x10 ), /* 16 */
/* 826 */ NdrFcShort( 0x0 ), /* 0 */
/* 828 */ NdrFcShort( 0x6 ), /* Offset= 6 (834) */
/* 830 */ 0x8, /* FC_LONG */
0x40, /*
FC_STRUCTPAD4 */
/* 832 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 834 */
0x12, 0x0, /*
FC_UP */
/* 836 */ NdrFcShort( 0xffffffe6 ), /* Offset= -
26 (810) */
/* 838 */
0x15, /*
FC_STRUCT */
0x3, /*
3 */
/* 840 */ NdrFcShort( 0x8 ), /* 8 */
/* 842 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 844 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 846 */
0x1b, /*
FC_CARRAY */
0x3, /*
3 */
/* 848 */ NdrFcShort( 0x8 ), /* 8 */
/* 850 */ 0x7, /* Corr desc: FC_USHORT
*/
0x0, /*
*/

```

```

/* 852 */ NdrFcShort( 0xffc8 ), /* -56 */
/* 854 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 856 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
*/
/* 858 */ NdrFcShort( 0xfffffec ), /* Offset= -
20 (838) */
/* 860 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 862 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 864 */ NdrFcShort( 0x38 ), /* 56 */
/* 866 */ NdrFcShort( 0xfffffec ), /* Offset= -
20 (846) */
/* 868 */ NdrFcShort( 0x0 ), /* Offset= 0 (868) */
/* 870 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 872 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 874 */ 0x40, /* FC_STRUCTPAD4 */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 876 */ 0x0, /* 0 */
NdrFcShort( 0xffffe0f
), /* Offset= -497 (380) */
0x5b, /*
FC_END */
/* 880 */
0x12, 0x0, /*
FC_UP */
/* 882 */ NdrFcShort( 0xfffff04 ), /* Offset= -
252 (630) */
/* 884 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 886 */ 0x1, /* FC_BYTE */
0x5c, /*
FC_PAD */
/* 888 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 890 */ 0x6, /* FC_SHORT */
0x5c, /*
FC_PAD */
/* 892 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 894 */ 0x8, /* FC_LONG */
0x5c, /*
FC_PAD */
/* 896 */
0x12, 0x8, /*
FC_UP [simple_pointer] */

```

```

/* 898 */ 0xb, /* FC_HYPER */
0x5c, /*
FC_PAD */
/* 900 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 902 */ 0xa, /* FC_FLOAT */
0x5c, /*
FC_PAD */
/* 904 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 906 */ 0xc, /* FC_DOUBLE */
0x5c, /*
FC_PAD */
/* 908 */
0x12, 0x0, /*
FC_UP */
/* 910 */ NdrFcShort( 0xffffda2 ), /* Offset= -
606 (304) */
/* 912 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 914 */ NdrFcShort( 0xffffda4 ), /* Offset= -
604 (310) */
/* 916 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 918 */ NdrFcShort( 0xffffdba ), /* Offset= -
582 (336) */
/* 920 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 922 */ NdrFcShort( 0xffffdc8 ), /* Offset= -
568 (354) */
/* 924 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 926 */ NdrFcShort( 0xffffdd6 ), /* Offset= -
554 (372) */
/* 928 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 930 */ NdrFcShort( 0x2 ), /* Offset= 2 (932) */
/* 932 */
0x12, 0x0, /*
FC_UP */
/* 934 */ NdrFcShort( 0x14 ), /* Offset= 20 (954) */
/* 936 */
0x15, /*
FC_STRUCT */
0x7, /*
7 */
/* 938 */ NdrFcShort( 0x10 ), /* 16 */
/* 940 */ 0x6, /* FC_SHORT */
0x1, /*
FC_BYTE */
/* 942 */ 0x1, /* FC_BYTE */
0x8, /*
FC_LONG */
/* 944 */ 0xb, /* FC_HYPER */

```

```

                                0x5b,          /*
FC_END */
/* 946 */
                                0x12, 0x0,      /*
FC_UP */
/* 948 */ NdrFcShort( 0xffffffff4 ), /* Offset= -
12 (936) */
/* 950 */
                                0x12, 0x8,      /*
FC_UP [simple_pointer] */
/* 952 */ 0x2,          /* FC_CHAR */
                                0x5c,          /*
FC_PAD */
/* 954 */
                                0x1a,          /*
FC_BOGUS_STRUCT */
                                0x7,          /*
7 */
/* 956 */ NdrFcShort( 0x20 ), /* 32 */
/* 958 */ NdrFcShort( 0x0 ), /* 0 */
/* 960 */ NdrFcShort( 0x0 ), /* Offset= 0 (960) */
/* 962 */ 0x8,          /* FC_LONG */
                                0x8,          /*
FC_LONG */
/* 964 */ 0x6,          /* FC_SHORT */
                                0x6,          /*
FC_SHORT */
/* 966 */ 0x6,          /* FC_SHORT */
                                0x6,          /*
FC_SHORT */
/* 968 */ 0x4c,        /* FC_EMBEDDED_COMPLEX
*/
                                0x0,          /*
0 */
/* 970 */ NdrFcShort( 0xfffffc3c ), /* Offset= -
964 (6) */
/* 972 */ 0x5c,          /* FC_PAD */
                                0x5b,          /*
FC_END */
/* 974 */ 0xb4,        /* FC_USER_MARSHAL */
                                0x83,          /*
131 */
/* 976 */ NdrFcShort( 0x0 ), /* 0 */
/* 978 */ NdrFcShort( 0x18 ), /* 24 */
/* 980 */ NdrFcShort( 0x0 ), /* 0 */
/* 982 */ NdrFcShort( 0xfffffc2c ), /* Offset= -
980 (2) */
/* 984 */
                                0x11, 0x4,      /*
FC_RP [allocated_on_stack] */
/* 986 */ NdrFcShort( 0x6 ), /* Offset= 6 (992) */
/* 988 */
                                0x13, 0x0,      /*
FC_OP */
/* 990 */ NdrFcShort( 0xfffffcdc ), /* Offset= -
36 (954) */
/* 992 */ 0xb4,          /* FC_USER_MARSHAL */
                                0x83,          /*
131 */
/* 994 */ NdrFcShort( 0x0 ), /* 0 */
/* 996 */ NdrFcShort( 0x18 ), /* 24 */

```

```

/* 998 */ NdrFcShort( 0x0 ), /* 0 */
/* 1000 */ NdrFcShort( 0xffffffff4 ), /*
Offset= -12 (988) */
                                }
                                };
static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        VARIANT_UserSize
        ,VARIANT_UserMarshal
        ,VARIANT_UserUnmarshal
        ,VARIANT_UserFree
    }
};

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000,
ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xc0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEED6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,0xBF,0xE0,0x8B}} */

#pragma code_seg(".orpc")
static const unsigned short
ITPCC_FormatStringOffsetTable[] =
{
    0,
    44,
    88,
    132,
    176,
    220
};

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo
=
{
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,

```

```

0
};

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
{
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0;
};
CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *) (INT_PTR) -1 /* ITPCC::NewOrder */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Payment */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Delivery */ ,
    (void *) (INT_PTR) -1 /* ITPCC::StockLevel */ ,
    (void *) (INT_PTR) -1 /* ITPCC::OrderStatus */ ,
    (void *) (INT_PTR) -1 /* ITPCC::CallSetComplete
*/
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x60015b, /* MIDL Version 6.0.347 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0, /* proxy/server info */
    0 /* Reserved5 */

```



```

};

const CInterfaceProxyVtbl *
_tpcc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl *
_tpcc_com_ps_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const
_tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};

#define _tpcc_com_ps_CHECK_IID(n)
IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID,
n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID *
pIID, int * pIndex )
{
    if(! _tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo
=
{
    (PCInterfaceProxyVtblList *) &
_tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) &
_tpcc_com_ps_StubVtblList,
    (const PCInterfaceName *) &
_tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    & _tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};

```

```

#endif /* defined(_M_IA64) || defined(_M_AMD64) */

tpcc_com_sl.rgs
HKCR
{
    TPCC.StockLevel.1 = s 'StockLevel Class'
    {
        CLSID = s '{2668369E-A50D-11D2-
BA4E-00C04FBFE08B}'
    }
    TPCC.StockLevel = s 'StockLevel Class'
    {
        CurVer = s 'TPCC.StockLevel.1'
    }
    NoRemove CLSID
    {
        ForceRemove {2668369E-A50D-11D2-
BA4E-00C04FBFE08B} = s 'StockLevel Class'
    {
        ProgID = s
'TPCC.StockLevel.1'
        VersionIndependentProgID = s
'TPCC.StockLevel'
        InprocServer32 = s
'&MODULE%'
        {
            val
ThreadingModel = s 'Both'
        }
    }
}

tpcc_dblib.cpp
/* FILE: TPCC_DBLIB.CPP
* Microsoft
* TPC-C Kit Ver. 4.42.000
* Copyright
* Microsoft, 2002
* All Rights Reserved
*
* Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
*
* PURPOSE: Implements dblib calls for TPC-C
txns.
* Contact: Charles Levine
(clevine@microsoft.com)
*
* Change history:
* 4.42.000 - changed w_id fields
from short to long to support >32K warehouses
* 4.20.000 - updated rev number to
match kit

```

```

* 4.10.001 - not deleting error
class in catch handler on deadlock retry;
* not a
functional bug, but a memory leak
* - had to
tweak some declarations to compile with latest SDK;
no functional change
*/

#include <windows.h>
#include <stdio.h>
#include <assert.h>

#define DBNTWIN32
#include <sqlfront.h>
#include <sqldb.h>

#ifdef ICECAP
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_dblib.h"

#define DEFCLPCKSIZE
4096

// version string; must match return value from
tpcc_version stored proc
const char sVersion[] = "4.20.000";

const iMaxRetries = 10;
// how many retries on deadlock
static long iConnectionCount = 0; // number
of current dblib connections

const int iErrOleDbProvider = 7312;
const char sErrTimeoutExpired[] = "Timeout expired";

BOOL WINAPI DllMain(HMODULE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
            DisableThreadLibraryCalls(hModule);
            dbinit(); //
            initialize dblib
            break;
        case DLL_PROCESS_DETACH:
            dbexit(); //
            close all dblib structures/connections
            break;
    }
}

```

```

        default:
        } /* nothing */;
    }
    return TRUE;
}

int err_handler(DBPROCESS *dbproc, int severity, int
dberr, int oserr, LPCSTR dberrstr, LPCSTR oserrstr)
{
    CTPCC_DBLIB
    *pConn;

    assert(dbproc != NULL);
    pConn =
(CTPCC_DBLIB*)dbgetuserdata(dbproc);

    if (pConn != NULL)
    {
        pConn->SetDbLibError( severity,
dberr, oserr, dberrstr, oserrstr );
    }
    return INT_CANCEL;
}

/* FUNCTION: int msg_handler(DBPROCESS *dbproc, DBINT
msgno, int msgstate, int severity, char *msgtext)
*
* PURPOSE: This function handles DB-Library
SQL Server error messages
*
* ARGUMENTS: DBPROCESS *dbproc
DBPROCESS id pointer DBINT
message number
message state int
message severity int
message description severity
char *msgtext printable
*
* RETURNS: int
INT_CONTINUE continue if
error is SQLETIME else INT_CANCEL action
*
cancel operation INT_CANCEL
*
* COMMENTS: This function also sets the dead
lock dbproc variable if necessary.
*
*/

// typedef INT (SQLAPI *DBMSGHANDLE_PROC) (DBPROCESS,
DBINT, INT, INT, LPCSTR, LPCSTR, LPCSTR,
DBUSMALLINT);

```

```

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int
msgstate, int severity,
LPCSTR
msgtext, LPCSTR srvname, LPCSTR procname, DBUSMALLINT
line)
{
    CTPCC_DBLIB
    *pConn;

    assert(dbproc != NULL);
    pConn =
(CTPCC_DBLIB*)dbgetuserdata(dbproc);

    if (pConn != NULL)
    {
        pConn->SetSqlError( msgno,
msgstate, severity, msgtext );
    }
    return 0;
}

/* FUNCTION: void UtilStrCpy(char * pDest, char *
pSrc, int n)
*
* PURPOSE: This function copies n characters
from string pSrc to pDest and places a
* null character at the
end of the destination string.
*
* ARGUMENTS: char
*pDest destination string pointer
char
*pSrc source string pointer
int
n
number of characters to copy
*
* RETURNS: None
*
* COMMENTS: Unlike strncpy this function
ensures that the result string is
always null
terminated.
*/

inline static void UtilStrCpy(char * pDest, const
BYTE * pSrc, int n)
{
    strncpy(pDest, (char *)pSrc, n);
    pDest[n] = '\0';

    return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
*
*/

```

```

char* CTPCC_DBLIB_ERR::ErrorText(void)
{
    int i;

    static SERRORMSG errorMsgs[] =
    {
        { ERR_WRONG_SP_VERSION,
"Wrong version of stored procs on database
server" },
        { ERR_INVALID_CUST,
"Invalid Customer id,name." },
        { ERR_NO_SUCH_ORDER,
"No orders found for customer." },
        { ERR_RETRIED_TRANS,
"Retries before transaction succeeded." },
        { 0, "" }
    };
}

static char szNotFound[] = "Unknown error
number.";

for(i=0; errorMsgs[i].szMsg[0]; i++)
{
    if ( m_errno ==
errorMsgs[i].iError )
        break;
}
if ( !errorMsgs[i].szMsg[0] )
    return szNotFound;
else
    return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_DBLIB* CTPCC_DBLIB_new(
LPCSTR szServer, // name of
SQL server
LPCSTR szUser, //
user name for login
LPCSTR szPassword, // password
for login
LPCSTR szHost, //
workstation name; shows up in sp_who; max 30 chars,
only first 10 kept by SQL Server
LPCSTR szDatabase ) // name of
database to use
{
    return new CTPCC_DBLIB( szServer, szUser,
szPassword, szHost, szDatabase );
}

CTPCC_DBLIB::CTPCC_DBLIB (

```

```

        LPCSTR szServer,          // name of
SQL server
        LPCSTR szUser,          //
user name for login
        LPCSTR szPassword,      // password
for login
        LPCSTR szHost,          //
workstation name; shows up in sp_who; max 30 chars,
only first 10 kept by SQL Server
        LPCSTR szDatabase )     // name of
database to use
{
    LOGINREC *login;
    const BYTE *pData;

    // initialization
    m_dbproc = NULL;
    m_DbLibErr = (CDBLIBERR*)NULL;
    m_SqlErr = (CSQLERR*)NULL;

    m_MaxRetries = 10;          // how many
retries on deadlock

    // increase max number of connections if
getting close
    if ( dbgetmaxprocs() < (iConnectionCount+5)
)
    {
        if (
dbsetmaxprocs(iConnectionCount+10) == FAIL )

            ThrowError(CDBLIBERR::eDbSetMaxProcs);
    }

    // allocate a login structure
    login = dblogin();
    if (login == NULL)
        ThrowError(CDBLIBERR::eLogin);
    InterlockedIncrement( &iConnectionCount );

    // register error and message handler
functions
    if (dbprocerrhandle(login, err_handler) ==
NULL)

        ThrowError(CDBLIBERR::eDbProcHandler);

    if (dbprocmsghandle(login, msg_handler) ==
NULL)

        ThrowError(CDBLIBERR::eDbProcHandler);

    DBSETLUSER(login, szUser);
    DBSETLPWD(login, szPassword);
    DBSETLHOST(login, szHost);
    DBSETLPACKET(login, (unsigned
short)DEFCLPACKSIZE);
    DBSETLVERSION(login, DBVER60);
    // use dblib ver 6.0 client behavior

```

```

        // set time to wait for login
if (dbsetlogintime(60) == FAIL)
    ThrowError(CDBLIBERR::eDbSet);

        // set time to wait for statement execution
if (dbsettime(180) == FAIL)
    ThrowError(CDBLIBERR::eDbSet);

    m_dbproc = dbopen(login, szServer);

    // deallocate login structure before
checking for success
    dbfreelogin( login );

    if (m_dbproc == NULL)
        ThrowError(CDBLIBERR::eDbOpen);

    // save address of class instance so that
the message and error handler
    // can get to data.
    dbsetuserdata(m_dbproc, (LPVOID)this);

    // Use the the right database
if (dbuse(m_dbproc, szDatabase) == FAIL)
    ThrowError(CDBLIBERR::eDbUse);

    dbcmd(m_dbproc, "set nocount on ");
    // do not return row counts
    dbcmd(m_dbproc, "set XACT_ABORT ON");
    // rollback transaction on abort

    if (dbsqlexec(m_dbproc) == FAIL)
        ThrowError(CDBLIBERR::eDbSqlExec);
    DiscardNextResults(2);

    // verify that version of stored procs on
server is correct
    dbrpcinit(m_dbproc, "tpcc_version", 0);

    if (dbrpcexec(m_dbproc) == FAIL)
        ThrowError(CDBLIBERR::eDbRpcExec);

    if (dbresults(m_dbproc) != SUCCEED)
        ThrowError(CDBLIBERR::eDbResults);

    if (dbnextrow(m_dbproc) != REG_ROW)
        ThrowError(CDBLIBERR::eDbNextRow);

    char szSrvVersion[16];
    pData=dbdata(m_dbproc, 1);
    if (pData)
        UtilStrCpy(szSrvVersion, pData,
dbdatlen(m_dbproc, 1));
    else
        szSrvVersion[0]=0;
    if (strcmp(szSrvVersion,sVersion)

```

```

        throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_WRONG_SP_VERSION );

        DiscardNextRows(0);
        DiscardNextResults(0);
    }

CTPCC_DBLIB::~CTPCC_DBLIB( void )
{
    // close db connection and deallocate
resources
    dbclose(m_dbproc);
    InterlockedDecrement( &iConnectionCount );
    if (m_DbLibErr != NULL)
        delete m_DbLibErr;
    if (m_SqlErr != NULL)
        delete m_SqlErr;
}

void CTPCC_DBLIB::SetDbLibError(int severity, int
dberr, int oserr, LPCSTR dberrstr, LPCSTR oserrstr)
{
    delete m_DbLibErr;
    m_DbLibErr = new
CDBLIBERR(CDBLIBERR::eUnknown, severity, dberr,
oserr);

    if (dberrstr != NULL)
    {
        m_DbLibErr->m_dberrstr = new
char[ strlen(dberrstr)+1 ];
        strcpy( m_DbLibErr->m_dberrstr,
dberrstr );
    }

    if (oserrstr != NULL)
    {
        m_DbLibErr->m_oserrstr = new
char[ strlen(oserrstr)+1 ];
        strcpy( m_DbLibErr->m_oserrstr,
oserrstr );
    }
}

void CTPCC_DBLIB::SetSqlError( int /*DBINT*/ msgno,
int msgstate, int severity, LPCSTR msgtext )
{
    if (m_SqlErr == NULL)
        m_SqlErr = new CSQLERR();

    m_SqlErr->m_msgno = msgno;
    m_SqlErr->m_msgstate = msgstate;
    m_SqlErr->m_severity = severity;

    delete [] m_SqlErr->m_msgtext;
    if (msgtext != NULL)
    {

```

```

        m_SqlErr->m_msgtext = new char[
strlen(msgtext)+1 ];
        strcpy( m_SqlErr->m_msgtext,
msgtext );
    }
}

void CTPCC_DBLIB::ThrowError( CDBLIBERR::ACTION
eAction )
{
    // discard anything still in return buffer
    DiscardNextRows(-1);
    DiscardNextResults(-1);

    // check for SQL Server error first; if
yes, throw it and ignore any DBLib error.
    if (m_SqlErr != NULL)
    {
        CSQLERR          *pSqlErr;
        pSqlErr = m_SqlErr;
        m_SqlErr = NULL; // clear our
pointer to instance; catch handler will delete
        throw pSqlErr;
    }

    CDBLIBERR *pDbLibErr;
    if (m_DbLibErr == NULL)
    // this case isn't expected to
happen, since it means that an error was returned
not called.
    // but the error handlers were
        pDbLibErr = new
CDBLIBERR(eAction);
    else
    {
        pDbLibErr = m_DbLibErr;
        pDbLibErr->m_eAction = eAction;
        m_DbLibErr = NULL; //
clear our pointer to instance; catch handler will
delete
    }

    throw pDbLibErr;
}

// Read and discard rows until no more. Throw an
exception if number of rows read doesn't
// match number of rows expected. The row count will
be ignored if the expected count value
// passed in is negative. A typical use of this
routine is to verify that there are no more
// rows to be read.
void CTPCC_DBLIB::DiscardNextRows(int iExpectedCount)
{
    int          iRowsRead = 0;
    RETCODE     rc;

    while (TRUE)
    {
        rc = dbnextrow(m_dbproc);
        if (rc == NO_MORE_ROWS)

```

```

            break;
        if (rc == FAIL)
        {
            if (iExpectedCount >=
0)
                ThrowError(CDBLIBERR::eDbNextRow);
            else
                break;
        }
        iRowsRead++;
    }

    if ((iExpectedCount >= 0) &&
(iExpectedCount != iRowsRead))
        ThrowError(CDBLIBERR::eWrongRowCount);
}

// Read and discard results until no more. Throw an
exception if number of result sets read doesn't
// match number expected. The result set count will
be ignored if the expected count value
// passed in is negative. A typical use of this
routine is to verify that there are no more
// result sets to be read.
void CTPCC_DBLIB::DiscardNextResults(int
iExpectedCount)
{
    int          iResultsRead = 0;
    RETCODE     rc;

    while (TRUE)
    {
        rc = dbresults(m_dbproc);
        if (rc == NO_MORE_RESULTS)
            break;
        if (rc == FAIL)
        {
            if (iExpectedCount >=
0)
                ThrowError(CDBLIBERR::eDbResults);
            else
                break;
        }
        DiscardNextRows(-1);
        iResultsRead++;
    }

    if ((iExpectedCount >= 0) &&
(iExpectedCount != iResultsRead))
        ThrowError(CDBLIBERR::eWrongRowCount);
}

void CTPCC_DBLIB::StockLevel()
{
    int          iTryCount =

```

```

        const BYTE          *pData;

        ResetError();

        while (TRUE)
        {
            try
            {
                dbrpcinit(m_dbproc,
"tpcc_stocklevel", 0);

                dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.StockLevel.w_id); // @w_id int
                dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.StockLevel.d_id); // @d_id
                tinyint
                dbrpcparam(m_dbproc,
NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.StockLevel.threshold); // @threshold
                smallint

                if (dbrpcexec(m_dbproc)
== FAIL)
                    ThrowError(CDBLIBERR::eDbRpcExec);

                if (dbresults(m_dbproc)
!= SUCCEEDED)
                    ThrowError(CDBLIBERR::eDbResults);

                if (dbnextrow(m_dbproc)
!= REG_ROW)
                    ThrowError(CDBLIBERR::eDbNextRow);

                if
(pData=dbdata(m_dbproc, 1))
                    m_txn.StockLevel.low_stock = *((long *)
pData);

                DiscardNextRows(0);
                DiscardNextResults(0);

                m_txn.StockLevel.exec_status_code = eOK;
                return;
            }
            catch (CSQLERR *e)
            {
                if ((e->m_msgno == 1205
||
== iErrOleDbProvider &&
(e->m_msgno
>m_msgtext, sErrTimeoutExpired) != NULL) &&
(++iTryCount
<= iMaxRetries))

```

```

        {
            // hit
            deadlock; backoff for increasingly longer period
            delete e;
            Sleep(10 *
iTryCount);
        }
        else
            throw;
    }
    // while (TRUE)
    //if (iTryCount)
    //    throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_DBLIB::NewOrder()
{
    int                i;
    DBINT              commit_flag;
    DBDATETIME         datetime;
    DBDATEREC          daterec;

    int                iTryCount =
0;
    const BYTE         *pData;

    ResetError();

    while (TRUE)
    {
        try
        {
            dbrpcinit(m_dbproc,
"tpcc_neworder", 0);

            dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.w_id);
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.d_id);
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.c_id);
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.o_ol_cnt);

            // check whether any
order lines are for a remote warehouse
            m_txn.NewOrder.o_all_local = 1;
            for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; i++)
            {

```

```

                if
(m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
                {
                    m_txn.NewOrder.o_all_local = 0; // at
least one remote warehouse
                    break;
                }
            }
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.o_all_local);

            for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; i++)
            {
                dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1,
-1, (BYTE *) &m_txn.NewOrder.OL[i].ol_i_id);
                dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1,
-1, (BYTE *) &m_txn.NewOrder.OL[i].ol_supply_w_id);
                dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1,
-1, (BYTE *) &m_txn.NewOrder.OL[i].ol_quantity);
            }

            if (dbrpcexec(m_dbproc)
== FAIL)
                ThrowError(CDBLIBERR::eDbRpcExec);

            // Get order line
            results
                m_txn.NewOrder.total_amount = 0;
                for (i = 0;
i<m_txn.NewOrder.o_ol_cnt; i++)
                {
                    if
(dbresults(m_dbproc) != SUCCEED)
                        ThrowError(CDBLIBERR::eDbResults);

                    if
(dbnumcols(m_dbproc) != 5)
                        ThrowError(CDBLIBERR::eWrongNumCols);

                    if
(dbnextrow(m_dbproc) != REG_ROW)
                        ThrowError(CDBLIBERR::eDbNextRow);

                    if (pData=dbdata(m_dbproc, 1))

```

```

                UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name,
pData, dbdatlen(m_dbproc, 1));
                if (pData=dbdata(m_dbproc, 2))
                    m_txn.NewOrder.OL[i].ol_stock =
(* (DBSMALLINT *) pData);
                if (pData=dbdata(m_dbproc, 3))
                    UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_ge
neric, pData, dbdatlen(m_dbproc, 3));
                if (pData=dbdata(m_dbproc, 4))
                    dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc, 4),
SQLFLT8, (BYTE
*)&m_txn.NewOrder.OL[i].ol_i_price, 8);
                if (pData=dbdata(m_dbproc, 5))
                    dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc, 5),
SQLFLT8, (BYTE
*)&m_txn.NewOrder.OL[i].ol_amount, 8);

                m_txn.NewOrder.total_amount =
m_txn.NewOrder.total_amount +
m_txn.NewOrder.OL[i].ol_amount;

                DiscardNextRows(0);
            }

            // get remaining values
for w_tax, d_tax, o_id, c_last, c_discount, c_credit,
o_entry_d, commit_flag
            if (dbresults(m_dbproc)
!= SUCCEED)
                ThrowError(CDBLIBERR::eDbResults);

            if (dbnextrow(m_dbproc)
!= REG_ROW)
                ThrowError(CDBLIBERR::eDbNextRow);

            if (dbnumcols(m_dbproc)
!= 8)
                ThrowError(CDBLIBERR::eWrongNumCols);

```

```

    if
    (pData=dbdata(m_dbproc, 1))

        dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,1), SQLFLT8, (BYTE
*)&m_txn.NewOrder.w_tax, 8);
    if
    (pData=dbdata(m_dbproc, 2))

        dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,2), SQLFLT8, (BYTE
*)&m_txn.NewOrder.d_tax, 8);
    if
    (pData=dbdata(m_dbproc, 3))
        m_txn.NewOrder.o_id = (*(DBINT *) pData);
    if
    (pData=dbdata(m_dbproc, 4))
        UtilStrCpy(m_txn.NewOrder.c_last, pData,
dbdatlen(m_dbproc, 4));
    if
    (pData=dbdata(m_dbproc, 5))
        dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,5), SQLFLT8, (BYTE
*)&m_txn.NewOrder.c_discount, 8);
    if
    (pData=dbdata(m_dbproc, 6))
        UtilStrCpy(m_txn.NewOrder.c_credit, pData,
dbdatlen(m_dbproc, 6));
    if
    {
        datetime =
        (*(DBDATETIME *) pData);
        dbdatecrack(m_dbproc, &daterec, &datetime);
        m_txn.NewOrder.o_entry_d.year =
daterec.year;
        m_txn.NewOrder.o_entry_d.month =
daterec.month;
        m_txn.NewOrder.o_entry_d.day =
daterec.day;
        m_txn.NewOrder.o_entry_d.hour =
daterec.hour;
        m_txn.NewOrder.o_entry_d.minute =
daterec.minute;
        m_txn.NewOrder.o_entry_d.second =
daterec.second;

```

```

    }
    if
    (pData=dbdata(m_dbproc, 8))
        commit_flag =
        (*(DBTINYINT *) pData);
        DiscardNextRows(0);
        DiscardNextResults(0);
    if (commit_flag == 1)
    {
        m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));
        m_txn.NewOrder.exec_status_code = eOK;
    }
    else
        m_txn.NewOrder.exec_status_code =
eInvalidItem;
        return;
    }
    catch (CSQLERR *e)
    {
        if ((e->m_msgno == 1205
        ||
        == iErrOleDbProvider &&
        strstr(e-
        >m_msgtext, sErrTimeoutExpired) != NULL) &&
        (++iTryCount
        <= iMaxRetries))
        {
            // hit
            deadlock; backoff for increasingly longer period
            delete e;
            Sleep(10 *
            iTryCount);
        }
        else
            throw;
    }
    } while (TRUE)
    //
    // if (iTryCount)
    //     throw new
    CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
    iTryCount);
}

void CTPCC_DBLIB::Payment()
{
    DBDATETIME datetime;
    DBDATEREC daterec;

```

```

    int
    iTryCount =
    0;
    const BYTE
    *pData;
    ResetError();
    while (TRUE)
    {
        try
        {
            dbrpcinit(m_dbproc,
            "tpcc_payment", 0);
            dbrpcparam(m_dbproc,
            NULL, 0, SQLINT4, -1, -1, (BYTE *)
            &m_txn.Payment.w_id);
            dbrpcparam(m_dbproc,
            NULL, 0, SQLINT4, -1, -1, (BYTE *)
            &m_txn.Payment.c_w_id);
            dbrpcparam(m_dbproc,
            NULL, 0, SQLFLT8, -1, -1, (BYTE *)
            &m_txn.Payment.h_amount);
            dbrpcparam(m_dbproc,
            NULL, 0, SQLINT1, -1, -1, (BYTE *)
            &m_txn.Payment.d_id);
            dbrpcparam(m_dbproc,
            NULL, 0, SQLINT1, -1, -1, (BYTE *)
            &m_txn.Payment.c_d_id);
            dbrpcparam(m_dbproc,
            NULL, 0, SQLINT4, -1, -1, (BYTE *)
            &m_txn.Payment.c_id);
            // if customer id is
            zero, then payment is by name
            if (m_txn.Payment.c_id
            == 0)
                dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
            strlen(m_txn.Payment.c_last), (unsigned char
            *)m_txn.Payment.c_last);
            if (dbrpcexec(m_dbproc)
            == FAIL)
                ThrowError(CDBLIBERR::eDbRpcExec);
            if (dbresults(m_dbproc)
            != SUCCEEDED)
                ThrowError(CDBLIBERR::eDbResults);
            if (dbnextrow(m_dbproc)
            != REG_ROW)
                ThrowError(CDBLIBERR::eDbNextRow);
            if (dbnumcols(m_dbproc)
            != 27)
                ThrowError(CDBLIBERR::eWrongNumCols);

```

```

        if
(pData=dbdata(m_dbproc, 1))
            m_txn.Payment.c_id = *((DBINT *) pData);
        if
(pData=dbdata(m_dbproc, 2))
            UtilStrCpy(m_txn.Payment.c_last, pData,
dbdatlen(m_dbproc, 2));
        if
(pData=dbdata(m_dbproc, 3))
        {
            datetime =
*((DBDATETIME *) pData);
            dbdatecrack(m_dbproc, &daterec, &datetime);
            m_txn.Payment.h_date.year = daterec.year;
            m_txn.Payment.h_date.month =
daterec.month;
            m_txn.Payment.h_date.day = daterec.day;
            m_txn.Payment.h_date.hour = daterec.hour;
            m_txn.Payment.h_date.minute =
daterec.minute;
            m_txn.Payment.h_date.second =
daterec.second;
        }
        if
(pData=dbdata(m_dbproc, 4))
            UtilStrCpy(m_txn.Payment.w_street_1, pData,
dbdatlen(m_dbproc, 4));
        if
(pData=dbdata(m_dbproc, 5))
            UtilStrCpy(m_txn.Payment.w_street_2, pData,
dbdatlen(m_dbproc, 5));
        if
(pData=dbdata(m_dbproc, 6))
            UtilStrCpy(m_txn.Payment.w_city, pData,
dbdatlen(m_dbproc, 6));
        if
(pData=dbdata(m_dbproc, 7))
            UtilStrCpy(m_txn.Payment.w_state, pData,
dbdatlen(m_dbproc, 7));
        if
(pData=dbdata(m_dbproc, 8))
            UtilStrCpy(m_txn.Payment.w_zip, pData,
dbdatlen(m_dbproc, 8));
        if
(pData=dbdata(m_dbproc, 9))

```

```

            UtilStrCpy(m_txn.Payment.d_street_1, pData,
dbdatlen(m_dbproc, 9));
        if
(pData=dbdata(m_dbproc, 10))
            UtilStrCpy(m_txn.Payment.d_street_2, pData,
dbdatlen(m_dbproc, 10));
        if
(pData=dbdata(m_dbproc, 11))
            UtilStrCpy(m_txn.Payment.d_city, pData,
dbdatlen(m_dbproc, 11));
        if
(pData=dbdata(m_dbproc, 12))
            UtilStrCpy(m_txn.Payment.d_state, pData,
dbdatlen(m_dbproc, 12));
        if
(pData=dbdata(m_dbproc, 13))
            UtilStrCpy(m_txn.Payment.d_zip, pData,
dbdatlen(m_dbproc, 13));
        if
(pData=dbdata(m_dbproc, 14))
            UtilStrCpy(m_txn.Payment.c_first, pData,
dbdatlen(m_dbproc, 14));
        if
(pData=dbdata(m_dbproc, 15))
            UtilStrCpy(m_txn.Payment.c_middle, pData,
dbdatlen(m_dbproc, 15));
        if
(pData=dbdata(m_dbproc, 16))
            UtilStrCpy(m_txn.Payment.c_street_1, pData,
dbdatlen(m_dbproc, 16));
        if
(pData=dbdata(m_dbproc, 17))
            UtilStrCpy(m_txn.Payment.c_street_2, pData,
dbdatlen(m_dbproc, 17));
        if
(pData=dbdata(m_dbproc, 18))
            UtilStrCpy(m_txn.Payment.c_city, pData,
dbdatlen(m_dbproc, 18));
        if
(pData=dbdata(m_dbproc, 19))
            UtilStrCpy(m_txn.Payment.c_state, pData,
dbdatlen(m_dbproc, 19));
        if
(pData=dbdata(m_dbproc, 20))
            UtilStrCpy(m_txn.Payment.c_zip, pData,
dbdatlen(m_dbproc, 20));
        if
(pData=dbdata(m_dbproc, 21))

```

```

            UtilStrCpy(m_txn.Payment.c_phone, pData,
dbdatlen(m_dbproc, 21));
        if
(pData=dbdata(m_dbproc, 22))
        {
            datetime =
*((DBDATETIME *) pData);
            dbdatecrack(m_dbproc, &daterec, &datetime);
            m_txn.Payment.c_since.year =
daterec.year;
            m_txn.Payment.c_since.month =
daterec.month;
            m_txn.Payment.c_since.day = daterec.day;
            m_txn.Payment.c_since.hour =
daterec.hour;
            m_txn.Payment.c_since.minute =
daterec.minute;
            m_txn.Payment.c_since.second =
daterec.second;
        }
        if (pData=dbdata(m_dbproc, 23))
            UtilStrCpy(m_txn.Payment.c_credit, pData,
dbdatlen(m_dbproc, 23));
        if (pData=dbdata(m_dbproc, 24))
            dbconvert(m_dbproc, SQLNUMERIC,
(LPBYTE)pData, dbdatlen(m_dbproc,24), SQLFLT8, (BYTE
*)&m_txn.Payment.c_credit_lim, 8);
        if (pData=dbdata(m_dbproc, 25))
            dbconvert(m_dbproc, SQLNUMERIC,
(LPBYTE)pData, dbdatlen(m_dbproc,25), SQLFLT8, (BYTE
*)&m_txn.Payment.c_discount, 8);
        if (pData=dbdata(m_dbproc, 26))
            dbconvert(m_dbproc, SQLNUMERIC,
(LPBYTE)pData, dbdatlen(m_dbproc,26), SQLFLT8, (BYTE
*)&m_txn.Payment.c_balance, 8);
        if (pData=dbdata(m_dbproc, 27))
            UtilStrCpy(m_txn.Payment.c_data, pData,
dbdatlen(m_dbproc, 27));
            DiscardNextRows(0);
            DiscardNextResults(0);

```

```

        if (m_txn.Payment.c_id
== 0)
            throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
            else
                m_txn.Payment.exec_status_code = eOK;
                return;
            }
        catch (CSQLERR *e)
        {
            if ((e->m_msgno == 1205
||
== iErrOleDbProvider &&
>m_msgtext, sErrTimeoutExpired) != NULL) &&
(++iTryCount
<= iMaxRetries))
            {
                // hit
                deadlock; backoff for increasingly longer period
                delete e;
                Sleep(10 *
iTryCount);
            }
            else
                throw;
        }
        // while (TRUE)
        {
            if (iTryCount)
                throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
        }

void CTPCC_DBLIB::OrderStatus()
{
    int i;
    DBDATETIME datetime;
    DBDATEREC daterec;

    int iTryCount =
0;
    RETCODE rc;
    const BYTE *pData;

    ResetError();

    while (TRUE)
    {
        try
        {
            dbrpcinit(m_dbproc,
"tpcc_orderstatus", 0);

```

```

            dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.OrderStatus.w_id);
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.OrderStatus.d_id);
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.OrderStatus.c_id);

            // if customer id is
            zero, then order status is by name
            if
            (m_txn.OrderStatus.c_id == 0)
                dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
strlen(m_txn.OrderStatus.c_last), (unsigned char
*)m_txn.OrderStatus.c_last);

            if (dbrpcexec(m_dbproc)
== FAIL)
                ThrowError(CDBLIBERR::eDbRpcExec);

            // Get order lines
            if (dbresults(m_dbproc)
!= SUCCEEDED)
            {
                if
                ((m_DbLibErr == NULL) && (m_SqlErr == NULL))
                    throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
                else
                    ThrowError(CDBLIBERR::eDbResults);
            }

            if (dbnumcols(m_dbproc)
!= 5)
                ThrowError(CDBLIBERR::eWrongNumCols);

            i = 0;
            while (TRUE)
            {
                rc =
dbnextrow(m_dbproc);
                if (rc ==
NO_MORE_ROWS)
                    break;
                if (rc !=
REG_ROW)
                    ThrowError(CDBLIBERR::eDbNextRow);

                if (pData=dbdata(m_dbproc, 1))

```

```

                m_txn.OrderStatus.OL[i].ol_supply_w_id =
(* (DBSMALLINT *) pData);
                if (pData=dbdata(m_dbproc, 2))
                    m_txn.OrderStatus.OL[i].ol_i_id = (* (DBINT
*) pData);
                if (pData=dbdata(m_dbproc, 3))
                    m_txn.OrderStatus.OL[i].ol_quantity =
(* (DBSMALLINT *) pData);
                if (pData=dbdata(m_dbproc, 4))
                    dbconvert(m_dbproc, SQLNUMERIC,
(LPCTSTR)pData, dbdatlen(m_dbproc, 4),
SQLFLT8, (BYTE
*)&m_txn.OrderStatus.OL[i].ol_amount, 8);
                if (pData=dbdata(m_dbproc, 5))
                {
                    datetime = *((DBDATETIME *) pData);
                    dbdatecrack(m_dbproc, &daterec, &datetime);
                    m_txn.OrderStatus.OL[i].ol_delivery_d.year
= daterec.year;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.month
= daterec.month;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.day
= daterec.day;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.hour
= daterec.hour;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.minut
e = daterec.minute;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.secon
d = daterec.second;
                }
                ++i;
                m_txn.OrderStatus.o_ol_cnt = i;
            }
            if (dbresults(m_dbproc)
!= SUCCEEDED)
                ThrowError(CDBLIBERR::eDbResults);
            if (dbnextrow(m_dbproc)
!= REG_ROW)

```



```

ThrowError(CDBLIBERR::eDbNextRow);
        if (dbnumcols(m_dbproc)
!= 8)
        ThrowError(CDBLIBERR::eWrongNumCols);

        if(pData=dbdata(m_dbproc, 1))
        m_txn.OrderStatus.c_id = (*(DBINT *)
pData);
        if(pData=dbdata(m_dbproc, 2))
        UtilStrCpy(m_txn.OrderStatus.c_last, pData,
dbdatlen(m_dbproc,2));
        if(pData=dbdata(m_dbproc, 3))
        UtilStrCpy(m_txn.OrderStatus.c_first,
pData, dbdatlen(m_dbproc,3));
        if(pData=dbdata(m_dbproc, 4))
        UtilStrCpy(m_txn.OrderStatus.c_middle,
pData, dbdatlen(m_dbproc, 4));
        if(pData=dbdata(m_dbproc, 5))
        {
                datetime =
*(DBDATETIME *) pData);
                dbdatecrack(m_dbproc, &daterec, &datetime);
                m_txn.OrderStatus.o_entry_d.year =
daterec.year;
                m_txn.OrderStatus.o_entry_d.month =
daterec.month;
                m_txn.OrderStatus.o_entry_d.day =
daterec.day;
                m_txn.OrderStatus.o_entry_d.hour =
daterec.hour;
                m_txn.OrderStatus.o_entry_d.minute =
daterec.minute;
                m_txn.OrderStatus.o_entry_d.second =
daterec.second;
        }
        if(pData=dbdata(m_dbproc, 6))
        m_txn.OrderStatus.o_carrier_id =
(*(DBSMALLINT *) pData);
        if(pData=dbdata(m_dbproc, 7))

```

```

        dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,7),
        SQLFLT8, (BYTE
*)&m_txn.OrderStatus.c_balance, 8);
        if(pData=dbdata(m_dbproc, 8))
        m_txn.OrderStatus.o_id = (*(DBINT *)
pData);
        DiscardNextRows(0);
        DiscardNextResults(0);
        if
(m_txn.OrderStatus.o_ol_cnt == 0)
        throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER
);
        else if
(m_txn.OrderStatus.c_id == 0 &&
m_txn.OrderStatus.c_last[0] == 0)
        throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
        else
        m_txn.OrderStatus.exec_status_code = eOK;
        return;
    }
    catch (CSQLERR *e)
    {
        if ((e->m_msgno == 1205
||
== iErrOleDbProvider &&
(e->m_msgno
>m_msgtext, sErrTimeoutExpired) != NULL) &&
(++iTryCount
<= iMaxRetries))
        {
            // hit
            deadlock; backoff for increasingly longer period
            delete e;
            Sleep(10 *
iTryCount);
        }
        else
            throw;
    }
    // while (TRUE)
    if (iTryCount)
    throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

```

```

void CTPCC_DBLIB::Delivery()
{
    int i;
    int iTryCount =
0;
    const BYTE *pData;
    ResetError();
    while (TRUE)
    {
        try
        {
            dbrpcinit(m_dbproc,
"tpcc_delivery", 0);
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.Delivery.w_id);
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.Delivery.o_carrier_id);
            if (dbrpcexec(m_dbproc)
== FAIL)
                ThrowError(CDBLIBERR::eDbRpcExec);
            if (dbresults(m_dbproc)
!= SUCCEEDED)
                ThrowError(CDBLIBERR::eDbResults);
            if (dbnextrow(m_dbproc)
!= REG_ROW)
                ThrowError(CDBLIBERR::eDbNextRow);
            if (dbnumcols(m_dbproc)
!= 10)
                ThrowError(CDBLIBERR::eWrongNumCols);
            for (i=0; i<10; i++)
            {
                if (pData =
dbdata(m_dbproc, i+1))
                m_txn.Delivery.o_id[i] = (*(DBINT *)pData);
            }
            DiscardNextRows(0);
            DiscardNextResults(0);
            m_txn.Delivery.exec_status_code = eOK;
            return;
        }
        catch (CSQLERR *e)
        {

```

```

        if ((e->m_msgno == 1205
||
== iErrOleDbProvider &&
>m_msgtext, sErrTimeoutExpired) != NULL) &&
<= iMaxRetries))
    {
        // hit
        deadlock; backoff for increasingly longer period
        delete e;
        Sleep(10 *
iTryCount);
    }
    else
        throw;
}
// while (TRUE)
}
if (iTryCount)
    throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_DBLIB::ResetError()
{
    if (m_DbLibErr != NULL)
    {
        delete m_DbLibErr;
        m_DbLibErr = (CDBLIBERR*)NULL;
    }

    if (m_SqlErr != NULL)
    {
        delete m_SqlErr;
        m_SqlErr = (CSQLERR*)NULL;
    }

    return;
}

```

tpcc_dblib.h

```

/* FILE: TPC_C_DBLIB.H
 * Microsoft
 * TPC-C Kit Ver. 4.20.000
 * Copyright
 * Microsoft, 1999
 * All Rights Reserved
 * Version
 * 4.10.000 audited by Richard Gimarc, Performance
 * Metrics, 3/17/99
 * PURPOSE: Header file for TPC-C txn class
 * implementation.
 * Change history:

```

```

* 4.20.000 - updated rev number to
match kit
*/
#pragma once

#ifndef PDBPROCESS
#define DBPROCESS void // dbprocess structure type
typedef DBPROCESS * PDBPROCESS;
#endif

// need to declare functions for import, unless
define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl __declspec( dllimport )
#endif

class CSQLERR : public CBaseErr
{
public:
    CSQLERR(void)
    {
        m_msgno = 0;
        m_msgstate = 0;
        m_severity = 0;
        m_msgtext = NULL;
    };

    ~CSQLERR()
    {
        delete [] m_msgtext;
    };

    int m_msgno;
    int m_msgstate;
    int m_severity;
    char *m_msgtext;

    int ErrorType()
    {return ERR_TYPE_SQL;};
    char* ErrorTypeStr() { return
"SQL"; }

    int ErrorNum()
    {return m_msgno;};
    char* ErrorText() {return
m_msgtext;};
};

class CDBLIBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eLogin,
        // error from dblogin
        eDbOpen,
        // error from dlopen

```

```

        eDbUse,
        // error from dbuse
        eDbSqlExec,
        // error from dbsqlexec
        eDbSet,
        // error from one of the dbset*
    };

    routines
    {
        eDbNextRow,
        // error from dbnextrow
        eWrongRowCount,
        // more or less rows returned than expected
        eWrongNumCols,
        // more or less columns returned than
        expected
    };

    eDbResults,
    // error from dbresults
    eDbRpcExec,
    // error from dbrpcexec
    eDbSetMaxProcs,
    // error from dbsetmaxprocs
    eDbProcHandler,
    // error from either dbprocerrhandle or
    dbprocmsghandle
    };

    CDBLIBERR(ACTION eAction, int
severity = 0, int dberror = 0, int oserr = 0)
    {
        m_eAction = eAction;
        m_severity = severity;
        m_dberror = dberror;
        m_oserr = oserr;

        m_dberrstr = NULL;
        m_oserrstr = NULL;
    };

    ~CDBLIBERR()
    {
        delete [] m_dberrstr;
        delete [] m_oserrstr;
    };

    ACTION m_eAction;
    int m_severity;
    int m_dberror;
    int m_oserr;
    char *m_dberrstr;
    char *m_oserrstr;

    int ErrorType()
    {return ERR_TYPE_DBLIB;};
    char* ErrorTypeStr() { return
"DBLIB"; }

    int ErrorNum()
    {return m_dberror;};
    char* ErrorText() {return
m_dberrstr;};
    int ErrorAction()
    { return (int)m_eAction; }

```

```

};

class CTPCC_DBLIB_ERR : public CBaseErr
{
public:
    enum CTPCC_DBLIB_ERRS
    {
        ERR_WRONG_SP_VERSION =
1, // "Wrong version of stored procs on
database server"
        ERR_INVALID_CUST,
// "Invalid Customer id,name."
        ERR_NO_SUCH_ORDER,
// "No orders found for
customer."
        ERR_RETRIED_TRANS,
// "Retries before transaction
succeeded."
    };

    CTPCC_DBLIB_ERR( int iErr ) {
m_errno = iErr; m_iTryCount = 0; };

    CTPCC_DBLIB_ERR( int iErr, int
iTryCount ) { m_errno = iErr; m_iTryCount =
iTryCount; };

    int m_errno;
    int m_iTryCount;

    int ErrorType()
{return ERR_TYPE_TPCC_DBLIB;};
    char* ErrorTypeStr() { return
"TPCC_DBLIB"; }
    int ErrorNum()
{return m_errno;};

    char* ErrorText();
};

class DllDecl CTPCC_DBLIB : public CTPCC_BASE
{
private: // declare variables and private
functions here...
    PDBPROCESS m_dbproc;
    CDBLIBERR *m_DbLibErr;
// not allocated until needed (maybe never)
    CSQLErr *m_SqlErr;
// not allocated until
needed (maybe never)
    int m_MaxRetries; // retry
count on deadlock

    void DiscardNextRows(int
iExpectedCount);
    void DiscardNextResults(int
iExpectedCount);
    void ThrowError(
CDBLIBERR::ACTION eAction );

```

```

void ResetError();

union
{
    NewOrder;          NEW_ORDER_DATA
    Payment;          PAYMENT_DATA
    Delivery;         DELIVERY_DATA
    StockLevel;       STOCK_LEVEL_DATA
    OrderStatus;      ORDER_STATUS_DATA
};
m_txn;

public:
    CTPCC_DBLIB(LPCSTR szServer,
LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost,
LPCSTR szDatabase );
    ~CTPCC_DBLIB(void);

    inline PNEW_ORDER_DATA
BuffAddr_NewOrder() { return
&m_txn.NewOrder; };
    inline PPAYMENT_DATA
BuffAddr_Payment() { return
&m_txn.Payment; };
    inline PDELIVERY_DATA
BuffAddr_Delivery() { return
&m_txn.Delivery; };
    inline PSTOCK_LEVEL_DATA
BuffAddr_StockLevel() { return
&m_txn.StockLevel; };
    inline PORDER_STATUS_DATA
BuffAddr_OrderStatus() { return
&m_txn.OrderStatus; };

    void NewOrder ();
    void Payment ();
    void Delivery ();
    void StockLevel ();
    void OrderStatus ();

// these are public because they
must be called from the dblib err_handler and
msg_hangler
// outside of the class
void SetDbLibError(int severity,
int dberr, int oserr, LPCSTR dberrstr, LPCSTR
oserrstr);
void SetSqlError( int msgno, int
msgstate, int severity, LPCSTR msgtext );
};

extern "C" DllDecl CTPCC_DBLIB* CTPCC_DBLIB_new
( LPCSTR szServer, LPCSTR szUser, LPCSTR
szPassword, LPCSTR szHost, LPCSTR szDatabase );

```

```

typedef CTPCC_DBLIB* (TYPE_CTPCC_DBLIB)(LPCSTR,
LPCSTR, LPCSTR, LPCSTR, LPCSTR);

```

tpcc_enc.cpp

```

// tpcc_enc.cpp: implementation of the CTPCC_ENCINA
class.
//
//
//
#include <windows.h>
#include <process.h>
#include <stdio.h>
#include <stdarg.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <io.h>

#ifdef ICECAP
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\trans.h"
//tpckit transaction header contains definitions of
structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_enc.h"
#include "..\include\tpcc_type.h"
#include "mon_client.h"
#include "client_utils.h"

static CRITICAL_SECTION TpCriticalSection;
extern "C" char *errFile;

BOOL APIENTRY DllMain(HANDLE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:

DisableThreadLibraryCalls(hModule);
InitializeCriticalSection(&TpCriticalSection);
break;

        case DLL_PROCESS_DETACH:
DeleteCriticalSection(&TpCriticalSection);

```

```

        break;
        default:
            /* nothing */;
    }
    return TRUE;
}

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_ENCINA*
CTPCC_ENCINA_new()
{
    return new CTPCC_ENCINA();
}

// wrapper routine for enroll_client
__declspec(dllexport) CTPCC_ENCINA*
CTPCC_ENCINA_post_init()
{
    enroll_client();
    return NULL;
}

// constructor and destructor
CTPCC_ENCINA::CTPCC_ENCINA()
{
    // Add initialization of ENCINA
    Structures if any
        m_txn = (ENC_DATA
* )malloc(sizeof(ENC_DATA));
        if (m_txn == NULL)
            throw new
CENCERR(ERR_TYPE_MEMORY, ERR_FATAL_LEVEL);
}

CTPCC_ENCINA::~CTPCC_ENCINA()
{
    // free the data structure allocated with
    tpalloc
        free((char *)m_txn);
}

void CTPCC_ENCINA::NewOrder()
{
    // question: if we need to prepare the
    data?
    if (send_new_order(sizeof(ENC_DATA), (unsigned
char *)m_txn) == TRPC_ERROR)
        throw new CENCERR(TRPC_ERROR);

    if ( m_txn->ErrorType != ERR_SUCCESS )
        throw new CENCERR( m_txn-
>ErrorType, m_txn->error );
}

void CTPCC_ENCINA::Payment()
{
    if (send_payment(sizeof(ENC_DATA), (unsigned char
*)m_txn) == TRPC_ERROR)
        throw new CENCERR(TRPC_ERROR);
}

```

```

    if ( m_txn->ErrorType != ERR_SUCCESS )
        throw new CENCERR( m_txn-
>ErrorType, m_txn->error );
}

void CTPCC_ENCINA::Delivery()
{
    // Note: Delivery txn code in the tuxedo
    server does not implement logging of the delivery
    txn results, so cannot be used as
    is to run an auditable TPC-C result. For that
    // reason, delivery txns should not
    be done via Tuxedo.
    // The code is included for
    completeness.
    //m_txn->u.Delivery.exec_status_code =
    eDeliveryFailed;
    //return;

    // Note: If we use the delivery thread in
    tpcc.dll, it is not possible to get to this
    // point for delivery txns. But if we
    use Encina delivery server, the code is
    // needed. It is suggested using the
    delivery thread in tpcc.dll since it is
    // convenient and provides best
    performance.
    GetLocalTime(&m_txn-
>u.Delivery.queue_time);

    if (send_delivery(sizeof(ENC_DATA), (unsigned
char *)m_txn) == TRPC_ERROR)
        m_txn-
>u.Delivery.exec_status_code = eDeliveryFailed;
    else
        m_txn-
>u.Delivery.exec_status_code = eOK;
}

void CTPCC_ENCINA::StockLevel()
{
    if (send_stock_level(sizeof(ENC_DATA), (unsigned
char *)m_txn) == TRPC_ERROR)
        throw new CENCERR(TRPC_ERROR);

    if ( m_txn->ErrorType != ERR_SUCCESS )
        throw new CENCERR( m_txn-
>ErrorType, m_txn->error );
}

void CTPCC_ENCINA::OrderStatus()
{
    if (send_order_status(sizeof(ENC_DATA), (unsigned
char *)m_txn) == TRPC_ERROR)
        throw new CENCERR(TRPC_ERROR);

    if ( m_txn->ErrorType != ERR_SUCCESS )

```

```

        throw new CENCERR( m_txn-
>ErrorType, m_txn->error );
    }
}

char *CENCERR::ErrorText()
{
    if (m_iErrorType == TRPC_ERROR)
    {
        sprintf( m_szErrorText, "Error:
ENCINA TRPC error (see log file %s for details)",
errFile);
    }
    else
        sprintf( m_szErrorText, "Error:
Class %d, error # %d", m_iErrorType, m_iError );
    return m_szErrorText;
};

```

tpcc_enc.h

```

/* FILE: TPCC_ENCINA.H
 * Microsoft
TPC-C Kit Ver. 4.10.000
 * not yet
audited
 *
 * PURPOSE: Header file for TPC-C Encina
class implementation.
 * Copyright
Microsoft, 1999
 * All Rights Reserved
 *
*/

#ifndef _TPCC_ENCINA_H_
#define _TPCC_ENCINA_H_

#pragma once

// need to declare functions for import, unless
define has already been created
// by the DLL's .cpp module for export.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

class CTPCC_ENCINA : public CTPCC_BASE
{
private:
    struct ENC_DATA
    {
        int
        ErrorType;
        int
        error;
        union
    {

```

```

NEW_ORDER_DATA      NewOrder;
                    PAYMENT_DATA
                    Payment;
                    DELIVERY_DATA
                    Delivery;

STOCK_LEVEL_DATA    StockLevel;

ORDER_STATUS_DATA   OrderStatus;
                    } u;
                    } *m_txn;

public:

    CTPCC_ENCINA();
    virtual ~CTPCC_ENCINA();

    inline PNEW_ORDER_DATA
    BuffAddr_NewOrder() { return
&m_txn->u.NewOrder; };
    inline PPAYMENT_DATA
    BuffAddr_Payment() { return
&m_txn->u.Payment; };
    inline PDELIVERY_DATA
    BuffAddr_Delivery() { return
&m_txn->u.Delivery; };
    inline PSTOCK_LEVEL_DATA
    BuffAddr_StockLevel() { return
&m_txn->u.StockLevel; };
    inline PORORDER_STATUS_DATA
    BuffAddr_OrderStatus() { return
&m_txn->u.OrderStatus; };

    void NewOrder      ();
    void Payment       ();
    void Delivery      ();
    void StockLevel    ();
    void OrderStatus   ();
};

class CENCERR : public CBaseErr
{
private:
    char    m_szErrorText[64];
public:
    int     m_errno;
    int     m_iErrorType;
// match ErrorType in CTPCC_ENCINA
    int     m_iError;
// machine error in CTPCC_ENCINA

// use this interface for genuine
Encina errors
    CENCERR( int iErr )
    {
        m_errno = iErr; //
ENCINA error

```

```

ERR_TYPE_ENCINA;
                    m_iErrorType =
                    m_iError = 0; //
only meaningful if m_errno == TPEOS
                    };
// use this interface to
impersonate a non-Encina error type
                    CENCERR( int iErrorType, int
iError )
                    {
                    m_iErrorType =
iErrorType;
                    m_iError = iError;
                    m_errno = iError; //
???
                    }
// A CENCERR class can
impersonate another class, which happens if the error
error, but was simply transmitted back via Tuxedo.
                    int ErrorType()
                    {
                        return m_iErrorType;
                    }
                    int ErrorNum() {return m_errno;};
                    char *ErrorText();
};

// wrapper routine for class constructor:
extern "C" __declspec(dllexport) CTPCC_ENCINA*
CTPCC_ENCINA_new();
extern "C" __declspec(dllexport) CTPCC_ENCINA*
CTPCC_ENCINA_post_init();

typedef CTPCC_ENCINA* (TYPE_CTPCC_ENCINA)();

#endif // !defined(TPCC_ENCINA_H)

tpcc_odbc.cpp
/* FILE: TPCC_ODBC.CPP
* Microsoft
TPC-C Kit Ver. 4.42.000
* Copyright
Microsoft, 2002
* All Rights Reserved
*
* Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
*
* PURPOSE: Implements ODBC calls for TPC-C
txns.
* Contact: Charles Levine
(clevine@microsoft.com)
*
* Change history:

```

```

* 4.42.000 - changed w_id fields
from short to long to support >32K warehouses
* 4.20.000 - updated rev number to
match kit
* 4.10.001 - not deleting error
class in catch handler on deadlock retry;
* not a
functional bug, but a memory leak
*/

#include <windows.h>
#include <stdio.h>
#include <assert.h>

#define DBNTWIN32
#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>

// #define COMPILE_FOR_SNAC // define that to
compile for SQL Native Client; comment out to use
MDAC

#ifdef COMPILE_FOR_SNAC
#include <odbcss.h>
#else
// Compile for SNAC
#include <sqlncli.h>
#endif

#ifdef ICECAP
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_odbc.h"

// version string; must match return value from
tpcc_version stored proc
const char sVersion[] = "4.20.000";

const iMaxRetries = 10; // how many
retries on deadlock

const int iErrOleDbProvider = 7312;
const char sErrTimeoutExpired[] = "Timeout expired";

static SQLHENV henv = SQL_NULL_HENV;
// ODBC environment handle

BOOL APIENTRY DllMain(HMODULE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:

```

```

        DisableThreadLibraryCalls(hModule);
        if (
SQLAllocHandleStd(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
&henv) != SQL_SUCCESS )
            return FALSE;
        break;
        case DLL_PROCESS_DETACH:
            if (henv != NULL)
                SQLFreeEnv(henv);
            break;
        default:
            /* nothing */;
    }
    return TRUE;
}

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
 *
 */
char* CTPCC_ODBC_ERR::ErrorText(void)
{
    int i;

    static SERRORMSG errorMsgs[] =
    {
        { ERR_WRONG_SP_VERSION,
"Wrong version of stored procs on database
server" },
        { ERR_INVALID_CUST,
"Invalid Customer id,name." },
        { ERR_NO_SUCH_ORDER,
"No orders found for customer." },
        { ERR_RETRIED_TRANS,
"Retries before transaction succeeded." },
        { 0, "" }
    };

    static char szNotFound[] = "Unknown error
number.";

    for(i=0; errorMsgs[i].szMsg[0]; i++)
    {
        if ( m_errno ==
errorMsgs[i].iError )
            break;
    }
    if ( !errorMsgs[i].szMsg[0] )
        return szNotFound;
    else

```

```

        return errorMsgs[i].szMsg;
    }

    // wrapper routine for class constructor
    __declspec(dllexport) CTPCC_ODBC* CTPCC_ODBC_new(
        LPCWSTR szServer, // name of
SQL server
        LPCWSTR szUser, //
user name for login
        LPCWSTR szPassword, // password
for login
        LPCWSTR szHost, //
not used
        LPCWSTR szDatabase, // name of
database to use
        LPCWSTR szSPPrefix, // prefix to
append to the stored procedure names
        BOOL bCallNoDuplicatesNewOrder ) // whether
to check for non-duplicate items in NewOrder and call
a new SP
    {
        return new CTPCC_ODBC( szServer, szUser,
szPassword, szHost, szDatabase, szSPPrefix,
bCallNoDuplicatesNewOrder );
    }

CTPCC_ODBC::CTPCC_ODBC (
    LPCWSTR szServer,
// name of SQL server
    LPCWSTR szUser,
// user name for login
    LPCWSTR szPassword,
// password for login
    LPCWSTR szHost,
// not used
    LPCWSTR szDatabase,
// name of database to use
    LPCWSTR szSPPrefix,
// prefix to append to the stored procedure
names
    BOOL bCallNoDuplicatesNewOrder //
whether to check for non-duplicate items in NewOrder
and call a new SP
)
:
m_bCallNoDuplicatesNewOrder(bCallNoDuplicatesNewOrder)
{
    RETCODE rc;

    // initialization
    m_hdbc = SQL_NULL_HDBC;
    m_hstmt = SQL_NULL_HSTMT;

    m_hstmtNewOrder = SQL_NULL_HSTMT;
    m_hstmtPayment = SQL_NULL_HSTMT;
    m_hstmtDelivery = SQL_NULL_HSTMT;
    m_hstmtOrderStatus = SQL_NULL_HSTMT;
    m_hstmtStockLevel = SQL_NULL_HSTMT;

```

```

    m_descNewOrderCols1 = SQL_NULL_HDESC;
    m_descNewOrderCols2 = SQL_NULL_HDESC;
    m_descOrderStatusCols1 = SQL_NULL_HDESC;
    m_descOrderStatusCols2 = SQL_NULL_HDESC;

    wcsncpy(m_szSPPrefix, szSPPrefix,
sizeof(m_szSPPrefix)/sizeof(m_szSPPrefix[0]));

    if ( SQLAllocHandle(SQL_HANDLE_DBC, henv,
&m_hdbc) != SQL_SUCCESS )
        ThrowError(CODBCERR::eAllocHandle);

    if ( SQLSetConnectOption(m_hdbc,
SQL_PACKET_SIZE, 4096) != SQL_SUCCESS )
        ThrowError(CODBCERR::eConnOption);

    {
        char
szConnectStr[256];
        char
szOutStr[1024];
        SQLSMALLINT
iOutStrLen;

#ifdef COMPILE_FOR_SNAC
        sprintf( szConnectStr,
"DRIVER=SQL
Server;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
szServer, szUser,
szPassword, szDatabase );
#else
        // Compile for SNAC
        sprintf( szConnectStr,
"DRIVER=SQL Native
Client;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
szServer, szUser,
szPassword, szDatabase );
#endif
        rc = SQLDriverConnect(m_hdbc,
NULL, (SQLCHAR*)szConnectStr, sizeof(szConnectStr),
(SQLCHAR*)szOutStr,
sizeof(szOutStr), &iOutStrLen, SQL_DRIVER_NOPROMPT );

        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
            ThrowError(CODBCERR::eConnect);
    }

    if (SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmt) != SQL_SUCCESS)
        ThrowError(CODBCERR::eAllocHandle);

    {
        char
buffer[128];

```

```

        // set some options affecting
connection behavior
        strcpy(buffer, "set nocount on
set XACT_ABORT ON");
        rc = SQLExecDirect(m_hstmt,
(unsigned char *)buffer, SQL_NTS);
        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);

        // verify that version of stored
procs on server is correct
        char db_sp_version[10];
        strcpy(buffer, "{call
tpcc_version}");
        rc = SQLExecDirect(m_hstmt,
(unsigned char *)buffer, SQL_NTS);
        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);
        if ( SQLBindCol(m_hstmt, 1,
SQL_C_CHAR, &db_sp_version, sizeof(db_sp_version),
NULL) != SQL_SUCCESS )

                ThrowError(CODBCERR::eBindCol);
        if ( SQLFetch(m_hstmt) ==
SQL_ERROR )

                ThrowError(CODBCERR::eFetch);
        if
        (strcmp(db_sp_version,sVersion))
                throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_WRONG_SP_VERSION
);

        SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmt);
}

// Bind parameters for each of the
transactions
        InitNewOrderParams();
        InitPaymentParams();
        InitOrderStatusParams();
        InitDeliveryParams();
        InitStockLevelParams();
}

CTPCC_ODBC::~CTPCC_ODBC( void )
{
        // note: descriptors are automatically
released when the connection is dropped
        SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtNewOrder);
        SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtPayment);
        SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtDelivery);

```

```

        SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtOrderStatus);
        SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtStockLevel);

        SQLDisconnect(m_hdbc);
        SQLFreeHandle(SQL_HANDLE_DBC, m_hdbc);
}

void CTPCC_ODBC::ThrowError( CODBCERR::ACTION eAction
)
{
        RETCODE                rc;
        SDWORD                lNativeError;
        char                  szState[6];
        char                  szMsg[SQL_MAX_MESSAGE_LENGTH];
        char                  szTmp[6*SQL_MAX_MESSAGE_LENGTH];
        CODBCERR              *pODBCErr;
        // not allocated until needed (maybe never)

        pODBCErr = new CODBCERR();

        pODBCErr->m_NativeError = 0;
        pODBCErr->m_eAction = eAction;
        pODBCErr->m_bDeadLock = FALSE;

        szTmp[0] = 0;
        while (TRUE)
        {
                rc = SQLError(henv, m_hdbc,
m_hstmt, (BYTE *)&szState, &lNativeError,
(BYTE *)&szMsg, sizeof(szMsg), NULL);
                if (rc == SQL_NO_DATA)
                        break;

                // check for deadlock
                if (lNativeError == 1205 ||
(lNativeError == iErrOleDbProvider &&
strstr(szMsg,
sErrTimeoutExpired) != NULL))
                        pODBCErr->m_bDeadLock =
TRUE;

                error                // capture the (first) database
                if (pODBCErr->m_NativeError == 0
&& lNativeError != 0)
                        pODBCErr->m_NativeError
= lNativeError;

                // quit if there isn't enough
room to concatenate error text
                if ( (strlen(szMsg) + 2) >
(sizeof(szTmp) - strlen(szTmp)) )
                        break;

                // include line break after first
error msg

```

```

        if (szTmp[0] != 0)
                strcat( szTmp, "\n");
        strcat( szTmp, szMsg );
}

        if (pODBCErr->m_odbcerrstr != NULL)
        {
                delete [] pODBCErr->m_odbcerrstr;
                pODBCErr->m_odbcerrstr = NULL;
        }

        if (strlen(szTmp) > 0)
        {
                pODBCErr->m_odbcerrstr = new
char[ strlen(szTmp)+1 ];
                strcpy( pODBCErr->m_odbcerrstr,
szTmp );
        }

        SQLFreeStmt(m_hstmt, SQL_CLOSE);
        throw pODBCErr;
}

void CTPCC_ODBC::InitStockLevelParams()
{
        if ( SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtStockLevel) != SQL_SUCCESS )

                ThrowError(CODBCERR::eAllocHandle);

        m_hstmt = m_hstmtStockLevel;

        int i = 0;
        if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.StockLevel.w_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.StockLevel.d_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.StockLevel.threshold, 0, NULL) != SQL_SUCCESS
)
                ThrowError(CODBCERR::eBindParam);

        if ( SQLBindCol(m_hstmt, 1, SQL_C_SLONG,
&m_txn.StockLevel.low_stock, 0, NULL) != SQL_SUCCESS
)
                ThrowError(CODBCERR::eBindCol);

        //Compose Stock Level statement
        _snprintf(m_szStockLevelCommand,
sizeof(m_szStockLevelCommand)/sizeof(m_szStockLevelCo
mmand[0]),
                L"{call %stpcck_stocklevel
(?,?,?)}", m_szSPPrefix);
}

void CTPCC_ODBC::StockLevel()
{
        RETCODE                rc;

```

```

int iTryCount =
0;

m_hstmt = m_hstmtStockLevel;

while (TRUE)
{
    try
    {
        rc =
SQLExecDirectW(m_hstmt, m_szStockLevelCommand,
SQL_NTS);
        if (rc != SQL_SUCCESS
&& rc != SQL_SUCCESS_WITH_INFO)
            ThrowError(CODBCERR::eExecDirect);

        if ( SQLFetch(m_hstmt)
== SQL_ERROR )
            ThrowError(CODBCERR::eFetch);

        SQLFreeStmt(m_hstmt,
SQL_CLOSE);

        m_txn.StockLevel.exec_status_code = eOK;
        break;
    }
    catch (CODBCERR *e)
    {
        if (!e->m_bDeadLock)
            throw;

        // hit deadlock;
        backoff for increasingly longer period
        delete e;
        Sleep(10 * iTryCount);
    }
}

// if (iTryCount)
// throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_ODBC::InitNewOrderParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtNewOrder) != SQL_SUCCESS
||
SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtNewOrderNoDuplicates) != SQL_SUCCESS
||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols1) != SQL_SUCCESS
||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols2) != SQL_SUCCESS

```

```

||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderNoDuplicatesCols1) != SQL_SUCCESS
||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderNoDuplicatesCols2) != SQL_SUCCESS
)
    ThrowError(CODBCERR::eAllocHandle);

m_hstmt = m_hstmtNewOrder;

if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )
    ThrowError(CODBCERR::eSetStmtAttr);

int i = 0;
if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.w_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.d_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.c_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.o_ol_cnt, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.o_all_local, 0, NULL) != SQL_SUCCESS
)
    ThrowError(CODBCERR::eBindParam);

for (int j=0; j<MAX_OL_NEW_ORDER_ITEMS;
j++)
{
    if ( SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.OL[j].ol_i_id, 0, NULL) !=
SQL_SUCCESS
||
SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.OL[j].ol_supply_w_id, 0, NULL) !=
SQL_SUCCESS
||
SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.NewOrder.OL[j].ol_quantity, 0, NULL) !=
SQL_SUCCESS
)
        ThrowError(CODBCERR::eBindParam);
}

// set the bind offset pointer

```

```

if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_BIND_OFFSET_PTR, &m_BindOffset,
SQL_IS_POINTER ) != SQL_SUCCESS )
    ThrowError(CODBCERR::eSetStmtAttr);

i = 0;
if ( SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_i_name,
sizeof(m_txn.NewOrder.OL[0].ol_i_name), NULL) !=
SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.NewOrder.OL[0].ol_stock, 0,
NULL) != SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.OL[0].ol_brand_generic,
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), NULL)
!= SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_i_price, 0,
NULL) != SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_amount, 0,
NULL) != SQL_SUCCESS
)
    ThrowError(CODBCERR::eBindCol);

// associate the column bindings for the
second result set
if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )
    ThrowError(CODBCERR::eSetStmtAttr);

i = 0;
if ( SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.w_tax, 0, NULL) !=
SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.d_tax, 0, NULL) !=
SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.NewOrder.o_id, 0, NULL) !=
SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.c_last,
sizeof(m_txn.NewOrder.c_last), NULL) != SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.c_discount, 0, NULL)
!= SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.c_credit,
sizeof(m_txn.NewOrder.c_credit), NULL) != SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.NewOrder.o_entry_d, 0,
NULL) != SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_no_commit_flag, 0, NULL) !=
SQL_SUCCESS
)
}

```



```

        ThrowError(CODBCERR::eBindCol);

        //Compose the New Order statement
        _snwprintf(m_szNewOrderCommand,
sizeof(m_szNewOrderCommand)/sizeof(m_szNewOrderCommand),
                // 0      1      2
                //
012345678901234567890123456789
                L"call
%stpc_neworder(?,?,?,?,?,?,?,?,?,?,?,?,?,?
,?,?,?,?,?,?,"
                L"?,?,?,?,?,?,?,?,?,?,?,?,?,?
,?,?,?,?,?)", m_szSPPrefix);

        m_iBeginNewOrderVariablePart = 29 +
wcslen(m_szSPPrefix); // fixed part + prefix
part

        //////////////////////////////////////
        //////////////////////////////////////
        //
        //      Now initialize New Order that
works on no duplicate (w_id,i_id) pairs
        //      and returns one result set for
lineitem details.
        //
        //
        m_hstmt = m_hstmtNewOrderNoDuplicates;

        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols1, SQL_IS_POINTER ) !=
SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.o_ol_cnt, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.o_all_local, 0, NULL) != SQL_SUCCESS
        )
        ThrowError(CODBCERR::eBindParam);

        for (int j=0; j<MAX_OL_NEW_ORDER_ITEMS;
j++)
        {

```

```

                if ( SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.OL[j].ol_i_id, 0, NULL) !=
SQL_SUCCESS
                ||
SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.OL[j].ol_supply_w_id, 0, NULL) !=
SQL_SUCCESS
                ||
SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.NewOrder.OL[j].ol_quantity, 0, NULL) !=
SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindParam);
        }

        // set row-wise binding
        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.NewOrder.OL[0]),
SQL_IS_UIINTEGER) != SQL_SUCCESS
        || SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROWS_FETCHED_PTR, &m_RowsFetched, 0) !=
SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if ( SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_i_name,
sizeof(m_txn.NewOrder.OL[0].ol_i_name), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.NewOrder.OL[0].ol_stock, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.OL[0].ol_brand_generic,
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_i_price, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_amount, 0,
NULL) != SQL_SUCCESS
        )
        ThrowError(CODBCERR::eBindCol);

        // associate the column bindings for the
second result set
        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols2, SQL_IS_POINTER ) !=
SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;

```

```

        if ( SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.w_tax, 0, NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.d_tax, 0, NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.NewOrder.o_id, 0, NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.c_last,
sizeof(m_txn.NewOrder.c_last), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.c_discount, 0, NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.c_credit,
sizeof(m_txn.NewOrder.c_credit), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.NewOrder.o_entry_d, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_no_commit_flag, 0, NULL) !=
SQL_SUCCESS
        )
        ThrowError(CODBCERR::eBindCol);

        //Compose the New Order statement
        _snwprintf(m_szNewOrderNoDuplicatesCommand,
sizeof(m_szNewOrderNoDuplicatesCommand)/sizeof(m_szNew
OrderNoDuplicatesCommand[0]),
                L"call
%stpc_neworder_new(?,?,?,?,?,?,?,?,?,?
,?,?,?,?,?,?,"
                L"?,?,?,?,?,?,?,?,?,?
,?,?,?,?,?)", m_szSPPrefix);

        m_iBeginNewOrderNoDuplicatesVariablePart =
33 + wcslen(m_szSPPrefix); // fixed part + prefix
part
        }

        //
        //      Returns true if there are duplicate
(warehouse_id, item_id)
        //      lineitem pairs in New Order input
parameters.
        //
bool CTPCC_ODBC::DuplicatesInNewOrder()
{
        int i, j;

        for (i = 0; i < m_txn.NewOrder.o_ol_cnt;
++i)
        {
                for (j = i+1; j<
m_txn.NewOrder.o_ol_cnt; ++j)
                {

```

```

        if
(m_txn.NewOrder.OL[i].ol_i_id ==
m_txn.NewOrder.OL[j].ol_i_id)
        {
            return true;
        }
    }
    return false;
}

void CTPCC_ODBC::NewOrder()
{
    if (m_bCallNoDuplicatesNewOrder)
    {
        if (DuplicatesInNewOrder())
        {
            NewOrderDuplicates();
        }
        else
        {
            NewOrderNoDuplicates();
        }
    }
    else
    {
        NewOrderDuplicates();
    }
}

void CTPCC_ODBC::NewOrderDuplicates()
{
    int
    i;
    RETCODE          rc;
    int
    iTryCount = 0;

    0          1          2          //
    012345678901234567890123456789          //
    wchar_t
    szSqlTemplate[iMAX_SP_NAME_LEN];

    tpcc_neworder(?,?,?,?,"          // L"{call
    L"?,?,?,?,?,?,?,?,?,?,?,?,?",          //
    L"?,?,?,?,?,?,?,?,?,?,?,?,?",          //
    L"?,?,?,?,?,?,?,?,?,?,?,?,?)"          //
};
    m_hstmt = m_hstmtNewOrder;

```

```

        // associate the parameter and column
bindings for this transaction
        if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )
            ThrowError(CODBCERR::eSetStmtAttr);

        // clip statement buffer based on number of
parameters
        // fixed part is 29 chars and variable part
is 6 chars per line item
        wcsncpy(szSqlTemplate, m_szNewOrderCommand);
        i = m_iBeginNewOrderVariablePart +
m_txn.NewOrder.o_ol_cnt*6;
        wcsncpy( &szSqlTemplate[i], L" )" );

        // check whether any order lines are for a
remote warehouse
        m_txn.NewOrder.o_all_local = 1;
        for (i = 0; i < m_txn.NewOrder.o_ol_cnt;
i++)
        {
            if
(m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
            {
                m_txn.NewOrder.o_all_local = 0; // at
least one remote warehouse
                break;
            }
        }
        while (TRUE)
        {
            try
            {
                m_BindOffset = 0;
                rc =
SQLExecDirectW(m_hstmt, szSqlTemplate, SQL_NTS);
                if (rc != SQL_SUCCESS
&& rc != SQL_SUCCESS_WITH_INFO)
                    ThrowError(CODBCERR::eExecDirect);

                // Get order line
results
                m_txn.NewOrder.total_amount = 0;
                for (i = 0;
i < m_txn.NewOrder.o_ol_cnt; i++)
                {
                    // set the
bind offset value...
                    m_BindOffset
= i * sizeof(m_txn.NewOrder.OL[0]);

                    if (
SQLFetch(m_hstmt) == SQL_ERROR)

```

```

                    ThrowError(CODBCERR::eFetch);

                    // move to
the next resultset
                    if (
SQLMoreResults(m_hstmt) == SQL_ERROR )
                        ThrowError(CODBCERR::eMoreResults);

                    m_txn.NewOrder.total_amount +=
m_txn.NewOrder.OL[i].ol_amount;
                }
            }
            // associate the column
bindings for the second result set
            if ( SQLSetStmtAttrW(
m_hstmt, SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

            if ( SQLFetch(m_hstmt)
== SQL_ERROR)
                ThrowError(CODBCERR::eFetch);

            SQLFreeStmt(m_hstmt,
SQL_CLOSE);

            if (m_no_commit_flag ==
1)
            {
                m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

                m_txn.NewOrder.exec_status_code = eOK;
            }
            else
            {
                m_txn.NewOrder.exec_status_code =
eInvalidItem;

                break;
            }
        }
        catch (CODBCERR *e)
        {
            if (!(e->m_bDeadLock)
|| (++iTryCount > iMaxRetries))
                throw;

            // hit deadlock;
backoff for increasingly longer period
            delete e;
            Sleep(10 * iTryCount);
        }
    }
}

```

```

//      if (iTryCount)
//          throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

//
//      No lineitem duplicates optimized version.
//
void CTPCC_ODBC::NewOrderNoDuplications()
{
    int
    i;
    RETCODE          rc;
    int
    iTryCount = 0;

0      1      2      3          //
0123456789012345678901234567890123          //
    wchar_t
    szSqlTemplate[iMAX_SP_NAME_LEN];

tpcc_neworder_new(?,?,?,?,"          // L"(call
L"?,?,?,?,?,?,?,?,?,?,?,?,?",          //
L"?,?,?,?,?,?,?,?,?,?,?,?,?",          //
L"?,?,?,?,?,?,?,?,?,?,?,?,?)"");
    m_hstmt = m_hstmtNewOrderNoDuplications;
    // associate the parameter and column
bindings for this transaction
    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicationsCols1, SQL_IS_POINTER ) !=
SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    // clip statement buffer based on number of
parameters
    // fixed part is 33 chars and variable part
is 6 chars per line item
    wcsncpy(szSqlTemplate,
m_szNewOrderNoDuplicationsCommand);
    i =
m_iBeginNewOrderNoDuplicationsVariablePart +
m_txn.NewOrder.o_ol_cnt*6;
    wcsncpy( &szSqlTemplate[i], L" ) );

    // check whether any order lines are for a
remote warehouse

```

```

    m_txn.NewOrder.o_all_local = 1;
    for (i = 0; i < m_txn.NewOrder.o_ol_cnt;
i++)
    {
        if
(m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
        {
            m_txn.NewOrder.o_all_local = 0; // at
least one remote warehouse
            break;
        }
    }
    while (TRUE)
    {
        try
        {
            // configure block
            cursor
            if (
SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)1, 0) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

            rc =
SQLExecDirectW(m_hstmt, szSqlTemplate, SQL_NTS);
            if (rc != SQL_SUCCESS
&& rc != SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eExecDirect);

            // configure block
            cursor
            if
(SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_NEW_ORDER_ITEMS, 0) !=
SQL_SUCCESS)
                ThrowError(CODBCERR::eSetStmtAttr);

            // Get order line
            results
            if ( SQLFetch(m_hstmt)
== SQL_ERROR)
                ThrowError(CODBCERR::eFetch);

            m_txn.NewOrder.total_amount = 0;
            for (i = 0;
i < m_txn.NewOrder.o_ol_cnt; i++)
            {
                m_txn.NewOrder.total_amount +=
m_txn.NewOrder.OL[i].ol_amount;
            }

```

```

// associate the column
bindings for the second result set
    if ( SQLSetStmtAttrW(
m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicationsCols2, SQL_IS_POINTER ) !=
SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    // move to the next
resultset
    if (
SQLMoreResults(m_hstmt) == SQL_ERROR )
        ThrowError(CODBCERR::eMoreResults);

    if ( SQLFetch(m_hstmt)
== SQL_ERROR)
        ThrowError(CODBCERR::eFetch);

    SQLFreeStmt(m_hstmt,
SQL_CLOSE);

    if (m_no_commit_flag ==
1)
    {
        m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

        m_txn.NewOrder.exec_status_code = eOK;
    }
    else
        m_txn.NewOrder.exec_status_code =
eInvalidItem;

        break;
    }
    catch (CODBCERR *e)
    {
        if (!(e->m_bDeadLock)
|| (++iTryCount > iMaxRetries))
            throw;

        // hit deadlock;
backoff for increasingly longer period
        delete e;
        Sleep(10 * iTryCount);
    }

    //      if (iTryCount)
    //          throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

```

```

void CTPCC_ODBC::InitPaymentParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtPayment) != SQL_SUCCESS )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtPayment;

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Payment.c_w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_NUMERIC, 6, 2,
&m_txn.Payment.h_amount, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.Payment.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.Payment.c_d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Payment.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_txn.Payment.c_last), 0,
&m_txn.Payment.c_last, sizeof(m_txn.Payment.c_last),
NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindParam);

    i = 0;
    if ( SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.Payment.c_id,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_last,
sizeof(m_txn.Payment.c_last), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.Payment.h_date,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_street_1,
sizeof(m_txn.Payment.w_street_1), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_street_2,
sizeof(m_txn.Payment.w_street_2), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_city,
sizeof(m_txn.Payment.w_city), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_state,

```

```

sizeof(m_txn.Payment.w_state), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_zip,
sizeof(m_txn.Payment.w_zip), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.d_street_1,
sizeof(m_txn.Payment.d_street_1), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.d_street_2,
sizeof(m_txn.Payment.d_street_2), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.d_city,
sizeof(m_txn.Payment.d_city), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.d_state,
sizeof(m_txn.Payment.d_state), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.d_zip,
sizeof(m_txn.Payment.d_zip), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_first,
sizeof(m_txn.Payment.c_first), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_middle,
sizeof(m_txn.Payment.c_middle), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_street_1,
sizeof(m_txn.Payment.c_street_1), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_street_2,
sizeof(m_txn.Payment.c_street_2), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_city,
sizeof(m_txn.Payment.c_city), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_state,
sizeof(m_txn.Payment.c_state), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_zip,
sizeof(m_txn.Payment.c_zip), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_phone,
sizeof(m_txn.Payment.c_phone), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.Payment.c_since,
0, NULL) != SQL_SUCCESS

```

```

        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_credit,
sizeof(m_txn.Payment.c_credit), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.Payment.c_credit_lim, 0, NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.Payment.c_discount,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.Payment.c_balance,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_data,
sizeof(m_txn.Payment.c_data), NULL) !=
SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindCol);

    //Compose Payment statement
    _snprintf(m_szPaymentCommand,
sizeof(m_szPaymentCommand)/sizeof(m_szPaymentCommand[
0]),
        L"{call %stpcc_payment
(?, ?, ?, ?, ?, ?, ?)}", m_szSPPrefix);

void CTPCC_ODBC::Payment()
{
    RETCODE rc;
    int iTryCount =

0;

    m_hstmt = m_hstmtPayment;

    if (m_txn.Payment.c_id != 0)
        m_txn.Payment.c_last[0] = 0;

    while (TRUE)
    {
        try
        {
            rc =
SQLExecDirectW(m_hstmt, m_szPaymentCommand, SQL_NTS);
            if (rc != SQL_SUCCESS
&& rc != SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eExecDirect);

            if ( SQLFetch(m_hstmt)
== SQL_ERROR)
                ThrowError(CODBCERR::eFetch);

            SQLFreeStmt(m_hstmt,
SQL_CLOSE);

            if (m_txn.Payment.c_id
== 0)

```

```

        throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_INVALID_CUST );
        else

            m_txn.Payment.exec_status_code = eOK;

            break;
        }
        catch (COBDCERR *e)
        {
            if (!(e->m_bDeadLock)
|| (++iTryCount > iMaxRetries))
                throw;

            // hit deadlock;
            backoff for increasingly longer period
            delete e;
            Sleep(10 * iTryCount);
        }
    }

//      if (iTryCount)
//          throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_ODBC::InitOrderStatusParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtOrderStatus) != SQL_SUCCESS
||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols1) != SQL_SUCCESS
||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols2) != SQL_SUCCESS
)

        ThrowError(COBDCERR::eAllocHandle);

    m_hstmt = m_hstmtOrderStatus;

    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )

        ThrowError(COBDCERR::eSetStmtAttr);

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.OrderStatus.w_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.OrderStatus.d_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,

```

```

sizeof(m_txn.OrderStatus.c_last), 0,
&m_txn.OrderStatus.c_last,
sizeof(m_txn.OrderStatus.c_last), NULL) !=
SQL_SUCCESS
        )
            ThrowError(COBDCERR::eBindParam);

        // configure block cursor
        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.OrderStatus.OL[0]), 0) !=
SQL_SUCCESS
|| SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROWS_FETCHED_PTR, &m_RowsFetched, 0) !=
SQL_SUCCESS
        )
            ThrowError(COBDCERR::eSetStmtAttr);

        i = 0;
        if ( SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.OL[0].ol_supply_w_id,
0, NULL) != SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.OL[0].ol_i_id, 0,
NULL) != SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.OrderStatus.OL[0].ol_quantity,
0, NULL) != SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.OrderStatus.OL[0].ol_amount, 0,
NULL) != SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP,
&m_txn.OrderStatus.OL[0].ol_delivery_d, 0, NULL) !=
SQL_SUCCESS
        )
            ThrowError(COBDCERR::eBindCol);

        if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )

            ThrowError(COBDCERR::eSetStmtAttr);

        i = 0;
        if ( SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.c_id, 0, NULL) !=
SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.OrderStatus.c_last,
sizeof(m_txn.OrderStatus.c_last), NULL) !=
SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.OrderStatus.c_first,
sizeof(m_txn.OrderStatus.c_first), NULL) !=
SQL_SUCCESS
|| SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.OrderStatus.c_middle,

```

```

sizeof(m_txn.OrderStatus.c_middle), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.OrderStatus.o_entry_d,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.OrderStatus.o_carrier_id, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.OrderStatus.c_balance, 0, NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.c_id, 0, NULL) !=
SQL_SUCCESS
        )
            ThrowError(COBDCERR::eBindCol);

        //Compose Order Status statement
        _snprintf(m_szOrderStatusCommand,
sizeof(m_szOrderStatusCommand)/sizeof(m_szOrderStatus
Command[0]),
            L" {call %stpcc_orderstatus
(?,?,?,?)", m_szSPPrefix);
    }

void CTPCC_ODBC::OrderStatus()
{
    int
        iTryCount = 0;

    RETCODE
        rc;

    m_hstmt = m_hstmtOrderStatus;

    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )

        ThrowError(COBDCERR::eSetStmtAttr);

    if (m_txn.OrderStatus.c_id != 0)
        m_txn.OrderStatus.c_last[0] = 0;

    while (TRUE)
    {
        try
        {
            // configure block
            cursor
                if (
SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)1, 0) != SQL_SUCCESS )

                    ThrowError(COBDCERR::eSetStmtAttr);

                rc =
SQLExecDirectW(m_hstmt, m_szOrderStatusCommand,
SQL_NTS);

                if ( ((rc ==
SQL_SUCCESS_WITH_INFO) && (m_RowsFetched != 0)) ||
(rc == SQL_ERROR) )

```

```

        ThrowError(CODBCERR::eExecDirect);

        // configure block
        cursor
            if (
                SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_ARRAY_SIZE,
                    (SQLPOINTER)MAX_OL_ORDER_STATUS_ITEMS, 0) !=
                SQL_SUCCESS )
            ThrowError(CODBCERR::eSetStmtAttr);

            rc = SQLFetchScroll(
                m_hstmt, SQL_FETCH_NEXT, 0 );
            if ( (rc ==
                SQL_SUCCESS_WITH_INFO) && (m_RowsFetched != 0) ||
                (rc == SQL_ERROR) )
            ThrowError(CODBCERR::eFetchScroll);

            m_txn.OrderStatus.o_ol_cnt =
                (short)m_RowsFetched;

            if
                (m_txn.OrderStatus.o_ol_cnt != 0)
            {
                if (
                    SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
                        m_descOrderStatusCols2, SQL_IS_POINTER ) !=
                        SQL_SUCCESS )
                    ThrowError(CODBCERR::eSetStmtAttr);

                    if (
                        SQLMoreResults(m_hstmt) == SQL_ERROR )
                        ThrowError(CODBCERR::eMoreResults);

                        if ( (rc =
                            SQLFetch(m_hstmt)) == SQL_ERROR)
                            ThrowError(CODBCERR::eFetch);
                        }

                        SQLFreeStmt(m_hstmt,
                            SQL_CLOSE);

                            if
                                (m_txn.OrderStatus.o_ol_cnt == 0)
                                throw new
                                    CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_NO_SUCH_ORDER );
                                else if
                                    (m_txn.OrderStatus.c_id == 0 &&
                                        m_txn.OrderStatus.c_last[0] == 0)
                                        throw new
                                            CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_INVALID_CUST );
                                        else

                                            m_txn.OrderStatus.exec_status_code = eOK;

```

```

                break;
            }
            catch (CODBCERR *e)
            {
                if (!e->m_bDeadLock)
                    throw;

                    // hit deadlock;
                    backoff for increasingly longer period
                    delete e;
                    Sleep(10 * iTryCount);
            }
        }

        // if (iTryCount)
        // throw new
        CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
            iTryCount);
    }

    void CTPCC_ODBC::InitDeliveryParams()
    {
        if ( SQLAllocHandle(SQL_HANDLE_STMT,
            m_hdbc, &m_hstmtDelivery) != SQL_SUCCESS )
            ThrowError(CODBCERR::eAllocHandle);

            m_hstmt = m_hstmtDelivery;

            int i = 0;
            if ( SQLBindParameter(m_hstmt, ++i,
                SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
                &m_txn.Delivery.w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i,
                SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
                &m_txn.Delivery.o_carrier_id, 0, NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindParam);

                for (i=0; i<10; i++)
                {
                    if ( SQLBindCol(m_hstmt,
                        (UWORD)(i+1), SQL_C_SLONG, &m_txn.Delivery.o_id[i],
                        0, NULL) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eBindCol);
                }

                //Compose Delivery statement
                _snprintf(m_szDeliveryCommand,
                    sizeof(m_szDeliveryCommand)/sizeof(m_szDeliveryComman
                    d[0]),
                    L"%sL"%{call %stpc_delivery (?,?)}",
                    m_szSPPrefix);
            }

            void CTPCC_ODBC::Delivery()
            {
                RETCODE          rc;

```

```

        int          iTryCount =
        0;

        m_hstmt = m_hstmtDelivery;

        while (TRUE)
        {
            try
            {
                rc =
                SQLExecDirectW(m_hstmt, m_szDeliveryCommand,
                    SQL_NTS);
                if (rc != SQL_SUCCESS
                    && rc != SQL_SUCCESS_WITH_INFO)
                    ThrowError(CODBCERR::eExecDirect);

                    if ( SQLFetch(m_hstmt)
                        == SQL_ERROR )
                        ThrowError(CODBCERR::eFetch);

                        SQLFreeStmt(m_hstmt,
                            SQL_CLOSE);

                            m_txn.Delivery.exec_status_code = eOK;
                            break;
                        }
                        catch (CODBCERR *e)
                        {
                            if (!e->m_bDeadLock)
                                throw;

                                    // hit deadlock;
                                    backoff for increasingly longer period
                                    delete e;
                                    Sleep(10 * iTryCount);
                                }
                            }

                            // if (iTryCount)
                            // throw new
                            CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
                                iTryCount);
                            }

                            tpcc_odbc.h
                            /* FILE: TPCC_ODBC.H
                             * Microsoft
                             * TPC-C Kit Ver. 4.20.000
                             * Copyright
                             * Microsoft, 1999
                             * All Rights Reserved
                             *
                             * Version
                             * 4.10.000 audited by Richard Gimarc, Performance
                             * Metrics, 3/17/99
                             *

```

```

*      PURPOSE: Header file for TPC-C txn class
implementation.
*
*      Change history:
*      4.20.000 - updated rev number to
match kit
*/
#pragma once

// need to declare functions for import, unless
define has already been created
// by the DLL's .cpp module for export.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

#define iMAX_SP_NAME_LEN 256 //maximum length of a
stored procedure name with parameters

class CODBCERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eAllocConn,
        // error from SQLAllocConnect
        eAllocHandle,
        // error from SQLAllocHandle
        eConnOption,
        // error from SQLSetConnectOption
        eConnect,
        // error from SQLConnect
        eAllocStmt,
        // error from SQLAllocStmt
        eExecDirect,
        // error from SQLExecDirect
        eBindParam,
        // error from SQLBindParameter
        eBindCol,
        // error from SQLBindCol
        eFetch,
        // error from SQLFetch
        eFetchScroll,
        // error from SQLFetchScroll
        eMoreResults,
        // error from SQLMoreResults
        ePrepare,
        // error from SQLPrepare
        eExecute,
        // error from SQLExecute
        eSetEnvAttr,
        // error from SQLSetEnvAttr
        eSetStmntAttr,
        // error from SQLSetStmntAttr
    };

    CODBCERR(void)
    {
        m_eAction = eNone;

```

```

        m_NativeError = 0;
        m_bDeadLock = FALSE;
        m_odbcerrstr = NULL;
    };

    ~CODBCERR()
    {
        if (m_odbcerrstr !=
NULL)
            delete []
m_odbcerrstr;
    };

    ACTION    m_eAction;
    int
m_NativeError;
    BOOL    m_bDeadLock;
    char    *m_odbcerrstr;

    int
ErrorType()
{return ERR_TYPE_ODBC;};
    char*
ErrorTypeStr() { return
"ODBC"; };
    int
ErrorNum()
{return m_NativeError;};
    char*
ErrorText() {return
m_odbcerrstr;};
    int
ErrorAction()
{ return (int)m_eAction; };
};

class CTPCC_ODBC_ERR : public CBaseErr
{
public:
    enum TPCC_ODBC_ERRS
    {
        ERR_WRONG_SP_VERSION =
1, // "Wrong version of stored procs on
database server"
        ERR_INVALID_CUST,
        // "Invalid Customer id,name."
        ERR_NO_SUCH_ORDER,
        // "No orders found for
customer."
        ERR_RETRIED_TRANS,
        // "Retries before transaction
succeeded."
    };

    CTPCC_ODBC_ERR( int iErr ) {
m_errno = iErr; m_iTryCount = 0; };

    CTPCC_ODBC_ERR( int iErr, int
iTryCount ) { m_errno = iErr; m_iTryCount =
iTryCount; };

    int
m_errno;
    int
m_iTryCount;

    int
ErrorType()
{return ERR_TYPE_TPCC_ODBC;};

```

```

    char*
ErrorTypeStr() { return
"TPCC ODBC"; };
    int
ErrorNum()
{return m_errno;};
    char*
ErrorText();
};

class DllDecl CTPCC_ODBC : public CTPCC_BASE
{
private:
    // declare variables and private
functions here...
    BOOL    m_bDeadlock;
    // transaction was selected as
deadlock victim
    int
m_MaxRetries; // retry
count on deadlock
    SQLHENV    m_henv;
    // ODBC environment
handle
    SQLHDBC    m_hdbc;
    SQLHSTMT    m_hstmt;
    // the current hstmt
    SQLHSTMT    m_hstmtNewOrder;
    SQLHSTMT
m_hstmtNewOrderNoDuplicates; // NewOrder
with one result set for lineitem details
    SQLHSTMT    m_hstmtPayment;
    SQLHSTMT    m_hstmtDelivery;
    SQLHSTMT    m_hstmtOrderStatus;
    SQLHSTMT    m_hstmtStockLevel;
    SQLHDESC    m_descNewOrderCols1;
    SQLHDESC    m_descNewOrderCols2;
    SQLHDESC
m_descNewOrderNoDuplicatesCols1; //
NewOrder with one result set for lineitem details
    SQLHDESC
m_descNewOrderNoDuplicatesCols2; //
NewOrder with one result set for lineitem details
    SQLHDESC    m_descOrderStatusCols1;
    SQLHDESC    m_descOrderStatusCols2;
    wchar_t
m_szSPPrefix[32]; // stored procedures
prefix
    wchar_t
m_szNewOrderCommand[iMAX_SP_NAME_LEN];
    wchar_t
m_szNewOrderNoDuplicatesCommand[iMAX_SP_NAM
E_LEN];
    int
m_iBeginNewOrderVariablePart; // begining
of the variable part in NewOrder statement
    int
m_iBeginNewOrderNoDuplicatesVariablePart;

```

```

// begining of the variable part in
NewOrder statement
    wchar_t
    m_szPaymentCommand[iMAX_SP_NAME_LEN];
    wchar_t
    m_szDeliveryCommand[iMAX_SP_NAME_LEN];
    wchar_t
    m_szOrderStatusCommand[iMAX_SP_NAME_LEN];
    wchar_t
    m_szStockLevelCommand[iMAX_SP_NAME_LEN];

    // new-order specific fields
    SQLINTEGER      m_BindOffset;
    m_RowsFetched;
    int
    m_no_commit_flag;

    // tpcc_neworder_new flag
    BOOL
    m_bCallNoDuplicatesNewOrder;

    void ThrowError( CODBCERR::ACTION
eAction );

    void InitNewOrderParams();
    void InitPaymentParams();
    void InitDeliveryParams();
    void InitStockLevelParams();
    void InitOrderStatusParams();

    union
    {
        NEW_ORDER_DATA
        NewOrder;
        PAYMENT_DATA
        Payment;
        DELIVERY_DATA
        Delivery;
        STOCK_LEVEL_DATA
        StockLevel;
        ORDER_STATUS_DATA
        OrderStatus;
    }
    m_txn;

    bool DuplicatesInNewOrder();
    void NewOrderDuplicates();
    void NewOrderNoDuplicates();

public:
    CTPCC_ODBC( LPCWSTR
szServer, LPCWSTR szUser, LPCWSTR szPassword,
LPCWSTR szHost, LPCWSTR szDatabase,
LPCWSTR szSPPrefix, BOOL
bCallNoDuplicatesNewOrder);
    ~CTPCC_ODBC(void);

```

```

inline PNEW_ORDER_DATA
BuffAddr_NewOrder() { return
&m_txn.NewOrder;
};
inline PPAYMENT_DATA
BuffAddr_Payment() { return
&m_txn.Payment;
};
inline PDELIVERY_DATA
BuffAddr_Delivery() { return
&m_txn.Delivery;
};
inline PSTOCK_LEVEL_DATA
BuffAddr_StockLevel() { return
&m_txn.StockLevel;
};
inline PORDER_STATUS_DATA
BuffAddr_OrderStatus() { return
&m_txn.OrderStatus;
};

void NewOrder ();
void Payment ();
void Delivery ();
void StockLevel ();
void OrderStatus ();

};

// wrapper routine for class constructor
extern "C" DllDecl CTPCC_ODBC* CTPCC_ODBC_new
( LPCWSTR szServer, LPCWSTR szUser,
LPCWSTR szPassword,
LPCWSTR szHost, LPCWSTR szDatabase,
LPCWSTR szSPPrefix, BOOL
bCallNoDuplicatesNewOrder );

typedef CTPCC_ODBC* (TYPE_CTPCC_ODBC) (LPCWSTR, LPCWSTR,
LPCWSTR, LPCWSTR, LPCWSTR, LPCWSTR, BOOL);

tpcc_tux.cpp
/* FILE: TPCC_TUX.CPP
* Microsoft
* TPC-C Kit Ver. 4.20.000
* Copyright
* Microsoft, 1999
* All Rights Reserved
*
* Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
*
* PURPOSE: Implementation for TPC-C Tuxedo
class.
* Contact: Charles Levine
(clevine@microsoft.com)
*
* Change history:
* 4.20.000 - updated rev number to
match kit
*/

#include <windows.h>

```

```

#include <process.h>
#include <stdio.h>
#include <stdarg.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <io.h>
#include <assert.h>

#include <tmenv.h>
#include <xa.h>
#include <atmi.h>

#ifdef ICECAP
// for IceCAP profiling
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\trans.h"
//tpckit transaction header contains
definitions of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_tux.h" // interface to Tuxedo libraries

static TPINIT
*tpinf;
static DWORD
TLSIsTpInitedKey;
static CRITICAL_SECTION
TpCriticalSection;

BOOL APIENTRY DllMain(HMODULE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
            DisableThreadLibraryCalls(hModule);

            // create thread local
storage to determine Tuxedo initialization per
thread.
            // it really should be
possible to do this in the DLL_THREAD_ATTACH call,
but
            // Ed says he could not
get it to work.
            // assumption: value
init'd to 0
            TLSIsTpInitedKey =
TlsAlloc();

```



```

        if ((tpinf = (TPINIT
*)tpalloc("TPINIT", NULL, sizeof(TPINIT))) == NULL)
        {
            // int TpRc =
tperrno;
            return FALSE;
        }
        tpinf->flags |=
TPMULTICONTEXTS;

        InitializeCriticalSection(&TpCriticalSection);
        break;
        case DLL_PROCESS_DETACH:
            TlsFree(TLSIsTpInitedKey);
            DeleteCriticalSection(&TpCriticalSection);
            break;
        default:
            /* nothing */;
    }
    return TRUE;
}

static void ThrTpInit()
{
    static int num_tpinit=0;
    int iRc, TpRc;

    // has this thread been initialized? check
    thread local storage
    if(!TlsGetValue(TLSIsTpInitedKey))
    {
        EnterCriticalSection(&TpCriticalSection);
        itoa(++num_tpinit, tpinf-
>cltname, 10);

        iRc = tpinit(tpinf);
        TpRc = tperrno;

        LeaveCriticalSection(&TpCriticalSection);

        if (iRc < 0)
            throw new CTUXERR(
tperrno );

        int value = 1;

        TlsSetValue(TLSIsTpInitedKey, &value);
    }

    // wrapper routine for class constructor
    __declspec(dllexport) CTPCC_TUXEDO*
    CTPCC_TUXEDO_new()
    {

```

```

        return new CTPCC_TUXEDO();
    }

    CTPCC_TUXEDO::CTPCC_TUXEDO()
    {
        // Add initialization of Tuxedo
        Structures
        m_txn = (TUX_DATA *)tpalloc("CARRAY", NULL,
sizeof(TUX_DATA));
        if (m_txn == NULL)
            throw new CTUXERR( tperrno );
    }

    CTPCC_TUXEDO::~CTPCC_TUXEDO()
    {
        // free the data structure allocated with
        tpalloc
        tpfree((char *)m_txn);
    }

    void CTPCC_TUXEDO::NewOrder()
    {
        long ilen, *olen;

        ThrTpInit();

        ilen = sizeof(TUX_DATA);
        olen = &ilen;

        if (tpcall("NEWORDER", (char *)m_txn, ilen,
(char **)&m_txn, (long *)olen, TPSIGRSTRT) == -1)
            throw new CTUXERR( tperrno );

        if ( m_txn->ErrorType != ERR_SUCCESS )
            throw new CTUXERR( m_txn-
>ErrorType, m_txn->error );
    }

    void CTPCC_TUXEDO::Payment()
    {
        long ilen, *olen;

        ThrTpInit();

        ilen = sizeof(TUX_DATA);
        olen = &ilen;

        if (tpcall("PAYMENT", (char *)m_txn, ilen,
(char **)&m_txn, (long *)olen, TPSIGRSTRT) == -1)
            throw new CTUXERR( tperrno );

        if ( m_txn->ErrorType != ERR_SUCCESS )
            throw new CTUXERR( m_txn-
>ErrorType, m_txn->error );
    }

    void CTPCC_TUXEDO::Delivery()
    {
        int iRc;
        long ilen, *olen;

```

```

        // Note: Delivery txn code in the tuxedo
server does not implement logging of the delivery
txn results, so cannot be used as
is to run an auditable TPC-C result. For that
reason, delivery txns should not
be done via tuxedo.
        // The code is included for
completeness.
        m_txn->u.Delivery.exec_status_code =
eDeliveryFailed;
        return;

        // normal path...
        ThrTpInit();

        GetLocalTime(&m_txn-
>u.Delivery.queue_time);

        ilen = sizeof(TUX_DATA);
        olen = &ilen;

        if ((iRc = tpacall("DELIVERY", (char
*)m_txn, ilen, TPNOREPLY)) == -1)
        {
            int TpRc = tperrno;
            m_txn-
>u.Delivery.exec_status_code = eDeliveryFailed;
        }
        else
            m_txn-
>u.Delivery.exec_status_code = eOK;
    }

    void CTPCC_TUXEDO::StockLevel()
    {
        long ilen, *olen;

        ThrTpInit();

        ilen = sizeof(TUX_DATA);
        olen = &ilen;

        if (tpcall("STOCKLEVEL", (char *)m_txn,
ilen, (char **)&m_txn, (long *)olen, TPSIGRSTRT) == -
1)
            throw new CTUXERR( tperrno );

        if ( m_txn->ErrorType != ERR_SUCCESS )
            throw new CTUXERR( m_txn-
>ErrorType, m_txn->error );
    }

    void CTPCC_TUXEDO::OrderStatus()
    {
        long ilen, *olen;

        ThrTpInit();

        ilen = sizeof(TUX_DATA);

```

```

        olen = &ilen;

        if (tpcall("ORDERSTATUS", (char *)m_txn,
ilen, (char **)&m_txn, (long *)olen, TPSIGRSTRT) == -
1)
            throw new CTUXERR( tperno );

        if ( m_txn->ErrorType != ERR_SUCCESS )
            throw new CTUXERR( m_txn-
>ErrorType, m_txn->error );
    }

char *CTUXERR::ErrorText()
{
    if (m_iErrorType == 0)
    {
        if (m_errno == TPEOS)
            sprintf( m_szErrorText,
"Error: TUXEDO error # %d, OS error # %d", m_errno,
m_iError );
        else
            sprintf( m_szErrorText,
"Error: TUXEDO error # %d", m_errno );
    }
    else
        sprintf( m_szErrorText, "Error:
Class %d, error # %d", m_iErrorType, m_iError );
    return m_szErrorText;
};

```

tpcc_tux.h

```

/* FILE: TPC_C_TUX.H
 * Microsoft
 * TPC-C Kit Ver. 4.20.000
 * Copyright
 * Microsoft, 1999
 * All Rights Reserved
 * Version
 * 4.10.000 audited by Richard Gimarc, Performance
 * Metrics, 3/17/99
 * PURPOSE: Header file for TPC-C Tuxedo
 * class implementation.
 * Change history:
 * 4.20.000 - updated rev number to
 * match kit
 */

#pragma once

// need to declare functions for import, unless
// define has already been created
// by the DLL's .cpp module for export.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

```

```

class DllDecl CTPCC_TUXEDO : public CTPCC_BASE
{
private:
    struct TUX_DATA
    {
        int
        ErrorType;
        int
        error;

        union
        {
            NEW_ORDER_DATA      NewOrder;
            PAYMENT_DATA        Payment;
            DELIVERY_DATA       Delivery;
            STOCK_LEVEL_DATA     StockLevel;
            ORDER_STATUS_DATA    OrderStatus;
        } u;
    } *m_txn;

public:
    CTPCC_TUXEDO();
    ~CTPCC_TUXEDO(void);

    inline PNEW_ORDER_DATA
    BuffAddr_NewOrder() { return
&m_txn->u.NewOrder; };
    inline PPAYMENT_DATA
    BuffAddr_Payment() { return
&m_txn->u.Payment; };
    inline PDELIVERY_DATA
    BuffAddr_Delivery() { return
&m_txn->u.Delivery; };
    inline PSTOCK_LEVEL_DATA
    BuffAddr_StockLevel() { return
&m_txn->u.StockLevel; };
    inline PORDER_STATUS_DATA
    BuffAddr_OrderStatus() { return
&m_txn->u.OrderStatus; };

    void NewOrder      ();
    void Payment       ();
    void Delivery      ();
    void StockLevel    ();
    void OrderStatus   ();
};

class CTUXERR : public CBaseErr
{
private:
    // TODO: should use the sz_Msg
    // field of the base class instead
    char m_szErrorText[64];
};

```

```

public:
    // use this interface for genuine
    Tuxedo errors
    CTUXERR( int iErr )
    {
        m_errno = iErr;
        m_iErrorType = 0;
        m_iError =
};

    GetLastError(); // only meaningful if m_errno ==
    TPEOS

    // use this interface to
    // impersonate a non-Tuxedo error type
    CTUXERR( int iErrorType, int
iError )
    {
        m_iErrorType =
iErrorType;
        m_iError = iError;
        m_errno = 0;
    }

    int m_errno;
    int m_iErrorType;
    int m_iError;

    // A CTUXERR class can
    // impersonate another
    // class, which happens if the error
    // was not actually a Tuxedo
    // error, but was simply transmitted back via Tuxedo.
    int ErrorType()
    {
        if (m_iErrorType == 0)
            return
ERR_TYPE_TUXEDO;
        else
            return
m_iErrorType;
    }

    int ErrorNum() {return m_errno;};
    char *ErrorText();
};

// wrapper routine for class constructor
extern "C" __declspec(dlllexport) CTPCC_TUXEDO*
CTPCC_TUXEDO_new();

typedef CTPCC_TUXEDO* (TYPE_CTPCC_TUXEDO)();

tpcc_type.h
/* Generated by IDL compiler version DEC DCE V2.0.0-6
*/
#ifndef tpcc_types_v1_0_included
#define tpcc_types_v1_0_included
#endif
#include IDLBASE_H
#include <dce\idlbase.h>
#endif

```

```

#ifndef __cplusplus
extern "C" {
#endif

#ifndef nbase_v0_0_included
#include "dce\nbase.h"
#endif
#define NAME_LENGTH (32)
#define NEWO_INTERFACE (1)
#define PAYMENT_INTERFACE (2)
#define ORDER_STAT_INTERFACE (4)
#define DELIVERY_INTERFACE (8)
#define STOCK_INTERFACE (16)
#define ONLINE_INTERFACES (23)
#define ALL_INTERFACE (65535)
#define NEWO_TRANS (1)
#define PAYMENT_TRANS (2)
#define ORDER_STAT_TRANS (3)
#define DELIVERY_TRANS (4)
#define STOCK_TRANS (5)
#define MAX_TRAN_TYPE (5)
#define TPCC_SUCCESS (0)
#define TRPC_ERROR (1)
#define INVALID_NEWO (100)
typedef struct {
    idl_long_int sec;
    idl_long_int usec;
} time_type;
typedef struct {
    idl_short_int returncode;
    idl_short_int stats;
    time_type srv_start;
    time_type srv_end;
    time_type clnt_start;
    time_type clnt_end;
} data_header;
typedef struct {
    idl_long_int first_wh;
    idl_long_int last_wh;
    idl_long_int server_id;
} dbInfo_data_t;

#endif
}

#endif

```

trans.h

```

/* FILE: TRANS.H Microsoft
 * Copyright
 * TPC-C Kit Ver. 4.42.000
 * Microsoft, 2002
 * All Rights Reserved
 */

```

```

* Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
* PURPOSE: Header file for TPC-C structure
templates.
* Change history:
* 4.42.000 - changed w_id fields
from short to long to support >32K warehouses
* 4.20.000 - updated rev number to
match kit
*/
#pragma once

// String length constants
#define SERVER_NAME_LEN 20
#define DATABASE_NAME_LEN 20
#define USER_NAME_LEN 20
#define PASSWORD_LEN 20
#define TABLE_NAME_LEN 20
#define I_DATA_LEN 50
#define I_NAME_LEN 24
#define BRAND_LEN 1
#define LAST_NAME_LEN 16
#define W_NAME_LEN 10
#define ADDRESS_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9
#define S_DIST_LEN 24
#define S_DATA_LEN 50
#define D_NAME_LEN 10
#define FIRST_NAME_LEN 16
#define MIDDLE_NAME_LEN 2
#define PHONE_LEN 16
#define DATETIME_LEN 30
#define CREDIT_LEN 2
#define C_DATA_LEN 250
#define H_DATA_LEN 24
#define DIST_INFO_LEN 24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN 25
#define OL_DIST_INFO_LEN 24

// TIMESTAMP_STRUCT is provided by the ODBC header
file sqltypes.h, but is not available
// when compiling with dblib, so redefined here.
Note: we are using the symbol "_SQLTYPES"
// (declared in sqltypes.h) as a way to determine if
TIMESTAMP_STRUCT has been declared.
#ifndef _SQLTYPES
typedef struct
{
    short
    year;
    unsigned short /*
SQLSMALLINT */ month;
    unsigned short /*
SQLSMALLINT */ day;

```

```

    unsigned short /*
SQLSMALLINT */ hour;
    unsigned short /*
SQLSMALLINT */ minute;
    unsigned short /*
SQLSMALLINT */ second;
    unsigned long /*
SQLINTEGER */ fraction;
} TIMESTAMP_STRUCT;
#endif

// possible values for exec_status_code after
transaction completes
enum EXEC_STATUS
{
    eOK, // 0
    "Transaction committed."
    eInvalidItem, // 1 "Item number
is not valid."
    eDeliveryFailed // 2 "Delivery
Post Failed."
};

// transaction structures
typedef struct
{
    // input params
    long
    ol_supply_w_id;
    long
    ol_i_id;
    short
    ol_quantity;

    // output params
    char
    ol_i_name[I_NAME_LEN+1];
    char
    ol_brand_generic[BRAND_LEN+1];
    double
    ol_i_price;
    double
    ol_amount;
    short
    ol_stock;
} OL_NEW_ORDER_DATA;

typedef struct
{
    // input params
    long
    short
    long
    short
    w_id;
    d_id;
    c_id;
    o_ol_cnt;

    // output params
    EXEC_STATUS
    exec_status_code;
    char
    c_last[LAST_NAME_LEN+1];
    char
    c_credit[CREDIT_LEN+1];

```

```

double          c_discount;
double          w_tax;
double          d_tax;
long            o_id;
short
o_commit_flag;
TIMESTAMP_STRUCT o_entry_d;
short          o_all_local;
double        total_amount;
OL_NEW_ORDER_DATA
OL[MAX_OL_NEW_ORDER_ITEMS];
} NEW_ORDER_DATA, *PNEW_ORDER_DATA;

typedef struct
{
    // input params
    long
w_id;
short
d_id;
long
c_id;
short
c_d_id;
long
c_w_id;
double
h_amount;
char
c_last[LAST_NAME_LEN+1];

    // output params
EXEC_STATUS
exec_status_code;
TIMESTAMP_STRUCT h_date;
char
w_street_1[ADDRESS_LEN+1];
char
w_street_2[ADDRESS_LEN+1];
char
w_city[ADDRESS_LEN+1];
char
w_state[STATE_LEN+1];
char
w_zip[ZIP_LEN+1];
char
d_street_1[ADDRESS_LEN+1];
char
d_street_2[ADDRESS_LEN+1];
char
d_city[ADDRESS_LEN+1];
char
d_state[STATE_LEN+1];
char
d_zip[ZIP_LEN+1];
char
c_first[FIRST_NAME_LEN+1];
char
c_middle[MIDDLE_NAME_LEN + 1];
char
c_street_1[ADDRESS_LEN+1];

```

```

char
c_street_2[ADDRESS_LEN+1];
char
c_city[ADDRESS_LEN+1];
char
c_state[STATE_LEN+1];
char
c_zip[ZIP_LEN+1];
char
c_phone[PHONE_LEN+1];
TIMESTAMP_STRUCT c_since;
char
c_credit[CREDIT_LEN+1];
double
c_credit_lim;
double
c_discount;
double
c_balance;
char
c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

typedef struct
{
    long
ol_i_id;
long
ol_supply_w_id;
short
ol_quantity;
double
ol_amount;
TIMESTAMP_STRUCT ol_delivery_d;
} OL_ORDER_STATUS_DATA;

typedef struct
{
    // input params
    long
w_id;
short
d_id;
long
c_id;
char
c_last[LAST_NAME_LEN+1];

    // output params
EXEC_STATUS
exec_status_code;
char
c_first[FIRST_NAME_LEN+1];
char
c_middle[MIDDLE_NAME_LEN+1];
double
c_balance;
long
o_id;
TIMESTAMP_STRUCT o_entry_d;
short
o_carrier_id;
OL_ORDER_STATUS_DATA
OL[MAX_OL_ORDER_STATUS_ITEMS];
short
o_ol_cnt;
} ORDER_STATUS_DATA, *PORDER_STATUS_DATA;

```

```

typedef struct
{
    // input params
    long
short
w_id;
o_carrier_id;

    // output params
EXEC_STATUS
exec_status_code;
SYSTEMTIME
queue_time;
long
o_id[10];
// id's of delivered
orders for districts 1 to 10
} DELIVERY_DATA, *PDELIVERY_DATA;

//This structure is used for posting delivery
transactions and for writing them to the delivery
server.
typedef struct _DELIVERY_TRANSACTION
{
    SYSTEMTIME
queue;
//time delivery transaction queued
long
w_id;
//delivery warehouse
short
o_carrier_id;
//carrier id
} DELIVERY_TRANSACTION;

typedef struct
{
    // input params
    long
w_id;
short
d_id;
short
threshold;

    // output params
EXEC_STATUS
exec_status_code;
long
low_stock;
} STOCK_LEVEL_DATA, *PSTOCK_LEVEL_DATA;

```

tuxapp.cpp

```

/* FILE: TUXAPP.CPP
* Microsoft
TPC-C Kit Ver. 4.20.000
* Copyright
Microsoft, 1999
* All Rights Reserved
* Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
*
* PURPOSE: Implementation for TPC-C Tuxedo
server.
* Contact: Charles Levine
(clevine@microsoft.com)

```

```

*
* Change history:
* 4.20.000 - updated rev number to
match kit
*/

#include <windows.h>
#include <process.h>
#include <tchar.h>
#include <stdio.h>
#include <stdarg.h>
#include <iostream.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <io.h>
#include <assert.h>

#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>

#include <tmenv.h>
#include <xa.h>
#include <atmi.h>

#include "..\..\common\src\trans.h"
//tpckit transaction header contains
definitions of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\ReadRegistry.h"
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"
// DBLIB implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"
// ODBC implementation of TPC-C txns
#include "tuxapp.h"

char
    szMyComputerName[MAX_COMPUTERNAME_LENGTH+1]
;

// configuration settings from registry
TPCCREGISTRYDATA    Reg;

CTPCC_BASE          *pTxn = NULL;

#include "..\..\common\src\ReadRegistry.cpp"

/* FUNCTION: tpsvrinit ( int argc, char *argv[] )
*
* PURPOSE:          Initialize the Server to Database
connection.
*
* RETURNS:          int          0
                    Success
*
                    Failure          -1
*/

```

```

*/
int tpsvrinit ( int argc, char *argv[] )
{
    try
    {
        DWORD dwSize =
MAX_COMPUTERNAME_LENGTH+1;
        GetComputerName(szMyComputerName,
&dwSize);

        szMyComputerName[dwSize] = 0;

        if ( ReadTPCCRegistrySettings(
&Reg ) )
            throw new CTUXAPP_ERR(
ERR_MISSING_REGISTRY_ENTRIES );

        GetParameters(argc, argv);

        switch (Reg.eDB_Protocol)
        {
            case ODBC:
                pTxn = new CTPCC_ODBC(
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );
                break;
            case DBLIB:
                pTxn = new CTPCC_DBLIB(
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );
                break;
        }
    }
    catch (CBaseErr *e)
    {
        WriteMessageToEventLog(e-
>ErrorText());
        delete e;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception."));
    }
    return 0;
}

/* FUNCTION: tpsvrdone ( void )
*
*
*/
void tpsvrdone ( void )
{
    delete pTxn;
    pTxn = NULL;
}

/* FUNCTION: BOOL GetParameters(int argc, char
*argv[])

```

```

*
* PURPOSE:          This function parses the command
line passed in to the delivery executable,
initializing
*
*
* and filling in global
variable parameters.
*
* ARGUMENTS:        int          argc
                    number of command line arguments passed to
delivery
*
                    char
*argv[]          array of command line argument
pointers
*
*/

static void GetParameters(int argc, char *argv[])
{
    // advance through args until "--" is found
for(int j=0; j<argc; j++)
    {
        if (strcmp(argv[j],"--") == 0)
            break;
    }

for(int i=j+1; i<argc; i++)
    {
        if ( argv[i][0] == '-' ||
argv[i][0] == '/' )
            {
                switch(argv[i][1])
                {
                    case 'S':
                        strcpy(Reg.szDbServer, argv[i+2]);
                        break;
                    case 'D':
                        strcpy(Reg.szDbName, argv[i+2]);
                        break;
                    case 'P':
                        strcpy(Reg.szDbPassword, argv[i+2]);
                        break;
                    case 'U':
                        strcpy(Reg.szDbUser, argv[i+2]);
                        break;
                    default:
                        cout << "Microsoft TPC-C Kit" << endl;
                        cout << "Tuxedo Server" << endl << endl;
                        cout << "Usage:" << endl;
                        cout << "    tuxapp [<tuxedo-args>] - -

```

```

S<sql-server> [-D<database>] [-U<user>] [-
P<password>]" << endl << endl;

        cout << "All parameters default to values
in registry." << endl;

        throw new CTUXAPP_ERR( ERR_BAD_SYNTAX );
    }
}

static void WriteMessageToEventLog(LPTSTR lpszMsg)
{
    TCHAR    szMsg[256];
    HANDLE   hEventSource;
    LPTSTR   lpszStrings[2];

    // Use event logging to log the error.
    //
    hEventSource = RegisterEventSource(NULL,
TEXT("TUXAPP.EXE"));

    _stprintf(szMsg, TEXT("Error in TUXAPP.EXE: "));
    lpszStrings[0] = szMsg;
    lpszStrings[1] = lpszMsg;

    if (hEventSource != NULL)
    {
        ReportEvent(hEventSource, // handle of event
source
            EVENTLOG_ERROR_TYPE, // event type
            0, // event category
            0, // event ID
            NULL, // current user's
SID
            2, // strings in
lpszStrings
            0, // no bytes of raw
data
            (LPCTSTR *)lpszStrings, // array of
error strings
            NULL); // no raw data

        (VOID) DeregisterEventSource(hEventSource);
    }
}

void NEWORDER( TPSVCINFO *rqst )
{
    PNEW_ORDER_DATA    pNewOrder;
    TUX_DATA            *pData;
    const int          iSize = sizeof(pData-
>u.NewOrder);

    try
    {
        pData = (TUX_DATA*)rqst->data;
        pData->retval = ERR_SUCCESS;
        pData->error = 0;

```

```

        pNewOrder = pTxn-
>BuffAddr_NewOrder();
        assert( rqst->len ==
sizeof(TUX_DATA) );
        memcpy(pNewOrder, &pData-
>u.NewOrder, iSize );

        pTxn->NewOrder();
        memcpy( &pData->u.NewOrder,
pNewOrder, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
    }
    catch (CBaseErr *e)
    {
        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        memcpy( &pData->u.NewOrder,
pNewOrder, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
        delete e;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception."));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        memcpy( &pData->u.NewOrder,
pNewOrder, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
    }
}

void PAYMENT( TPSVCINFO *rqst )
{
    PPAYMENT_DATA    pPayment;
    TUX_DATA          *pData;
    const int          iSize = sizeof(pData-
>u.Payment);

    try
    {
        pData = (TUX_DATA*)rqst->data;
        pData->retval = ERR_SUCCESS;
        pData->error = 0;

        pPayment = pTxn-
>BuffAddr_Payment();
        assert( rqst->len ==
sizeof(TUX_DATA) );
        memcpy(pPayment, &pData-
>u.Payment, iSize );

        pTxn->Payment();
        memcpy( &pData->u.Payment,
pPayment, iSize );

```

```

        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
    }
    catch (CBaseErr *e)
    {
        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        memcpy( &pData->u.Payment,
pPayment, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
        delete e;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception."));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        memcpy( &pData->u.Payment,
pPayment, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
    }
}

// Note: Delivery txn code below does not implement
logging of the delivery
// txn results, so cannot be used as is to run
an auditable TPC-C result.
// The code is included for completeness.
void DELIVERY( TPSVCINFO *rqst )
{
    PDELIVERY_DATA    pDelivery;
    TUX_DATA          *pData;
    const int          iSize = sizeof(pData-
>u.Delivery);

    try
    {
        pData = (TUX_DATA*)rqst->data;
        pData->retval = ERR_SUCCESS;
        pData->error = 0;

        pDelivery = pTxn-
>BuffAddr_Delivery();
        assert( rqst->len ==
sizeof(TUX_DATA) );
        memcpy(pDelivery, &pData-
>u.Delivery, iSize );

        pTxn->Delivery();

        memcpy( &pData->u.Delivery,
pDelivery, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
    }
    catch (CBaseErr *e)
    {

```

```

        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        memcpy( &pData->u.Delivery,
pDelivery, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
        delete e;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception."));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        memcpy( &pData->u.Delivery,
pDelivery, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
    }
}

void STOCKLEVEL( TPSVCINFO *rqst )
{
    PSTOCK_LEVEL_DATA  pStockLevel;
    TUX_DATA            *pData;
    const int           iSize =
sizeof(pData->u.StockLevel);

    try
    {
        pData = (TUX_DATA*)rqst->data;
        pData->retval = ERR_SUCCESS;
        pData->error = 0;

        pStockLevel = pTxn-
>BuffAddr_StockLevel();
        assert( rqst->len ==
sizeof(TUX_DATA) );
        memcpy(pStockLevel, &pData-
>u.StockLevel, iSize );

        pTxn->StockLevel();
        memcpy( &pData->u.StockLevel,
pStockLevel, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
    }
    catch (CBaseErr *e)
    {
        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        memcpy( &pData->u.StockLevel,
pStockLevel, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
        delete e;
    }
    catch (...)
    {

```

```

        WriteMessageToEventLog(TEXT("Unhandled
exception."));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        memcpy( &pData->u.StockLevel,
pStockLevel, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
    }
}

void ORDERSTATUS( TPSVCINFO *rqst )
{
    PORDER_STATUS_DATA pOrderStatus;
    TUX_DATA            *pData;
    const int iSize = sizeof(pData-
>u.OrderStatus);

    try
    {
        pData = (TUX_DATA*)rqst->data;
        pData->retval = ERR_SUCCESS;
        pData->error = 0;

        pOrderStatus = pTxn-
>BuffAddr_OrderStatus();
        assert( rqst->len ==
sizeof(TUX_DATA) );
        memcpy(pOrderStatus, &pData-
>u.OrderStatus, iSize );

        pTxn->OrderStatus();
        memcpy( &pData->u.OrderStatus,
pOrderStatus, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
    }
    catch (CBaseErr *e)
    {
        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        memcpy( &pData->u.OrderStatus,
pOrderStatus, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
        delete e;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception."));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        memcpy( &pData->u.OrderStatus,
pOrderStatus, iSize );
        tpreturn( TPSUCCESS, 0, rqst-
>data, sizeof(TUX_DATA), 0);
    }
}

```

```

/* FUNCTION: CTUXAPP_ERR::ErrorText
 *
 */
char* CTUXAPP_ERR::ErrorText(void)
{
    int i;

    static SERRORMSG errorMsgs[] =
    {
        { ERR_MISSING_REGISTRY_ENTRIES,
"Required entries missing from registry."
},
        { ERR_BAD_SYNTAX,
"Syntax error in input
parameters."
},
        { ERR_UNKNOWN_DB_PROTOCOL,
"Unknown database protocol specified in
registry."
},
        { 0, ""
}
};

    static char szNotFound[] = "Unknown error
number.";

    for(i=0; errorMsgs[i].szMsg[0]; i++)
    {
        if ( m_Error ==
errorMsgs[i].iError )
            break;
    }
    if ( !errorMsgs[i].szMsg[0] )
        return szNotFound;
    else
        return errorMsgs[i].szMsg;
}

```

tuxapp.h

```

/* FILE: TUXAPP.H
 *
 * Microsoft
TPC-C Kit Ver. 4.20.000
 * Copyright
Microsoft, 1999
 * All Rights Reserved
 *
 * Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
 *
 * PURPOSE: Header file for TPC-C Tuxedo
server.

```

```

*
* Change history:
* 4.20.000 - updated rev number to
match kit
*/

enum TUXERROR
{
    ERR_MISSING_REGISTRY_ENTRIES = 1,
    ERR_BAD_SYNTAX,
    ERR_UNKNOWN_DB_PROTOCOL
};

class CTUXAPP_ERR : public CBaseErr
{
public:
    TUXERROR m_Error;

    CTUXAPP_ERR(TUXERROR Err) {
        m_Error = Err; };
    ~CTUXAPP_ERR() {};

    int ErrorType() {return
ERR_TYPE_TUXEDO;};
    int ErrorNum() {return m_Error;};
    char *ErrorText();
};

struct TUX_DATA
{
    int
    retval;
    int
    error;

    union
    {
        NEW_ORDER_DATA
    NewOrder;
        PAYMENT_DATA
    Payment;
        DELIVERY_DATA
    Delivery;
        STOCK_LEVEL_DATA    StockLevel;
        ORDER_STATUS_DATA    OrderStatus;
    } u;
};

static void GetParameters(int argc, char *argv[]);
static void WriteMessageToEventLog(LPTSTR lpszMsg);

#if defined(__cplusplus)
extern "C" {
#endif

void NEWORDER( TPSVCINFO *rqst );
void PAYMENT( TPSVCINFO *rqst );
void DELIVERY( TPSVCINFO *rqst );
void STOCKLEVEL( TPSVCINFO *rqst );
void ORDERSTATUS( TPSVCINFO *rqst );

```

```

#if defined(__cplusplus)
}
#endif

tuxmain.c


---


/* FILE: TUXMAIN.C
* Microsoft
TPC-C Kit Ver. 4.20.000
* Copyright
Microsoft, 1999
* All Rights Reserved
* Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
*
* PURPOSE: Implementation for TPC-C Tuxedo
server.
* Contact: Charles Levine
(levine@microsoft.com)
* Change history:
* 4.20.000 - updated rev number to
match kit
*/

#include <stdio.h>
#include <xa.h>
#include <atml.h>

#if defined(__cplusplus)
extern "C" {
#endif
extern int _tmrunserver_((int));
extern void DELIVERY_((TPSVCINFO *));
extern void NEWORDER_((TPSVCINFO *));
extern void ORDERSTATUS_((TPSVCINFO *));
extern void PAYMENT_((TPSVCINFO *));
extern void STOCKLEVEL_((TPSVCINFO *));
#if defined(__cplusplus)
}
#endif

static struct tmdsptchtbl_t_tmdsptchtbl[] = {
    { "DELIVERY", "DELIVERY", (void (*)
_((TPSVCINFO *)) DELIVERY, 0, 0 },
    { "NEWORDER", "NEWORDER", (void (*)
_((TPSVCINFO *)) NEWORDER, 1, 0 },
    { "ORDERSTATUS", "ORDERSTATUS", (void (*)
_((TPSVCINFO *)) ORDERSTATUS, 2, 0 },
    { "PAYMENT", "PAYMENT", (void (*)
_((TPSVCINFO *)) PAYMENT, 3, 0 },
    { "STOCKLEVEL", "STOCKLEVEL", (void (*)
_((TPSVCINFO *)) STOCKLEVEL, 4, 0 },
    { NULL, NULL, NULL, 0, 0 }
};

```

```

#ifndef _TMDLLIMPORT
#define _TMDLLIMPORT
#endif

_TMDLLIMPORT extern struct xa_switch_t tnull_switch;

struct tmsvrgs_t tmsvrgs = {
    NULL,
    &tmdsptchtbl[0],
    0,
    tpsvrinit,
    tpsvrdone,
    _tmrunserver, /* PRIVATE */
    NULL, /* RESERVED
*/
    NULL, /* RESERVED
*/
    NULL, /* RESERVED
*/
    NULL /* RESERVED
*/
};

struct tmsvrgs_t *
#ifdef _TMPROTOTYPES
_tmgetsvrgs(void)
#else
_tmgetsvrgs()
#endif
{
    tmsvrgs.xa_switch = &tnull_switch;
    return(&tmsvrgs);
}

int
#ifdef _TMPROTOTYPES
main(int argc, char **argv)
#else
main(argc,argv)
int argc;
char **argv;
#endif
{
#ifdef TMMAINEXIT
#include "mainexit.h"
#endif

    return( _tmstartserver( argc, argv,
_tmgetsvrgs()));
}

txn_base.h


---


/* FILE: TXN_BASE.H
* Microsoft
TPC-C Kit Ver. 4.20.000
* Copyright
Microsoft, 1999
* All Rights Reserved

```



```

*
*                                     Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
*
*   PURPOSE: Header file for TPC-C txn class
implementation.
*
*   Change history:
*   4.20.000 - updated rev number to
match kit
*/

#pragma once

// need to declare functions for import, unless
define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl __declspec( dllimport )
#endif

class DllDecl CTPCC_BASE
{
public:
    CTPCC_BASE(void) {};
    virtual ~CTPCC_BASE(void) {};

    virtual PNEW_ORDER_DATA
BuffAddr_NewOrder() = 0;
    virtual PPAYMENT_DATA
BuffAddr_Payment() = 0;
    virtual PDELIVERY_DATA
BuffAddr_Delivery() = 0;
    virtual PSTOCK_LEVEL_DATA
BuffAddr_StockLevel() = 0;
    virtual PORDER_STATUS_DATA
BuffAddr_OrderStatus() = 0;

    virtual void NewOrder
() = 0;
    virtual void Payment
() = 0;
    virtual void Delivery
() = 0;
    virtual void StockLevel
() = 0;
    virtual void OrderStatus
() = 0;
};

```

tpcc_oledb.cpp

```

/* FILE: TPCC_OLEDB.CPP
* Microsoft
TPC-C Kit Ver. 4.42.000
* Copyright
Microsoft, 2004
* Written by
Sergey Vasilevskiy

```

```

* All Rights Reserved
*
*
*   PURPOSE: Implements OLEDB calls for TPC-C
txns.
*   Contact: Charles Levine
(clevine@microsoft.com)
*
*/

#include <windows.h>
#include <stdio.h>
#include <assert.h>
#include <stddef.h>

#define DBINITCONSTANTS
#include <oledb.h>
// #include <sqloledb.h> // Use MDAC
#include <sqlncli.h> // Use SNAC
#include <oledberr.h>

#ifdef ICECAP
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_oledb.h"

#ifdef SQL_MAX_MESSAGE_LENGTH
#define SQL_MAX_MESSAGE_LENGTH 512
#endif

// version string; must match return value from
tpcc_version stored proc
const char sVersion[] = "4.20.000";

const iMaxRetries = 10; // how many
retries on deadlock

const int iErrOleDbProvider = 7312;
const char sErrTimeoutExpired[] = "Timeout expired";

// this needs to be the same as the max length of
machine/database/user/password in Benchcraft
(engstut.h)
const static int iMaxNameLen = 32;

BOOL WINAPI DllMain(HMODULE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:

```

```

        DisableThreadLibraryCalls(hModule);

        break;

        case DLL_PROCESS_DETACH:
            break;

        default:
            /* nothing */;
    }
    return TRUE;
}

/* FUNCTION: CTPCC_OLEDB_ERR::ErrorText
*
*/
char* CTPCC_OLEDB_ERR::ErrorText(void)
{
    int i;

    static SERRORMSG errorMsgs[] =
    {
        { ERR_WRONG_SP_VERSION,
"Wrong version of stored procs on database
server" },
        { ERR_INVALID_CUST,
"Invalid Customer id,name." },
        { ERR_NO_SUCH_ORDER,
"No orders found for customer." },
        { ERR_RETRIED_TRANS,
"Retries before transaction succeeded." },
        { 0, "" }
    };

    static char szNotFound[] = "Unknown error
number.";

    for(i=0; errorMsgs[i].szMsg[0]; i++)
    {
        if ( m_errno ==
errorMsgs[i].iError )
            break;
    }
    if ( !errorMsgs[i].szMsg[0] )
        return szNotFound;
    else
        return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_OLEDB* CTPCC_OLEDB_new(

```

```

        LPCWSTR szServer,          // name of
SQL server: LPCWSTR szUser,          //
user name for login: LPCWSTR szPassword, // password
for login: LPCWSTR szHost,          //
not used: LPCWSTR szDatabase,       // name of
database to use: LPCWSTR szSPPrefix ) //
prefix to append to the stored procedure names
{
    return new CTPCC_OLEDB( szServer, szUser,
szPassword, szHost, szDatabase, szSPPrefix );
}

CTPCC_OLEDB::CTPCC_OLEDB (
    LPCWSTR szServer,
    // name of SQL server
    LPCWSTR szUser,
    // user name for login
    LPCWSTR szPassword,
    // password for login
    LPCWSTR szHost,
    // not used
    LPCWSTR szDatabase,
    // name of database to use
    LPCWSTR szSPPrefix
    // prefix to append to the stored procedure
names
)
: m_pIMalloc(NULL)
{
    int
    iRc;
    int
    i;
    HRESULT hr;

    IDBInitialize*
    pIDBInitialize = NULL; //
data source interface
    IDBProperties*
    pIDBProperties = NULL;
    ICommandText*
    pICommandText;
    // SQL command without parameters
    wchar_t
    szwServer[iMaxNameLen]; //
Unicode string used to convert to BSTR
    wchar_t
    szwDatabase[iMaxNameLen]; // Unicode
string used to convert to BSTR
    wchar_t
    szwUser[iMaxNameLen]; //
Unicode string used to convert to BSTR
    wchar_t
    szwPassword[iMaxNameLen]; // Unicode
string used to convert to BSTR

```

```

        // Copy stored procedures prefix
        wcsncpy(m_szwSPPrefix, szSPPrefix,
sizeof(m_szwSPPrefix)/sizeof(m_szwSPPrefix[0]));

        // Convert single byte ANSI strings to
Unicode (for later conversion to BSTR)
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szServer, (int)strlen(szServer)+1,
szwServer, iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szDatabase,
(int)strlen(szDatabase)+1, szwDatabase, iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szUser, (int)strlen(szUser)+1,
szwUser, iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szPassword,
(int)strlen(szPassword)+1, szwPassword, iMaxNameLen);

        // Initialize COM library to be able to use
OLE-DB interfaces
        CoInitialize(NULL);

        // Initialization - create SQLOLEDB
component
        //hr = CoCreateInstance(CLSID_SQLOLEDB, //
GUID of SQLOLEDB component
        // Compile for SNAC
        hr = CoCreateInstance(CLSID_SQLNCLI, //
GUID of SQLNCLI component
        NULL,
        // not defining an aggregate
component, so NULL
        CLSCTX_INPROC_SERVER, //
run the component in our process
        IID_IDBInitialize,
        (void **) &pIDBInitialize);

        /*
        Initialize the property values needed
        to establish the connection.
        */
        for(i = 0; i < 4; i++)
            VariantInit(&m_InitProperties[i].vValue);
        //Server name.
        m_InitProperties[0].dwPropertyID =
DBPROP_INIT_DATASOURCE;
        m_InitProperties[0].vValue.vt = VT_BSTR;
        m_InitProperties[0].vValue.bstrVal=
SysAllocString(szwServer);
        m_InitProperties[0].dwOptions =
DBPROPOPTIONS_REQUIRED;
        m_InitProperties[0].colid = DB_NULLID;
        //Database.
        m_InitProperties[1].dwPropertyID =
DBPROP_INIT_CATALOG;
        m_InitProperties[1].vValue.vt = VT_BSTR;
        m_InitProperties[1].vValue.bstrVal=
SysAllocString(szwDatabase);
        m_InitProperties[1].dwOptions =
DBPROPOPTIONS_REQUIRED;

```

```

        m_InitProperties[1].colid = DB_NULLID;
        //Username (login).
        m_InitProperties[2].dwPropertyID =
DBPROP_AUTH_USERID;
        m_InitProperties[2].vValue.vt = VT_BSTR;
        m_InitProperties[2].vValue.bstrVal=
SysAllocString(szwUser);
        m_InitProperties[2].dwOptions =
DBPROPOPTIONS_REQUIRED;
        m_InitProperties[2].colid = DB_NULLID;
        //Password.
        m_InitProperties[3].dwPropertyID =
DBPROP_AUTH_PASSWORD;
        m_InitProperties[3].vValue.vt = VT_BSTR;
        m_InitProperties[3].vValue.bstrVal=
SysAllocString(szwPassword);
        m_InitProperties[3].dwOptions =
DBPROPOPTIONS_REQUIRED;
        m_InitProperties[3].colid = DB_NULLID;
        /*
        Construct the DBPROPSET
        structure(m_rgInitPropSet). The
        DBPROPSET structure is used to pass an array of
        DBPROP
        structures (m_InitProperties) to the
        SetProperties method.
        */
        m_rgInitPropSet.guidPropertySet =
DBPROPSET_DBINIT;
        m_rgInitPropSet.cProperties = 4;
        m_rgInitPropSet.rgProperties =
m_InitProperties;
        //Set initialization properties.
        if (FAILED(hr = pIDBInitialize-
>QueryInterface(IID_IDBProperties,
        (void **) &pIDBProperties)))
        {
            ThrowError(pIDBInitialize,
COLEDBERR::eQueryInterface, "CTPCC_OLEDB()");
        }

        hr = pIDBProperties->SetProperties(1,
&m_rgInitPropSet);

        pIDBProperties->Release();
        //Now establish the connection to the data
source.
        hr = pIDBInitialize->Initialize();

        // Free BSTR property strings
        for(i = 0; i < 4; i++)
        {
            SysFreeString(m_InitProperties[i].vValue.bstrVal);
        }

```

```

        hr = pIDBInitialize-
>QueryInterface(IID_IDBCreateSession, (void
**) &m_pIDBCreateSession);

        // Releasing this has no effect on the SQL
Server connection
        // of the data source object because of the
reference maintained by
        // m_pIDBCreateSession.
        pIDBInitialize->Release();
        pIDBInitialize = NULL;

        hr = m_pIDBCreateSession-
>CreateSession(NULL, IID_IDBCreateCommand, (IUnknown
**) &m_pIDBCreateCommand);
        if (FAILED(hr))
        {
            ThrowError(m_pIDBCreateSession,
COLEDBERR::eCreateSession, "CTPCC_OLEDB()");
        }

        hr = m_pIDBCreateCommand-
>CreateCommand(NULL, IID ICommandText, (IUnknown
**) &pICommandText);
        if (FAILED(hr))
        {
            ThrowError(m_pIDBCreateCommand,
COLEDBERR::eCreateCommand, "CTPCC_OLEDB()");
        }

        hr = pICommandText-
>SetCommandText(DBGUID_SQL, L"set nocount on set
XACT_ABORT ON");
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eSetCommandText, "CTPCC_OLEDB()");
        }

        hr = pICommandText->Execute(NULL, IID_NULL,
NULL, NULL, NULL);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eExecute, "CTPCC_OLEDB()");
        }

        pICommandText->Release();

        // verify that version of stored procs on
server is correct
        CheckSPVersion();

        // Get IMalloc interface
        hr = CoGetMalloc(1, (LPMAALLOC
*) &m_pIMalloc);

        // Bind parameters for each of the
transactions
        InitNewOrderParams();
        InitPaymentParams();

```

```

        InitOrderStatusParams();
        InitDeliveryParams();
        InitStockLevelParams();
    }

CTPCC_OLEDB::~CTPCC_OLEDB( void )
{
    if (m_pIMalloc != NULL)
    {
        m_pIMalloc->Release();
    }
    m_pIPaymentCommand->Release();
    m_pIDBCreateCommand->Release();
    m_pIDBCreateSession->Release();

    CoUninitialize(); // uninitialized COM
library
}

/*
* Check stored procedures version on the
server.
*/
void CTPCC_OLEDB::CheckSPVersion()
{
    HRESULT                hr;
    char                   db_sp_version[10];
    ICommandText*         pICommandText;
    IAccessor*            pIAccessor;
    IRowset*              pRowset;
    const ULONG           nOutputParams
= 1;
    // output 1st result set columns
    HACCESSOR             hTpccVersionOutputAccessor;
    // Structure to bind in accessor
    DBBINDING            acOutputDBBinding[nOutputParams];
    DBBINDSTATUS          acOutputDBBindStatus[nOutputParams];
    LONG                  cRows = 1;
    // number of rows returned in the rowset
    ULONG                 cRowsObtained;
    HROW                  rghRow;
    //returned row handles
    HROW*                 prghRow =
&rghRow;

    hr = m_pIDBCreateCommand-
>CreateCommand(NULL, IID ICommandText, (IUnknown
**) &pICommandText);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateCommand,
COLEDBERR::eCreateCommand, "CheckSPVersion()");
    }

    hr = pICommandText-
>SetCommandText(DBGUID_SQL, L"{call tpcc_version}");

```

```

        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eSetCommandText, "CheckSPVersion()");
        }

        hr = pICommandText-
>QueryInterface(IID IAccessor, (void **) &pIAccessor);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eQueryInterface, "CheckSPVersion()");
        }

        // Now fill the binding information for
result set 1 output columns
        InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

        // Binding for a rowset
        SetBinding(&acOutputDBBinding[0], 0,
sizeof(db_sp_version), DBTYPE_STR);

        hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA,
nOutputParams,
acOutputDBBinding,
sizeof(db_sp_version),
&hTpccVersionOutputAccessor,
acOutputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "CheckSPVersion()");
        }

        hr = pICommandText->Execute(NULL,
IID_IRowset, NULL, NULL, (IUnknown **) &pRowset);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eExecute, "CheckSPVersion()");
        }

        // Fetch the result row handle(s)
        hr = pRowset->GetNextRows(DB_NULL_HCHAPTER,
0, cRows, &cRowsObtained, &prghRow);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eGetNextRows, "CheckSPVersion()");
        }

        // Fetch the actual row data by handle
        hr = pRowset->GetData(rghRow,
hTpccVersionOutputAccessor, &db_sp_version);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eGetData, "CheckSPVersion()");
        }

```

```

    }

    // Release row(s)
    hr = pRowset->Release();

    piCommandText->Release();

    // Check the retrieved version
    if (strcmp(db_sp_version,sVersion))
        throw new

CTPCC_OLEDB_ERR(
CTPCC_OLEDB_ERR::ERR_WRONG_SP_VERSION );
}

void CTPCC_OLEDB::ThrowError( IUnknown*
pObjectWithError, COLEDBERR::ACTION eAction, LPCTSTR
szLocation)
{
    HRESULT
    hr;
    //char
    szState[6];
    char
    szMsg[SQL_MAX_MESSAGE_LENGTH];
    char
    szTmp[6*SQL_MAX_MESSAGE_LENGTH];
    COLEDBERR
    *pOLEDBErr;
not allocated until needed (maybe never)
    int
    iLen;
    // Interfaces
    IErrorInfo*          piErrorInfoAll
= NULL;
    IErrorInfo*          piErrorInfoRecord
= NULL;
    IErrorRecords*      piErrorRecords
= NULL;
    ISupportErrorInfo*  piSupportErrorInfo
= NULL;
    ISQLServerErrorInfo*
piSQLServerErrorInfo = NULL;
    ISQLErrorInfo*
piSQLErrorInfo = NULL;

    // Information used when cannot get custom
error object
    ERRORINFO
    BasicErrorInfo;
    BSTR
    bstrDescription;
    // Number of error records.
    ULONG
    nRecs;
    ULONG
    nRec;

    // SQL Server error information from
ISQLServerErrorInfo.
    SSERRORINFO*        pSSErrorInfo =
NULL;
    OLECHAR*            pSSErrorStrings =
NULL;

```

```

    assert(pObjectWithError != NULL);
    pOLEDBErr = new COLEDBERR(szLocation);

    pOLEDBErr->m_NativeError = 0;
    pOLEDBErr->m_eAction = eAction;
    pOLEDBErr->m_bDeadLock = FALSE;

    szTmp[0] = 0;

    // Only ask for error information if the
interface supports it.
    // Note: SQLOLEDB provider supports error
interface, so this check is
    // for good style only.
    hr = pObjectWithError-
>QueryInterface(IID_ISupportErrorInfo, (void**)
&piSupportErrorInfo);
    if (FAILED(hr))
    {
        _snprintf(szMsg, sizeof(szMsg),
"SupportErrorInfo interface not supported (hr=0x%X)",
hr);
        pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
        strcpy(pOLEDBErr->m_OLEDBErrStr,
szMsg);
        throw pOLEDBErr;
    }
    /*if (FAILED(piSupportErrorInfo-
>InterfaceSupportsErrorInfo(IID_InterfaceWithError))
    {
        _snprintf(szMsg, sizeof(szMsg),
"InterfaceWithError interface not supported");
        pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
        strcpy(pOLEDBErr->m_OLEDBErrStr,
szMsg);
        return;
    }*/

    // Do not test the return of GetErrorInfo.
It can succeed and return
    // a NULL pointer in piErrorInfoAll. Simply
test the pointer.
    GetErrorInfo(0, &piErrorInfoAll);

    if (piErrorInfoAll != NULL)
    {
        // Test to see if it's a valid
OLE DB IErrorInfo interface
        // exposing a list of records.
        if (SUCCEEDED(piErrorInfoAll-
>QueryInterface(IID_IErrorRecords, (void**)
&piErrorRecords)))
        {
            piErrorRecords-
>GetRecordCount(&nRecs);

```

```

// Within each record,
retrieve information from each
// of the defined
interfaces.
for (nRec = 0; nRec <
nRecs; nRec++)
{
    // Request
the generic SQL error interface.
    piErrorRecords->GetCustomErrorObject(nRec,

    IID_ISQLErrorInfo, // generic SQL error
interface
    (IUnknown**) &piSQLErrorInfo);

    if
(piSQLErrorInfo != NULL)
    {
        //
Request SQL Server-specific error interface, not the
generic SQL error interface.
        piSQLErrorInfo->QueryInterface(

        IID_ISQLServerErrorInfo, // SQL Server
error interface
        (void**) &piSQLServerErrorInfo);

        // Test to
ensure the reference is valid, then
// get error
information from ISQLServerErrorInfo.
        if
(piSQLServerErrorInfo != NULL)
        {
            piSQLServerErrorInfo-
>GetErrorInfo(&pSSErrorInfo, &pSSErrorStrings);

            //
ISQLServerErrorInfo::GetErrorInfo succeeds
//
even when it has nothing to return. Test the
//
pointers before using.
            if
(piSSErrorInfo)
            {
                // First, add the error message.

                // Convert Unicode error string to ANSI.
WideCharToMultiByte(CP_THREAD_ACP, 0,

```

```

        pSSErrorInfo->pwszMessage, -1,
        szMsg, sizeof(szMsg),
        NULL, NULL);

    // quit if there isn't enough room to
    concatenate error text
    if ( (strlen(szMsg) + 2) > (sizeof(szTmp) -
    strlen(szTmp)) )
        break;

    // include line break after first error msg
    if (szTmp[0] != 0)
        strcat( szTmp, "\r\n");

    // concatenate the error record to the
    overall error message
    strcat( szTmp, szMsg );

    // Second, add the stored procedure name
    and line number, if available.

    if (wcslen(pSSErrorInfo->pwszProcedure)>0)
    {
        // Prefix with a line break
        iLen = sprintf(szMsg,
        "\r\nProcedure: ");

        // Convert Unicode error string
        to ANSI.

        WideCharToMultiByte(CP_THREAD_ACP, 0,
        pSSErrorInfo->pwszProcedure, -1,
        sizeof(szMsg) - iLen,
        &szMsg[iLen],
        NULL, NULL);

        // Check if have space to add the
        line number.

```

```

        // Assume the line number takes
        no more than 3 digits.
        if ((strlen(szMsg) + 4)<
        sizeof(szMsg))
        {
            _snprintf(&szMsg[strlen(szMsg)],
            sizeof(szMsg),
            "%d",
            pSSErrorInfo->wLineNumber);
        }

        // quit if there isn't enough
        room to concatenate error text
        if ( (strlen(szMsg) + 2) >
        (sizeof(szTmp) - strlen(szTmp)) )
            break;

        // concatenate the error record
        to the overall error message
        strcat( szTmp, szMsg );

        // copy the overall error string
        to the exception
        pOLEDBErr->m_OLEDBErrStr = new
        char[strlen(szTmp)+1];
        strcpy(pOLEDBErr->m_OLEDBErrStr,
        szTmp);
    }

    // Third, capture the (first) database
    error
    if (pOLEDBErr->m_NativeError == 0 &&
    pSSErrorInfo->lNative != 0)
    {
        pOLEDBErr->m_NativeError =
        pSSErrorInfo->lNative;

        // Check for deadlock error code
        and set the deadlock flag

```

```

        if (pSSErrorInfo->lNative ==
        1205)
        {
            pOLEDBErr->m_bDeadLock
            = TRUE;
        }

        // IMalloc::Free needed to release
        references
        // on returned values.
        if (m_pIMalloc != NULL)
        {
            m_pIMalloc->Free(pSSErrorStrings);
            m_pIMalloc->Free(pSSErrorInfo);
        }

        pISQLServerErrorInfo->Release();
    }
    else
    {
        //
        Custom error object is not supported.
        Use general OLE-DB error interface.
        //
        //
        Get the numeric error code
        pIErrorRecords->GetBasicErrorInfo(nRec,
        &BasicErrorInfo);

        if
        (pOLEDBErr->m_NativeError == 0)
        {
            // Get the failed call HRESULT code, which
            is not really the native error
            pOLEDBErr->m_NativeError =
            BasicErrorInfo.hrError;
        }

        //
        Try to get the string description of the error.

```

```

        pIErrorRecords->GetErrorInfo(nRec,
        LOCALE_USER_DEFAULT,
        (IErrorInfo**) &pIErrorInfoRecord);

        if
        {
            pIErrorInfoRecord-
            >GetDescription(&bstrDescription);

            // Convert Unicode error string to ANSI.
            WideCharToMultiByte(CP_THREAD_ACP, 0,
            bstrDescription, -1,
            szMsg, sizeof(szMsg),
            NULL, NULL);

            pOLEDBErr->m_OLEDBErrStr = new
            char[strlen(szMsg)+1];
            strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
        }
        } // for()
        } // if
        (SUCCEEDED(pIErrorInfoAll-
        >QueryInterface(IID_IErrorRecords, (void**)
        &pIErrorRecords)))
        else
        { // No IErrorRecords
        interface supported. Use default IErrorInfo.
        // Note: SQLOLEDB
        supports IErrorRecords, so this check is for good
        style only.
            _snprintf(szMsg,
            sizeof(szMsg), "IErrorRecords interface not
            supported");
            pOLEDBErr-
            >m_OLEDBErrStr = new char[strlen(szMsg)+1];
            strcpy(pOLEDBErr-
            >m_OLEDBErrStr, szMsg);
        }

        pIErrorInfoAll->Release();
        } // if (pIErrorInfoAll != NULL)
        else
        {
            // No IErrorInfo interface
            supported.
            // Note: SQLOLEDB supports
            IErrorInfo, so this check is for good style only.

```

```

            _snprintf(szMsg, sizeof(szMsg),
            "IErrorInfo interface not supported");
            pOLEDBErr->m_OLEDBErrStr = new
            char[strlen(szMsg)+1];
            strcpy(pOLEDBErr->m_OLEDBErrStr,
            szMsg);
        }
        }
        throw pOLEDBErr;
    }
    /*
    *
    * Create a new command object from the SQL
    text passed in.
    */
    void CTPCC_OLEDB::CreateCommand(wchar_t*
    szSQLCommand, // I: SQL
    query for the command
    ICommandText**
    ppICommandText // O: returned command object
    )
    {
        HRESULT hr;
        // Create a new command object
        hr = m_pIDBCreateCommand-
        >CreateCommand(NULL, IID ICommandText, (IUnknown
        **)ppICommandText);
        if (FAILED(hr))
        {
            ThrowError(m_pIDBCreateCommand,
            COLEDBERR::eCreateCommand,
            "CTPCC_OLEDB::CreateCommand");
        }
        // Set command text
        hr = (*ppICommandText)-
        >SetCommandText(DBGUID_SQL, szSQLCommand);
        if (FAILED(hr))
        {
            ThrowError(*ppICommandText,
            COLEDBERR::eSetCommandText,
            "CTPCC_OLEDB::CreateCommand");
        }
        // Prepare the command
        PrepareCommand(*ppICommandText);
    }
    /*
    * QueryInterface and Prepare in one function
    for simplicity.
    * DEFERRED PREPARE property is set to off to
    prepare immediately.
    */
    void CTPCC_OLEDB::PrepareCommand(ICommandText*
    pICommandText)

```

```

    {
        HRESULT hr;
        ICommandPrepare* pICommandPrepare;
        ICommandProperties* pICommandProperties;
        DBPROPSET
        rowSetPropSet;
        DBPROP
        rowSetProp;

        // Set the deferred prepare property to
        false.
        rowSetProp.dwPropertyID =
        SSPROP_DEFERPREPARE;
        memset(&rowSetProp.vValue, 0,
        sizeof(rowSetProp.vValue));
        rowSetProp.dwOptions =
        DBPROPOPTIONS_REQUIRED;
        rowSetProp.colid = DB_NULLID;

        rowSetPropSet.cProperties = 1;
        rowSetPropSet.guidPropertySet =
        DBPROPSET_SQLSERVERROWSET;
        rowSetPropSet.rgProperties = &rowSetProp;

        // Query interface for setting properties
        hr = pICommandText-
        >QueryInterface(IID ICommandProperties, (void
        **)&pICommandProperties);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
            COLEDBERR::eQueryInterface,
            "CTPCC_OLEDB::PrepareCommand");
        }
        // Set the property set
        hr = pICommandProperties->SetProperties(1,
        &rowSetPropSet);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
            COLEDBERR::eQueryInterface,
            "CTPCC_OLEDB::PrepareCommand");
        }
        // Get interface for preparing commands
        hr = pICommandText-
        >QueryInterface(IID ICommandPrepare, (void
        **)&pICommandPrepare);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
            COLEDBERR::eQueryInterface,
            "CTPCC_OLEDB::PrepareCommand");
        }
        // Prepare Payment command
        hr = pICommandPrepare->Prepare(0xFFFFFFFF);
        if (FAILED(hr))
        {

```

```

        ThrowError(pICommandPrepare,
COLEDBERR::ePrepare, "CTPCC_OLEDB::PrepareCommand");
    }
}

/*
 * Initialize fields of an array of bindings
 * structures.
 * Needs to be called before setting
 * individual parameter/column bindings.
 */
void CTPCC_OLEDB::InitBindings(DBBINDING*
pDBBindings, // IO: array of bindings

    int iCount, // I: number of
elements in the array

    eBindingType BindingType) //
I: what the bindings will be used for
(parameters/columns)
{
    int i;

    for(i = 0; i < iCount; i++)
    {
        pDBBindings[i].iOrdinal = i + 1;
        pDBBindings[i].obLength = 0;
        pDBBindings[i].obStatus = 0;
        pDBBindings[i].pTypeInfo = NULL;
        pDBBindings[i].pObject = NULL;
        pDBBindings[i].pBindExt = NULL;
        pDBBindings[i].dwPart = DBPART_VALUE;

        switch (BindingType)
        {
            case eInputParameter:
                pDBBindings[i].eParamIO
= DBPARAMIO_INPUT;
                break;
            case eOutputParameter:
                pDBBindings[i].eParamIO
= DBPARAMIO_OUTPUT;
                break;
            case eInputOutputParameter:
                pDBBindings[i].eParamIO
= DBPARAMIO_INPUT | DBPARAMIO_OUTPUT;
                break;
            case eOutputColumn:
                pDBBindings[i].eParamIO
= DBPARAMIO_NOTPARAM;
                break;
            default:
                assert(false); //
this should never happen
        }

        pDBBindings[i].dwMemOwner =
DBMEMOWNER_CLIENTOWNED;
        pDBBindings[i].dwFlags = 0;
        pDBBindings[i].bPrecision = 0;

```

```

        pDBBindings[i].bScale = 0;
    }
}

/*
 * Perform binding for one parameter or output
 * column.
 */
void CTPCC_OLEDB::SetBinding(DBBINDING* pDBBinding,
// I: binding row structure

    size_t obValue, // I: parameter (column) offset in the user
buffer

    size_t cbMaxLen, //
I: parameter (column) length

    DBTYPE wType // I: parameter (column) type
)
{
    pDBBinding->obValue = (ULONG)obValue;
    pDBBinding->cbMaxLen = (ULONG)cbMaxLen;
    pDBBinding->wType = wType;
}

void CTPCC_OLEDB::InitStockLevelParams()
{
    int i;

    HRESULT hr;
    wchar_t
szName[IMAX_SP_NAME_LEN];
    IAccessor*
pIAccessor;

    const ULONG
nInputParams = 3; // input parameters
const ULONG
nOutputParams = 1; // output 1st result

    set columns
    // Structure to bind in accessor
    DBBINDING
acInputDBBinding[nInputParams];
    DBBINDSTATUS
acInputDBBindStatus[nInputParams];
    DBBINDING
acOutputDBBinding[nOutputParams];
    DBBINDSTATUS
acOutputDBBindStatus[nOutputParams];

    // Set command text
    snprintf(szName,
sizeof(szName)/sizeof(szName[0]),
L"{call
%stpcck_stocklevel (?,?,?)}", m_szSPPrefix);

```

```

    // Create and Prepare a new command object
    for StockLevel.
    CreateCommand(szName,
&m_pIStockLevelCommand);

    // Describe the consumer buffer by filling
    in the array
    // of DBBINDING structures. Each binding
    associates
    // a single parameter to the consumer's buffer.
    InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

    i = 0;
    // StockLevel parameter 1
    SetBinding(&acInputDBBinding[i++],
offsetof(STOCK_LEVEL_DATA, w_id),
sizeof(m_txn.StockLevel.w_id), DBTYPE_I4);

    // StockLevel parameter 2
    SetBinding(&acInputDBBinding[i++],
offsetof(STOCK_LEVEL_DATA, d_id),
sizeof(m_txn.StockLevel.d_id), DBTYPE_UI1);

    // StockLevel parameter 3
    SetBinding(&acInputDBBinding[i++],
offsetof(STOCK_LEVEL_DATA, threshold),
sizeof(m_txn.StockLevel.threshold), DBTYPE_I2);

    hr = m_pIStockLevelCommand-
>QueryInterface(IID_IAccessor, (void **)&pIAccessor);
    if (FAILED(hr))
    {
        ThrowError(m_pIStockLevelCommand,
COLEDBERR::eQueryInterface,
"InitStockLevelParams()");
    }

    hr = pIAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
nInputParams,
acInputDBBinding,
sizeof(STOCK_LEVEL_DATA),

&m_hStockLevelInputAccessor,
acInputDBBindStatus);

    if (FAILED(hr))
    {
        ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitStockLevelParams()");
    }

    m_StockLevelExecuteParams.cParamSets = 1;
    m_StockLevelExecuteParams.hAccessor =
m_hStockLevelInputAccessor;
    m_StockLevelExecuteParams.pData =
&m_txn.StockLevel;

    // Now fill the binding information for
    result set 1 output columns

```

```

        InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

        // Binding for a rowset that may return
more than one row.
        i = 0;
        // StockLevel output column 1
        SetBinding(&acOutputDBBinding[i++],
offsetof(STOCK_LEVEL_DATA, low_stock),
sizeof(m_txn.StockLevel.low_stock), DBTYPE_I4);

        hr = piAccessor->CreateAccessor(
DBACCESSOR_OPTIMIZED,
        nOutputParams,
        acOutputDBBinding,
        sizeof(STOCK_LEVEL_DATA),

&m_hStockLevelOutputAccessor,
        acOutputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(piAccessor,
COLEDBERR::eCreateAccessor,
"InitStockLevelParams()");
        }
    }

void CTPCC_OLEDB::StockLevel()
{
    HRESULT                hr;
    int                    iTryCount = 0;
    IRowset*               pRowset;
    LONG                   cRows = 1;
    // number of rows returned in the rowset
    ULONG                  cRowsObtained;
    HROW                   rghRow;
    //returned row handles
    HROW*                  prghRow =
&rghRow;

    while (TRUE)
    {
        try
        {
            // Execute the prepared
command
            hr =
m_pIStockLevelCommand->Execute(NULL, IID_IRowset,
&m_StockLevelExecuteParams, NULL,

(IUnknown **)&pRowset);
            if (FAILED(hr))
            {
                ThrowError(m_pIStockLevelCommand,
COLEDBERR::eExecute, "StockLevel()");
            }
        }
    }
}

```

```

        // Fetch the result row
handle(s)
        hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRow);
        if (FAILED(hr))
        {
            ThrowError(m_pIStockLevelCommand,
COLEDBERR::eGetNextRows, "StockLevel()");
        }

        // Fetch the actual row
data by handle
        hr = pRowset-
>GetData(rghRow, m_hStockLevelOutputAccessor,
&m_txn.StockLevel);
        if (FAILED(hr))
        {
            ThrowError(m_pIStockLevelCommand,
COLEDBERR::eGetData, "StockLevel()");
        }

        // Release row(s)
        hr = pRowset-
>ReleaseRows(cRowsObtained, prghRow, NULL, NULL,
NULL);
        // Release rowset
        hr = pRowset-
>Release();

        m_txn.StockLevel.exec_status_code = eOK;
        break;
    }
    catch (COLEDBERR *e)
    {
        if (!(e->m_bDeadLock)
|| (++iTryCount > iMaxRetries))
            throw;

        // hit deadlock;
backoff for increasingly longer period
        delete e;
        Sleep(10 * iTryCount);
    }
}

// if (iTryCount)
// throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_OLEDB::InitNewOrderParams()
{
    int                    i, j, iOlCount;
}

```

```

        HRESULT
        hr;
        wchar_t
        szName[IMAX_SP_NAME_LEN];
        IAccessor*
        piAccessor;
        const ULONG
        nInputParams = 5 +
3*MAX_OL_NEW_ORDER_ITEMS; // input parameters
        const ULONG
        nOutputParams = 5; // output 1st result
set columns
        const ULONG
        nOutputParams2 = 8; // output 2nd result
set columns
        // Structure to bind in accessor
        DBBINDING
        acInputDBBinding[nInputParams];
        DBBINDSTATUS
        acInputDBBindStatus[nInputParams];
        DBBINDING
        acOutputDBBinding[nOutputParams];
        DBBINDSTATUS
        acOutputDBBindStatus[nOutputParams];
        DBBINDING
        acOutputDBBinding2[nOutputParams2];
        DBBINDSTATUS
        acOutputDBBindStatus2[nOutputParams2];

        // Describe the consumer buffer by filling
in the array
        // of DBBINDING structures. Each binding
associates
        // a single parameter to the consumer's buffer.
        InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

        i = 0;
        // NewOrder parameter 1
        SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, w_id),
sizeof(m_txn.NewOrder.w_id), DBTYPE_I4);

        // NewOrder parameter 2
        SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, d_id),
sizeof(m_txn.NewOrder.d_id), DBTYPE_UI1);

        // NewOrder parameter 3
        SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, c_id),
sizeof(m_txn.NewOrder.c_id), DBTYPE_I4);

        // NewOrder parameter 4
        SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, o_ol_cnt),
sizeof(m_txn.NewOrder.o_ol_cnt), DBTYPE_UI1);

        // NewOrder parameter 5
}

```



```

        SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, o_all_local),
sizeof(m_txn.NewOrder.o_all_local), DBTYPE_UI1);

        for (j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
        {
            SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, OL[j].ol_i_id),
sizeof(m_txn.NewOrder.OL[j].ol_i_id), DBTYPE_I4);

            SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, OL[j].ol_supply_w_id),
sizeof(m_txn.NewOrder.OL[j].ol_supply_w_id),
DBTYPE_I4);

            SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, OL[j].ol_quantity),
sizeof(m_txn.NewOrder.OL[j].ol_quantity), DBTYPE_I2);
        }

        // Now fill the binding information for
result set 1 output columns
        InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

        // Binding for the order line rowsets (each
consist of one row).
        // Bind to offsets of the OL_NEW_ORDER_DATA
structure
instead of NEW_ORDER_DATA.
        // IRowset::GetData() will be passed
individual array slots OL[i] to fetch the data
// from the row set.

        i = 0;
        // NewOrder output column 1
        SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_i_name),
sizeof(m_txn.NewOrder.OL[0].ol_i_name), DBTYPE_STR);

        // NewOrder output column 2
        SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_stock),
sizeof(m_txn.NewOrder.OL[0].ol_stock), DBTYPE_I2);

        // NewOrder output column 3
        SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_brand_generic),
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic),
DBTYPE_STR);

        // NewOrder output column 4
        SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_i_price),
sizeof(m_txn.NewOrder.OL[0].ol_i_price), DBTYPE_R8);

        // NewOrder output column 5
        SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_amount),

```

```

sizeof(m_txn.NewOrder.OL[0].ol_amount), DBTYPE_R8);

        // Now fill the binding information for
result set 2 output columns
        InitBindings(&acOutputDBBinding2[0],
nOutputParams2, eOutputColumn);

        i = 0;
        // NewOrder output column 1
        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, w_tax),
sizeof(m_txn.NewOrder.w_tax), DBTYPE_R8);

        // NewOrder output column 2
        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, d_tax),
sizeof(m_txn.NewOrder.d_tax), DBTYPE_R8);

        // NewOrder output column 3
        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, o_id),
sizeof(m_txn.NewOrder.o_id), DBTYPE_I4);

        // NewOrder output column 4
        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, c_last),
sizeof(m_txn.NewOrder.c_last), DBTYPE_STR);

        // NewOrder output column 5
        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, c_discount),
sizeof(m_txn.NewOrder.c_discount), DBTYPE_R8);

        // NewOrder output column 6
        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, c_credit),
sizeof(m_txn.NewOrder.c_credit), DBTYPE_STR);

        // NewOrder output column 7
        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, o_entry_d),
sizeof(m_txn.NewOrder.o_entry_d),
DBTYPE_DBTIMESTAMP);

        // NewOrder output column 8
        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, o_commit_flag),
sizeof(m_txn.NewOrder.o_commit_flag), DBTYPE_I2);

        for (j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
        {
            // Set command text first

            // Print the fixed first portion
of parameters
            i = _snprintf(szName,
sizeof(szName)/sizeof(szName[0]),

```

```

        L"call %stpcc_neworder (?,?,?,?,"
m_szSPPrefix);

        // Now print the variable portion
depending on the number of order line parameters
        for (iOlCount = 0; iOlCount <= j;
++iOlCount)
        {
            i +=
            _snprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L",?,?,?");
        }

        // Print the fixed end
if (j != MAX_OL_NEW_ORDER_ITEMS -
1)
        { // append 'default' for
the parameters that are not used
            i +=
            _snprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L",default}");
        }
        else // using all 15 order
line parameters
        {
            i +=
            _snprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L"}");
        }

        // Create and Prepare a new
command object for NewOrder.
        CreateCommand(szName,
&m_pINewOrderCommand[j]);

        // Now create the input accessor
for this prepared command
        hr = m_pINewOrderCommand[j]-
>QueryInterface(IID_IAccessor, (void **)&pIAccessor);
if (FAILED(hr))
        {
            ThrowError(m_pINewOrderCommand[j],
COLEDBERR::eQueryInterface, "InitNewOrderParams()");
        }

        hr = pIAccessor->CreateAccessor(

            DBACCESSOR_PARAMETERDATA,

            5 +
            3 * (j + 1),

            acInputDBBinding,

```

```

sizeof(NEW_ORDER_DATA),

&m_hNewOrderInputAccessor[j],

acInputDBBindStatus);
if (FAILED(hr))
{
    ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitNewOrderParams()");
}

m_NewOrderExecuteParams[j].cParamSets = 1;
//
m_NewOrderExecuteParams.hAccessor is set dynamically
at run-time
// based on the number of new
order items for the particular transaction call.
m_NewOrderExecuteParams[j].hAccessor =
m_hNewOrderInputAccessor[j];
m_NewOrderExecuteParams[j].pData
= &m_txn.NewOrder;

// Create accessor for the first
rowset
hr = pIAccessor->CreateAccessor(
DBACCESSOR_OPTIMIZED,
nOutputParams,
acOutputDBBinding,
sizeof(OL_NEW_ORDER_DATA),
&m_hNewOrderOutputAccessor[j],
acOutputDBBindStatus);
if (FAILED(hr))
{
    ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitNewOrderParams()");
}

// Create accessor for the second
rowset
hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA, //
cannot be optimized too because #1 accessor is
nOutputParams2,
acOutputDBBinding2,
sizeof(NEW_ORDER_DATA),
&m_hNewOrderOutputAccessor2[j],
acOutputDBBindStatus2);
if (FAILED(hr))
{

```

```

        ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitNewOrderParams()");
    }
}
pIAccessor->Release();
}

void CTPCC_OLEDB::NewOrder()
{
    HRESULT hr;
    int iTryCount = 0;
    IMultipleResults* pMultipleResults;
    IRowset* pRowset;
    IRowset* pRowset2;
    LONG cRows = 1; // number of rows
    returned in the 1st rowset
    ULONG cRowsObtained;
    HROW rghRows; //returned row handles
    for the 1st result set
    HROW* prghRows = &rghRows;
    LONG cRows2 = 1; // number of rows
    returned in the 2nd rowset
    ULONG cRowsObtained2;
    HROW rghRows2; //returned row handle
    for the 2nd result set
    HROW* prghRows2 = &rghRows2;
    int i;
    long lRowsAffected; // the number of
    affected rows for a rowset
    int iHandleIndex; // index into the
    handle arrays based on the orders count

    // check whether any order lines are for a
    remote warehouse
    m_txn.NewOrder.o_all_local = 1;
    for (i = 0; i < m_txn.NewOrder.o_ol_cnt;
i++)
    {
        if
(m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
        {
            m_txn.NewOrder.o_all_local = 0; // at
            least one remote warehouse
            break;
        }
    }
}

```

```

iHandleIndex = m_txn.NewOrder.o_ol_cnt - 1;
// for convenience
while (TRUE)
{
    try
    {
        // Execute the prepared
        command (according to the number of new orders)
        // Ask for
        IMultipleResults because it returns 2 rowsets.
        hr =
m_pINewOrderCommand[iHandleIndex]->Execute(
        NULL, IID_IMultipleResults,
        &m_NewOrderExecuteParams[iHandleIndex],
        NULL,
        (IUnknown **)&pMultipleResults);
        if (FAILED(hr))
        {
            ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eExecute, "NewOrder()");
        }

        //////////////////////////////////////
        // Get order line
        results
        //////////////////////////////////////
        m_txn.NewOrder.total_amount = 0;
        for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; ++i)
        {
            // Get the
            first rowset object
            hr =
pMultipleResults->GetResult(NULL, 0, IID_IRowset,
&lRowsAffected, (IUnknown **)&pRowset);
            if
(FAILED(hr))
            {
                char szTmp[256];

                _snprintf(szTmp, sizeof(szTmp), "NewOrder()
result set %d, hr=0x%X", i, hr);

```

```

        ThrowError(m_pINewOrderCommand[m_txn.NewOrder.o_ol_cnt - 1], COLEDBERR::eGetResult, szTmp);
    }

    // Fetch the
    result row handle(s)
    hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRows);
    if
    (FAILED(hr))
    {
        ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eGetNextRows, "NewOrder()");
    }

    // Fetch the
    actual row data by handle
    hr = pRowset-
>GetData(rghRows,
m_hNewOrderOutputAccessor[iHandleIndex],
&m_txn.NewOrder.OL[i]);
    if
    (FAILED(hr))
    {
        ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eGetData, "NewOrder()");
    }

    m_txn.NewOrder.total_amount +=
m_txn.NewOrder.OL[i].ol_amount;

    // Release
    row(s)
    hr = pRowset-
>ReleaseRows(cRowsObtained, prghRows, NULL, NULL,
NULL);
    // Release
    rowset
    hr = pRowset-
>Release();
}

////////////////////////////////////
// Get the second
rowset object
////////////////////////////////////
hr = pMultipleResults-
>GetResult(NULL, 0, IID_IRowset, &lRowsAffected,
(IUnknown **)&pRowset2);
    if (FAILED(hr))
    {
        char
szTmp[256];

```

```

        _snprintf(szTmp, sizeof(szTmp), "NewOrder()
result set %d, hr=%d", i, hr);

        ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eGetResult, szTmp);
    }

    // Fetch the result row
    handle(s)
    hr = pRowset2-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows2,
&cRowsObtained2, &prghRows2);
    if (FAILED(hr))
    {
        ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eGetNextRows, "NewOrder()");
    }

    // Fetch the actual row
    data by handle
    hr = pRowset2-
>GetData(rghRows2,
m_hNewOrderOutputAccessor2[iHandleIndex],
&m_txn.NewOrder);
    if (FAILED(hr))
    {
        ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eGetData, "NewOrder()");
    }

    // Release row(s)
    hr = pRowset2-
>ReleaseRows(cRowsObtained2, prghRows2, NULL, NULL,
NULL);
    // Release rowset
    hr = pRowset2-
>Release();

    // Release the common
    MultipleResults interface
    hr = pMultipleResults-
>Release();

    if
    (m_txn.NewOrder.o_all_local == 1)
    {
        m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

        m_txn.NewOrder.exec_status_code = eOK;
    }
    else
    {

```

```

        m_txn.NewOrder.exec_status_code =
eInvalidItem;
    }

    break;
}
catch (COLEDBERR *e)
{
    if (!e->m_bDeadLock)
    || (++iTryCount > iMaxRetries))
        throw;

    // hit deadlock;
    backoff for increasingly longer period
    delete e;
    Sleep(10 * iTryCount);
}

// if (iTryCount)
// throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_OLEDB::InitPaymentParams()
{
    int
i;
    HRESULT
hr;
    wchar_t
szName[IMAX_SP_NAME_LEN];
    IAccessor*
pIAccessor;

    const
ULONG
nInputParams = 7; // input parameters
const ULONG
nOutputParams = 27; // output result set

    columns
    // Structure to bind in accessor
    DBBINDING
acInputDBBinding[nInputParams];
    DBBINDSTATUS
acInputDBBindStatus[nInputParams];
    DBBINDING
acOutputDBBinding[nOutputParams];
    DBBINDSTATUS
acOutputDBBindStatus[nOutputParams];

    // Set command text
    _snprintf(szName,
sizeof(szName)/sizeof(szName[0]), L"(call
%stpc_payment(?,?,?,?))", m_szSPPrefix);

    // Create and Prepare a new command object
    for Payment.
    CreateCommand(szName, &m_pIPaymentCommand);
}

```

```

        // Describe the consumer buffer by filling
in the array
// of DBBINDING structures. Each binding
associates
// a single parameter to the consumer's buffer.
InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

    i = 0;
// Payment parameter 1
SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, w_id),
sizeof(m_txn.Payment.w_id), DBTYPE_I4);

// Payment parameter 2
SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, c_w_id),
sizeof(m_txn.Payment.c_w_id), DBTYPE_I4);

// Payment parameter 3
SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, h_amount),
sizeof(m_txn.Payment.h_amount), DBTYPE_R8);

// Payment parameter 4
SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, d_id),
sizeof(m_txn.Payment.d_id), DBTYPE_UI1);

// Payment parameter 5
SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, c_d_id),
sizeof(m_txn.Payment.c_d_id), DBTYPE_UI1);

// Payment parameter 6
SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, c_id),
sizeof(m_txn.Payment.c_id), DBTYPE_I4);

// Payment parameter 7
SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, c_last),
sizeof(m_txn.Payment.c_last), DBTYPE_STR);

    hr = m_pIPaymentCommand-
>QueryInterface(IID_IAccessor, (void **)&IAccessor);
    if (FAILED(hr))
    {
        ThrowError(m_pIPaymentCommand,
COLEDBERR::eQueryInterface, "InitPaymentParams()");
    }

    hr = pIAccessor->CreateAccessor(
        DBACCESSOR_PARAMETERDATA,
        nInputParams,
        acInputDBBinding,
        sizeof(PAYMENT_DATA),
        &m_hPaymentInputAccessor,
        acInputDBBindStatus);
    if (FAILED(hr))

```

```

    {
        ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitPaymentParams()");
    }

    m_PaymentExecuteParams.cParamSets = 1;
    m_PaymentExecuteParams.hAccessor =
m_hPaymentInputAccessor;
    m_PaymentExecuteParams.pData =
&m_txn.Payment;

// Now fill the binding information for
output columns
InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

    i = 0;
// Payment output column 1
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_id),
sizeof(m_txn.Payment.c_id), DBTYPE_I4);

// Payment output column 2
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_last),
sizeof(m_txn.Payment.c_last), DBTYPE_STR);

// Payment output column 3
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, h_date),
sizeof(m_txn.Payment.h_date), DBTYPE_DBTIMESTAMP);

// Payment output column 4
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_street_1),
sizeof(m_txn.Payment.w_street_1), DBTYPE_STR);

// Payment output column 5
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_street_2),
sizeof(m_txn.Payment.w_street_2), DBTYPE_STR);

// Payment output column 6
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_city),
sizeof(m_txn.Payment.w_city), DBTYPE_STR);

// Payment output column 7
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_state),
sizeof(m_txn.Payment.w_state), DBTYPE_STR);

// Payment output column 8
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_zip),
sizeof(m_txn.Payment.w_zip), DBTYPE_STR);

// Payment output column 9
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_1),
sizeof(m_txn.Payment.d_street_1), DBTYPE_STR);

```

```

// Payment output column 10
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_2),
sizeof(m_txn.Payment.d_street_2), DBTYPE_STR);

// Payment output column 11
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_city),
sizeof(m_txn.Payment.d_city), DBTYPE_STR);

// Payment output column 12
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_state),
sizeof(m_txn.Payment.d_state), DBTYPE_STR);

// Payment output column 13
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_zip),
sizeof(m_txn.Payment.d_zip), DBTYPE_STR);

// Payment output column 14
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_first),
sizeof(m_txn.Payment.c_first), DBTYPE_STR);

// Payment output column 15
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_middle),
sizeof(m_txn.Payment.c_middle), DBTYPE_STR);

// Payment output column 16
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_1),
sizeof(m_txn.Payment.d_street_1), DBTYPE_STR);

// Payment output column 17
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_2),
sizeof(m_txn.Payment.d_street_2), DBTYPE_STR);

// Payment output column 18
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_city),
sizeof(m_txn.Payment.d_city), DBTYPE_STR);

// Payment output column 19
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_state),
sizeof(m_txn.Payment.d_state), DBTYPE_STR);

// Payment output column 20
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_zip),
sizeof(m_txn.Payment.d_zip), DBTYPE_STR);

// Payment output column 21
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_phone),
sizeof(m_txn.Payment.c_phone), DBTYPE_STR);

```

```

        // Payment output column 22
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_since),
sizeof(m_txn.Payment.c_since), DBTYPE_DBTIMESTAMP);

        // Payment output column 23
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_credit),
sizeof(m_txn.Payment.c_credit), DBTYPE_STR);

        // Payment output column 24
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_credit_lim),
sizeof(m_txn.Payment.c_credit_lim), DBTYPE_R8);

        // Payment output column 25
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_discount),
sizeof(m_txn.Payment.c_discount), DBTYPE_R8);

        // Payment output column 26
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_balance),
sizeof(m_txn.Payment.c_balance), DBTYPE_R8);

        // Payment output column 27
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_data),
sizeof(m_txn.Payment.c_data), DBTYPE_STR);

        hr = pIAccessor->CreateAccessor(
DBACCESSOR_OPTIMIZED,
        nOutputParams,
        acOutputDBBinding,
        sizeof(PAYMENT_DATA),
&m_hPaymentOutputAccessor,
        acOutputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitPaymentParams()");
        }

void CTPCC_OLEDB::Payment()
{
    HRESULT          hr;
    int              i;
    iTryCount = 0;
    IRowset*         pRowset;
    LONG             cRows = 1;
    // number of rows returned in the rowset
    ULONG           cRowsObtained;
    HROW             rghRow;
    //returned row handles
    HROW*            prghRow =
&rghRow;

```

```

        if (m_txn.Payment.c_id != 0)
            m_txn.Payment.c_last[0] = 0;

        while (TRUE)
        {
            try
            {
                // Execute the prepared
                command
                hr =
m_pIPaymentCommand->Execute(NULL, IID_IRowset,
&m_PaymentExecuteParams, NULL,

                (IUnknown **)&pRowset);
                if (FAILED(hr))
                {
                    ThrowError(m_pIPaymentCommand,
COLEDBERR::eExecute, "Payment()");
                }

                // Fetch the result row
                handle(s)
                hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRow);
                if (FAILED(hr))
                {
                    ThrowError(m_pIPaymentCommand,
COLEDBERR::eGetNextRows, "Payment()");
                }

                // Fetch the actual row
                data by handle
                hr = pRowset-
>GetData(rghRow, m_hPaymentOutputAccessor,
&m_txn.Payment);
                if (FAILED(hr))
                {
                    ThrowError(m_pIPaymentCommand,
COLEDBERR::eGetData, "Payment()");
                }

                // Release row(s)
                hr = pRowset-
>ReleaseRows(cRowsObtained, prghRow, NULL, NULL,
NULL);
                // Release rowset
                hr = pRowset-
>Release();

                if (m_txn.Payment.c_id
== 0)
                    throw new
CTPCC_OLEDB_ERR( CTPCC_OLEDB_ERR::ERR_INVALID_CUST );
                else

```

```

        m_txn.Payment.exec_status_code = eOK;
        break;
    }
    catch (COLEDBERR *e)
    {
        if (!e->m_bDeadLock)
        {
            if (++iTryCount > iMaxRetries)
                throw;

            // hit deadlock;
            backoff for increasingly longer period
            delete e;
            Sleep(10 * iTryCount);
        }
    }

    // if (iTryCount)
    // throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_OLEDB::InitOrderStatusParams()
{
    int              i;
    HRESULT          hr;
    wchar_t         szName[iMAX_SP_NAME_LEN];
    IAccessor*       pIAccessor;
    const ULONG     nInputParams = 4; // input parameters
    const ULONG     nOutputParams = 5; // output 1st result
set columns
    const ULONG     nOutputParams2 = 8; // output 2nd result
set columns
    // Structure to bind in accessor
    DBBINDING
    acInputDBBinding[nInputParams];
    DBBINDSTATUS
    acInputDBBindStatus[nInputParams];
    DBBINDING
    acOutputDBBinding[nOutputParams];
    DBBINDSTATUS
    acOutputDBBindStatus[nOutputParams];
    DBBINDING
    acOutputDBBinding2[nOutputParams2];
    DBBINDSTATUS
    acOutputDBBindStatus2[nOutputParams2];

    // Set command text
    _snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),
L"call
%stpcc_orderstatus (?,?,?)", m_szSPPrefix);

```

```

// Create and Prepare a new command object
for OrderStatus.
CreateCommand(szName,
&m_pIOrderStatusCommand);

// Describe the consumer buffer by filling
in the array
// of DBBINDING structures. Each binding
associates
// a single parameter to the consumer's buffer.
InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

i = 0;
// OrderStatus parameter 1
SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, w_id),
sizeof(m_txn.OrderStatus.w_id), DBTYPE_I4);

// OrderStatus parameter 2
SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, d_id),
sizeof(m_txn.OrderStatus.d_id), DBTYPE_UI1);

// OrderStatus parameter 3
SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, c_id),
sizeof(m_txn.OrderStatus.c_id), DBTYPE_I4);

// OrderStatus parameter 4
SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, c_last),
sizeof(m_txn.OrderStatus.c_last), DBTYPE_STR);

hr = m_pIOrderStatusCommand-
>QueryInterface(IID_IAccessor, (void **)&piAccessor);
if (FAILED(hr))
{
ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eQueryInterface,
"InitOrderStatusParams()");
}

hr = piAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
nInputParams,
acInputDBBinding,
sizeof(ORDER_STATUS_DATA),
&m_hOrderStatusInputAccessor,
acInputDBBindStatus);
if (FAILED(hr))
{
ThrowError(piAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
}

```

```

m_OrderStatusExecuteParams.cParamSets = 1;
m_OrderStatusExecuteParams.hAccessor =
m_hOrderStatusInputAccessor;
m_OrderStatusExecuteParams.pData =
&m_txn.OrderStatus;

// Now fill the binding information for
result set 1 output columns
InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

// Binding for a rowset that may return
more than one row.
// Bind to offsets of the
OL_ORDER_STATUS_DATA structure instead of
ORDER_STATUS_DATA.
// IRowset::GetData() will be passed
individual array slots OL[i] to fetch the data
// from the row set.

i = 0;
// OrderStatus output column 1
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_supply_w_id),
sizeof(m_txn.OrderStatus.OL[0].ol_supply_w_id),
DBTYPE_I4);

// OrderStatus output column 2
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_i_id),
sizeof(m_txn.OrderStatus.OL[0].ol_i_id),
DBTYPE_I4);

// OrderStatus output column 3
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_quantity),
sizeof(m_txn.OrderStatus.OL[0].ol_quantity),
DBTYPE_I2);

// OrderStatus output column 4
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_amount),
sizeof(m_txn.OrderStatus.OL[0].ol_amount),
DBTYPE_R8);

// OrderStatus output column 5
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_delivery_d),
sizeof(m_txn.OrderStatus.OL[0].ol_delivery_d),
DBTYPE_DBTIMESTAMP);

hr = piAccessor->CreateAccessor(
DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
nOutputParams,
acOutputDBBinding,
sizeof(OL_ORDER_STATUS_DATA),
&m_hOrderStatusOutputAccessor,
acOutputDBBindStatus);
if (FAILED(hr))

```

```

{
ThrowError(piAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
}

// Now fill the binding information for
result set 2 output columns
InitBindings(&acOutputDBBinding2[0],
nOutputParams2, eOutputColumn);

i = 0;
// OrderStatus output column 1
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_id),
sizeof(m_txn.OrderStatus.c_id), DBTYPE_I4);

// OrderStatus output column 2
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_last),
sizeof(m_txn.OrderStatus.c_last), DBTYPE_STR);

// OrderStatus output column 3
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_first),
sizeof(m_txn.OrderStatus.c_first), DBTYPE_STR);

// OrderStatus output column 4
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_middle),
sizeof(m_txn.OrderStatus.c_middle), DBTYPE_STR);

// OrderStatus output column 5
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_entry_d),
sizeof(m_txn.OrderStatus.o_entry_d),
DBTYPE_DBTIMESTAMP);

// OrderStatus output column 7
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_carrier_id),
sizeof(m_txn.OrderStatus.o_carrier_id), DBTYPE_I2);

// OrderStatus output column 8
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_balance),
sizeof(m_txn.OrderStatus.c_balance),
DBTYPE_R8);

// OrderStatus output column 9
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_id),
sizeof(m_txn.OrderStatus.o_id), DBTYPE_I4);

hr = piAccessor->CreateAccessor(
DBACCESSOR_ROWDATA, //
cannot be optimized too because #1 accessor is
nOutputParams2,
acOutputDBBinding2,
sizeof(NEW_ORDER_DATA),
&m_hOrderStatusOutputAccessor2,

```

```

        acOutputDBBindStatus2);
    if (FAILED(hr))
    {
        ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
    }
}

void CTPCC_OLEDB::OrderStatus()
{
    HRESULT                hr;
    int
    iTryCount = 0;
    IMultipleResults*     pMultipleResults;
    IRowset*              pRowset;
    IRowset*              pRowset2;
    LONG
    cRows = MAX_OL_ORDER_STATUS_ITEMS; //
number of rows returned in the 1st rowset
    ULONG
    cRowsObtained;
    HROW
    rghRows[MAX_OL_ORDER_STATUS_ITEMS];
//returned row handles for the 1st result
set
    HROW*
    prghRows = &rghRows[0];
    LONG
    cRows2 = 1; // number of rows
returned in the 2nd rowset
    ULONG
    cRowsObtained2;
    HROW
    rghRows2; //returned row handle
for the 2nd result set
    HROW*
    prghRows2 = &rghRows2;
    int
    i;
    long
    lRowsAffected; // the number of
affected rows for a rowset

    if (m_txn.OrderStatus.c_id != 0)
        m_txn.OrderStatus.c_last[0] = 0;

    while (TRUE)
    {
        try
        {
            // Execute the prepared
command
            // Ask for
IMultipleResults because it returns 2 rowsets.
            hr =
m_pIOrderStatusCommand->Execute(NULL,
IID_IMultipleResults, &m_OrderStatusExecuteParams,
NULL,

```

```

(IUnknown **) &pMultipleResults);
        if (FAILED(hr))
        {
            ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eExecute, "OrderStatus()");
        }

        ////////////////
        // Get order line

        results

        ////////////////

        // Get the first rowset
        object
            hr = pMultipleResults-
>GetResult(NULL, 0, IID_IRowset, &lRowsAffected,
(IUnknown **) &pRowset);
            if (FAILED(hr))
            {
                ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetResult, "OrderStatus()");
            }

            // Fetch the result row
handle(s)
            hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRows);
            if (FAILED(hr))
            {
                ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetNextRows, "OrderStatus()");
            }

            m_txn.OrderStatus.o_ol_cnt =
(short)cRowsObtained;

            // Get the data from
multiple rows in this rowset
            for (i = 0; i <
m_txn.OrderStatus.o_ol_cnt; ++i)
            {
                // Fetch the
actual row data by handle
                hr = pRowset-
>GetData(rghRows[i], m_hOrderStatusOutputAccessor,
&m_txn.OrderStatus.OL[i]);
                if
                (FAILED(hr))
                {

```

```

                ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetData, "OrderStatus()");
            }
        }
        // Release row(s)
        hr = pRowset-
>ReleaseRows(cRowsObtained, prghRows, NULL, NULL,
NULL);
        // Release rowset
        hr = pRowset-
>Release();

        // Get the second
rowset object

        ////////////////

        if
(m_txn.OrderStatus.o_ol_cnt > 0)
        {
            hr =
pMultipleResults->GetResult(NULL, 0, IID_IRowset,
&lRowsAffected, (IUnknown **) &pRowset2);
            if
            (FAILED(hr))
            {
                ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetResult, "OrderStatus()");
            }

            // Fetch the
result row handle(s)
            hr =
pRowset2->GetNextRows(DB_NULL_HCHAPTER, 0, cRows2,
&cRowsObtained2, &prghRows2);
            if
            (FAILED(hr))
            {
                ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetNextRows, "OrderStatus()");
            }

            // Fetch the
actual row data by handle
            hr =
pRowset2->GetData(rghRows2,
m_hOrderStatusOutputAccessor2, &m_txn.OrderStatus);
            if
            (FAILED(hr))
            {
                ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetData, "OrderStatus()");
            }
        }
    }
}

```

```

// Release
row(s)
hr =
pRowset2->Release();
}
// Release the common
MultipleResults interface
hr = pMultipleResults-
>Release();
if
(m_txn.OrderStatus.o_ol_cnt == 0)
throw new
CTPCC_OLEDB_ERR( CTPCC_OLEDB_ERR::ERR_NO_SUCH_ORDER
);
else if
(m_txn.OrderStatus.c_id == 0 &&
m_txn.OrderStatus.c_last[0] == 0)
throw new
CTPCC_OLEDB_ERR( CTPCC_OLEDB_ERR::ERR_INVALID_CUST );
else
m_txn.OrderStatus.exec_status_code = eOK;
break;
}
catch (COLEDBERR *e)
{
if (!(e->m_bDeadLock)
|| (++iTryCount > iMaxRetries))
throw;
// hit deadlock;
backoff for increasingly longer period
delete e;
Sleep(10 * iTryCount);
}
}
// if (iTryCount)
// throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}
void CTPCC_OLEDB::InitDeliveryParams()
{
int
i;
HRESULT
hr;
wchar_t
szName[IMAX_SP_NAME_LEN];
IAccessor*
pIAccessor;
const ULONG
nInputParams = 2; // input parameters

```

```

const ULONG
nOutputParams = 10; // output 1st result
set columns
// Structure to bind in accessor
DBBINDING
acInputDBBinding[nInputParams];
DBBINDSTATUS
acInputDBBindStatus[nInputParams];
DBBINDING
acOutputDBBinding[nOutputParams];
DBBINDSTATUS
acOutputDBBindStatus[nOutputParams];
// Set command text
_snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),
L"%call %stpcc_delivery
(?,?)", m_szSPPrefix);
// Create and Prepare a new command object
for Delivery.
CreateCommand(szName,
&m_pIDeliveryCommand);
// Describe the consumer buffer by filling
in the array
// of DBBINDING structures. Each binding
associates
// a single parameter to the consumer's buffer.
InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);
i = 0;
// Delivery parameter 1
SetBinding(&acInputDBBinding[i++],
offsetof(DELIVERY_DATA, w_id),
sizeof(m_txn.Delivery.w_id), DBTYPE_I4);
// Delivery parameter 2
SetBinding(&acInputDBBinding[i++],
offsetof(DELIVERY_DATA, o_carrier_id),
sizeof(m_txn.Delivery.o_carrier_id), DBTYPE_I2);
hr = m_pIDeliveryCommand-
>QueryInterface(IID_IAccessor, (void **)&pIAccessor);
if (FAILED(hr))
{
ThrowError(m_pIDeliveryCommand,
COLEDBERR::eQueryInterface, "InitDeliveryParams()");
}
hr = pIAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
nInputParams,
acInputDBBinding,
sizeof(DELIVERY_DATA),
&m_hDeliveryInputAccessor,
acInputDBBindStatus);
if (FAILED(hr))
{

```

```

ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitDeliveryParams()");
}
m_DeliveryExecuteParams.cParamSets = 1;
m_DeliveryExecuteParams.hAccessor =
m_hDeliveryInputAccessor;
m_DeliveryExecuteParams.pData =
&m_txn.Delivery;
// Now fill the binding information for
result set 1 output columns
InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);
// Binding for a rowset that may return
more than one row.
for (i = 0; i < 10; ++i)
{
// Delivery output column 1
SetBinding(&acOutputDBBinding[i],
offsetof(DELIVERY_DATA, o_id[i]),
sizeof(m_txn.Delivery.o_id[i]), DBTYPE_I4);
}
hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
nOutputParams,
acOutputDBBinding,
sizeof(DELIVERY_DATA),
&m_hDeliveryOutputAccessor,
acOutputDBBindStatus);
if (FAILED(hr))
{
ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitDeliveryParams()");
}
}
void CTPCC_OLEDB::Delivery()
{
HRESULT
hr;
int
iTryCount = 0;
IRowset*
pRowset;
LONG
cRows = 1;
// number of rows returned in the rowset
ULONG
cRowsObtained;
HROW
rghRow;
//returned row handles
HROW*
prghRow =
&rghRow;
while (TRUE)
{
try
{

```



```

        // Execute the prepared
command
        hr =
m_pIDeliveryCommand->Execute(NULL, IID_IRowset,
&m_DeliveryExecuteParams, NULL,

        (IUnknown **) &pRowset);
        if (FAILED(hr))
        {

                ThrowError(m_pIDeliveryCommand,
COLEDBERR::eExecute, "Delivery()");
        }

        // Fetch the result row
handle(s)
        hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRow);
        if (FAILED(hr))
        {

                ThrowError(m_pIDeliveryCommand,
COLEDBERR::eGetNextRows, "Delivery()");
        }

        // Fetch the actual row
data by handle
        hr = pRowset-
>GetData(rghRow, m_hDeliveryOutputAccessor,
&m_txn.Delivery);
        if (FAILED(hr))
        {

                ThrowError(m_pIDeliveryCommand,
COLEDBERR::eGetData, "Delivery()");
        }

        // Release row(s)
        hr = pRowset-
>ReleaseRows(cRowsObtained, prghRow, NULL, NULL,
NULL);
        // Release rowset
        hr = pRowset-
>Release();

        m_txn.Delivery.exec_status_code = eOK;

        break;
    }
    catch (COLEDBERR *e)
    {
        if (!(e->m_bDeadLock)
|| (++iTryCount > iMaxRetries))
            throw;

        // hit deadlock;
backoff for increasingly longer period
        delete e;

```

```

        Sleep(10 * iTryCount);
    }
}
    if (iTryCount)
        throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

```

tpcc_oledb.h

```

/* FILE: TPC_C_OLEDB.H
 * Microsoft
 * TPC-C Kit Ver. 4.20.000
 * Copyright
 * Microsoft, 1999-2004
 * Written by
 * Sergey Vasilevskiy
 * All Rights Reserved
 *
 * PURPOSE: Header file for TPC-C txn class
OLE DB implementation.
 */
#pragma once

// need to declare functions for import, unless
define has already been created
// by the DLL's .cpp module for export.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

#define IMAX_SP_NAME_LEN 256 //maximum length of a
stored procedure name with parameters

// Type of parameter and result set column bindings.
enum eBindingType
{
    eInputParameter,
    eOutputParameter,
    eInputOutputParameter,
    eOutputColumn
};

class COLEDBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eQueryInterface,
        // error from QueryInterface
        eCreateSession,
        eCreateCommand,

```

```

        eSetCommandText,
eExecute,

        // = 6

        eCreateAccessor,
ePrepare,
eGetNextRows,
eGetData,
eGetResult

        // = 11
    };

COLEDBERR(LPCTSTR szLoc)
: CBaseErr(szLoc)
{
    m_eAction = eNone;
    m_NativeError = 0;
    m_bDeadLock = FALSE;
    m_OLEDBErrStr = NULL;
};

~COLEDBERR()
{
    if (m_OLEDBErrStr !=
NULL)
        delete []
m_OLEDBErrStr;
};

ACTION m_eAction;
int
m_NativeError;
BOOL m_bDeadLock;
char *m_OLEDBErrStr;

int
ErrorType()
{return ERR_TYPE_OLEDB;};
char*
ErrorTypeStr() { return
"OLEDB"; }

int
ErrorNum()
{return m_NativeError;};
char*
ErrorText() {return
m_OLEDBErrStr;};
int
ErrorAction()
{ return (int)m_eAction; }
};

class CTPCC_OLEDB_ERR : public CBaseErr
{
public:
    enum TPC_C_OLEDB_ERRS
    {
        ERR_WRONG_SP_VERSION =
1, // "Wrong version of stored procs on
database server"

        ERR_INVALID_CUST,
// "Invalid Customer id,name."
        ERR_NO_SUCH_ORDER,
// "No orders found for
customer."

```

```

        ERR_RETRIED_TRANS,
    // "Retries before transaction
succeeded."
    };
    CTPCC_OLEDB_ERR( int iErr ) {
m_errno = iErr; m_iTryCount = 0; };
    CTPCC_OLEDB_ERR( int iErr, int
iTryCount ) { m_errno = iErr; m_iTryCount =
iTryCount; };
        int                m_errno;
        int                m_iTryCount;
    int                ErrorType()
{return ERR_TYPE_TPCC_OLEDB;};
    char*              ErrorTypeStr() { return
"TPCC OLEDB"; }
        int                ErrorNum()
{return m_errno;};
    char*              ErrorText();
};
class DllDecl CTPCC_OLEDB : public CTPCC_BASE
{
    private:
    // declare variables and private
functions here...
        BOOL                m_bDeadlock;
transaction was selected as deadlock victim
        int                m_MaxRetries;
// retry count on deadlock
        DBPROPSET
m_rgInitPropSet;
// initialization property set used to establish a
connection
        DBPROP
m_InitProperties[4];
// individual initialization properties
        IDBCreateSession*
m_pIDBCreateSession;
// session
(connection) interface
        IDBCreateCommand*
m_pIDBCreateCommand;
// SQL
command creation interface
        IMalloc*
m_pIMalloc;
// Needed to release error strings.
        // StockLevel
ICommandText*
m_pIStockLevelCommand;

```

```

        HACCESSOR
m_hStockLevelInputAccessor; // accessor
to bind input parameters
        HACCESSOR
m_hStockLevelOutputAccessor; // accessor
to bind output columns
        DBPARAMS
m_StockLevelExecuteParams;
// parameter structure for Execute
        // NewOrder
// One prepared command for each
possible number of new order line items
        ICommandText*
m_pINewOrderCommand[MAX_OL_NEW_ORDER_ITEMS]
;
        // accessors to bind input
parameters
        // one for each possible number
of new order line items
        HACCESSOR
m_hNewOrderInputAccessor[MAX_OL_NEW_ORDER_I
TEMS];
        // accessor to bind output
columns of the first rowset
        HACCESSOR
m_hNewOrderOutputAccessor[MAX_OL_NEW_ORDER_
ITEMS];
        // accessor to bind output
columns of the second rowset
        HACCESSOR
m_hNewOrderOutputAccessor2[MAX_OL_NEW_ORDER
_ITEMS];
        // parameter structure for
Execute
        DBPARAMS
m_NewOrderExecuteParams[MAX_OL_NEW_ORDER_IT
EMS];
        // Payment
ICommandText*
m_pIPaymentCommand;
        HACCESSOR
m_hPaymentInputAccessor; // accessor
to bind input parameters
        HACCESSOR
m_hPaymentOutputAccessor; // accessor
to bind output columns
        DBPARAMS
m_PaymentExecuteParams;
// parameter structure for Execute
        // OrderStatus
ICommandText*
m_pIOrderStatusCommand;
        HACCESSOR
m_hOrderStatusInputAccessor; // accessor
to bind input parameters
        HACCESSOR
m_hOrderStatusOutputAccessor; // accessor
to bind output columns

```

```

        HACCESSOR
m_hOrderStatusOutputAccessor2;
// accessor to bind output columns
        DBPARAMS
m_OrderStatusExecuteParams;
// parameter structure for Execute
        // Delivery
ICommandText*
m_pIDeliveryCommand;
        HACCESSOR
m_hDeliveryInputAccessor; // accessor
to bind input parameters
        HACCESSOR
m_hDeliveryOutputAccessor; // accessor
to bind output columns
        DBPARAMS
m_DeliveryExecuteParams;
// parameter structure for Execute
        wchar_t
m_szSPPrefix[32];
// stored
procedures prefix
        // new-order specific fields
        int
m_no_commit_flag;
        void ThrowError( IUnknown*
pObjectWithError, COLEDBERR::ACTION eAction, LPCTSTR
szLocation );
        void CheckSPVersion();
        void InitNewOrderParams();
        void InitPaymentParams();
        void InitDeliveryParams();
        void InitStockLevelParams();
        void InitOrderStatusParams();
        // Helper function to create and
prepare a command
        void CreateCommand(wchar_t*
szSQLCommand, ICommandText** ppICommandText);
        // Helper function to prepare a
command
        void PrepareCommand(ICommandText*
pICommand);
        // Helper function to fill one
binding
        // Used for both input parameter
and output column bindings
        void SetBinding(DBBINDING*
pDBBinding, size_t obValue, size_t cbMaxLen, DBTYPE
wType);
        // Helper function to initialize
an array of bindings

```

```

        void InitBindings(DBBINDING*
pDBBindings, int iCount, eBindingType BindingType);

        union
        {
                NEW_ORDER_DATA
NewOrder;
                PAYMENT_DATA
Payment;
                DELIVERY_DATA
Delivery;
                STOCK_LEVEL_DATA
StockLevel;
                ORDER_STATUS_DATA
OrderStatus;
        }
m_txn;

public:
        CTPCC_OLEDB(LPCSTR szServer,
LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost,
LPCSTR szDatabase, LPCWSTR szSPPrefix);
        ~CTPCC_OLEDB(void);

        inline PNEW_ORDER_DATA
BuffAddr_NewOrder() { return
&m_txn.NewOrder; };
        inline PPAYMENT_DATA
BuffAddr_Payment() { return
&m_txn.Payment; };
        inline PDELIVERY_DATA
BuffAddr_Delivery() { return
&m_txn.Delivery; };
        inline PSTOCK_LEVEL_DATA
BuffAddr_StockLevel() { return
&m_txn.StockLevel; };
        inline PORDER_STATUS_DATA
BuffAddr_OrderStatus() { return
&m_txn.OrderStatus; };

        void NewOrder        ();
        void Payment        ();
        void Delivery        ();
        void StockLevel     ();
        void OrderStatus    ();
};

// wrapper routine for class constructor
extern "C" DllDecl CTPCC_OLEDB* CTPCC_OLEDB_new
( LPCSTR szServer, LPCSTR szUser, LPCSTR
szPassword, LPCSTR szHost, LPCSTR szDatabase, LPCWSTR
szSPPrefix );

typedef CTPCC_OLEDB* (TYPE_CTPCC_OLEDB) (LPCSTR,
LPCSTR, LPCSTR, LPCSTR, LPCSTR, LPCWSTR);

```

tpcc_com_all ***resource.h***

```

//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by tpcc_com_all.rc
//
#define IDS_PROJNAME            100
#define IDR_TPCC                101
#define IDR_NEWORDER           102
#define IDR_ORDERSTATUS        103
#define IDR_PAYMENT            104
#define IDR_STOCKLEVEL         105

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        202
#define _APS_NEXT_COMMAND_VALUE        32768
#define _APS_NEXT_CONTROL_VALUE        201
#define _APS_NEXT_SYMED_VALUE          106
#endif
#endif

```

isapi_dll ***resource.h***

```

//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by tpcc.rc
//
#define IDD_DIALOG1            101

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        102
#define _APS_NEXT_COMMAND_VALUE        40001
#define _APS_NEXT_CONTROL_VALUE        1000
#define _APS_NEXT_SYMED_VALUE          101
#endif
#endif

```

Appendix B: Database Design

The TPC-C database was created with the following Transact-SQL scripts:

backup.sql

```
-----
--
-- File: BACKUP.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.61
--
-- Copyright Microsoft, 2005
--
-----
DECLARE @startdate DATETIME,
        @enddate DATETIME
SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,
21)
DUMP DATABASE tpcc TO tpccback1, tpccback2,
tpccback3, tpccback4 WITH init, stats = 1
SELECT @enddate = GETDATE()
SELECT 'End date: ',
       CONVERT(VARCHAR(30),@enddate, 21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate,
@enddate)
GO
```

backupdev.sql

```
-----
--
--
```

```
-- File: BACKUPDEV.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.61
--
-- Copyright Microsoft, 2005
--
-----
```

```
USE master
GO
```

```
-----
-- create backup devices
-----
EXEC sp_addumpdevice
'disk','tpccback1','W:\tpccback1.dmp'
GO
EXEC sp_addumpdevice
'disk','tpccback2','X:\tpccback2.dmp'
GO
EXEC sp_addumpdevice
'disk','tpccback3','Y:\tpccback3.dmp'
GO
EXEC sp_addumpdevice
'disk','tpccback4','Z:\tpccback4.dmp'
GO
```

config.sql

```
-- File: CONFIG.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.00
-- Copyright Microsoft, 1996
-- Purpose: Collects SQL Server configuration
parameters
print " "
select convert(char(30), getdate(),9)
print " "
go
sp_configure "show advanced",1
go
reconfigure with override
go
exec sp_configure "affinity mask", 15
exec sp_configure "cost threshold for parallelism",
5
exec sp_configure "index create memory", 704
exec sp_configure "lightweight pooling", 1
exec sp_configure "awe enabled", 0
exec sp_configure "c2 audit mode", 0
exec sp_configure "locks", 0
exec sp_configure "max degree of parallelism", 1
exec sp_configure "max server memory", 31700
exec sp_configure "max worker threads", 496
exec sp_configure "min memory per query", 512
```

```
exec sp_configure "min server memory", 0
exec sp_configure "nested triggers", 1
exec sp_configure "network packet size", 2048
exec sp_configure "open objects", 0
exec sp_configure "priority boost", 1
exec sp_configure "recovery interval", 32767
exec sp_configure "set working set size", 0
exec sp_configure "user connections", 0
exec sp_configure "default trace", 1
```

```
go
```

```
reconfigure with override
go
sp_configure
go
```

createdb.sql

```
-----
--
-- File: CREATEDB.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.61
--
-- Copyright Microsoft, 2005
--
-----
SET ANSI_NULL_DFLT_OFF ON
GO
USE master
GO
-----
-- Create temporary table for timing
-----
IF EXISTS( SELECT name FROM sysobjects WHERE name =
'tpcc_timer' )
DROP TABLE tpcc_timer
GO
CREATE TABLE tpcc_timer
(start_date CHAR(30),
end_date CHAR(30))
GO
INSERT INTO tpcc_timer VALUES(0,0)
GO
-----
-- Store starting time
```

```

-----
UPDATE tpcc_timer
SET start_date = (SELECT CONVERT(CHAR(30),
GETDATE(), 21))
GO

-----
-- create main database files
-----
CREATE DATABASE tpcc
ON PRIMARY
(
    NAME = MSSQL_tpcc_root,
    FILENAME = 'c:\MSSQL_tpcc_root.mdf',
    SIZE = 8MB,
    FILEGROWTH = 0),
FILEGROUP MSSQL_cs_fg
(
    NAME = MSSQL_cs1,
    FILENAME = 'c:\cs\cs1\',
    SIZE = 64000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs2,
    FILENAME = 'c:\cs\cs2\',
    SIZE = 64000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs3,
    FILENAME = 'c:\cs\cs3\',
    SIZE = 64000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs4,
    FILENAME = 'c:\cs\cs4\',
    SIZE = 64000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs5,
    FILENAME = 'c:\cs\cs5\',
    SIZE = 64000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs6,
    FILENAME = 'c:\cs\cs6\',
    SIZE = 64000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs7,
    FILENAME = 'c:\cs\cs7\',
    SIZE = 64000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs8,
    FILENAME = 'c:\cs\cs8\',
    SIZE = 64000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs9,
    FILENAME = 'c:\cs\cs9\',
    SIZE = 64000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs10,
    FILENAME = 'c:\cs\cs10\',
    SIZE = 64000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs11,
    FILENAME = 'c:\cs\cs11\',
    SIZE = 64000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs12,
    FILENAME = 'c:\cs\cs12\',

```

```

    SIZE = 64000MB,
    FILEGROWTH = 0),
FILEGROUP MSSQL_misc_fg
(
    NAME = MSSQL_misc1,
    FILENAME = 'c:\misc\misc1\',
    SIZE = 34000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc2,
    FILENAME = 'c:\misc\misc2\',
    SIZE = 34000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc3,
    FILENAME = 'c:\misc\misc3\',
    SIZE = 34000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc4,
    FILENAME = 'c:\misc\misc4\',
    SIZE = 34000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc5,
    FILENAME = 'c:\misc\misc5\',
    SIZE = 34000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc6,
    FILENAME = 'c:\misc\misc6\',
    SIZE = 34000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc7,
    FILENAME = 'c:\misc\misc7\',
    SIZE = 34000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc8,
    FILENAME = 'c:\misc\misc8\',
    SIZE = 34000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc9,
    FILENAME = 'c:\misc\misc9\',
    SIZE = 34000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc10,
    FILENAME = 'c:\misc\misc10\',
    SIZE = 34000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc11,
    FILENAME = 'c:\misc\misc11\',
    SIZE = 34000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc12,
    FILENAME = 'c:\misc\misc12\',
    SIZE = 34000MB,
    FILEGROWTH = 0)
LOG ON
(
    NAME = MSSQL_tpcc_log,
    FILENAME = 'E:',
    SIZE = 520000MB,
    FILEGROWTH = 0)
COLLATE Latin1_General_BIN
GO

```

```

-----
-- Store ending time
-----
UPDATE tpcc_timer
SET end_date = (SELECT CONVERT(CHAR(30),
GETDATE(), 21))
GO

SELECT DATEDIFF(second, (SELECT start_date FROM
tpcc_timer), (SELECT end_date FROM tpcc_timer))
GO

-----
-- remove temporary table
-----
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_timer' )
DROP TABLE tpcc_timer
GO

dbopt1.sql
-----
--
-- File: DBOPT1.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
-- Sets database options for load
--
-----
USE master
GO

ALTER DATABASE tpcc SET RECOVERY BULK_LOGGED
GO

EXEC sp_dboption tpcc, 'trunc. log on chkpt.', TRUE
GO

ALTER DATABASE tpcc SET TORN_PAGE_DETECTION OFF
GO

ALTER DATABASE tpcc SET PAGE_VERIFY NONE
GO

USE tpcc
GO

CHECKPOINT
GO

```

dbopt2.sql

```
-----
--
-- File:      DBOPT2.SQL
--
--           Microsoft TPC-C Benchmark Kit Ver. 4.63
--
--           Copyright Microsoft, 2005
--
--           Sets database options after load
--
-----
ALTER DATABASE tpcc SET RECOVERY FULL
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE @msg          varchar(50)

-----
--           OPTIONS FOR SQL SERVER 2000
-- Set option values for user-defined indexes
-----

SET @msg = ' '
PRINT @msg
SET @msg = 'Setting SQL Server indexoptions'
PRINT @msg
SET @msg = ' '
PRINT @msg

EXEC sp_indexoption 'customer',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'district',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'warehouse',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'stock',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'order_line',
'DisAllowRowLocks', TRUE
EXEC sp_indexoption 'orders',
'DisAllowRowLocks', TRUE
```

```
EXEC sp_indexoption 'new_order',
'DisAllowRowLocks', TRUE
EXEC sp_indexoption 'item',
'DisAllowRowLocks', TRUE
EXEC sp_indexoption 'item',
'DisAllowPageLocks', TRUE
GO

Print ' '
Print '*****'
Print 'Pre-specified Locking Hierarchy:'
Print '      Lockflag = 0 ==> No pre-specified
hierarchy'
Print '      Lockflag = 1 ==> Lock at Page-level then
Table-level'
Print '      Lockflag = 2 ==> Lock at Row-level then
Table-level'
Print '      Lockflag = 3 ==> Lock at Table-level'
Print ' '

SELECT name,
lockflags
FROM sysindexes
WHERE object_id('warehouse') = id OR
object_id('district') = id OR
object_id('customer') = id OR
object_id('stock') = id OR
object_id('orders') = id OR
object_id('order_line') = id OR
object_id('history') = id OR
object_id('new_order') = id OR
object_id('item') = id

ORDER BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc, 'auto update
statistics', FALSE
EXEC sp_dboption tpcc, 'auto create
statistics', FALSE
GO

DECLARE @db_id int,
@tbl_id int

SET @db_id = DB_ID('tpcc')
SET @tbl_id = OBJECT_ID('tpcc..warehouse')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..district')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..new_order')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..item')
DBCC PINTABLE (@db_id, @tbl_id)
GO
```

delivery.sql

```
-----
--
-- File:      DELIVERY.SQL
--
--           Microsoft TPC-C Benchmark Kit Ver. 4.63
--
--           Copyright Microsoft, 2005
--
--           Creates delivery stored procedure
--
--           Interface Level: 4.20.000
--
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_delivery' )
DROP PROCEDURE tpcc_delivery
GO

CREATE PROC tpcc_delivery
@w_id int,
@o_carrier_id smallint

AS

DECLARE @d_id tinyint,
@o_id int,
@c_id int,
@tTotal money,
@oid1 int,
@oid2 int,
@oid3 int,
@oid4 int,
@oid5 int,
@oid6 int,
@oid7 int,
@oid8 int,
@oid9 int,
@oid10 int

SELECT @d_id = 0
```

```

BEGIN TRANSACTION d
WHILE (@d_id < 10)
  BEGIN
    SELECT @d_id = @d_id + 1,
           @total = 0,
           @o_id = 0

    SELECT TOP 1
           @o_id = no_o_id
    FROM new_order WITH (serializable
updlock)
    WHERE no_w_id = @w_id AND
           no_d_id = @d_id
    ORDER BY no_o_id ASC

    IF (@@rowcount <> 0)
      BEGIN
        -- claim the order for this district
        DELETE new_order
        WHERE no_w_id = @w_id AND
              no_d_id = @d_id AND
              no_o_id = @o_id

        -- set carrier_id on this order (and get
customer id)
        UPDATE orders
        SET o_carrier_id = @o_carrier_id,
            @c_id = o_c_id
        WHERE o_w_id = @w_id AND
              o_d_id = @d_id AND
              o_id = @o_id

        -- set date in all lineitems for this
order (and sum amounts)
        UPDATE order_line
        SET ol_delivery_d = GETDATE(),
            @total = @total +
ol_amount
        WHERE ol_w_id = @w_id AND
              ol_d_id = @d_id AND
              ol_o_id = @o_id

        -- accumulate lineitem amounts for this
order into customer
        UPDATE customer
        SET c_balance = c_balance +
@total,
            c_delivery_cnt = c_delivery_cnt
+ 1
        WHERE c_w_id = @w_id AND
              c_d_id = @d_id AND
              c_id = @c_id

      END

      SELECT @oid1 = CASE @d_id WHEN 1 THEN
@o_id ELSE @oid1 END,
           @oid2 = CASE @d_id WHEN 2 THEN
@o_id ELSE @oid2 END,
           @oid3 = CASE @d_id WHEN 3 THEN
@o_id ELSE @oid3 END,
           @oid4 = CASE @d_id WHEN 4 THEN
@o_id ELSE @oid4 END,

```

```

           @oid5 = CASE @d_id WHEN 5 THEN
@o_id ELSE @oid5 END,
           @oid6 = CASE @d_id WHEN 6 THEN
@o_id ELSE @oid6 END,
           @oid7 = CASE @d_id WHEN 7 THEN
@o_id ELSE @oid7 END,
           @oid8 = CASE @d_id WHEN 8 THEN
@o_id ELSE @oid8 END,
           @oid9 = CASE @d_id WHEN 9 THEN
@o_id ELSE @oid9 END,
           @oid10 = CASE @d_id WHEN 10 THEN
@o_id ELSE @oid10 END
        END

      COMMIT TRANSACTION d

      -- return delivery data to client

      SELECT @oid1,
             @oid2,
             @oid3,
             @oid4,
             @oid5,
             @oid6,
             @oid7,
             @oid8,
             @oid9,
             @oid10
      GO

      SET QUOTED_IDENTIFIER OFF
      GO

      SET ANSI_NULLS ON
      GO

getargs.c

// File: GETARGS.C
// Microsoft
// TPC-C Kit Ver. 4.51
// Copyright
// Microsoft, 1996, 1997, 1998, 1999, 2000, 2001, 2002,
// 2003
// Purpose: Source file for command line
// processing

// Includes
#include "tpcc.h"

//=====
//
// Function name: GetArgsLoader
//
//=====

```

```

void GetArgsLoader(int argc, char **argv,
TPCC_LDR_ARGS *pargs)
{
    int i;
    char *ptr;

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoader()\n",
(int) GetCurrentThreadId());
#endif

    /* init args struct with some useful values */
    pargs->server = SERVER;
    pargs->user = USER;
    pargs->password = PASSWORD;
    pargs->database = DATABASE;
    pargs->batch = BATCH;
    pargs->num_warehouses = UNDEF;
    pargs->tables_all =
TRUE;
    pargs->table_item =
FALSE;
    pargs->table_warehouse =
FALSE;
    pargs->table_customer =
FALSE;
    pargs->table_orders =
FALSE;
    pargs->loader_res_file =
LOADER_RES_FILE;
    pargs->log_path =
LOADER_LOG_PATH;
    pargs->pack_size =
DEFLDPACKSIZE;
    pargs->starting_warehouse =
DEF_STARTING_WAREHOUSE;
    pargs->build_index =
BUILD_INDEX;
    pargs->index_order =
INDEX_ORDER;
    pargs->index_script_path =
INDEX_SCRIPT_PATH;
    pargs->scale_down =
SCALE_DOWN;

    /* check for zero command line args */
    if ( argc == 1 )
        GetArgsLoaderUsage();

    for ( i = 1; i < argc; ++i )
    {
        if ( argv[i][0] != '-' &&
argv[i][0] != '/' )
        {
            printf("\nUnrecognized command");
            GetArgsLoaderUsage();
            exit(1);
        }

        ptr = argv[i];

        switch ( ptr[1] )
        {

```

```

        case '?': /* Fall through */
        GetArgsLoaderUsage();
                break;

        case 'D':
                pargs->database = ptr+2;
                break;

        case 'P':
                pargs->password = ptr+2;
                break;

        case 'S':
                pargs->server = ptr+2;
                break;

        case 'U':
                pargs->user = ptr+2;
                break;

        case 'b':
                pargs->batch = atol(ptr+2);
                break;

        case 'W':
                pargs->num_warehouses = atol(ptr+2);
                break;

        case 's':
                pargs->starting_warehouse = atol(ptr+2);
                break;

        case 't':
                {
                pargs->tables_all = FALSE;
                if (strcmp(ptr+2,"item") == 0)
                pargs->table_item = TRUE;
                else if (strcmp(ptr+2,"warehouse") == 0)
                pargs->table_warehouse = TRUE;
                else if (strcmp(ptr+2,"customer") == 0)
                pargs->table_customer = TRUE;
                else if (strcmp(ptr+2,"orders") == 0)
                pargs->table_orders = TRUE;
                else
                {

```

```

                printf("\nUnrecognized command");
                GetArgsLoaderUsage();
                exit(1);
                break;
                }

        case 'f':
                pargs->loader_res_file = ptr+2;
                break;

        case 'L':
                pargs->log_path = ptr+2;
                break;

        case 'p':
                pargs->pack_size = atol(ptr+2);
                break;

        case 'i':
                pargs->build_index = atol(ptr+2);
                break;

        case 'o':
                pargs->index_order = atol(ptr+2);
                break;

        case 'c':
                pargs->scale_down = atol(ptr+2);
                break;

        case 'd':
                pargs->index_script_path = ptr+2;
                break;

        default:
                GetArgsLoaderUsage();
                exit(-1);
                break;
        }

        /* check for required args */
        if (pargs->num_warehouses == UNDEF)
        {
                printf("Number of Warehouses is
                required\n");
                exit(-2);
        }

        return;

```

```

        }

        //=====
        //
        // Function name: GetArgsLoaderUsage
        //
        //=====
        void GetArgsLoaderUsage()
        {
        #ifdef DEBUG
                printf("[%ld]DBG: Entering
                GetArgsLoaderUsage()\n", (int) GetCurrentThreadId());
        #endif

                printf("TPCCCLR:\n\n");
                printf("Parameter
                Default\n");
                printf("-----\n");
                printf("-W Number of Warehouses to Load
                Required \n");
                printf("-S Server
                %s\n", SERVER);
                printf("-U Username
                %s\n", USER);
                printf("-P Password
                %s\n", PASSWORD);
                printf("-D Database
                %s\n", DATABASE);
                printf("-b Batch Size
                %ld\n", (long) BATCH);
                printf("-p TDS packet size
                %ld\n", (long) DEFLDPPACKSIZE);
                printf("-L Loader BCP Log Path
                %s\n", LOADER_LOG_PATH);
                printf("-f Loader Results Output Filename
                %s\n", LOADER_RES_FILE);
                printf("-s Starting Warehouse
                %ld\n", (long) DEF_STARTING_WAREHOUSE);
                printf("-i Build Option (data = 0, data and
                index = 1)
                %ld\n", (long) BUILD_INDEX);
                printf("-o Cluster Index Build Order
                (before = 1, after = 0) %ld\n", (long) INDEX_ORDER);
                printf("-c Build Scaled Database (normal =
                0, tiny = 1) %ld\n", (long) SCALE_DOWN);
                printf("-d Index Script Path
                %s\n", INDEX_SCRIPT_PATH);
                printf("-t Table to Load
                all tables \n");
                printf(" [item|warehouse|customer|orders]\n");
                printf(" Notes: \n");
                printf(" - the '-t' parameter may be included
                multiple times to \n");
                printf(" specify multiple tables to be
                loaded \n");
                printf(" - 'item' loads ITEM table \n");

```



```

        printf("    - 'warehouse' loads WAREHOUSE,
DISTRICT, and STOCK tables \n");
        printf("    - 'customer' loads CUSTOMER and
HISTORY tables \n");
        printf("    - 'orders' load NEW-ORDER, ORDERS,
ORDER-LINE tables \n");

        printf("\nNote: Command line switches are
case sensitive.\n");

        exit(0);
}

```

idxcuscl.sql

```

-----
--
--
-- File:    IDXCUSCL.SQL
--
--          Microsoft TPC-C Benchmark Kit Ver. 4.63
--
--          Copyright Microsoft, 2005
--
--          Creates clustered index on customer table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name =
'customer_c1' )
    DROP INDEX customer.customer_c1

CREATE UNIQUE CLUSTERED INDEX customer_c1 ON
customer(c_w_id, c_d_id, c_id)
ON MSSQL_cs_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxcusnc.sql

```

-----
--
--
-- File:    IDXCUSNC.SQL
--
--          Microsoft TPC-C Benchmark Kit Ver. 4.63
--
--          Copyright Microsoft, 2005
--
--          Creates non-clustered index on customer
table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name =
'customer_nc1' )
    DROP INDEX customer.customer_nc1

CREATE UNIQUE NONCLUSTERED INDEX customer_nc1 ON
customer(c_w_id, c_d_id, c_last, c_first, c_id)
ON MSSQL_cs_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxdiscl.sql

```

-----
--
--
-- File:    IDXDISCL.SQL
--
--          Microsoft TPC-C Benchmark Kit Ver. 4.63
--
--          Copyright Microsoft, 2005
--
--          Creates clustered index on district table
--
-----

```

```

-----
--
--
-- USE tpcc
-- GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name =
'district_c1' )
    DROP INDEX district.district_c1

CREATE UNIQUE CLUSTERED INDEX district_c1 ON
district(d_w_id, d_id)
WITH FILLFACTOR=100 ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxhiscl.sql

```

-----
--
--
-- File:    IDXHISCL.SQL
--
--          Microsoft TPC-C Benchmark Kit Ver. 4.63
--
--          Copyright Microsoft, 2005
--
--          Creates clustered index on history table
--
--
--          CAUTION: This index is only beneficial
for systems --
--          CAUTION: with 8 or more processors.
--
--          CAUTION: It may negatively impact
performance on --
--          CAUTION: systems with less than 8
processors. --
--
-----
USE tpcc
GO

```

```

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name =
            'history_c1' )
    DROP INDEX history.history_c1

CREATE UNIQUE CLUSTERED INDEX history_c1 ON
history(h_c_w_id, h_date, h_c_d_id, h_c_id, h_amount)
ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxitmcl.sql

```

-----
--
-- File:   IDXITMCL.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
--
-- Creates clustered index on item table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name =
            'item_c1' )
    DROP INDEX item.item_c1

CREATE UNIQUE CLUSTERED INDEX item_c1 ON item(i_id)
ON MSSQL_misc_fg

SELECT @enddate = GETDATE()

```

```

SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxnodcl.sql

```

-----
--
-- File:   IDXNODCL.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
--
-- Creates clustered index on new-order
table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name =
            'new_order_c1' )
    DROP INDEX new_order.new_order_c1

CREATE UNIQUE CLUSTERED INDEX new_order_c1 ON
new_order(no_w_id, no_d_id, no_o_id)
ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxodlcl.sql

```

-----
--
--

```

```

-- File:   IDXODLCL.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
--
-- Creates clustered index on order-line
table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name =
            'order_line_c1' )
    DROP INDEX order_line.order_line_c1

CREATE UNIQUE CLUSTERED INDEX order_line_c1 ON
order_line(ol_w_id, ol_d_id, ol_o_id, ol_number)
ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxordcl.sql

```

-----
--
-- File:   IDXORDCL.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
--
-- Creates clustered index on orders table
--
-----
USE tpcc

```

```

GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT  @startdate = GETDATE()
SELECT  'Start date:',
        CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name =
'orders_c1' )
        DROP INDEX orders.orders_c1

CREATE UNIQUE CLUSTERED INDEX orders_c1 ON
orders(o_w_id, o_d_id, o_id)
ON MSSQL_misc_fg

SELECT  @enddate = GETDATE()
SELECT  'End date:',
        CONVERT(VARCHAR(30),@enddate,21)
SELECT  'Elapsed time (in seconds): ',
        DATEDIFF(second, @startdate, @enddate)
GO

```

idxstkcl.sql

```

-----
--
-- File:   IDXSTKCL.SQL
--
--       Microsoft TPC-C Benchmark Kit Ver. 4.63
--
--       Copyright Microsoft, 2005
--
--       Creates clustered index on stock table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT  @startdate = GETDATE()
SELECT  'Start date:',
        CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name =
'stock_c1' )
        DROP INDEX stock.stock_c1

CREATE UNIQUE CLUSTERED INDEX stock_c1 ON
stock(s_i_id, s_w_id)

```

```

ON MSSQL_cs_fg

SELECT  @enddate = GETDATE()
SELECT  'End date:',
        CONVERT(VARCHAR(30),@enddate,21)
SELECT  'Elapsed time (in seconds): ',
        DATEDIFF(second, @startdate, @enddate)
GO

```

idxwarcl.sql

```

-----
--
-- File:   IDXWARCL.SQL
--
--       Microsoft TPC-C Benchmark Kit Ver. 4.63
--
--       Copyright Microsoft, 2005
--
--       Creates clustered index on warehouse
table --
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT  @startdate = GETDATE()
SELECT  'Start date:',
        CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name =
'warehouse_c1' )
        DROP INDEX warehouse.warehouse_c1

CREATE UNIQUE CLUSTERED INDEX warehouse_c1 ON
warehouse(w_id)
WITH FILLFACTOR=100 ON MSSQL_misc_fg

SELECT  @enddate = GETDATE()
SELECT  'End date:',
        CONVERT(VARCHAR(30),@enddate,21)
SELECT  'Elapsed time (in seconds): ',
        DATEDIFF(second, @startdate, @enddate)
GO

```

neword.sql

```

--
-- File:   NEWORD.SQL
--
--       Microsoft TPC-C Benchmark Kit Ver. 4.63
--
--       Copyright Microsoft, 2005
--
--       Creates neworder stored procedure
--
--       Interface Level:   4.20.000
--
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_neworder' )
        DROP PROCEDURE tpcc_neworder
GO

CREATE PROCEDURE tpcc_neworder
        @w_id      int,
        @d_id      tinyint,
        @c_id      int,
        @o_ol_cnt  tinyint,
        @o_all_local tinyint,
        @i_id1     int = 0, @s_w_id1
int = 0, @ol_qty1 smallint = 0,
        @i_id2     int = 0, @s_w_id2
int = 0, @ol_qty2 smallint = 0,
        @i_id3     int = 0, @s_w_id3
int = 0, @ol_qty3 smallint = 0,
        @i_id4     int = 0, @s_w_id4
int = 0, @ol_qty4 smallint = 0,
        @i_id5     int = 0, @s_w_id5
int = 0, @ol_qty5 smallint = 0,
        @i_id6     int = 0, @s_w_id6
int = 0, @ol_qty6 smallint = 0,
        @i_id7     int = 0, @s_w_id7
int = 0, @ol_qty7 smallint = 0,
        @i_id8     int = 0, @s_w_id8
int = 0, @ol_qty8 smallint = 0,
        @i_id9     int = 0, @s_w_id9
int = 0, @ol_qty9 smallint = 0,
        @i_id10    int = 0, @s_w_id10
int = 0, @ol_qty10 smallint = 0,
        @i_id11    int = 0, @s_w_id11
int = 0, @ol_qty11 smallint = 0,

```

```

        @i_id12 int = 0, @s_w_id12
int = 0, @ol_qty12 smallint = 0,
        @i_id13 int = 0, @s_w_id13
int = 0, @ol_qty13 smallint = 0,
        @i_id14 int = 0, @s_w_id14
int = 0, @ol_qty14 smallint = 0,
        @i_id15 int = 0, @s_w_id15
int = 0, @ol_qty15 smallint = 0

AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN
BEGIN TRANSACTION n
-----
-- get district tax and next available order id and
update
-- plus initialize local variables
-----
UPDATE district
SET
    @d_tax          = d_tax,
    @o_id           = d_next_o_id,
    d_next_o_id     = d_next_o_id + 1,
    @o_entry_d      = GETDATE(),
    @li_no          = 0,
    @commit_flag    = 1
WHERE
    d_w_id          = @w_id AND
    d_id            = @d_id

-----
-- process orderlines
-----
WHILE (@li_no < @@ol_cnt)
BEGIN
    SELECT @li_no = @li_no + 1

```

```

-- set i_id, s_w_id, and qty for this lineitem
-----
SELECT @li_id = CASE @li_no
    WHEN 1 THEN @i_id1
    WHEN 2 THEN @i_id2
    WHEN 3 THEN @i_id3
    WHEN 4 THEN @i_id4
    WHEN 5 THEN @i_id5
    WHEN 6 THEN @i_id6
    WHEN 7 THEN @i_id7
    WHEN 8 THEN @i_id8
    WHEN 9 THEN @i_id9
    WHEN 10 THEN @i_id10
    WHEN 11 THEN @i_id11
    WHEN 12 THEN @i_id12
    WHEN 13 THEN @i_id13
    WHEN 14 THEN @i_id14
    WHEN 15 THEN @i_id15
END,

@li_s_w_id = CASE @li_no
    WHEN 1 THEN @s_w_id1
    WHEN 2 THEN @s_w_id2
    WHEN 3 THEN @s_w_id3
    WHEN 4 THEN @s_w_id4
    WHEN 5 THEN @s_w_id5
    WHEN 6 THEN @s_w_id6
    WHEN 7 THEN @s_w_id7
    WHEN 8 THEN @s_w_id8
    WHEN 9 THEN @s_w_id9
    WHEN 10 THEN
    WHEN 11 THEN
    WHEN 12 THEN
    WHEN 13 THEN
    WHEN 14 THEN
    WHEN 15 THEN

END,

@li_qty = CASE @li_no
    WHEN 1 THEN @ol_qty1
    WHEN 2 THEN @ol_qty2
    WHEN 3 THEN @ol_qty3
    WHEN 4 THEN @ol_qty4
    WHEN 5 THEN @ol_qty5
    WHEN 6 THEN @ol_qty6
    WHEN 7 THEN @ol_qty7
    WHEN 8 THEN @ol_qty8
    WHEN 9 THEN @ol_qty9
    WHEN 10 THEN
    WHEN 11 THEN
    WHEN 12 THEN
    WHEN 13 THEN
    WHEN 14 THEN
    WHEN 15 THEN

```

```

    WHEN 14 THEN
    WHEN 15 THEN
END

-----
-- get item data (no one updates item)
-----
SELECT @i_price = i_price,
       @i_name  = i_name,
       @i_data  = i_data
FROM item WITH (repeatableread)
WHERE i_id = @li_id

-----
-- update stock values
-----
UPDATE stock
SET
    s_ytd          = s_ytd + @li_qty,
    @s_quantity    = s_quantity -
s_quantity - @li_qty +
    CASE WHEN
(s_quantity - @li_qty < 10) THEN 91 ELSE 0 END,
    s_order_cnt    = s_order_cnt + 1,
    s_remote_cnt   = s_remote_cnt +
    CASE WHEN
(@li_s_w_id = @w_id) THEN 0 ELSE 1 END,
    @s_data        = s_data,
    @s_dist        = CASE @d_id
    WHEN 1 THEN
s_dist_01
    WHEN 2 THEN
s_dist_02
    WHEN 3 THEN
s_dist_03
    WHEN 4 THEN
s_dist_04
    WHEN 5 THEN
s_dist_05
    WHEN 6 THEN
s_dist_06
    WHEN 7 THEN
s_dist_07
    WHEN 8 THEN
s_dist_08
    WHEN 9 THEN
s_dist_09
    WHEN 10 THEN
s_dist_10
    END
WHERE
    s_i_id         = @li_id AND
    s_w_id         = @li_s_w_id

-----
-- if there actually is a stock (and item) with
these ids, go to work
-----
IF (@@rowcount > 0)
BEGIN

```

```

-----
-- insert order_line data (using data from item and
stock)
-----
INSERT INTO order_line VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @li_no,
                                @li_id,
                                'dec 31,
1899',
                                @i_price
* @li_qty,
@li_s_w_id,
                                @li_qty,
@s_dist)

-----
-- send line-item data to client
-----
SELECT @i_name,
       @s_quantity,
       b_g = CASE WHEN (
(patindex('%ORIGINAL%',@i_data) > 0) AND
(patindex('%ORIGINAL%',@s_data) > 0) )
THEN 'B' ELSE 'G' END,
       @i_price,
       @i_price * @li_qty
END
ELSE
BEGIN
-----
-- no item (or stock) found - triggers rollback
condition
-----
SELECT '',0, '',0,0
SELECT @commit_flag = 0
END

-----
-- get customer last name, discount, and credit
rating
-----
SELECT @c_last = c_last,
       @c_discount = c_discount,
       @c_credit = c_credit,
       @c_id_local = c_id
FROM customer WITH (repeatableread)
WHERE c_id = @c_id AND
      c_w_id = @w_id AND
      c_d_id = @d_id

```

```

-----
-- insert fresh row into orders table
-----
INSERT INTO orders VALUES ( @o_id,
                             @d_id,
                             @w_id,
                             @c_id_local,
                             0,
                             @o_ol_cnt,
                             @o_all_local,
                             @o_entry_d)

-----
-- insert corresponding row into new-order table
-----
INSERT INTO new_order VALUES ( @o_id,
                                @d_id,
                                @w_id)

-----
-- select warehouse tax
-----
SELECT @w_tax = w_tax
FROM warehouse WITH (repeatableread)
WHERE w_id = @w_id

IF (@commit_flag = 1)

COMMIT TRANSACTION n

ELSE

-----
-- all that work for nuthn!!!
-----
ROLLBACK TRANSACTION n

-----
-- return order data to client
-----
SELECT @w_tax,
       @d_tax,
       @o_id,
       @c_last,
       @c_discount,
       @c_credit,
       @o_entry_d,
       @commit_flag

END
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

null-txns.sql

```

--
-- File: NULL-TXNS.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
-- This script will create stored procs
which
--
-- accept the same parameters and return
correctly
--
-- formed results sets to match the standard
TPC-C
--
-- stored procs. Of course, the advantage
is that
--
-- these stored procs place almost no load
on
--
-- SQL Server and do not require a database.
--
-- Interface Level: 4.10.000
--
-----
USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_delivery' )
DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_neworder' )
DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_orderstatus' )
DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_payment' )
DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_stocklevel' )
DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_version' )
DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'order_line_null' )
DROP PROCEDURE order_line_null
GO

CREATE PROCEDURE tpcc_delivery

```

```

                @w_id          int,
                @o_carrier_id  smallint
AS
DECLARE @d_id    tinyint,
        @o_id    int,
        @c_id    int,
        @total   numeric(12,2),
        @oid1    int,
        @oid2    int,
        @oid3    int,
        @oid4    int,
        @oid5    int,
        @oid6    int,
        @oid7    int,
        @oid8    int,
        @oid9    int,
        @oid10   int,
        @delaytime varchar(30)

-----
-- uniform random delay of 0 - 1 second; avg = 0.50
-----
SELECT @delaytime = '00:00:0' +
CAST(CAST((RAND()*1.00) AS decimal(4,3)) AS char(5))

WAITFOR delay @delaytime

SELECT 3001, 3001, 3001, 3001, 3001, 3001, 3001,
3001, 3001, 3001
GO

CREATE PROCEDURE tpcc_neworder
        @w_id          int,
        @d_id          tinyint,
        @c_id          int,
        @o_ol_cnt      tinyint,
        @o_all_local   tinyint,
        @i_id1 int = 0, @s_w_id1 int
= 0, @ol_qty1 smallint = 0,
        @i_id2 int = 0, @s_w_id2 int
= 0, @ol_qty2 smallint = 0,
        @i_id3 int = 0, @s_w_id3 int
= 0, @ol_qty3 smallint = 0,
        @i_id4 int = 0, @s_w_id4 int
= 0, @ol_qty4 smallint = 0,
        @i_id5 int = 0, @s_w_id5 int
= 0, @ol_qty5 smallint = 0,
        @i_id6 int = 0, @s_w_id6 int
= 0, @ol_qty6 smallint = 0,
        @i_id7 int = 0, @s_w_id7 int
= 0, @ol_qty7 smallint = 0,
        @i_id8 int = 0, @s_w_id8 int
= 0, @ol_qty8 smallint = 0,
        @i_id9 int = 0, @s_w_id9 int
= 0, @ol_qty9 smallint = 0,
        @i_id10 int = 0, @s_w_id10
int = 0, @ol_qty10 smallint = 0,
        @i_id11 int = 0, @s_w_id11
int = 0, @ol_qty11 smallint = 0,

```

```

        @i_id12 int = 0, @s_w_id12
int = 0, @ol_qty12 smallint = 0,
        @i_id13 int = 0, @s_w_id13
int = 0, @ol_qty13 smallint = 0,
        @i_id14 int = 0, @s_w_id14
int = 0, @ol_qty14 smallint = 0,
        @i_id15 int = 0, @s_w_id15
int = 0, @ol_qty15 smallint = 0

AS
DECLARE @w_tax      numeric(4,4),
        @d_tax      numeric(4,4),
        @c_last     char(16),
        @c_credit   char(2),
        @c_discount numeric(4,4),
        @i_price    numeric(5,2),
        @i_name     char(24),
        @o_entry_d  datetime,
        @li_no      int,
        @o_id       int,
        @commit_flag tinyint,
        @li_id      int,
        @li_qty     smallint,
        @delaytime  varchar(30)

BEGIN
-----
-- uniform random delay of 0 - 0.6 second; avg =
0.3
-----
SELECT @delaytime = '00:00:0' +
CAST(CAST((RAND()*0.60) AS decimal(4,3)) AS char(5))

WAITFOR delay @delaytime

-----
-- process orderlines
-----
SELECT @commit_flag = 1,
        @li_no       = 0

WHILE (@li_no < @o_ol_cnt)
BEGIN
        SELECT @li_id = CASE @li_no
                WHEN 1 THEN @i_id1
                WHEN 2 THEN @i_id2
                WHEN 3 THEN @i_id3
                WHEN 4 THEN @i_id4
                WHEN 5 THEN @i_id5
                WHEN 6 THEN @i_id6
                WHEN 7 THEN @i_id7
                WHEN 8 THEN @i_id8
                WHEN 9 THEN @i_id9
                WHEN 10 THEN @i_id10
                WHEN 11 THEN @i_id11
                WHEN 12 THEN @i_id12
                WHEN 13 THEN @i_id13
                WHEN 14 THEN @i_id14
                WHEN 15 THEN @i_id15

```

```

                END
SELECT @li_no      = @li_no + 1

SELECT @i_price    = 23.45, @li_qty = @li_no

IF (@li_id = 999999)
BEGIN
        SELECT '',0,',',0,0

        SELECT @commit_flag = 0
END
ELSE
BEGIN
        SELECT 'Item Name blah',
                17,
                'G',
                @i_price,
                @i_price * @li_qty
END
END

-----
-- return order data to client
-----
SELECT @w_tax      = 0.1234,
        @d_tax      = 0.0987,
        @o_id       = 3001,
        @c_last     = 'BAROUGHTABLE',
        @c_discount = 0.2198,
        @c_credit   = 'GC',
        @o_entry_d  = GETDATE()

SELECT @w_tax,
        @d_tax,
        @o_id,
        @c_last,
        @c_discount,
        @c_credit,
        @o_entry_d,
        @commit_flag

END
GO

CREATE PROCEDURE tpcc_orderstatus
        @w_id          int,
        @d_id          tinyint,

        @c_id          int,
        @c_last        char(16) = ''

AS
DECLARE @c_balance  numeric(12,2),
        @c_first    char(16),
        @c_middle   char(2),
        @o_id       int,
        @o_entry_d  datetime,
        @o_carrier_id smallint,
        @ol_cnt     smallint,

```

```

        @delaytime      varchar(30)
-----
-- uniform random delay of 0 - 0.2 second; avg = 0.1
-----
SELECT @delaytime = '00:00:0' +
CAST(CAST((RAND()*0.20) AS decimal(4,3)) AS char(5))

WAITFOR delay @delaytime

SELECT @c_id          = 113,
@c_balance          = -10.00,
@c_first           = '8YCodgytqCj8',
@c_middle          = 'OE',
@c_last            = 'OUGHTOUGHTABLE',
@o_id              = 3456,
@o_entry_d         = GETDATE(),
@o_carrier_id      = 1

SELECT @ol_cnt = (RAND() * 11) + 5

SET ROWCOUNT @ol_cnt

SELECT ol_supply_w_id,
ol_i_id,
ol_quantity,
ol_amount,
ol_delivery_d
FROM order_line_null

SELECT @c_id,
@c_last,
@c_first,
@c_middle,
@o_entry_d,
@o_carrier_id,
@c_balance,
@o_id

GO

CREATE PROCEDURE tpcc_payment
    @w_id int,
    @c_w_id int,
    @h_amount numeric(6,2),
    @d_id tinyint,
    @c_d_id tinyint,
    @c_id int,
    @c_last char(16) = ''

AS
DECLARE @w_street_1 char(20),
@w_street_2 char(20),
@w_city char(20),
@w_state char(2),
@w_zip char(9),
@w_name char(10),
@d_street_1 char(20),
@d_street_2 char(20),
@d_city char(20),
@d_state char(2),
@d_zip char(9),
@d_name char(10),

```

```

@c_first char(16),
@c_middle char(2),
@c_street_1 char(20),
@c_street_2 char(20),
@c_city char(20),
@c_state char(2),
@c_zip char(9),
@c_phone char(16),
@c_phone datetime,
@c_since char(2),
@c_credit numeric(12,2),
@c_credit_lim numeric(12,2),
@c_balance numeric(4,4),
@c_discount char(500),
@c_data char(500),
@c_data datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int,
@delaytime varchar(30)

-----
-- uniform random delay of 0 - 0.3 second; avg = 0.15
-----
SELECT @delaytime = '00:00:0' +
CAST(CAST((RAND()*0.20) AS decimal(4,3)) AS char(5))

WAITFOR delay @delaytime

SELECT @screen_data = ''

-----
-- get customer info and update balances
-----
SELECT @d_street_1 = 'rqSHhakqyV',
@d_street_2 = 'zZ98nW3BR2s',
@d_city = 'ArNr4GNFV9',
@d_state = 'aV',
@d_zip = '453511111'

-----
-- get warehouse data and update year-to-date
-----
SELECT @w_street_1 = 'rqSHhakqyV',
@w_street_2 = 'zZ98nW3BR2s',
@w_city = 'ArNr4GNFV9',
@w_state = 'aV',
@w_zip = '453511111'

SELECT @c_id = 123,
@c_balance = -10000.00,
@c_first = 'KmR03Xureb',
@c_middle = 'OE',
@c_last = 'BAROUGHTBAR',
@c_street_1 = 'QpGdOHjv8mR9vNI8V',
@c_street_2 = 'dzKoCObBqbc3yu',
@c_city = 'zAKZXdc037FQxq',
@c_state = 'QA',

```

```

@c_zip = '700311111',
@c_phone = '2967264064528555',
@c_credit = 'GC',
@c_credit_lim = 50000.00,
@c_discount = 0.3069,
@c_since = GETDATE(),
@datetime = GETDATE()

-----
-- return data to client
-----
SELECT @c_id,
@c_last,
@datetime,
@w_street_1,
@w_street_2,
@w_city,
@w_state,
@w_zip,
@d_street_1,
@d_street_2,
@d_city,
@d_state,
@d_zip,
@c_first,
@c_middle,
@c_street_1,
@c_street_2,
@c_city,
@c_state,
@c_zip,
@c_phone,
@c_since,
@c_credit,
@c_credit_lim,
@c_discount,
@c_balance,
@screen_data

GO

CREATE PROCEDURE tpcc_stocklevel
    @w_id int,
    @d_id tinyint,
    @threshold smallint

AS
DECLARE @delaytime varchar(30)

-----
-- uniform random delay of 0 - 3.6 second; avg = 1.8
-----
SELECT @delaytime = '00:00:0' +
CAST(CAST((RAND()*0.20) AS decimal(4,3)) AS char(5))

WAITFOR delay @delaytime

SELECT 49

GO

CREATE PROCEDURE tpcc_version

AS
DECLARE @version char(8)

```

```

BEGIN
    SELECT @version = '4.10.000'
END
SELECT @version AS 'Version'
END
GO

CREATE TABLE order_line_null (
    [ol_i_id] [int]
NOT NULL ,
    [ol_supply_w_id]
[int] NOT NULL ,
    [ol_delivery_d]
[datetime] NOT NULL ,
    [ol_quantity]
[smallint] NOT NULL ,
    [ol_amount]
[numeric](6, 2) NOT NULL
) ON [PRIMARY]
GO

INSERT INTO order_line_null VALUES ( 101, 1,
GETDATE(), 1, 123.45 )
INSERT INTO order_line_null VALUES ( 102, 1,
GETDATE(), 2, 123.45 )
INSERT INTO order_line_null VALUES ( 103, 1,
GETDATE(), 3, 123.45 )
INSERT INTO order_line_null VALUES ( 104, 1,
GETDATE(), 4, 123.45 )
INSERT INTO order_line_null VALUES ( 105, 1,
GETDATE(), 5, 123.45 )
INSERT INTO order_line_null VALUES ( 106, 1,
GETDATE(), 1, 123.45 )
INSERT INTO order_line_null VALUES ( 107, 1,
GETDATE(), 2, 123.45 )
INSERT INTO order_line_null VALUES ( 108, 1,
GETDATE(), 3, 123.45 )
INSERT INTO order_line_null VALUES ( 109, 1,
GETDATE(), 4, 123.45 )
INSERT INTO order_line_null VALUES ( 110, 1,
GETDATE(), 5, 123.45 )
INSERT INTO order_line_null VALUES ( 111, 1,
GETDATE(), 1, 123.45 )
INSERT INTO order_line_null VALUES ( 112, 1,
GETDATE(), 2, 123.45 )
INSERT INTO order_line_null VALUES ( 113, 1,
GETDATE(), 3, 123.45 )
INSERT INTO order_line_null VALUES ( 114, 1,
GETDATE(), 4, 123.45 )
INSERT INTO order_line_null VALUES ( 115, 1,
GETDATE(), 5, 123.45 )
GO

```

ordstat.sql

```

-----
--
--

```

```

-- File: ORDSTAT.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
--
--
-- Creates order status stored procedure
--
--
-- Interface Level: 4.20.000
--
--
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_orderstatus' )
    DROP PROCEDURE tpcc_orderstatus
GO

CREATE PROCEDURE tpcc_orderstatus
    @w_id int,
    @d_id tinyint,
    @c_id int,
    @c_last char(16) = ''
AS
DECLARE @c_balance money,
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint
BEGIN TRANSACTION o
    IF (@c_id = 0)
        BEGIN
            -----
            -- get customer id and info using last name
            -----
            SELECT @cnt = (count(*)+1)/2
            FROM customer WITH (repeatableread)
            WHERE c_last = @c_last AND
                  c_w_id = @w_id AND
                  c_d_id = @d_id
            SET rowcount @cnt

```

```

SELECT @c_id = c_id,
        @c_balance = c_balance,
        @c_first = c_first,
        @c_last = c_last,
        @c_middle = c_middle
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
       c_w_id = @w_id AND
       c_d_id = @d_id
ORDER BY c_w_id, c_d_id, c_last, c_first

SET rowcount 0
END
ELSE
BEGIN
    -----
    -- get customer info if by id
    -----
    SELECT @c_balance = c_balance,
           @c_first = c_first,
           @c_middle = c_middle,
           @c_last = c_last
    FROM customer WITH (repeatableread)
    WHERE c_id = @c_id AND
          c_d_id = @d_id AND
          c_w_id = @w_id

    SELECT @cnt = @@rowcount
END

-----
-- if no such customer
-----
IF (@cnt = 0)
BEGIN
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
END

-----
-- get order info
-----
SELECT @o_id = o_id,
        @o_entry_d = o_entry_d,
        @o_carrier_id = o_carrier_id
FROM orders WITH (serializable)
WHERE o_c_id = @c_id AND
       o_d_id = @d_id AND
       o_w_id = @w_id
ORDER BY o_id ASC

-----
-- select order lines for the current order
-----
SELECT ol_supply_w_id,
       ol_i_id,
       ol_quantity,
       ol_amount,
       ol_delivery_d
FROM order_line WITH (repeatableread)
WHERE ol_o_id = @o_id AND
       ol_d_id = @d_id AND

```



```

        ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

-----
-- return data to client
-----
SELECT  @c_id,
        @c_last,
        @c_first,
        @c_middle,

        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id
GO

```

payment.sql

```

-----
--
-- File:      PAYMENT.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
-- Creates payment stored procedure
--
--
-- Interface Level:      4.20.000
--
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_payment' )
    DROP PROCEDURE tpcc_payment
GO

CREATE PROCEDURE    tpcc_payment
        @w_id          int,

```

```

        @c_w_id      int,
        @h_amount    smallmoney,
        @d_id        tinyint,
        @c_d_id      tinyint,
        @c_id        int,
        @c_last      char(16) = ""

AS

DECLARE @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city      char(20),
        @w_state     char(2),
        @w_zip       char(9),
        @w_name      char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city      char(20),
        @d_state     char(2),
        @d_zip       char(9),
        @d_name      char(10),
        @c_first     char(16),
        @c_middle    char(2),
        @c_street_1 char(20),
        @c_street_2 char(20),
        @c_city      char(20),
        @c_state     char(2),
        @c_zip       char(9),
        @c_phone     char(16),
        @c_since     datetime,
        @c_credit    char(2),
        @c_credit_lim money,
        @c_balance   money,
        @c_discount  smallmoney,
        @c_data      char(42),
        @datetime    datetime,
        @w_ytd       money,
        @d_ytd       money,
        @cnt         smallint,
        @val         smallint,
        @screen_data char(200),
        @d_id_local  tinyint,
        @w_id_local  int,
        @c_id_local  int

SELECT @screen_data = ""

BEGIN TRANSACTION p
-- get payment date
SELECT @datetime = GETDATE()

IF (@c_id = 0)
BEGIN
-- get customer id and info using last name
SELECT @cnt = COUNT(*)
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
      c_w_id = @c_w_id AND
      c_d_id = @c_d_id

SELECT @val = (@cnt + 1) / 2

```

```

SET rowcount @val

SELECT @c_id = c_id
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
      c_w_id = @c_w_id AND
      c_d_id = @c_d_id

ORDER BY c_last, c_first

SET rowcount 0

END

-- get customer info and update balances

UPDATE customer
SET @c_balance = c_balance -
@h_amount,
    c_payment_cnt = c_payment_cnt + 1,
    c_ytd_payment = c_ytd_payment +
@h_amount,
    @c_first = c_first,
    @c_middle = c_middle,
    @c_last = c_last,
    @c_street_1 = c_street_1,
    @c_street_2 = c_street_2,
    @c_city = c_city,
    @c_state = c_state,
    @c_zip = c_zip,
    @c_phone = c_phone,
    @c_credit = c_credit,
    @c_credit_lim = c_credit_lim,
    @c_discount = c_discount,
    @c_since = c_since,
    @c_id_local = c_id
WHERE c_id = @c_id AND
      c_w_id = @c_w_id AND
      c_d_id = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
-- compute new info
SELECT @c_data = convert(char(5),@c_id) +
convert(char(4),@c_d_id)
+
convert(char(5),@c_w_id)
+
convert(char(4),@d_id) +
convert(char(5),@w_id) +
convert(char(19),@h_amount)

-- update customer info
UPDATE customer
SET c_data = @c_data +
substring(c_data, 1, 458),
    @screen_data = @c_data +
substring(c_data, 1, 158)
WHERE c_id = @c_id AND
      c_w_id = @c_w_id AND
      c_d_id = @c_d_id

END

```

```

-- get district data and update year-to-date
UPDATE district
SET
  d_ytd      = d_ytd + @h_amount,
  @d_street_1 = d_street_1,
  @d_street_2 = d_street_2,
  @d_city    = d_city,
  @d_state   = d_state,
  @d_zip     = d_zip,
  @d_name    = d_name,
  @d_id_local = d_id
WHERE
  d_w_id     = @w_id AND
  d_id      = @d_id

-- get warehouse data and update year-to-date
UPDATE warehouse
SET
  w_ytd      = w_ytd + @h_amount,
  @w_street_1 = w_street_1,
  @w_street_2 = w_street_2,
  @w_city    = w_city,
  @w_state   = w_state,
  @w_zip     = w_zip,
  @w_name    = w_name,
  @w_id_local = w_id
WHERE
  w_id      = @w_id

-- create history record
INSERT INTO
  history VALUES
  (@c_id_local,
   @c_d_id,
   @c_w_id,
   @d_id_local,
   @w_id_local,
   @datetime,
   @h_amount,
   @w_name + ' ' +
   @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT
  @c_id,
  @c_last,
  @datetime,
  @w_street_1,
  @w_street_2,
  @w_city,
  @w_state,
  @w_zip,
  @d_street_1,
  @d_street_2,
  @d_city,
  @d_state,
  @d_zip,
  @c_first,
  @c_middle,
  @c_street_1,
  @c_street_2,
  @c_city,
  @c_state,
  @c_zip,
  @c_phone,
  @c_since,

```

```

@c_credit,
@c_credit_lim,
@c_discount,
@c_balance,
@screen_data

GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

random.c

// File: RANDOM.C Microsoft
// TPC-C Kit Ver. 4.62
// Copyright
// Microsoft, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2005
// Purpose: Random number generation routines
// for database loader

// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A 16807
#define M 2147483647
#define Q 127773 /* M div A */
#define R 2836 /* M mod A */
#define Thread __declspec(thread)

// Globals
long Thread Seed = 0; /* thread local seed */

/*****
*****
*
*
* random -
*
* Implements a GOOD pseudo random number
generator. This generator
* will/should? run the complete period before
repeating.
*
* Copied from:
*
* Random Numbers Generators: Good Ones Are Hard
to Find.
* Communications of the ACM - October 1988
Volume 31 Number 10
*
*
*/

```

```

* Machine Dependencies:
*
* long must be 2 ^ 31 - 1 or greater.
*
*
*
*****
*****/

/*****
*****
* seed - load the Seed value used in irand and drand.
Should be used before *
* first call to irand or drand.
*
*****
*****/

void seed(long val)
{
#ifdef DEBUG
  printf("[%ld]DBG: Entering seed()...\n", (int)
  GetCurrentThreadId());
  printf("Old Seed %ld New Seed %ld\n",Seed,
  val);
#endif

  if ( val < 0 )
    val = abs(val);

  Seed = val;
}

/*****
*****
*
*
* irand - returns a 32 bit integer pseudo random
number with a period of
* 1 to 2 ^ 32 - 1.
*
*
* parameters:
*
* none.
*
* returns:
*
* 32 bit integer - defined as long ( see above
* ).
*
* side effects:
*
* seed get recomputed.
*
*/

```

```

*****
*****/

long irand()
{
    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
*/

#ifdef DEBUG
    printf("[%ld]DBG: Entering irand()...\n", (int)
GetCurrentThreadId());
#endif

    s = Seed;
    hi = s / Q;
    lo = s % Q;

    test = A * lo - R * hi;
    if ( test > 0 )
        Seed = test;
    else
        Seed = test + M;

    return( Seed );
}

/*****
*****
*
* drand - returns a double pseudo random number
between 0.0 and 1.0.
* See irand.
*
*****
*****/

double drand()
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering drand()...\n", (int)
GetCurrentThreadId());
#endif

    return( (double)irand() / 2147483647.0);
}

//=====
// Function : RandomNumber
// Description:
//=====
long RandomNumber(long lower, long upper)
{
    long rand_num;

```

```

#ifdef DEBUG
    printf("[%ld]DBG: Entering RandomNumber()...\n",
(int) GetCurrentThreadId());
#endif

    if ( upper == lower ) /* pgd 08-13-
96 perf enhancement */
        return lower;

    upper++;

    if ( upper <= lower )
        rand_num = upper;
    else
        rand_num = lower + irand() %
(upper - lower); /* pgd 08-13-96 perf enhancement */

#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld
=> %ld\n",
(int)
GetCurrentThreadId(), lower, upper, rand_num);
#endif

    return rand_num;
}

#if 0
//Original code pgd 08/13/96

long RandomNumber(long lower,
long upper)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering RandomNumber()...\n",
(int) GetCurrentThreadId());
#endif

    upper++;

    if ((upper <= lower))
        rand_num = upper;
    else
        rand_num = lower + irand() %
((upper > lower) ? upper - lower : upper);

#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld
=> %ld\n",
(int)
GetCurrentThreadId(), lower, upper, rand_num);
#endif

    return rand_num;
}

```

```

}
#endif

//=====
// Function : NURand
// Description:
//=====
long NURand(int iConst,
long x,
long y,
long C)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering NURand()...\n", (int)
GetCurrentThreadId());
#endif

    rand_num = (((RandomNumber(0,iConst) |
RandomNumber(x,y) + C) % (y-x+1))+x);

#ifdef DEBUG
    printf("[%ld]DBG: NURand: num = %d\n", (int)
GetCurrentThreadId(), rand_num);
#endif

    return rand_num;
}

-----
removedb.sql
-----
--
-- File: REMOVEDB.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.61
--
-- Copyright Microsoft, 2005
--
--
-----
USE master
GO

-----
-- remove any existing database and backup files
-----
EXEC sp_dbremove tpcc, dropdev

```

```

GO

EXEC sp_dropdevice 'tpcback1'
EXEC sp_dropdevice 'tpcback2'
EXEC sp_dropdevice 'tpcback3'
EXEC sp_dropdevice 'tpcback4'
GO

```

restore.sql

```

-----
--
-- File: RESTORE.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.61
--
-- Copyright Microsoft, 2005
--
-----
--
--
--
-----
DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
        CONVERT(VARCHAR(30),@startdate,
21)

LOAD DATABASE tpcc FROM tpcback1, tpcback2,
tpcback3, tpcback4 WITH stats = 1, replace

SELECT @enddate = GETDATE()
SELECT 'End date: ',
        CONVERT(VARCHAR(30),@enddate, 21)
SELECT 'Elapsed time (in seconds): ',
        DATEDIFF(second, @startdate, @enddate)

```

RunSQLCfg.sql

```

-----
--
-- File: RUNSQLCFG.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
-- Sets suggested runtime server
configuration --

```

```

-- parameters
--
--
-----
EXEC sp_configure 'show advanced option', 1
GO

RECONFIGURE WITH OVERRIDE
GO

-----
-- change this value to approximately the number of
connected users
-----
EXEC sp_configure 'max worker threads',255

-----
-- increase priority of user threads
-----
EXEC sp_configure 'priority boost',1

-----
-- disable automatic checkpointing
-----
EXEC sp_configure 'recovery interval',32767

-----
-- change to a mask appropriate for the number of
processors on the server
-----
EXEC sp_configure 'affinity mask',0xf

-----
-- enable fibers
-----
EXEC sp_configure 'lightweight pooling',1
GO

RECONFIGURE WITH OVERRIDE
GO

```

sqlshutdown.sql

```

-----
--
--

```

```

-- File: SQLSHUTDOWN.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
-- Checkpoints tpcc database and issues a
shutdown --
--
-----
USE tpcc
GO

CHECKPOINT
GO

SHUTDOWN
GO

```

stocklev.sql

```

-----
--
-- File: STOCKLEV.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
-- Creates stock level stored procedure
--
-- Interface Level: 4.20.000
--
-----
--
--
--
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_stocklevel' )
DROP PROCEDURE tpcc_stocklevel
GO

```

```

CREATE PROCEDURE tpcc_stocklevel
    @w_id int,
    @d_id tinyint,
    @threshold smallint

AS
DECLARE @o_id_low int,
        @o_id_high int

SELECT @o_id_low = (d_next_o_id - 20),
       @o_id_high = (d_next_o_id - 1)
FROM district
WHERE d_w_id = @w_id AND
      d_id = @d_id

SELECT COUNT(DISTINCT(s_i_id))
FROM stock,
      order_line
WHERE ol_w_id = @w_id AND
      ol_d_id = @d_id and
      ol_o_id BETWEEN @o_id_low AND
                    @o_id_high AND
      s_w_id = ol_w_id AND
      s_i_id = ol_i_id AND
      s_quantity < @threshold

OPTION(OORDER GROUP)
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

strings.c

```

// File: STRINGS.C
// Microsoft
TPC-C Kit Ver. 4.51
// Copyright
Microsoft, 1996, 1997, 1998, 1999, 2000, 2001, 2002,
2003
// Purpose: Source file for database loader
string functions

// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//=====
//
// Function name: MakeAddress
//=====

```

```

void MakeAddress(char *street_1,
                char
                *street_2,
                char *city,
                char *state,
                char *zip)
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeAddress()\n",
           (int) GetCurrentThreadId());
#endif

    MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
    MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
    MakeAlphaString (10, 20, ADDRESS_LEN, city);
    MakeAlphaString ( 2,  2, STATE_LEN, state);
    MakeZipNumberString( 9,  9, ZIP_LEN, zip);

#ifdef DEBUG
    printf("[%ld]DBG: MakeAddress: street_1: %s,
           street_2: %s, city: %s, state: %s, zip: %s\n",
           (int)
           GetCurrentThreadId(), street_1, street_2, city,
           state, zip);
#endif

    return;
}

//=====
//
// Function name: LastName
//=====
void LastName(int num,
             char *name)
{
    static char *n[] =
    {
        "BAR" , "OUGHT" , "ABLE" , "PRI"
    , "PRES" ,
        "ESE" , "ANTI" , "CALLY" ,
    "ATION" , "EING"
    };

#ifdef DEBUG
    printf("[%ld]DBG: Entering LastName()\n", (int)
           GetCurrentThreadId());
#endif

    if ((num >= 0) && (num < 1000))
    {
        strcpy(name, n[(num/100)%10]);
        strcat(name, n[(num/10)%10]);
        strcat(name, n[(num/1)%10]);

        if (strlen(name) < LAST_NAME_LEN)

```

```

    {
        PaddString(LAST_NAME_LEN, name);
    }
    else
    {
        printf("\nError in LastName()...
           num < %ld> out of range (0,999)\n", num);
        exit(-1);
    }
}

#ifdef DEBUG
    printf("[%ld]DBG: LastName: num = [%d] ==>
           [%d][%d][%d]\n",
           (int)
           GetCurrentThreadId(), num, num/100, (num/10)%10,
           num%10);
    printf("[%ld]DBG: LastName: String = %s\n",
           (int) GetCurrentThreadId(), name);
#endif

    return;
}

//=====
//
// Function name: MakeAlphaString
//=====
//philipdu 08/13/96 Changed MakeAlphaString to use A-
Z, a-z, and 0-9 in
//accordance with spec see below:
//The spec says:
//4.3.2.2 The notation random a-string [x .. y]
//(respectively, n-string [x .. y]) represents a
string of random alphanumeric
//(respectively, numeric) characters of a random
length of minimum x, maximum y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and
0..9. The only other
//requirement is that the character set used "must be
able to represent a minimum
//of 128 different characters". We are using 8-bit
chars, so this is a non issue.
//It is completely unreasonable to stuff non-printing
chars into the text fields.
//-Clevine 08/13/96

int MakeAlphaString( int x, int y, int z, char
*str)
{
    int len;
    int i;
    char cc = 'a';

```

```

        static char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
        static int chArrayMax = 61;

#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeAlphaString()\n",
(int) GetCurrentThreadId());
#endif

    len= RandomNumber(x, y);

    for (i=0; i<len; i++)
        str[i] =
chArray[RandomNumber(0,chArrayMax)];
    str[len] = 0;

    return len;
}

int MakeAlphaStringPadded( int minLen, int maxLen,
int padLen, char *str)
{
    int len;
    int i;
    char cc = 'a';
    static char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
    static int chArrayMax = 61;

#ifdef DEBUG
    printf("[%ld]DBG: Entering
MakeAlphaStringPadded()\n", (int)
GetCurrentThreadId());
#endif

    len= RandomNumber(minLen, maxLen);

    for (i=0; i<len; i++)
        str[i] =
chArray[RandomNumber(0,chArrayMax)];
    if (len < padLen)
        memset(str+len, ' ', padLen -
len);
    str[padLen] = 0;
    return padLen;
}

//=====
//
// Function name: MakeOriginalAlphaString
//
//=====
int MakeOriginalAlphaString(int x,
int y,

```

```

int z,
char *str,
int percent)
{
    int len;
    int val;
    int start;

#ifdef DEBUG
    printf("[%ld]DBG: Entering
MakeOriginalAlphaString()\n", (int)
GetCurrentThreadId());
#endif

    // verify percentage is valid
    if ((percent < 0) || (percent > 100))
    {
        printf("MakeOriginalAlphaString:
Invalid percentage: %d\n", percent);
        exit(-1);
    }

    // verify string is at least 8 chars in length
    if (x < 8)
    {
        printf("MakeOriginalAlphaString:
string length must be >= 8\n");
        exit(-1);
    }

    // Make Alpha String
    len = MakeAlphaString(x,y, z, str);

    val = RandomNumber(1,100);
    if (val <= percent)
    {
        start = RandomNumber(0, len - 8);
        strncpy(str + start, "ORIGINAL",
8);
    }

#ifdef DEBUG
    printf("[%ld]DBG: MakeOriginalAlphaString: :
%s\n", (int)
GetCurrentThreadId(), str);
#endif

    return len;
}

//=====
//
// Function name: MakeNumberString
//
//=====

```

```

int MakeNumberString(int x, int y, int z, char
*str)
{
    char tmp[16];

    //MakeNumberString is always called
MakeZipNumberString(16, 16, 16, string)

    memset(str, '0', 16);
    itoa(RandomNumber(0, 99999999), tmp, 10);
    memcpy(str, tmp, strlen(tmp));

    itoa(RandomNumber(0, 99999999), tmp, 10);
    memcpy(str+8, tmp, strlen(tmp));

    str[16] = 0;

    return 16;
}

//=====
//
// Function name: MakeZipNumberString
//
//=====
int MakeZipNumberString(int x, int y, int z, char
*str)
{
    char tmp[16];

    //MakeZipNumberString is always called
MakeZipNumberString(9, 9, 9, string)

    strcpy(str, "000011111");
    itoa(RandomNumber(0, 9999), tmp, 10);
    memcpy(str, tmp, strlen(tmp));

    return 9;
}

//=====
//
// Function name: InitString
//
//=====
void InitString(char *str, int len)
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering InitString()\n", (int)
GetCurrentThreadId());
#endif

    memset(str, ' ', len);
    str[len] = 0;
}

```

```

//=====
// Function name: InitAddress
//
// Description:
//
//=====
void InitAddress(char *street_1, char *street_2, char
*city, char *state, char *zip)
{
    memset(street_1, ' ', ADDRESS_LEN+1);
    memset(street_2, ' ', ADDRESS_LEN+1);
    memset(city, ' ', ADDRESS_LEN+1);

    street_1[ADDRESS_LEN+1] = 0;
    street_2[ADDRESS_LEN+1] = 0;
    city[ADDRESS_LEN+1] = 0;

    memset(state, ' ', STATE_LEN+1);
    state[STATE_LEN+1] = 0;

    memset(zip, ' ', ZIP_LEN+1);
    zip[ZIP_LEN+1] = 0;
}

//=====
//
// Function name: PaddString
//
//=====
void PaddString(int max, char *name)
{
    int len;

    len = strlen(name);
    if ( len < max )
        memset(name+len, ' ', max - len);
    name[max] = 0;

    return;
}

```

tables.sql

```

-----
--
-- File: TABLES.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--

```

```

-- Copyright Microsoft, 2005
--
--
-- Creates TPC-C tables
--
-----
SET ANSI_NULL_DFLT_OFF ON
GO

USE tpcc
GO

-----
-- Remove all existing TPC-C tables
-----
if exists ( select name from sysobjects where name =
'warehouse' )
    drop table warehouse
go
if exists ( select name from sysobjects where name =
'district' )
    drop table district
go
if exists ( select name from sysobjects where name =
'customer' )
    drop table customer
go
if exists ( select name from sysobjects where name =
'history' )
    drop table history
go
if exists ( select name from sysobjects where name =
'new_order' )
    drop table new_order
go
if exists ( select name from sysobjects where name =
'orders' )
    drop table orders
go
if exists ( select name from sysobjects where name =
'order_line' )
    drop table order_line
go
if exists ( select name from sysobjects where name =
'item' )
    drop table item
go
if exists ( select name from sysobjects where name =
'stock' )
    drop table stock
go

-----
-- Create new tables
-----
create table warehouse
(
    w_id          int,
    w_ytd         money,
    w_tax         smallmoney,

```

```

    w_name        char(10),
    w_street_1    char(20),
    w_street_2    char(20),
    w_city        char(20),
    w_state       char(2),
    w_zip         char(9)
) on MSSQL_misc_fg
go

create table district
(
    d_id          tinyint,
    d_w_id        int,
    d_ytd         money,
    d_next_o_id  int,

    d_tax         smallmoney,
    d_name        char(10),
    d_street_1    char(20),
    d_street_2    char(20),
    d_city        char(20),
    d_state       char(2),
    d_zip         char(9)
) on MSSQL_misc_fg
go

create table customer
(
    c_id          int,
    c_d_id        tinyint,
    c_w_id        int,
    c_discount    smallmoney,
    c_credit_lim  money,
    c_last        char(16),
    c_first       char(16),
    c_credit      char(2),
    c_balance     money,
    c_ytd_payment money,
    c_payment_cnt smallint,
    c_delivery_cnt smallint,
    c_street_1    char(20),
    c_street_2    char(20),
    c_city        char(20),
    c_state       char(2),
    c_zip         char(9),
    c_phone       char(16),
    c_since       datetime,
    c_middle      char(2),
    c_data        char(500)
) on MSSQL_cs_fg
go

-- Use the following table option if using c_data
varchar(max)
-- sp_tableoption 'customer','large value types out
of row','1'
-- go

create table history
(
    h_c_id        int,
    h_c_d_id      tinyint,
    h_c_w_id      int,

```

```

        h_d_id      tinyint,
        h_w_id      int,
        h_date      datetime,
        h_amount    smallmoney,
        h_data      char(24)
    ) on MSSQL_misc_fg
go

create table new_order
(
    no_o_id      int,
    no_d_id      tinyint,
    no_w_id      int
) on MSSQL_misc_fg
go

create table orders
(
    o_id          int,
    o_d_id        tinyint,
    o_w_id        int,
    o_c_id        int,
    o_carrier_id  tinyint,
    o_ol_cnt      tinyint,
    o_all_local   tinyint,
    o_entry_d     datetime
) on MSSQL_misc_fg
go

create table order_line
(
    ol_o_id      int,
    ol_d_id      tinyint,
    ol_w_id      int,
    ol_number    tinyint,
    ol_i_id      int,
    ol_delivery_d datetime,
    ol_amount    smallmoney,
    ol_supply_w_id int,
    ol_quantity  smallint,
    ol_dist_info char(24)
) on MSSQL_misc_fg
go

create table item
(
    i_id          int,
    i_name        char(24),
    i_price       smallmoney,
    i_data        char(50),
    i_im_id       int
) on MSSQL_misc_fg
go

create table stock
(
    s_i_id      int,
    s_w_id      int,
    s_quantity  smallint,
    s_ytd       int,
    s_order_cnt smallint,
    s_remote_cnt smallint,
    s_data      char(50),

```

```

        s_dist_01  char(24),
        s_dist_02  char(24),
        s_dist_03  char(24),
        s_dist_04  char(24),
        s_dist_05  char(24),
        s_dist_06  char(24),
        s_dist_07  char(24),
        s_dist_08  char(24),
        s_dist_09  char(24),
        s_dist_10  char(24)
    ) on MSSQL_cs_fg
go

```

time.c

```

//      File:                TIME.C                Microsoft
//
//      TPC-C Kit Ver. 4.62
//
//      Copyright
//      Microsoft, 1996, 1997, 1998, 1999, 2000, 2001, 2002,
//      2005
//      Purpose:  Source file for time functions

// Includes
#include "tpcc.h"

// Globals
static long start_sec;

//=====
//
// Function name: TimeNow
//=====
long TimeNow()
{
    long          time_now;
    struct _timeb el_time;

#ifdef DEBUG
    printf("[%ld]DBG: Entering TimeNow()\n", (int)
GetCurrentThreadId());
#endif
    _ftime(&el_time);

    time_now = ((el_time.time - start_sec) * 1000) +
el_time.millitm;

    return time_now;
}

```

tpcc.h

```

//      File:                TPCC.H                Microsoft
//
//      TPC-C Kit Ver. 4.51
//
//      Copyright
//      Microsoft, 1996, 1997, 1998, 1999, 2000, 2001, 2002,
//      2003, 2005
//      Purpose:  Header file for TPC-C database
//      loader

// Build number of TPC Benchmark Kit
#define TPCKIT_VER "4.51"

// General headers
#include <windows.h>
#include <winbase.h>
#include <stdlib.h>
#include <stdio.h>
#include <process.h>
#include <stddef.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <sys\types.h>
#include <math.h>

// ODBC headers
#include <sql.h>
#include <sqlext.h>
#include <odbc.h>

// General constants
#define MILLI          1000
#define FALSE          0
#define TRUE           1
#define UNDEF         -1
#define MINPRINTASCII 32
#define MAXPRINTASCII 126

// Default environment constants
#define SERVER          ""
#define DATABASE        "tpcc"
#define USER            "sa"
#define PASSWORD        ""

// Default loader arguments
#define BATCH           10000
#define DEFLDPACKSIZE  32768
#define LOADER_RES_FILE "C:\\MSTPCC.450\\SETUP\\LOGS\\load.out"

```



```

#define LOADER_LOG_PATH
    "C:\\MSTPCC.450\\SETUP\\LOGS\\"
#define LOADER_NURAND_C
    123
#define DEF_STARTING_WAREHOUSE
    1
#define BUILD_INDEX
    1 // build both data and indexes
#define INDEX_ORDER
    1 // build indexes before load
#define SCALE_DOWN
    0 // build a normal scale database
#define INDEX_SCRIPT_PATH
    "scripts"

typedef struct
{
    char
        *server;
    char
        *database;
    char
        *user;
    char
        *password;
    BOOL
        tables_all;
    set if loading all tables
        BOOL
        table_item;
    set if loading ITEM table specifically
        BOOL
        table_warehouse; // set if loading
        WAREHOUSE, DISTRICT, and STOCK
        BOOL
        table_customer; // set if
        loading CUSTOMER and HISTORY
        BOOL
        table_orders; // set if
        loading NEW-ORDER, ORDERS, ORDER-LINE
        long
        num_warehouses;
        long
        batch;
        long
        verbose;
        long
        pack_size;
        char
        *loader_res_file;
        char
        *log_path;
        char
        *synch_servername;
        long
        case_sensitivity;
        long
        starting_warehouse;
        long
        build_index;
        long
        index_order;
        long
        scale_down;

```

```

    char
        *index_script_path;
} TPCCLDR_ARGS;

// String length constants
#define SERVER_NAME_LEN
    20
#define DATABASE_NAME_LEN
    20
#define USER_NAME_LEN
    20
#define PASSWORD_LEN
    20
#define TABLE_NAME_LEN
    20
#define I_DATA_LEN
    50
#define I_NAME_LEN
    24
#define BRAND_LEN
    1
#define LAST_NAME_LEN
    16
#define W_NAME_LEN
    10
#define ADDRESS_LEN
    20
#define STATE_LEN
    2
#define ZIP_LEN
    9
#define S_DIST_LEN
    24
#define S_DATA_LEN
    50
#define D_NAME_LEN
    10
#define FIRST_NAME_LEN
    16
#define MIDDLE_NAME_LEN
    2
#define PHONE_LEN
    16
#define CREDIT_LEN
    2
#define C_DATA_LEN
    500
#define H_DATA_LEN
    24
#define DIST_INFO_LEN
    24
#define MAX_OL_NEW_ORDER_ITEMS
    15
#define MAX_OL_ORDER_STATUS_ITEMS
    15
#define STATUS_LEN
    25
#define OL_DIST_INFO_LEN
    24
#define C_SINCE_LEN
    23
#define H_DATE_LEN
    23
#define OL_DELIVERY_D_LEN
    23
#define O_ENTRY_D_LEN
    23

// Functions in random.c
void
    seed();
long
    irand();
double
    drand();
void
    WUcreate();
short
    WURand();
long
    RandomNumber(long lower, long upper);

// Functions in getargs.c;
void
    GetArgsLoader();
void
    GetArgsLoaderUsage();

// Functions in time.c
long
    TimeNow();

// Functions in strings.c
void
    MakeAddress();
void
    LastName();
int
    MakeAlphaString();
int
    MakeAlphaStringPadded();
int
    MakeOriginalAlphaString();

```

```

int
    MakeNumberString();
int
    MakeZipNumberString();
void
    InitString();
void
    InitAddress();
void
    PaddString();

```

tpccldr.c

```

//=====
// File:
    TPCCLDR.C
// Microsoft
TPC-C Kit Ver. 4.51
// Copyright
Microsoft, 1996, 1997, 1998, 1999,
    2000, 2001,
    2002, 2003
// Purpose: Source file for TPC-C database
loader
//=====
// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS
    100000
#define MAXITEMS_SCALE_DOWN
    100
#define CUSTOMERS_PER_DISTRICT
    3000
#define CUSTOMERS_SCALE_DOWN
    30
#define DISTRICT_PER_WAREHOUSE
    10
#define ORDERS_PER_DISTRICT
    3000
#define ORDERS_SCALE_DOWN
    30
#define MAX_CUSTOMER_THREADS
    2
#define MAX_ORDER_THREADS
    3
#define MAX_MAIN_THREADS
    4
#define MAX_SQL_ERRORS
    10

// Functions declarations
void
    HandleErrorDBC(SQLHDBC hdbc1);
long
    NURand();
void
    LoadItem();
void
    LoadWarehouse();
void
    Stock();
void
    District();
void
    LoadCustomer();
void
    CustomerBufInit();
void
    CustomerBufLoad();
void
    LoadCustomerTable();
void
    LoadHistoryTable();
void
    LoadOrders();
void
    OrdersBufInit();
void
    OrdersBufLoad();
void
    LoadOrdersTable();
void
    LoadNewOrderTable();
void
    LoadOrderLineTable();
void
    GetPermutation();
void
    CheckForCommit();
void
    CheckForCommit_Big();
void
    OpenConnections();

```

```

void BuildIndex();
void FormatDate ();

// Shared memory structures
typedef struct
{
    double          ol;
    long            ol_i_id;
    long            ol_supply_w_id;
    short           ol_quantity;
    double          ol_amount;
    char
ol_dist_info[DIST_INFO_LEN+1];
    char
    ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
    long            o_id;
    short           o_d_id;
    long            o_w_id;
    long            o_c_id;
    short           o_carrier_id;
    short           o_ol_cnt;
    short           o_all_local;
    ORDER_LINE_STRUCT o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
    long            c_id;
    short           c_d_id;
    long            c_w_id;
    char            c_first[FIRST_NAME_LEN+1];
    char            c_middle[MIDDLE_NAME_LEN+1];
    char            c_last[LAST_NAME_LEN+1];
    char            c_street_1[ADDRESS_LEN+1];
    char            c_street_2[ADDRESS_LEN+1];
    char            c_city[ADDRESS_LEN+1];
    char            c_state[STATE_LEN+1];
    char            c_zip[ZIP_LEN+1];
    char            c_phone[PHONE_LEN+1];
    char            c_credit[CREDIT_LEN+1];
    double          c_credit_lim;
    double          c_discount;
    char            c_balance[6];
}

```

```

double
    c_ytd_payment;
short
    c_payment_cnt;
short
    c_delivery_cnt;
char
    c_data[C_DATA_LEN+1];
double
    h_amount;
char
    h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
    char            c_last[LAST_NAME_LEN+1];
    char            c_first[FIRST_NAME_LEN+1];
    long            c_id;
} CUSTOMER_SORT_STRUCT;

typedef struct
{
    long            time_start;
} LOADER_TIME_STRUCT;

// Global variables
char            szLastError[300];

HENV            henv;
HDBC            v_hdbc; // for SQL Server version

verification
HDBC            i_hdbc1; // for ITEM table
HDBC            w_hdbc1; // for WAREHOUSE, DISTRICT, STOCK
HDBC            c_hdbc1; // for CUSTOMER
HDBC            c_hdbc2; // for HISTORY
HDBC            o_hdbc1; // for ORDERS
HDBC            o_hdbc2; // for NEW-ORDER
HDBC            o_hdbc3; // for ORDER-LINE

HSTMT          v_hstmt; // for SQL Server version verification
HSTMT          i_hstmt1;
HSTMT          w_hstmt1;
HSTMT          c_hstmt1, c_hstmt2;
HSTMT          o_hstmt1, o_hstmt2, o_hstmt3;

int            total_db_errors;

ORDERS_STRUCT  orders_buf[ORDERS_PER_DISTRICT];

```

```

CUSTOMER_STRUCT
    customer_buf[CUSTOMERS_PER_DISTRICT];
long            orders_rows_loaded;
double          new_order_rows_loaded;
double          order_line_rows_loaded;
long            history_rows_loaded;
long            customer_rows_loaded;
double          stock_rows_loaded;
long            district_rows_loaded;
long            item_rows_loaded;
long            warehouse_rows_loaded;
long            main_time_start;
long            main_time_end;
long            max_items;
long            customers_per_district;
long            orders_per_district;
long            first_new_order;
long            last_new_order;

TPCCCLR_ARGS   *aptr, args;

//=====
//
// Function name: main
//
//=====
int main(int argc, char **argv)
{
    DWORD
dwThreadId[MAX_MAIN_THREADS];
HANDLE        hThread[MAX_MAIN_THREADS];
FILE          *fLoader;
char          buffer[255];
int           i;

    for (i=0; i<MAX_MAIN_THREADS; i++)
        hThread[i] = NULL;

    printf("\n*****\n");
    printf("\n*
*");
    printf("\n* Microsoft SQL Server
*");
    printf("\n*
*");
    printf("\n* TPC-C BENCHMARK KIT: Database
loader *");
    printf("\n* Version %s
*", TPCKIT_VER);
    printf("\n*
*");
    printf("\n*****\n\n");

    // process command line arguments
    aptr = &args;
    GetArgsLoader(argc, argv, aptr);
}

```

```

printf("Build interface is ODBC.\n");

if (aptr->build_index == 0)
    printf("Data load only - no index
creation.\n");
else
    printf("Data load and index
creation.\n");

if (aptr->index_order == 0)
    printf("Clustered indexes will be
created after bulk load.\n");
else
    printf("Clustered indexes will be
created before bulk load.\n");

// set database scale values
if (aptr->scale_down == 1)
{
    printf("*** Scaled Down Database
***\n");
    max_items = MAXITEMS_SCALE_DOWN;
    customers_per_district =
CUSTOMERS_SCALE_DOWN;
    orders_per_district =
ORDERS_SCALE_DOWN;
    first_new_order = 0;
    last_new_order = 30;
}
else
{
    max_items = MAXITEMS;
    customers_per_district =
CUSTOMERS_PER_DISTRICT;
    orders_per_district =
ORDERS_PER_DISTRICT;
    first_new_order = 2100;
    last_new_order = 3000;
}

// open connections to SQL Server
OpenConnections();

// open file for loader results
fLoader = fopen(aptr->loader_res_file,
"w");

if (fLoader == NULL)
{
    printf("Error, loader result file
open failed.");
    exit(-1);
}

// start loading data
sprintf(buffer, "TPC-C load started for %ld
warehouses.\n", aptr->num_warehouses);
if (aptr->scale_down == 1)
{
    sprintf(buffer, "SCALED DOWN
DATABASE.\n");
}

```

```

printf("%s", buffer);
fprintf(fLoader, "%s", buffer);

main_time_start = (TimeNow() / MILLI);

// start parallel load threads
if (aptr->tables_all || aptr->table_item)
{
    fprintf(fLoader, "\nStarting
loader threads for: item\n");

    hThread[0] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadItem,
NULL,
0,
&dwThreadID[0]);

    if (hThread[0] == NULL)
    {
        printf("Error, failed
in creating creating thread = 0.\n");
        exit(-1);
    }

    if (aptr->tables_all || aptr-
>table_warehouse)
    {
        fprintf(fLoader, "Starting loader
threads for: warehouse\n");

        hThread[1] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadWarehouse,
NULL,
0,
&dwThreadID[1]);

        if (hThread[1] == NULL)
        {
            printf("Error, failed
in creating creating thread = 1.\n");
            exit(-1);
        }

        if (aptr->tables_all || aptr-
>table_customer)

```

```

{
    fprintf(fLoader, "Starting loader
threads for: customer\n");

    hThread[2] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadCustomer,
NULL,
0,
&dwThreadID[2]);

    if (hThread[2] == NULL)
    {
        printf("Error, failed
in creating creating main thread = 2.\n");
        exit(-1);
    }

    if (aptr->tables_all || aptr->table_orders)
    {
        fprintf(fLoader, "Starting loader
threads for: orders\n");

        hThread[3] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadOrders,
NULL,
0,
&dwThreadID[3]);

        if (hThread[3] == NULL)
        {
            printf("Error, failed
in creating creating main thread = 3.\n");
            exit(-1);
        }

        // Wait for threads to finish...
        for (i=0; i<MAX_MAIN_THREADS; i++)
        {
            if (hThread[i] != NULL)
            {
                WaitForSingleObject(
hThread[i], INFINITE);

                CloseHandle(hThread[i]);
                hThread[i] = NULL;
            }
        }
    }
}

```

```

        main_time_end = (TimeNow() / MILLI);

        sprintf(buffer, "\nTPC-C load completed
successfully in %ld minutes.\n",
                (main_time_end -
main_time_start)/60);

        printf("%s",buffer);
        fprintf(fLoader, "%s", buffer);

        fclose(fLoader);

        SQLFreeEnv(henv);

        exit(0);

        return 0;
}

//=====
//
// Function name: LoadItem
//
//=====
void LoadItem()
{
    int            i;
    long           i_id;
    long           i_im_id;
    char           i_name[I_NAME_LEN+1];
    double         i_price;
    char           i_data[I_DATA_LEN+1];
    char           name[20];
    long           time_start;
    RETCODE        rc;
    DBINT          rcint;
    char           bcp hint[128];
    char           err_log_path[256];

    // Seed with unique number
    seed(11);

    printf("Loading item table...\n");

    //if build index before load
    if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        BuildIndex("idxitmcl");

    InitString(i_name, I_NAME_LEN+1);
    InitString(i_data, I_DATA_LEN+1);

    sprintf(name, "%s..%s", aptr->database,
"item");

    strcpy(err_log_path, aptr->log_path);
    strcat(err_log_path, "item.err");
    rc = bcp_init(i_hdbc1, name, NULL,
err_log_path, DB_IN);

```

```

        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);

        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        {
            sprintf(bcp hint, "tablock, order
(i_id), ROWS_PER_BATCH = 100000");
            rc = bcp_control(i_hdbc1,
BCPHINTS, (void*) bcp hint);
            if (rc != SUCCEED)

                HandleErrorDBC(i_hdbc1);
        }

        i = 0;
        rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0,
I_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) &i_price,
0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0,
SQL_VARLEN_DATA, "", 1, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);

        time_start = (TimeNow() / MILLI);

        item_rows_loaded = 0;

        for (i_id = 1; i_id <= max_items; i_id++)
        {
            i_im_id = RandomNumber(1L,
10000L);

            MakeAlphaStringPadded(14, 24,
I_NAME_LEN, i_name);

            i_price = ((float)
RandomNumber(100L, 10000L))/100.0;

            MakeOriginalAlphaString(26, 50,
I_DATA_LEN, i_data);

            rc = bcp_sendrow(i_hdbc1);

            if (rc != SUCCEED)

                HandleErrorDBC(i_hdbc1);

            item_rows_loaded++;

```

```

                CheckForCommit(i_hdbc1, i_hstmt1,
item_rows_loaded, "item", &time_start);
            }

            rcint = bcp_done(i_hdbc1);
            if (rcint < 0)
                HandleErrorDBC(i_hdbc1);

            printf("Finished loading item table.\n");

            SQLFreeStmt(i_hstmt1, SQL_DROP);
            SQLDisconnect(i_hdbc1);
            SQLFreeConnect(i_hdbc1);

            // if build index after load
            if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
                BuildIndex("idxitmcl");
        }

//=====
//
// Function : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District
as Warehouses are created
//
//=====
void LoadWarehouse()
{
    int            i;
    long           w_id;
    char           w_name[W_NAME_LEN+1];
    char           w_street_1[ADDRESS_LEN+1];
    char           w_street_2[ADDRESS_LEN+1];
    char           w_city[ADDRESS_LEN+1];
    char           w_state[STATE_LEN+1];
    char           w_zip[ZIP_LEN+1];
    double         w_tax;
    double         w_ytd;
    char           name[20];
    long           time_start;
    RETCODE        rc;
    DBINT          rcint;
    char           bcp hint[128];
    char           err_log_path[256];

    // Seed with unique number
    seed(2);

    printf("Loading warehouse table...\n");

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        BuildIndex("idxwarc1");

    InitString(w_name, W_NAME_LEN+1);

```

```

    InitAddress(w_street_1, w_street_2, w_city,
w_state, w_zip);

    sprintf(name, "%s.%s", aptr->database,
"warehouse");

    strcpy(err_log_path, aptr->log_path);
    strcat(err_log_path, "whouse.err");
    rc = bcp_init(w_hdbc1, name, NULL,
err_log_path, DB_IN);

    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
    {
        sprintf(bcphint, "tablock, order
(w_id), ROWS_PER_BATCH = %d", aptr->num_warehouses);
        rc = bcp_control(w_hdbc1,
BCPHINTS, (void*) bcphint);
        if (rc != SUCCEEDED)

            HandleErrorDBC(w_hdbc1);
    }

    i = 0;
    rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0,
W_NAME_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1,
0, ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2,
0, ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0,
STATE_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0,
ZIP_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

```

```

    time_start = (TimeNow() / MILLI);

    warehouse_rows_loaded = 0;

    for (w_id = (long)aptr->starting_warehouse;
w_id <= aptr->num_warehouses; w_id++)
    {
        MakeAlphaStringPadded(6,10,
W_NAME_LEN, w_name);

        MakeAddress(w_street_1,
w_street_2, w_city, w_state, w_zip);

        w_tax = ((float)
RandomNumber(0L,2000L))/10000.00;

        w_ytd = 300000.00;

        rc = bcp_sendrow(w_hdbc1);
        if (rc != SUCCEEDED)

            HandleErrorDBC(w_hdbc1);

        warehouse_rows_loaded++;
        CheckForCommit(w_hdbc1, i_hstmt1,
warehouse_rows_loaded, "warehouse", &time_start);
    }

    rcint = bcp_done(w_hdbc1);
    if (rcint < 0)
        HandleErrorDBC(w_hdbc1);

    printf("Finished loading warehouse
table.\n");

    // if build index after load...
    if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
        BuildIndex("idxwarc1");

    stock_rows_loaded = 0;
    district_rows_loaded = 0;

    District();
    Stock();
}

//=====
//
// Function : District
//
//=====
void District()
{
    int         i;
    short      d_id;
    long       d_w_id;
    char       d_name[D_NAME_LEN+1];
    char       d_street_1[ADDRESS_LEN+1];

```

```

    char       d_street_2[ADDRESS_LEN+1];
    char       d_city[ADDRESS_LEN+1];
    char       d_state[STATE_LEN+1];
    char       d_zip[ZIP_LEN+1];
    double     d_tax;
    double     d_ytd;
    long       name[20];
    long       d_next_o_id;
    long       time_start;
    long       w_id;
    RETCODE   rc;
    DBINT     rcint;
    char       bcphint[128];
    char       err_log_path[256];

    // Seed with unique number
    seed(4);

    printf("Loading district table...\n");

    // build index before load
    if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        BuildIndex("idxdiscl");

    InitString(d_name, D_NAME_LEN+1);
    InitAddress(d_street_1, d_street_2, d_city,
d_state, d_zip);
    sprintf(name, "%s.%s", aptr->database,
"district");

    strcpy(err_log_path, aptr->log_path);
    strcat(err_log_path, "district.err");
    rc = bcp_init(w_hdbc1, name, NULL,
err_log_path, DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
    {
        sprintf(bcphint, "tablock, order
(d_w_id, d_id), ROWS_PER_BATCH = %u", (aptr-
>num_warehouses * 10));
        rc = bcp_control(w_hdbc1,
BCPHINTS, (void*) bcphint);
        if (rc != SUCCEEDED)

            HandleErrorDBC(w_hdbc1);
    }

    i = 0;
    rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)

```

```

        HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&d_next_o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0,
D_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1,
0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2,
0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0,
STATE_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0,
ZIP_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);

        d_ytd = 30000.0;

        d_next_o_id = orders_per_district+1;

        time_start = (TimeNow() / MILLI);

        for (w_id = aptr->starting_warehouse; w_id
<= aptr->num_warehouses; w_id++)
        {
            d_w_id = w_id;

            for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
            {

                MakeAlphaStringPadded(6,10,D_NAME_LEN,
d_name);

                MakeAddress(d_street_1,
d_street_2, d_city, d_state, d_zip);

                d_tax = ((float)
RandomNumber(0L,2000L))/10000.00;

                rc =
bcp_sendrow(w_hdbc1);

                if (rc != SUCCEEDED)

```

```

        HandleErrorDBC(w_hdbc1);

        district_rows_loaded++;
        CheckForCommit(w_hdbc1,
w_hstmt1, district_rows_loaded, "district",
&time_start);
    }

    rcint = bcp_done(w_hdbc1);
    if (rcint < 0)
        HandleErrorDBC(w_hdbc1);

    printf("Finished loading district
table.\n");

    // if build index after load..
    if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
        BuildIndex("idxdiscl");

    return;
}

//=====
//
// Function : Stock
//
//=====
void Stock()
{
    int i;
    long s_i_id;
    long s_w_id;

    short s_quantity;
    char s_dist_01[S_DIST_LEN+1];
    char s_dist_02[S_DIST_LEN+1];
    char s_dist_03[S_DIST_LEN+1];
    char s_dist_04[S_DIST_LEN+1];
    char s_dist_05[S_DIST_LEN+1];
    char s_dist_06[S_DIST_LEN+1];
    char s_dist_07[S_DIST_LEN+1];
    char s_dist_08[S_DIST_LEN+1];
    char s_dist_09[S_DIST_LEN+1];
    char s_dist_10[S_DIST_LEN+1];
    long s_ytd;
    short s_order_cnt;
    short s_remote_cnt;
    char s_data[S_DATA_LEN+1];
    short len;
    char name[20];
    long time_start;
    RETCODE rc;
    DBINT rcint;
    char bcphint[128];
    char err_log_path[256];

    // Seed with unique number
    seed(3);

```

```

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
            BuildIndex("idxstkc1");

        sprintf(name, "%s.%s", aptr->database,
"stock");

        strcpy(err_log_path,aptr->log_path);
        strcat(err_log_path,"stock.err");
        rc = bcp_init(w_hdbc1, name, NULL,
err_log_path, DB_IN);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);

        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        {
            sprintf(bcp_hint, "tablock, order
(s_i_id, s_w_id), ROWS_PER_BATCH = %u", (aptr-
>num_warehouses * 100000));
            rc = bcp_control(w_hdbc1,
BCPHINTS, (void*) bcp_hint);
            if (rc != SUCCEEDED)

                HandleErrorDBC(w_hdbc1);
        }

        i = 0;
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&s_quantity, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&s_order_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&s_remote_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0,
SQL_VARLEN_DATA, "", 1, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01,
0, S_DIST_LEN, NULL, 0, 0, ++i);

```

```

        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02,
0, S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03,
0, S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04,
0, S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05,
0, S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06,
0, S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07,
0, S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08,
0, S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09,
0, S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10,
0, S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);

        s_ytd = s_order_cnt = s_remote_cnt = 0;
        time_start = (TimeNow() / MILLI);

        printf("...Loading stock table\n");
        for (s_i_id=1; s_i_id <= max_items;
s_i_id++)
        {
            for (s_w_id = (long)aptr-
>starting_warehouse; s_w_id <= aptr->num_warehouses;
s_w_id++)
                {
                    s_quantity =
(short)RandomNumber(10L,100L);
                    len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_01);
                    len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_02);
                    len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_03);

```

```

                    len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_04);
                    len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_05);
                    len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_06);
                    len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_07);
                    len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_08);
                    len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_09);
                    len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_10);

                    len =
MakeOriginalAlphaString(26,50, S_DATA_LEN,
s_data,10);

                    rc =
bcp_sendrow(w_hdbc1);

                    if (rc != SUCCEEDED)
                        HandleErrorDBC(w_hdbc1);

                    stock_rows_loaded++;

                    CheckForCommit_Big(w_hdbc1, w_hstmt1,
stock_rows_loaded, "stock", &time_start);
                }
            }

            rcint = bcp_done(w_hdbc1);
            if (rcint < 0)
                HandleErrorDBC(w_hdbc1);

            printf("Finished loading stock table.\n");

            SQLFreeStmt(w_hstmt1, SQL_DROP);
            SQLDisconnect(w_hdbc1);
            SQLFreeConnect(w_hdbc1);

            // if build index after load...
            if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
                BuildIndex("idxstkcl");

            return;
        }

//=====
//
// Function : LoadCustomer
//
//=====
void LoadCustomer()
{
    LOADER_TIME_STRUCT
customer_time_start;
    LOADER_TIME_STRUCT
history_time_start;

```

```

        long
w_id;
        short
d_id;
        DWORD
dwThreadId[MAX_CUSTOMER_THREADS];
        HANDLE
hThread[MAX_CUSTOMER_THREADS];
        char
name[20];
        RETCODE
rc;
        DBINT
rcint;
        char
bcphint[128];
        char
cmd[256];
        int
num_procs;
        char
err_log_path_cust[256];
        char
err_log_path_hist[256];

        // Seed with unique number
        seed(5);

        printf("Loading customer and history
tables...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        {
            BuildIndex("idxcusc1");
            // check the number of
processors on this system
            // if 8 or more processors, then
build index on History.
            // if less than 8 processors, do
not build the index
            num_procs = atoi(getenv(
"NUMBER_OF_PROCESSORS" ));
            if ( num_procs >= 8 )
                BuildIndex("idxhiscl");
        }

        // Initialize bulk copy
        sprintf(name, "%s..%s", aptr->database,
"customer");

        strcpy(err_log_path_cust,aptr->log_path);
        strcat(err_log_path_cust,"customer.err");
        rc = bcp_init(c_hdbc1, name, NULL,
err_log_path_cust, DB_IN);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        {
            sprintf(bcphint, "tablock, order
(c_w_id, c_d_id, c_id), ROWS_PER_BATCH = %u", (aptr-
>num_warehouses * 30000));

```

```

        rc = bcp_control(c_hdbc1,
BCPHINTS, (void*) bcphint);
        if (rc != SUCCEED)

            HandleErrorDBC(c_hdbc1);
    }

    sprintf(name, "%s..%s", aptr->database,
"history");

    rc = bcp_init(c_hdbc2, name, NULL,
"logs\\history.err", DB_IN);
    strcpy(err_log_path_hist, aptr->log_path);
    strcat(err_log_path_hist, "history.err");
    rc = bcp_init(c_hdbc2, name, NULL,
err_log_path_hist, DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    sprintf(bcphint, "tablock");
    rc = bcp_control(c_hdbc2, BCPHINTS, (void*)
bcphint);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    customer_rows_loaded = 0;
    history_rows_loaded = 0;

    CustomerBufInit();

    customer_time_start.time_start = (TimeNow()
/ MILLI);
    history_time_start.time_start = (TimeNow()
/ MILLI);

    for (w_id = (long)aptr->starting_warehouse;
w_id <= aptr->num_warehouses; w_id++)
    {
        for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
        {
            CustomerBufLoad(d_id,
w_id);

            // Start parallel
            // Start customer table
            thread
            printf("...Loading
customer table for: d_id = %d, w_id = %d\n", d_id,
w_id);

            hThread[0] =
CreateThread(NULL,

                0,

(LPTHREAD_START_ROUTINE) LoadCustomerTable,

&customer_time_start,

```

```

                0,

                &dwThreadID[0]);

                if (hThread[0] == NULL)
                {
                    printf("Error, failed in creating creating
thread = 0.\n");
                    exit(-1);
                }

                // Start History table
                thread
                printf("...Loading
history table for: d_id = %d, w_id = %d\n", d_id,
w_id);

                hThread[1] =
CreateThread(NULL,

                0,

(LPTHREAD_START_ROUTINE) LoadHistoryTable,

&history_time_start,

                0,

                &dwThreadID[1]);

                if (hThread[1] == NULL)
                {
                    printf("Error, failed in creating creating
thread = 1.\n");
                    exit(-1);
                }

                WaitForSingleObject(
hThread[0], INFINITE);
                WaitForSingleObject(
hThread[1], INFINITE);

                if
(CloseHandle(hThread[0]) == FALSE)
                {
                    printf("Error, failed in closing customer
thread handle with errno: %d\n", GetLastError());
                }

                if
(CloseHandle(hThread[1]) == FALSE)
                {
                    printf("Error, failed in closing history
thread handle with errno: %d\n", GetLastError());
                }

```

```

    }
}

// flush the bulk connection
rcint = bcp_done(c_hdbc1);
if (rcint < 0)
    HandleErrorDBC(c_hdbc1);

rcint = bcp_done(c_hdbc2);
if (rcint < 0)
    HandleErrorDBC(c_hdbc2);

printf("Finished loading customer
table.\n");

// if build index after load...
if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
{
    BuildIndex("idxcuscl");
    // check the number of processors
on this system
    // if 8 or more processors, then
build index on History.
    // if less than 8 processors, do
not build the index
    num_procs = atoi(getenv(
"NUMBER_OF_PROCESSORS"));
    if (num_procs >= 8)
        BuildIndex("idxhiscl");
}

// build non-clustered index
if (aptr->build_index == 1)
    BuildIndex("idxcusnc");

// Output the NURAND used for the loader
into C_FIRST for C_ID = 1,
// C_W_ID = 1, and C_D_ID = 1
sprintf(cmd, "osql -S%s -U%s -P%s -d%s -e -
Q\"update customer set c_first = 'C_LOAD = %d' where
c_id = 1 and c_w_id = 1 and c_d_id = 1\" >
%snurand_load.log",

aptr->server,
aptr->user,
aptr-
>password,
aptr-
>database,

LOADER_NURAND_C,

aptr-
>log_path);

system(cmd);

SQLFreeStmt(c_hstmt1, SQL_DROP);
SQLDisconnect(c_hdbc1);
SQLFreeConnect(c_hdbc1);

```



```

SQLFreeStmt(c_hstmt2, SQL_DROP);
SQLDisconnect(c_hdbc2);
SQLFreeConnect(c_hdbc2);

return;
}

//=====
//
// Function : CustomerBufInit
//
//=====
void CustomerBufInit()
{
    long i;

    for (i=0;i<customers_per_district;i++)
    {
        customer_buf[i].c_id = 0;
        customer_buf[i].c_d_id = 0;
        customer_buf[i].c_w_id = 0;

        strcpy(customer_buf[i].c_first,"");
        strcpy(customer_buf[i].c_middle,"");
        strcpy(customer_buf[i].c_last,"");
        strcpy(customer_buf[i].c_street_1,"");
        strcpy(customer_buf[i].c_street_2,"");
        strcpy(customer_buf[i].c_city,"");
        strcpy(customer_buf[i].c_state,"");
        strcpy(customer_buf[i].c_zip,"");
        strcpy(customer_buf[i].c_phone,"");
        strcpy(customer_buf[i].c_credit,"");

        customer_buf[i].c_credit_lim = 0;
        customer_buf[i].c_discount =
(float) 0;

        strcpy(customer_buf[i].c_balance,"");

        customer_buf[i].c_ytd_payment =
0;
        customer_buf[i].c_payment_cnt =
0;
        customer_buf[i].c_delivery_cnt =
0;

        strcpy(customer_buf[i].c_data,"");

        customer_buf[i].h_amount = 0;

```

```

        strcpy(customer_buf[i].h_data,"");
    }
}

//=====
//
// Function : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//=====
void CustomerBufLoad(int d_id, long w_id)
{
    long i;
    CUSTOMER_SORT_STRUCT
c[CUSTOMERS_PER_DISTRICT];

    for (i=0;i<customers_per_district;i++)
    {
        if (i < 1000)
            LastName(i,
c[i].c_last);
        else
            LastName(NURand(255,0,999,LOADER_NURAND_C),
c[i].c_last);

        MakeAlphaStringPadded(8,16,FIRST_NAME_LEN,
c[i].c_first);

        c[i].c_id = i+1;

        printf("...Loading customer buffer for:
d_id = %d, w_id = %d\n",
d_id, w_id);

        for (i=0;i<customers_per_district;i++)
        {
            customer_buf[i].c_d_id = d_id;
            customer_buf[i].c_w_id = w_id;
            customer_buf[i].h_amount = 10.0;
            customer_buf[i].c_ytd_payment =
10.0;
            customer_buf[i].c_payment_cnt =
1;
            customer_buf[i].c_delivery_cnt =
0;
            customer_buf[i].c_id = c[i].c_id;
            strcpy(customer_buf[i].c_first,
c[i].c_first);
            strcpy(customer_buf[i].c_last,
c[i].c_last);
            customer_buf[i].c_middle[0] =
'O';
            customer_buf[i].c_middle[1] =
'E';

            MakeAddress(customer_buf[i].c_street_1,

```

```

customer_buf[i].c_street_2,
customer_buf[i].c_city,
customer_buf[i].c_state,
customer_buf[i].c_zip);
            MakeNumberString(16, 16,
PHONE_LEN, customer_buf[i].c_phone);

            if (RandomNumber(1L, 100L) > 10)
                customer_buf[i].c_credit[0] = 'G';
            else
                customer_buf[i].c_credit[0] = 'B';
                customer_buf[i].c_credit[1] =
'C';
                customer_buf[i].c_credit_lim =
50000.0;
                customer_buf[i].c_discount =
((float) RandomNumber(0L, 5000L) / 10000.0;

                strcpy(customer_buf[i].c_balance,"-10.0");
                MakeAlphaStringPadded(300, 500,
C_DATA_LEN, customer_buf[i].c_data);

                // Generate HISTORY data
                MakeAlphaStringPadded(12, 24,
H_DATA_LEN, customer_buf[i].h_data);
            }
        }
    }

//=====
//
// Function : LoadCustomerTable
//
//=====
void LoadCustomerTable(LOADER_TIME_STRUCT
*customer_time_start)
{
    long i;
    long c_id;
    short c_d_id;
    long c_w_id;
    char c_first[FIRST_NAME_LEN+1];
    char c_middle[MIDDLE_NAME_LEN+1];
    char c_last[LAST_NAME_LEN+1];
    char c_street_1[ADDRESS_LEN+1];
    char c_street_2[ADDRESS_LEN+1];
    char c_city[ADDRESS_LEN+1];
    char c_state[STATE_LEN+1];
    char c_zip[ZIP_LEN+1];
    char c_phone[PHONE_LEN+1];
    char c_credit[CREDIT_LEN+1];
    double c_credit_lim;
    double c_discount;
    char c_balance[6];
    double c_ytd_payment;

```

```

short      c_payment_cnt;
short      c_delivery_cnt;
char       c_data[C_DATA_LEN+1];
char       char          c_since[C_SINCE_LEN+1];

RETCODE    rc;

i = 0;
rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0,
LAST_NAME_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0,
FIRST_NAME_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0,
CREDIT_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5,
NULL, 0, SQLCHARACTER, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment,
0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *)
&c_delivery_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);

```

```

if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0,
STATE_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0,
ZIP_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0,
PHONE_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) &c_since,
0, C_SINCE_LEN, NULL, 0, SQLCHARACTER, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_middle,
0, MIDDLE_NAME_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0,
C_DATA_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);

for (i = 0; i < customers_per_district; i++)
{
    c_id = customer_buf[i].c_id;
    c_d_id = customer_buf[i].c_d_id;
    c_w_id = customer_buf[i].c_w_id;

    strcpy(c_first,
customer_buf[i].c_first);
    strcpy(c_middle,
customer_buf[i].c_middle);
    strcpy(c_last,
customer_buf[i].c_last);
    strcpy(c_street_1,
customer_buf[i].c_street_1);
    strcpy(c_street_2,
customer_buf[i].c_street_2);
    strcpy(c_city,
customer_buf[i].c_city);
    strcpy(c_state,
customer_buf[i].c_state);
    strcpy(c_zip,
customer_buf[i].c_zip);
    strcpy(c_phone,
customer_buf[i].c_phone);
    strcpy(c_credit,
customer_buf[i].c_credit);

    FormatDate(&c_since);

    c_credit_lim =
customer_buf[i].c_credit_lim;

```

```

c_discount =
customer_buf[i].c_discount;
strcpy(c_balance,
customer_buf[i].c_balance);
c_ytd_payment =
customer_buf[i].c_ytd_payment;
c_payment_cnt =
customer_buf[i].c_payment_cnt;
c_delivery_cnt =
customer_buf[i].c_delivery_cnt;
strcpy(c_data,
customer_buf[i].c_data);

// Send data to server
rc = bcp_sendrow(c_hdbc1);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);

customer_rows_loaded++;
CheckForCommit(c_hdbc1, c_hstmt1,
customer_rows_loaded, "customer",
&customer_time_start->time_start);
}

//=====
//
// Function : LoadHistoryTable
//
//=====
void LoadHistoryTable(LOADER_TIME_STRUCT
*history_time_start)
{
    long          c_id;
    short         c_d_id;
    long          c_w_id;
    double        h_amount;
    char          h_data[H_DATA_LEN+1];
    char          h_date[H_DATE_LEN+1];

    RETCODE      rc;

    i = 0;
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc2);
rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc2);
rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc2);
rc = bcp_bind(c_hdbc2, (BYTE *) &c_delivery_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc2);

```

```

        rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0,
H_DATE_LEN, NULL, 0, SQLCHARACTER, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0,
H_DATA_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

        for (i = 0; i < customers_per_district; i++)
        {
            c_id = customer_buf[i].c_id;
            c_d_id = customer_buf[i].c_d_id;
            c_w_id = customer_buf[i].c_w_id;
            h_amount =
customer_buf[i].h_amount;
            strcpy(h_data,
customer_buf[i].h_data);

            FormatDate(&h_date);

            // send to server
            rc = bcp_sendrow(c_hdbc2);
            if (rc != SUCCEEDED)

                HandleErrorDBC(c_hdbc2);

            history_rows_loaded++;
            CheckForCommit(c_hdbc2, c_hstmt2,
history_rows_loaded, "history", &history_time_start-
>time_start);
        }
    }
//=====
//
// Function   : LoadOrders
//
//=====
void LoadOrders()
{
    LOADER_TIME_STRUCT    orders_time_start;
    LOADER_TIME_STRUCT
new_order_time_start;
    LOADER_TIME_STRUCT
order_line_time_start;
    long
w_id;
    short                d_id;

```

```

        DWORD
dwThreadId[MAX_ORDER_THREADS];
        HANDLE
hThread[MAX_ORDER_THREADS];
        char
name[20];
        RETCODE
rc;
        char
bcphint[128];
        char
err_log_path_ord[256];
        char
err_log_path_nord[256];
        char
err_log_path_ordl[256];

        // seed with unique number
        seed(6);

        printf("Loading orders...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        {
            BuildIndex("idxordc1");
            BuildIndex("idxnodc1");
            BuildIndex("idxodlcl");
        }

        // initialize bulk copy
        sprintf(name, "%s..%s", aptr->database,
"orders");

        rc = bcp_init(o_hdbc1, name, NULL,
"logs\\orders.err", DB_IN);
        strcpy(err_log_path_ord, aptr->log_path);
        strcat(err_log_path_ord, "orders.err");
        rc = bcp_init(o_hdbc1, name, NULL,
err_log_path_ord, DB_IN);
        if (rc != SUCCEEDED)
            HandleErrorDBC(o_hdbc1);

        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        {
            sprintf(bcphint, "tablock, order
(o_w_id, o_d_id, o_id), ROWS_PER_BATCH = %u", (aptr-
>num_warehouses * 30000));
            rc = bcp_control(o_hdbc1,
BCPHINTS, (void*) bcphint);
            if (rc != SUCCEEDED)

                HandleErrorDBC(o_hdbc1);
        }

        sprintf(name, "%s..%s", aptr->database,
"new_order");

        rc = bcp_init(o_hdbc2, name, NULL,
"logs\\neword.err", DB_IN);
        strcpy(err_log_path_nord, aptr->log_path);
        strcat(err_log_path_nord, "neword.err");

```

```

        rc = bcp_init(o_hdbc2, name, NULL,
err_log_path_nord, DB_IN);
        if (rc != SUCCEEDED)
            HandleErrorDBC(o_hdbc2);

        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        {
            sprintf(bcphint, "tablock, order
(no_w_id, no_d_id, no_o_id), ROWS_PER_BATCH = %u",
(aptr->num_warehouses * 9000));
            rc = bcp_control(o_hdbc2,
BCPHINTS, (void*) bcphint);
            if (rc != SUCCEEDED)

                HandleErrorDBC(o_hdbc2);
        }

        sprintf(name, "%s..%s", aptr->database,
"order_line");

        rc = bcp_init(o_hdbc3, name, NULL,
"logs\\ordline.err", DB_IN);
        strcpy(err_log_path_ordl, aptr->log_path);
        strcat(err_log_path_ordl, "ordline.err");
        rc = bcp_init(o_hdbc3, name, NULL,
err_log_path_ordl, DB_IN);
        if (rc != SUCCEEDED)
            HandleErrorDBC(o_hdbc3);

        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        {
            sprintf(bcphint, "tablock, order
(ol_w_id, ol_d_id, ol_o_id, ol_number),
ROWS_PER_BATCH = %u", (aptr->num_warehouses *
300000));
            rc = bcp_control(o_hdbc3,
BCPHINTS, (void*) bcphint);
            if (rc != SUCCEEDED)

                HandleErrorDBC(o_hdbc3);
        }

        orders_rows_loaded = 0;
        new_order_rows_loaded = 0;
        order_line_rows_loaded = 0;

        OrdersBufInit();

        orders_time_start.time_start = (TimeNow() /
MILLI);
        new_order_time_start.time_start =
(TimeNow() / MILLI);
        order_line_time_start.time_start =
(TimeNow() / MILLI);

        for (w_id = (long)aptr->starting_warehouse;
w_id <= aptr->num_warehouses; w_id++)
        {
            for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
            {

```

<pre> w_id); OrdersBufLoad(d_id, // start parallel loading threads here... thread printf(...Loading Order Table for: d_id = %d, w_id = %d\n", d_id, w_id); CreateThread(NULL, hThread[0] = 0, (LPCTSTR) LoadOrdersTable, &orders_time_start, 0, &dwThreadID[0]); if (hThread[0] == NULL) { printf("Error, failed in creating creating thread = 0.\n"); exit(-1); } // start NewOrder table thread printf(...Loading New- Order Table for: d_id = %d, w_id = %d\n", d_id, w_id); CreateThread(NULL, hThread[1] = 0, (LPCTSTR) LoadNewOrderTable, &new_order_time_start, 0, &dwThreadID[1]); if (hThread[1] == NULL) { printf("Error, failed in creating creating thread = 1.\n"); </pre>	<pre> } // start Order-Line table thread printf(...Loading Order-Line Table for: d_id = %d, w_id = %d\n", d_id, w_id); CreateThread(NULL, hThread[2] = 0, (LPCTSTR) LoadOrderLineTable, &order_line_time_start, 0, &dwThreadID[2]); if (hThread[2] == NULL) { printf("Error, failed in creating creating thread = 2.\n"); exit(-1); } WaitForSingleObject(hThread[0], INFINITE); WaitForSingleObject(hThread[1], INFINITE); WaitForSingleObject(hThread[2], INFINITE); if (CloseHandle(hThread[0]) == FALSE) { printf("Error, failed in closing Orders thread handle with errno: %d\n", GetLastError()); } if (CloseHandle(hThread[1]) == FALSE) { printf("Error, failed in closing NewOrder thread handle with errno: %d\n", GetLastError()); } if (CloseHandle(hThread[2]) == FALSE) { printf("Error, failed in closing OrderLine thread handle with errno: %d\n", GetLastError()); } </pre>	<pre> } printf("Finished loading orders.\n"); return; } //===== // // Function : OrdersBufInit // // Clears shared buffer for ORDERS, NEWORDER, and ORDERLINE //===== void OrdersBufInit() { int i; int j; for (i=0;i<orders_per_district;i++) { orders_buf[i].o_id = 0; orders_buf[i].o_d_id = 0; orders_buf[i].o_w_id = 0; orders_buf[i].o_c_id = 0; orders_buf[i].o_carrier_id = 0; orders_buf[i].o_ol_cnt = 0; orders_buf[i].o_all_local = 0; for (j=0;j<=14;j++) { orders_buf[i].o_ol[j].ol = 0; orders_buf[i].o_ol[j].ol_i_id = 0; orders_buf[i].o_ol[j].ol_supply_w_id = 0; orders_buf[i].o_ol[j].ol_quantity = 0; orders_buf[i].o_ol[j].ol_amount = 0; strcpy(orders_buf[i].o_ol[j].ol_dist_info," "); } } //===== // // Function : OrdersBufLoad // // Fills shared buffer for ORDERS, NEWORDER, and ORDERLINE //===== void OrdersBufLoad(short d_id, long w_id) </pre>
---	--	--

```

{
    int      cust[ORDERS_PER_DISTRICT+1];
    long     o_id;
    long     ol;

    printf("...Loading Order Buffer for: d_id =
%d, w_id = %d\n",
           d_id, w_id);

    GetPermutation(cust, orders_per_district);

    for
(o_id=0;o_id<orders_per_district;o_id++)
    {
        // Generate ORDER and NEW-ORDER
        data
        orders_buf[o_id].o_d_id = d_id;
        orders_buf[o_id].o_w_id = w_id;
        orders_buf[o_id].o_id = o_id+1;
        orders_buf[o_id].o_c_id =
cust[o_id+1];
        orders_buf[o_id].o_ol_cnt =
(short)RandomNumber(5L, 15L);

        if (o_id < first_new_order)
        {
            orders_buf[o_id].o_carrier_id =
(short)RandomNumber(1L, 10L);

            orders_buf[o_id].o_all_local = 1;
        }
        else
        {
            orders_buf[o_id].o_carrier_id = 0;

            orders_buf[o_id].o_all_local = 1;
        }

        for (ol=0;
ol<orders_buf[o_id].o_ol_cnt; ol++)
        {
            orders_buf[o_id].o_ol[ol].ol = ol+1;

            orders_buf[o_id].o_ol[ol].ol_i_id =
RandomNumber(1L, max_items);

            orders_buf[o_id].o_ol[ol].ol_supply_w_id =
w_id;

            orders_buf[o_id].o_ol[ol].ol_quantity = 5;
            MakeAlphaString(24, 24,
OL_DIST_INFO_LEN,
&orders_buf[o_id].o_ol[ol].ol_dist_info);

            // Generate ORDER-LINE
            data
            if (o_id <
first_new_order)

```

```

{
    orders_buf[o_id].o_ol[ol].ol_amount = 0;
    insure ol_delivery_d set properly during load

    FormatDate(&orders_buf[o_id].o_ol[ol].ol_de
livery_d);
    }
    else
    {
        orders_buf[o_id].o_ol[ol].ol_amount =
RandomNumber(1,999999)/100.0;
        // Added to
        insure ol_delivery_d set properly during load

        // odbc
        datetime format
        strcpy(orders_buf[o_id].o_ol[ol].ol_deliver
y_d,"1899-12-31 00:00:00.000");
    }
}

//=====
//
// Function : LoadOrdersTable
//
//=====
void LoadOrdersTable(LOADER_TIME_STRUCT
*orders_time_start)
{
    int      i;
    long     o_id;
    short    o_d_id;
    long     o_w_id;

    long     o_c_id;
    short    o_carrier_id;
    short    o_ol_cnt;
    short    o_all_local;

    char     o_entry_d[O_ENTRY_D_LEN+1];
    RETCODE  rc;
    DBINT    rcint;

    // bind ORDER data
    i = 0;
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);

```

```

    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d,
0, O_ENTRY_D_LEN, NULL, 0, SQLCHARACTER, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);

    for (i = 0; i < orders_per_district; i++)
    {
        o_id      =
orders_buf[i].o_id;
        o_d_id    =
orders_buf[i].o_d_id;
        o_w_id    =
orders_buf[i].o_w_id;
        o_c_id    =
orders_buf[i].o_c_id;
        o_carrier_id =
orders_buf[i].o_carrier_id;
        o_ol_cnt  =
orders_buf[i].o_ol_cnt;
        o_all_local =
orders_buf[i].o_all_local;

        FormatDate(&o_entry_d);

        // send data to server
        rc = bcp_sendrow(o_hdbc1);
        if (rc != SUCCEEDED)
            HandleErrorDBC(o_hdbc1);

        orders_rows_loaded++;
        CheckForCommit(o_hdbc1, o_hstmt1,
orders_rows_loaded, "orders", &orders_time_start-
>time_start);
    }

    if ((o_w_id == aptr->num_warehouses) &&
(o_d_id == 10))
    {
        rcint = bcp_done(o_hdbc1);

        if (rcint < 0)
            HandleErrorDBC(o_hdbc1);
    }

```

```

SQLFreeStmt(o_hstmt1, SQL_DROP);
SQLDisconnect(o_hdbc1);
SQLFreeConnect(o_hdbc1);

// if build index after load...
if ((aptr->build_index == 1) &&
(aptr->index_order == 0))
    BuildIndex("idxordc1");

// build non-clustered index
if (aptr->build_index == 1)
    BuildIndex("idxordnc");
}

//=====
// Function : LoadNewOrderTable
//
//=====
void LoadNewOrderTable(LOADER_TIME_STRUCT
*new_order_time_start)
{
    long    long    i;
    long    o_id;
    short   o_d_id;
    long    o_w_id;
    RETCODE rc;
    DBINT   rcint;

    // Bind NEW-ORDER data
    i = 0;
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

    for (i = first_new_order; i <
last_new_order; i++)
    {
        o_id = orders_buf[i].o_id;
        o_d_id = orders_buf[i].o_d_id;
        o_w_id = orders_buf[i].o_w_id;

        rc = bcp_sendrow(o_hdbc2);
        if (rc != SUCCEEDED)

            HandleErrorDBC(o_hdbc2);

        new_order_rows_loaded++;
    }
}

```

```

CheckForCommit_Big(o_hdbc2,
o_hstmt2, new_order_rows_loaded, "new_order",
&new_order_time_start->time_start);
}

if ((o_w_id == aptr->num_warehouses) &&
(o_d_id == 10))
{
    rcint = bcp_done(o_hdbc2);

    if (rcint < 0)
        HandleErrorDBC(o_hdbc2);

    SQLFreeStmt(o_hstmt2, SQL_DROP);
    SQLDisconnect(o_hdbc2);
    SQLFreeConnect(o_hdbc2);

    // if build index after load...
    if ((aptr->build_index == 1) &&
(aptr->index_order == 0))
        BuildIndex("idxmodc1");
}

//=====
// Function : LoadOrderLineTable
//
//=====
void LoadOrderLineTable(LOADER_TIME_STRUCT
*order_line_time_start)
{
    long    long    i;
    long    long    j;
    long    o_id;
    short   o_d_id;
    long    o_w_id;
    double  ol;
    long    ol_i_id;
    long    ol_supply_w_id;
    short   ol_quantity;
    double  ol_amount;
    char    ol_dist_info[DIST_INFO_LEN+1];
    char    ol_delivery_d[OL_DELIVERY_D_LEN+1];
    RETCODE rc;
    DBINT   rcint;

    // bind ORDER-LINE data
    i = 0;
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
}

```

```

if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);
rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);
rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);
rc = bcp_bind(o_hdbc3, (BYTE *)
&ol_delivery_d, 0, OL_DELIVERY_D_LEN, NULL, 0,
SQLCHARACTER, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);
rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);
rc = bcp_bind(o_hdbc3, (BYTE *)
&ol_supply_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);
rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);
rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0,
DIST_INFO_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

for (i = 0; i < orders_per_district; i++)
{
    o_id = orders_buf[i].o_id;
    o_d_id = orders_buf[i].o_d_id;
    o_w_id = orders_buf[i].o_w_id;

    for (j=0; j <
orders_buf[i].o_ol_cnt; j++)
    {
        ol =
orders_buf[i].o_ol[j].ol;
        ol_i_id =
orders_buf[i].o_ol[j].ol_i_id;
        ol_supply_w_id =
orders_buf[i].o_ol[j].ol_supply_w_id;
        ol_quantity =
orders_buf[i].o_ol[j].ol_quantity;
        ol_amount =
orders_buf[i].o_ol[j].ol_amount;

        strcpy(ol_delivery_d,orders_buf[i].o_ol[j].
ol_delivery_d);

        strcpy(ol_dist_info,orders_buf[i].o_ol[j].o
l_dist_info);

        rc =
bcp_sendrow(o_hdbc3);
    }
}

```

```

        if (rc != SUCCEEDED)

        HandleErrorDBC(o_hdbc3);

        order_line_rows_loaded++;

        CheckForCommit_Big(o_hdbc3, o_hstmt3,
order_line_rows_loaded, "order_line",
&order_line_time_start->time_start);
    }
    }
    if ((o_w_id == aptr->num_warehouses) &&
(o_d_id == 10))
    {
        rcint = bcp_done(o_hdbc3);
        if (rcint < 0)

        HandleErrorDBC(o_hdbc3);

        SQLFreeStmt(o_hstmt3, SQL_DROP);
        SQLDisconnect(o_hdbc3);
        SQLFreeConnect(o_hdbc3);

        // if build index after load...
        if ((aptr->build_index == 1) &&
(aptr->index_order == 0))
            BuildIndex("idxodlcl1");
    }
}

//=====
//
// Function   : GetPermutation
//
//=====
void GetPermutation(int perm[], int n)
{
    int i, r, t;

    for (i=1;i<=n;i++)
        perm[i] = i;

    for (i=1;i<=n;i++)
    {
        r = RandomNumber(i,n);
        t = perm[i];
        perm[i] = perm[r];
        perm[r] = t;
    }
}

//=====
//
// Function   : CheckForCommit
//

```

```

//=====
void CheckForCommit(HDBC hdbc,
                    HSTMT hstmt,
                    long rows_loaded,
                    char *table_name,
                    long
*time_start)
{
    long time_end, time_diff;

    if ( !(rows_loaded % aptr->batch) )
    {
        time_end = (TimeNow() / MILLI);
        time_diff = time_end -
*time_start;

        printf("-> Loaded %ld rows into
%s in %ld sec - Total = %d (%.2f rps)\n",
aptr->batch,
table_name,
time_diff,
rows_loaded,
(float) aptr-
>batch / (time_diff ? time_diff : 1L));

        *time_start = time_end;
    }
}

return;

//=====
//
// Function   : CheckForCommit_Big
//
//=====
void CheckForCommit_Big(HDBC hdbc,
                        HSTMT hstmt,
                        double rows_loaded,
                        char *table_name,
                        long
*time_start)
{
    long time_end, time_diff;

    if ( !(fmod(rows_loaded,aptr->batch) ) )
    {
        time_end = (TimeNow() / MILLI);
        time_diff = time_end -
*time_start;

```

```

        printf("-> Loaded %ld rows into
%s in %ld sec - Total = %.0f (%.2f rps)\n",
aptr->batch,
table_name,
time_diff,
rows_loaded,
(float) aptr-
>batch / (time_diff ? time_diff : 1L));
    }
    *time_start = time_end;
}

return;

//=====
//
// Function   : OpenConnections
//
//=====
void OpenConnections()
{
    RETCODE rc;

    char
szDriverString[300];
char
szDriverStringOut[1024];
SQLSMALLINT
cbDriverStringOut;

    SQLAllocHandle(SQL_HANDLE_ENV,
SQL_NULL_HANDLE, &henv );

    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
(void*)SQL_OV_ODBC3, 0 );

    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&i_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&w_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&c_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&c_hdbc2);
    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&o_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&o_hdbc2);
    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&o_hdbc3);

    SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
    SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
    SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );

```

```

        SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );

        // Open connections to SQL Server
        // Connection 1
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

        aptr->server,

        aptr->user,

        aptr->password,

        aptr->database );

        rc = SQLSetConnectOption (i_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEEDED)
            HandleErrorDBC(i_hdbc1);

        rc = SQLDriverConnect ( i_hdbc1,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEEDED) &&
(rc != SQL_SUCCESS_WITH_INFO) )
        {
            HandleErrorDBC(i_hdbc1);
            printf("TPC-C Loader
aborted!\n");
            exit(9);
        }

        // Connection 2
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

        aptr->server,

        aptr->user,

        aptr->password,

        aptr->database );

```

```

        rc = SQLSetConnectOption (w_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);

        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);

        rc = SQLDriverConnect ( w_hdbc1,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );

        if ( (rc != SUCCEEDED) &&
(rc != SQL_SUCCESS_WITH_INFO) )
        {
            HandleErrorDBC(w_hdbc1);
            printf("TPC-C Loader
aborted!\n");
            exit(9);
        }

        // Connection 3
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

        aptr->server,

        aptr->user,

        aptr->password,

        aptr->database );

        rc = SQLSetConnectOption (c_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = SQLDriverConnect ( c_hdbc1,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );

```

```

        if ( (rc != SUCCEEDED) &&
(rc != SQL_SUCCESS_WITH_INFO) )
        {
            HandleErrorDBC(c_hdbc1);
            printf("TPC-C Loader
aborted!\n");
            exit(9);
        }

        // Connection 4
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

        aptr->server,

        aptr->user,

        aptr->password,

        aptr->database );

        rc = SQLSetConnectOption (c_hdbc2,
SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

        rc = SQLDriverConnect ( c_hdbc2,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEEDED) &&
(rc != SQL_SUCCESS_WITH_INFO) )
        {
            HandleErrorDBC(c_hdbc2);
            printf("TPC-C Loader
aborted!\n");
            exit(9);
        }

        // Connection 5
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

        aptr->server,

        aptr->user,

        aptr->password,

        aptr->database );

```



```

rc = SQLSetConnectOption (o_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

rc = SQLDriverConnect ( o_hdbc1,

    NULL,

    (SQLCHAR*)&szDriverString[0] ,

    SQL_NTS,

    (SQLCHAR*)&szDriverStringOut[0],

    sizeof(szDriverStringOut),

    &cbDriverStringOut,

    SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEEDED) &&
(rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(o_hdbc1);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

// Connection 6
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

    aptr->server,

    aptr->user,

    aptr->password,

    aptr->database );

rc = SQLSetConnectOption (o_hdbc2,
SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc2);

rc = SQLDriverConnect ( o_hdbc2,

    NULL,

    (SQLCHAR*)&szDriverString[0] ,

    SQL_NTS,

    (SQLCHAR*)&szDriverStringOut[0],

    sizeof(szDriverStringOut),

    &cbDriverStringOut,

    SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEEDED) &&
(rc != SQL_SUCCESS_WITH_INFO) )

```

```

{
    HandleErrorDBC(o_hdbc2);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

// Connection 7
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

    aptr->server,

    aptr->user,

    aptr->password,

    aptr->database );

rc = SQLSetConnectOption (o_hdbc3,
SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = SQLDriverConnect ( o_hdbc3,

    NULL,

    (SQLCHAR*)&szDriverString[0] ,

    SQL_NTS,

    (SQLCHAR*)&szDriverStringOut[0],

    sizeof(szDriverStringOut),

    &cbDriverStringOut,

    SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEEDED) &&
(rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(o_hdbc3);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

//=====
//
// Function name: BuildIndex
//
//=====
void BuildIndex(char          *index_script)
{
    char    cmd[256];

    printf("Starting index creation:
%s\n",index_script);

```

```

    sprintf(cmd, "osql -S%s -U%s -P%s -e -
i%s\\%s.sql > %s%.log",

    aptr->server,
    aptr->user,
    aptr->password,
    aptr->index_script_path,
    aptr->log_path,
    index_script);

    system(cmd);

    printf("Finished index creation:
%s\n",index_script);
}

//=====
//
// Function name: HandleErrorDBC
//
//=====
void HandleErrorDBC (SQLHDBC hdbc1)
{
    SQLCHAR    SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLLEN    NativeError;
    SQLSMALLINT i, MsgLen;
    SQLRETURN rc2;
    char    timebuf[128];
    char    datebuf[128];
    char    err_log_path[256];
    FILE    *fpl;

    i = 1;
    while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC
, hdbc1, i, SqlState , &NativeError,
Msg,
sizeof(Msg) , &MsgLen) != SQL_NO_DATA )
    {
        printf( szLastError , "%s" ,
Msg );

        _strtime(timebuf);
        _strdate(datebuf);

        printf( "[%s : %s]
%s\n=>SQLState: %s\n" , datebuf, timebuf,
szLastError, SqlState);

        strcpy(err_log_path,aptr-
>log_path);

        strcat(err_log_path,"tpccldr.err");
        fpl = fopen(err_log_path,"a+");
        if (fpl == NULL)

```

```

                printf("ERROR:  Unable
to open errorlog file.\n");
            else
            {
                fprintf(fp1, "[%s : %s]
%s\nSQLState: %s\n" , datebuf, timebuf, szLastError,
SqlState);
                fclose(fp1);
            }
            i++;
        }
    }

//=====
//
// Function   : HandleErrorSTMT
//
//=====
void HandleErrorSTMT (HSTMT  hstmt1)
{
    SQLCHAR          SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLLEN           NativeError;
    SQLSMALLINT i, MsgLen;
    SQLRETURN        rc2;
    char              timebuf[128];
    char              datebuf[128];
    char              err_log_path[256];
    FILE              *fp1;

    i = 1;
    while (( rc2 =
SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i, SqlState ,
&NativeError,
                Msg,
sizeof(Msg) , &MsgLen )) != SQL_NO_DATA )
    {
        MAX_SQL_ERRORS)
        if (total_db_errors >=
        {
            printf(">>>> Maximum
SQL errors of %d exceeded. Terminating
TPCCLDR.<<<<<\n",total_db_errors);
            exit(9);
        }
        total_db_errors++;
        sprintf( szLastError , "%s" ,
Msg );
        _strtime(timebuf);
        _strdate(datebuf);
        printf( "[%s : %s] %s\nSQLState:
%s\n" , datebuf, timebuf, szLastError, SqlState);
        strcpy(err_log_path,aptr-
>log_path);
        strcat(err_log_path,"tpccldr.err");

```

```

                fp1 = fopen(err_log_path,"a+");
                if (fp1 == NULL)
                    printf("ERROR:  Unable
to open errorlog file.\n");
            else
            {
                fprintf(fp1, "[%s : %s]
%s\nSQLState: %s\n" , datebuf, timebuf, szLastError,
SqlState);
                fclose(fp1);
            }
            i++;
        }
    }

//=====
//
// Function   : FormatDate
//
//=====
void FormatDate ( char* szTimeCOutput )
{
    struct tm when;
    time_t now;

    time( &now );
    when = *localtime( &now );

    mktime( &when );

    // odbc datetime format
    strftime( szTimeCOutput , 30 , "%Y-%m-%d
%H:%M:%S.000" , &when );

    return;
}

```

tpcc_neworder_new.sql

```

-----
--
-- File:   TPCC_NEWORDER_NEW.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
--
-- This acid stored procedure implements the
neworder --
-- transaction.  It outputs timestamps at
the
--

```

```

-- beginning of the transaction, before the
commit --
-- delay, and after the commit.
--
--
-----
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_neworder_new' )
    DROP PROCEDURE tpcc_neworder_new
GO

-- neworder_new v2.5 6/23/05 PeterCa
-- lq stock/order_line/client. upd district & ins
neworder.
-- cust/warehouse select together, ins order
separate
-- uses rownumber to distinct w any transform
-- uses in-memory sort for distinct on iid,wid
-- uses charindex
-- will rollback if (@i_idX,@s_w_idX pairs not
unique) OR (@i_idX not unique).

CREATE PROCEDURE tpcc_neworder_new
    @w_id int,
    @d_id tinyint,
    @c_id int,
    @o_ol_cnt tinyint,
    @o_all_local tinyint,
    @i_id1 int = 0, @s_w_id1
int = 0, @ol_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2
int = 0, @ol_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3
int = 0, @ol_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4
int = 0, @ol_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5
int = 0, @ol_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6
int = 0, @ol_qty6 smallint = 0,
    @i_id7 int = 0, @s_w_id7
int = 0, @ol_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8
int = 0, @ol_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9
int = 0, @ol_qty9 smallint = 0,
    @i_id10 int = 0, @s_w_id10
int = 0, @ol_qty10 smallint = 0,
    @i_id11 int = 0, @s_w_id11
int = 0, @ol_qty11 smallint = 0,
    @i_id12 int = 0, @s_w_id12
int = 0, @ol_qty12 smallint = 0,

```

```

        @i_id13 int = 0, @s_w_id13
int = 0, @ol_qty13 smallint = 0,
        @i_id14 int = 0, @s_w_id14
int = 0, @ol_qty14 smallint = 0,
        @i_id15 int = 0, @s_w_id15
int = 0, @ol_qty15 smallint = 0

AS
BEGIN
DECLARE @o_id          int,
        @d_tax         smallmoney,
        @o_entry_d     datetime,
        @commit_flag   tinyint

BEGIN TRANSACTION n
-- get district tax and next available order id
and update
-- insert corresponding row into new-order table
-- plus initialize local variables

UPDATE district
SET    @d_tax         = d_tax,
        @o_id         = d_next_o_id,
        d_next_o_id   = d_next_o_id + 1,
        @o_entry_d    = GETDATE(),
        @commit_flag  = 1
OUTPUT deleted.d_next_o_id,
        @d_id,
        @w_id

INTO new_order
WHERE d_w_id = @w_id AND
      d_id   = @d_id

-- update stock from stock join (item join
(params))
-- output to orderline, output to client
-- NOTE: @@rowcount != @ol_o_cnt
-- if (@i_idX,@s_w_idX pairs not unique) OR
(@i_idX not unique).

UPDATE stock
SET    s_ytd         = s_ytd + info.ol_qty,
        s_quantity   = s_quantity -
info.ol_qty +
        CASE WHEN (s_quantity -
info.ol_qty < 10) THEN 91 ELSE 0 END,
        s_order_cnt  = s_order_cnt + 1,
        s_remote_cnt = s_remote_cnt +
        CASE
WHEN (info.w_id = @w_id) THEN 0 ELSE 1 END

OUTPUT @o_id,
        @d_id,
        @w_id,
        info.lino,
        info.i_id,
        "dec 31, 1899",
        info.i_price * info.ol_qty,
        info.w_id,
        info.ol_qty,

```

```

        CASE @d_id WHEN 1 THEN
inserted.s_dist_01
        WHEN 2 THEN
inserted.s_dist_02
        WHEN 3 THEN
inserted.s_dist_03
        WHEN 4 THEN
inserted.s_dist_04
        WHEN 5 THEN
inserted.s_dist_05
        WHEN 6 THEN
inserted.s_dist_06
        WHEN 7 THEN
inserted.s_dist_07
        WHEN 8 THEN
inserted.s_dist_08
        WHEN 9 THEN
inserted.s_dist_09
        WHEN 10 THEN
inserted.s_dist_10
        END
        INTO order_line
        OUTPUT info.i_name,inserted.s_quantity,
        CASE WHEN
((charindex("ORIGINAL",info.i_data) > 0) AND
(charindex("ORIGINAL",inserted.s_data) > 0) )
        THEN "B" ELSE "G" END,
        info.i_price,
        info.i_price*info.ol_qty
FROM stock INNER JOIN
(SELECT iid,
        wid,
        lino,
        ol_qty,
        i_price,
        i_name,
        i_data
FROM (SELECT iid,
        wid,
        lino,
        qty,
        row_number()
OVER (PARTITION BY iid,wid ORDER BY iid,wid)
FROM (SELECT
@i_id1,@s_w_id1,1,@ol_qty1
        UNION ALL
SELECT
@i_id2,@s_w_id2,2,@ol_qty2
        UNION ALL
SELECT
@i_id3,@s_w_id3,3,@ol_qty3
        UNION ALL
SELECT
@i_id4,@s_w_id4,4,@ol_qty4
        UNION ALL
SELECT
@i_id5,@s_w_id5,5,@ol_qty5
        UNION ALL
SELECT
@i_id6,@s_w_id6,6,@ol_qty6
        UNION ALL
SELECT
@i_id7,@s_w_id7,7,@ol_qty7
        UNION ALL
SELECT
@i_id8,@s_w_id8,8,@ol_qty8
        UNION ALL
SELECT
@i_id9,@s_w_id9,9,@ol_qty9
        UNION ALL

```

```

        SELECT
@i_id10,@s_w_id10,10,@ol_qty10
        UNION ALL
SELECT
@i_id11,@s_w_id11,11,@ol_qty11
        UNION ALL
SELECT
@i_id12,@s_w_id12,12,@ol_qty12
        UNION ALL
SELECT
@i_id13,@s_w_id13,13,@ol_qty13
        UNION ALL
SELECT
@i_id14,@s_w_id14,14,@ol_qty14
        UNION ALL
SELECT
@i_id15,@s_w_id15,15,@ol_qty15) AS
uol(iid,wid,lino,qty)
        ) AS
ol(iid,wid,lino,ol_qty,rownum)
        INNER JOIN
        item (repeatableread) ON
i_id = iid AND -- filters out invalid items
rownum = 1
        ) AS
info(i_id,w_id,lino,ol_qty,i_price,i_name,i_data)
        ON s_i_id = info.i_id AND
        s_w_id = info.w_id

IF (@@rowcount <> @ol_cnt) -- must have an
invalid item
        SELECT @commit_flag = 0 -- 2.4.2.3 requires
rest to proceed

-- insert fresh row into orders table
INSERT INTO orders VALUES ( @o_id,
        @d_id,
        @w_id,
        @c_id,
        0,
        @o_ol_cnt,
        @o_all_local,
        @o_entry_d)

-- get customer last name, discount, and credit
rating
-- get warehouse tax
-- return order_data to client

SELECT w_tax,
        @d_tax,
        @o_id,
        c_last,
        c_discount,
        c_credit,
        @o_entry_d,
        @commit_flag
FROM warehouse (repeatableread),
customer (repeatableread)
WHERE w_id = @w_id AND
      c_id = @c_id AND
      c_w_id = @w_id AND
      c_d_id = @d_id

-- @@rowcount checks that previous select
found a valid customer
IF ((@commit_flag = 1) AND (@@rowcount = 1))

```



```

        @NEW_ORDER_TARGET      = @NUM_WAREHOUSE *
9000,
        @NEW_ORDER_TARGET_LOW  = @NEW_ORDER_TARGET -
FLOOR(@NEW_ORDER_TARGET * .01),
        @NEW_ORDER_TARGET_HIGH = @NEW_ORDER_TARGET +
FLOOR(@NEW_ORDER_TARGET * .01),
        @HISTORY_TARGET        = @NUM_WAREHOUSE *
30000,
        @STOCK_TARGET          = @NUM_WAREHOUSE *
100000

--- insert the values into TPCC_INFO
INSERT INTO TPCC_INFO VALUES (GETDATE(),
    @NUM_WAREHOUSE,
    @WAREHOUSE_TARGET,
    @NUM_DISTRICT,
    @DISTRICT_TARGET,
    @NUM_ITEM,
    @ITEM_TARGET,
    @NUM_CUSTOMER,
    @CUSTOMER_TARGET,
    @NUM_ORDERS,
    @ORDERS_TARGET,
    @ORDERS_TARGET_LOW,
    @ORDERS_TARGET_HIGH,
    @NUM_ORDER_LINE,
    @ORDER_LINE_TARGET,

@ORDER_LINE_TARGET_LOW,

@ORDER_LINE_TARGET_HIGH,

@NUM_NEW_ORDER,
@NEW_ORDER_TARGET,

@NEW_ORDER_TARGET_LOW,

@NEW_ORDER_TARGET_HIGH,

    @NUM_HISTORY,
    @HISTORY_TARGET,
    @NUM_STOCK,
    @STOCK_TARGET)
GO

--- output the row counts from the build
PRINT ''
PRINT ''
PRINT '-----'
PRINT '| WAREHOUSE TABLE |'
PRINT '-----'
SELECT TOP 1
    CONVERT (CHAR (30), INFO_DATE, 21) AS 'Date',
    NUM_WAREHOUSE AS
'Warehouse Rows',
    WAREHOUSE_TARGET AS
'Warehouse Target',
    CASE WHEN (NUM_WAREHOUSE = WAREHOUSE_TARGET)
        THEN 'OK!'
        ELSE 'ERROR!!!'
    END AS
'Warehouse Message'
FROM TPCC_INFO
GO

```

```

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| DISTRICT TABLE |'
PRINT '-----'
SELECT TOP 1
    CONVERT (CHAR (30), INFO_DATE, 21) AS 'Date',
    NUM_DISTRICT AS 'District
Rows',
    DISTRICT_TARGET AS
'District Target',
    CASE WHEN (NUM_DISTRICT = DISTRICT_TARGET)
        THEN 'OK!'
        ELSE 'ERROR!!!'
    END AS 'District
Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| ITEM TABLE |'
PRINT '-----'
SELECT TOP 1
    CONVERT (CHAR (30), INFO_DATE, 21) AS 'Date',
    NUM_ITEM AS 'Item
Rows',
    ITEM_TARGET AS
'Item Target',
    CASE WHEN (NUM_ITEM = ITEM_TARGET)
        THEN 'OK!'
        ELSE 'ERROR!!!'
    END AS 'Item
Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| CUSTOMER TABLE |'
PRINT '-----'
SELECT TOP 1
    CONVERT (CHAR (30), INFO_DATE, 21) AS 'Date',
    NUM_CUSTOMER AS 'Customer
Rows',
    CUSTOMER_TARGET AS
'Customer Target',
    CASE WHEN (NUM_CUSTOMER = CUSTOMER_TARGET)
        THEN 'OK!'
        ELSE 'ERROR!!!'
    END AS 'Customer
Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| ORDERS TABLE |'
PRINT '-----'
SELECT TOP 1

```

```

    CONVERT (CHAR (30), INFO_DATE, 21) AS 'Date',
    NUM_ORDERS AS 'Orders
Rows',
    ORDERS_TARGET AS
'Orders Target',
    CASE WHEN (NUM_ORDERS = ORDERS_TARGET)
        THEN 'OK!'
        WHEN (NUM_ORDERS BETWEEN
ORDERS_TARGET_LOW AND ORDERS_TARGET_HIGH)
        THEN 'OK! (within 1%)'
        ELSE 'ERROR!!!'
    END AS 'Orders
Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| ORDER LINE TABLE |'
PRINT '-----'
SELECT TOP 1
    CONVERT (CHAR (30), INFO_DATE, 21) AS 'Date',
    NUM_ORDER_LINE AS 'Order
Line Rows',
    ORDER_LINE_TARGET AS
'Order Line Target',
    CASE WHEN (NUM_ORDER_LINE =
ORDER_LINE_TARGET)
        THEN 'OK!'
        WHEN (NUM_ORDER_LINE BETWEEN
ORDER_LINE_TARGET_LOW AND ORDER_LINE_TARGET_HIGH)
        THEN 'OK! (within 1%)'
        ELSE 'ERROR!!!'
    END AS 'Orders
Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| NEW ORDER TABLE |'
PRINT '-----'
SELECT TOP 1
    CONVERT (CHAR (30), INFO_DATE, 21) AS 'Date',
    NUM_NEW_ORDER AS 'New
Order Rows',
    NEW_ORDER_TARGET AS
'New Order Target',
    CASE WHEN (NUM_NEW_ORDER = NEW_ORDER_TARGET)
        THEN 'OK!'
        WHEN (NUM_NEW_ORDER BETWEEN
NEW_ORDER_TARGET_LOW AND NEW_ORDER_TARGET_HIGH)
        THEN 'OK! (within 1%)'
        ELSE 'ERROR!!!'
    END AS 'New
Order Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''

```

```

PRINT '-----'
PRINT '|      HISTORY TABLE      |'
PRINT '-----'
SELECT TOP 1
  CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
  NUM_HISTORY AS 'History'
Rows',
  HISTORY_TARGET AS
  'History Target',
  CASE WHEN (NUM_HISTORY = HISTORY_TARGET)
    THEN 'OK!'
    ELSE 'ERROR!!!'
  END AS 'New'
Order Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '|      STOCK TABLE      |'
PRINT '-----'
SELECT TOP 1
  CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
  NUM_STOCK AS 'Stock'
Rows',
  STOCK_TARGET AS
  'Stock Target',
  CASE WHEN (NUM_STOCK = STOCK_TARGET)
    THEN 'OK!'
    ELSE 'ERROR!!!'
  END AS 'Stock'
Message'
FROM TPCC_INFO
GO

-----
-- Check Indexes
-----
USE tpcc
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '|      TPC-C INDEXES      |'
PRINT '-----'
EXEC sp_helpindex warehouse
EXEC sp_helpindex district
EXEC sp_helpindex item
EXEC sp_helpindex customer
EXEC sp_helpindex orders
EXEC sp_helpindex order_line
EXEC sp_helpindex new_order
EXEC sp_helpindex history
EXEC sp_helpindex stock
GO

```

version.sql

```

-----
--
--
-- File:   VERSION.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
--
-- Extracts current version of SQL Server
--
-----
USE master
GO

SELECT CONVERT(char(20),
SERVERPROPERTY('ProductVersion')),
  CONVERT(char(20),
SERVERPROPERTY('ProductLevel')),
  CONVERT(char(29), SERVERPROPERTY('Edition'))
GO

SELECT CONVERT(char(30), GETDATE(), 21)
GO

```

Appendix C: Tunable Parameters

Microsoft SQL Server 2005 Startup Parameters

```
start sqlservr.exe -c -x -T3502 -T8011 -T8012 -T8018
-T8019 -T661 -T836 -T834
```

Where:

```
-c Start SQL Server independently of the
Windows NT Service Control Manager
-x Disables the keeping of CPU time and cache-
hit ratio statistics
-T3502 Prints a message to the SQL Server log at the
start and end of each checkpoint
-T8011 Disable diagnostics for resource monitor
-T8012 Disable ring buffer for scheduler
-T8018 Disable exceptions rung buffer
-T8019 Disable stack collection for exception ring
buffer
-T661 Disable ghost writer
-T836 Force max server memory
-T834 Enable large page support
```

File locations:

```
sqlserver.exe C:\Program
Files\Microsoft SQL Server\MSSQL.1\MSSQL\BINN
ERRORLOG C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\LOG
```

Boot.ini Parameters

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows
Server 2003 Enterprise x64 Edition" /noexecute=optout
/fastdetect
```

Microsoft SQL Server 2005 Configuration Parameters

```
l> 2> name minimum
maximum config_value run_value
-----
Ad Hoc Distributed Queries 0
1 0 0
affinity I/O mask -2147483648
2147483647 0 0
affinity mask -2147483648
2147483647 15 15
affinity64 I/O mask -2147483648
2147483647 0 0
affinity64 mask -2147483648
2147483647 0 0
Agent XPs 0
1 0 0
allow updates 0
1 0 0
awe enabled 0
1 0 0
blocked process threshold 0
86400 0 0
c2 audit mode 0
1 0 0
clr enabled 0
1 0 0
cost threshold for parallelism 0
32767 5 5
cross db ownership chaining 0
1 0 0
cursor threshold -1
2147483647 -1 -1
Database Mail XPs 0
1 0 0
default full-text language 0
2147483647 1033 1033
default language 0
9999 0 0
default trace enabled 0
1 1 1
disallow results from triggers 0
1 0 0
fill factor (%) 0
100 0 0
ft crawl bandwidth (max) 0
32767 100 100
ft crawl bandwidth (min) 0
32767 0 0
ft notify bandwidth (max) 0
32767 100 100
ft notify bandwidth (min) 0
32767 0 0
in-doubt xact resolution 0
2 0 0
```

```
index create memory (KB) 704 704
2147483647 704 704
lightweight pooling 0
1 1 1
locks 5000
2147483647 0 0
max degree of parallelism 0
64 1 1
max full-text crawl range 0
256 4 4
max server memory (MB) 16
2147483647 31700 31700
max text repl size (B) 0
2147483647 65536 65536
max worker threads 128
32767 496 496
media retention 0
365 0 0
min memory per query (KB) 512
2147483647 512 512
min server memory (MB) 0
2147483647 0 16
nested triggers 0
1 1 1
network packet size (B) 512
32767 2048 2048
Ole Automation Procedures 0
1 0 0
open objects 0
2147483647 0 0
PH timeout (s) 1
3600 60 60
precompute rank 0
1 0 0
priority boost 0
1 1 1
query governor cost limit 0
2147483647 0 0
query wait (s) -1
2147483647 -1 -1
recovery interval (min) 0
32767 32767 32767
remote access 0
1 1 1
remote admin connections 0
1 0 0
remote login timeout (s) 0
2147483647 20 20
remote proc trans 0
1 0 0
remote query timeout (s) 0
2147483647 600 600
Replication XPs 0
1 0 0
scan for startup procs 0
1 0 0
server trigger recursion 0
1 1 1
set working set size 0
1 0 0
show advanced options 0
1 1 1
```

```

SMO and DMO XPs                0
1 1 1                            0
SQL Mail XPs                    0
1 0 0                            0
transform noise words          0
1 0 0                            0
two digit year cutoff          1753
9999 2049 2049
user connections                0
32767 0 0
user options                    0
32767 0 0
Web Assistant Procedures        0
1 0 0
xp_cmdshell                     0
1 0 0
1> 2> 3>

```

Microsoft SQL Server 2005 Torn Page Detection Status

```

1> 2> OptionName
CurrentSetting
-----
torn page detection      OFF
1> 2> 3>

```

Benchcraft Profile

```

Profile: DL380_11200_8c1
File Path: C:\Program
Files\BenchCraft\DL380_11200_8c1.xml
Version: 5

```

Number of Engines: 16

```

Name: RTE2
Description:
Directory: c:\blog\rte2.log
Machine: n1
Parameter Set: 1.005 better
Index: 1600000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER53164609
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1

```

Additional Options:

```

Name: RTE1
Description:
Directory: c:\blog\rte1.log
Machine: n1
Parameter Set: 1.005 better
Index: 7000000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER44265281
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

```

```

Name: RTE3
Description:
Directory: c:\blog\rte3.log
Machine: n2
Parameter Set: 1.005 better
Index: 2000000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER3439676359
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

```

```

Name: RTE4
Description:
Directory: c:\blog\rte4.log
Machine: n2
Parameter Set: 1.005 better
Index: 3000000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER4439706187
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

```

```

Name: RTE5
Description:
Directory: c:\blog\rte5.log
Machine: n3
Parameter Set: 1.005 better
Index: 4000000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER5346413218
Connect Rate: 100000

```

```

Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

```

```

Name: RTE6
Description:
Directory: c:\blog\rte6.log
Machine: n3
Parameter Set: 1.005 better
Index: 5000000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER62226046
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

```

```

Name: RTE7
Description:
Directory: c:\blog\rte7.log
Machine: n4
Parameter Set: 1.005 better
Index: 6000000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER72289718
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

```

```

Name: RTE8
Description:
Directory: c:\blog\rte8.log
Machine: n4
Parameter Set: 1.005 better
Index: 1700000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER82325578
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

```

```

Name: RTE9
Description:
Directory: c:\blog\rte9.log
Machine: n5
Parameter Set: 1.005 better

```


Index: 800000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER92360187
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: RTE10
Description:
Directory: c:\blog\rte10.log
Machine: n5
Parameter Set: 1.005 better
Index: 900000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER102399796
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

Name: RTE11
Description:
Directory: c:\blog\rte11.log
Machine: n6
Parameter Set: 1.005 better
Index: 1000000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER1122682203
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: RTE12
Description:
Directory: c:\blog\rte12.log
Machine: n6
Parameter Set: 1.005 better
Index: 1100000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER1222731546
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

Name: RTE13
Description:
Directory: c:\blog\rte13.log
Machine: n8
Parameter Set: 1.005 better
Index: 1200000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER13-1439076421
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: RTE14
Description:
Directory: c:\blog\rte14.log
Machine: n8
Parameter Set: 1.005 better
Index: 1300000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER14-1438943656
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

Name: RTE15
Description:
Directory: c:\blog\rte15.log
Machine: n9
Parameter Set: 1.005 better
Index: 1400000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER15-1438852265
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: RTE16
Description:
Directory: c:\blog\rte16.log
Machine: n9
Parameter Set: 1.005 better
Index: 1500000000
Seed: 4678
Configured Users: 7000
Pipe Name: DRIVER16-1438790906
Connect Rate: 100000
Start Rate: 100000
Max. Concurrency: 7000

Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:
Number of User groups: 16
Driver Engine: RTE1
IIS Server: cr31
SQL Server: dl380
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1 - 700
w_id Min Warehouse: 1
w_id Max Warehouse: 11200
Scale: Normal
User Count: 7000
District id: 1
Scale Down: No
Driver Engine: RTE2
IIS Server: cr31
SQL Server: dl380
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 701 - 1400
w_id Min Warehouse: 1
w_id Max Warehouse: 11200
Scale: Normal
User Count: 7000
District id: 1
Scale Down: No
Driver Engine: RTE3
IIS Server: cr32
SQL Server: dl380
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1401 - 2100
w_id Min Warehouse: 1
w_id Max Warehouse: 11200
Scale: Normal
User Count: 7000
District id: 1
Scale Down: No
Driver Engine: RTE4
IIS Server: cr32
SQL Server: dl380
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2101 - 2800
w_id Min Warehouse: 1
w_id Max Warehouse: 11200
Scale: Normal
User Count: 7000
District id: 1
Scale Down: No

Driver Engine: RTE5
 IIS Server: cr33
 SQL Server: dl380
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 2801 - 3500
 w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Driver Engine: RTE6
 IIS Server: cr33
 SQL Server: dl380
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 3501 - 4200
 w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Driver Engine: RTE7
 IIS Server: cr34
 SQL Server: dl380
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 4201 - 4900
 w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Driver Engine: RTE8
 IIS Server: cr34
 SQL Server: dl380
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 4901 - 5600
 w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Driver Engine: RTE9
 IIS Server: cr35
 SQL Server: dl380b
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 5601 - 6300

w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Driver Engine: RTE10
 IIS Server: cr35
 SQL Server: dl380b
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 6301 - 7000
 w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Driver Engine: RTE11
 IIS Server: cr36
 SQL Server: dl380b
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 7001 - 7700
 w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Driver Engine: RTE12
 IIS Server: cr36
 SQL Server: dl380b
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 7701 - 8400
 w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Driver Engine: RTE13
 IIS Server: cr37
 SQL Server: dl380b
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 8401 - 9100
 w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Driver Engine: RTE14
 IIS Server: cr37
 SQL Server: dl380b
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 9101 - 9800
 w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Driver Engine: RTE15
 IIS Server: cr38
 SQL Server: dl380b
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 9801 - 10500
 w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Driver Engine: RTE16
 IIS Server: cr38
 SQL Server: dl380b
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 10501 - 11200
 w_id Min Warehouse: 1
 w_id Max Warehouse: 11200
 Scale: Normal
 User Count: 7000
 District id: 1
 Scale Down: No

Number of Parameter Sets: 65

Key	~Default		Default Parameter Set		Think
	RT	RT	Txn	Weight	
Time	Delay	Fence	Delay	Weight	Time
12.05	18.01		New Order	10.00	
			0.10	5.00	0.10
12.05	3.01		Payment	10.00	
			0.10	5.00	0.10
5.05	2.01		Delivery	1.00	
			0.10	5.00	0.10
5.05	2.01		Stock Level	1.00	
			0.10	20.00	0.10
10.05	2.01		Order Status	1.00	
			0.10	5.00	0.10
Tuned Distribution					

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
12.05	18.01		0.10	5.00	0.10
			Payment	43.10	
12.05	3.01		0.10	5.00	0.10
			Delivery	4.05	
5.05	2.01		0.10	5.00	0.10
			Stock Level	4.05	
5.05	2.01		0.10	20.00	0.10
			Order Status	4.05	
10.05	2.01		0.10	5.00	0.10

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	10.00	
0.00	0.00		0.00	5.00	0.00
			Payment	10.00	
0.00	0.00		0.00	5.00	0.00
			Delivery	1.00	
0.00	0.00		0.00	5.00	0.00
			Stock Level	1.00	
0.00	0.00		0.00	20.00	0.00
			Order Status	1.00	
0.00	0.00		0.00	5.00	0.00

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
13.00	18.01		0.10	5.00	0.10
			Payment	43.10	
13.00	3.01		0.10	5.00	0.10
			Delivery	4.05	
6.00	2.01		0.10	5.00	0.10
			Stock Level	4.05	
6.00	2.01		0.10	20.00	0.10
			Order Status	4.05	
11.00	2.01		0.10	5.00	0.10

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.83	
16.00	18.01		0.10	5.00	0.10
			Payment	43.05	
16.00	3.01		0.10	5.00	0.10
			Delivery	4.04	
9.00	2.01		0.10	5.00	0.10

9.00	2.01		Stock Level	4.04	
			0.10	20.00	0.10
			Order Status	4.04	
14.00	2.01		0.10	5.00	0.10
				3.0	

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
36.15	0.00		0.10	5.00	0.10
			Payment	43.10	
36.15	0.00		0.10	5.00	0.10
			Delivery	4.05	
15.15	0.00		0.10	5.00	0.10
			Stock Level	4.05	
15.15	0.00		0.10	20.00	0.10
			Order Status	4.05	
30.15	0.00		0.10	5.00	0.10

4.0
4.0 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
48.20	18.01		0.10	5.00	0.10
			Payment	43.10	
48.20	3.01		0.10	5.00	0.10
			Delivery	4.05	
20.20	2.01		0.10	5.00	0.10
			Stock Level	4.05	
20.20	2.01		0.10	20.00	0.10
			Order Status	4.05	
40.20	2.01		0.10	5.00	0.10

3.8
3.8 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
45.70	18.01		0.10	5.00	0.10
			Payment	43.10	
45.70	3.01		0.10	5.00	0.10
			Delivery	4.05	
19.10	2.01		0.10	5.00	0.10
			Stock Level	4.05	
19.10	2.01		0.10	20.00	0.10
			Order Status	4.05	
38.10	2.01		0.10	5.00	0.10

3.6
3.6 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
43.30	18.01		0.10	5.00	0.10
			Payment	43.10	
43.30	3.01		0.10	5.00	0.10
			Delivery	4.05	
18.10	2.01		0.10	5.00	0.10
			Stock Level	4.05	
18.10	2.01		0.10	20.00	0.10
			Order Status	4.05	
36.18	2.01		0.10	5.00	0.10

43.30	18.01		New Order	44.75	
			0.10	5.00	0.10
			Payment	43.10	
43.30	3.01		0.10	5.00	0.10
			Delivery	4.05	
18.10	2.01		0.10	5.00	0.10
			Stock Level	4.05	
18.10	2.01		0.10	20.00	0.10
			Order Status	4.05	
36.18	2.01		0.10	5.00	0.10

3.4
3.4 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
40.90	18.01		0.10	5.00	0.10
			Payment	43.10	
40.90	3.01		0.10	5.00	0.10
			Delivery	4.05	
17.10	2.01		0.10	5.00	0.10
			Stock Level	4.05	
17.10	2.01		0.10	20.00	0.10
			Order Status	4.05	
17.10	2.01		0.10	5.00	0.10

3.2
3.2 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
38.50	18.01		0.10	5.00	0.10
			Payment	43.10	
38.50	3.01		0.10	5.00	0.10
			Delivery	4.05	
16.10	2.01		0.10	5.00	0.10
			Stock Level	4.05	
16.10	2.01		0.10	20.00	0.10
			Order Status	4.05	
32.10	2.01		0.10	5.00	0.10

2.8
2.8 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
33.74	18.01		0.10	5.00	0.10
			Payment	43.10	
33.74	3.01		0.10	5.00	0.10
			Delivery	4.05	
14.14	2.01		0.10	5.00	0.10
			Stock Level	4.05	
14.14	2.01		0.10	20.00	0.10
			Order Status	4.05	
28.14	2.01		0.10	5.00	0.10

2.6

2.6 tt					
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
31.30	18.01		0.10	5.00	0.10
			Payment	43.10	
31.30	3.01		0.10	5.00	0.10
			Delivery	4.05	
13.10	2.01		0.10	5.00	0.10
			Stock Level	4.05	
13.10	2.01		0.10	20.00	0.10
			Order Status	4.05	
26.10	2.01		0.10	5.00	0.10
2.4					
2.4 tt					
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
28.90	18.01		0.10	5.00	0.10
			Payment	43.10	
28.90	3.01		0.10	5.00	0.10
			Delivery	4.05	
12.10	2.01		0.10	5.00	0.10
			Stock Level	4.05	
12.10	2.01		0.10	20.00	0.10
			Order Status	4.05	
24.10	2.01		0.10	5.00	0.10
2.2					
2.2 tt					
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
28.90	18.01		0.10	5.00	0.10
			Payment	43.10	
28.90	3.01		0.10	5.00	0.10
			Delivery	4.05	
12.10	2.01		0.10	5.00	0.10
			Stock Level	4.05	
12.10	2.01		0.10	20.00	0.10
			Order Status	4.05	
24.12	2.01		0.10	5.00	0.10
2.0					
2.0 tt					
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
24.10	18.01		0.10	5.00	0.10
			Payment	43.10	
24.10	3.01		0.10	5.00	0.10
			Delivery	4.05	
10.10	2.01		0.10	5.00	0.10

10.10	2.01		Stock Level	4.05	
			0.10	20.00	0.10
			Order Status	4.05	
20.10	2.01		0.10	5.00	0.10
			5.0		
			5.0 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
60.25	18.01		0.10	5.00	0.10
			Payment	43.10	
60.25	3.01		0.10	5.00	0.10
			Delivery	4.05	
25.25	2.01		0.10	5.00	0.10
			Stock Level	4.05	
25.25	2.01		0.10	20.00	0.10
			Order Status	4.05	
50.25	2.01		0.10	5.00	0.10
			4.5		
			4.5 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
54.20	18.01		0.10	5.00	0.10
			Payment	43.10	
54.20	3.01		0.10	5.00	0.10
			Delivery	4.05	
22.70	2.01		0.10	5.00	0.10
			Stock Level	4.05	
22.70	2.01		0.10	20.00	0.10
			Order Status	4.05	
45.20	2.01		0.10	5.00	0.10
			3.5		
			3.5 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
42.10	18.01		0.10	5.00	0.10
			Payment	43.10	
42.10	3.01		0.10	5.00	0.10
			Delivery	4.05	
17.60	2.01		0.10	5.00	0.10
			Stock Level	4.05	
17.60	2.01		0.10	20.00	0.10
			Order Status	4.05	
35.10	2.01		0.10	5.00	0.10
			1.8		
			1.8 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time

21.60	18.01		New Order	44.75	
			0.10	5.00	0.10
			Payment	43.10	
21.60	3.01		0.10	5.00	0.10
			Delivery	4.05	
9.09	2.01		0.10	5.00	0.10
			Stock Level	4.05	
9.09	2.01		0.10	20.00	0.10
			Order Status	4.05	
18.09	2.01		0.10	5.00	0.10
			4.2		
			4.2 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
54.20	18.01		0.10	5.00	0.10
			Payment	43.10	
54.20	3.01		0.10	5.00	0.10
			Delivery	4.05	
22.70	2.01		0.10	5.00	0.10
			Stock Level	4.05	
22.70	2.01		0.10	20.00	0.10
			Order Status	4.05	
45.20	2.01		0.10	5.00	0.10
			1.6		
			1.6 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
19.20	18.01		0.10	5.00	0.10
			Payment	43.10	
19.20	3.01		0.10	5.00	0.10
			Delivery	4.05	
8.08	2.01		0.10	5.00	0.10
			Stock Level	4.05	
8.08	2.01		0.10	20.00	0.10
			Order Status	4.05	
16.08	2.01		0.10	5.00	0.10
			1.4		
			1.4 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.75	
16.87	18.01		0.10	5.00	0.10
			Payment	43.10	
16.87	3.01		0.10	5.00	0.10
			Delivery	4.05	
7.07	2.01		0.10	5.00	0.10
			Stock Level	4.05	
7.07	2.01		0.10	20.00	0.10
			Order Status	4.05	
14.07	2.01		0.10	5.00	0.10
			1.2		

1.2 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.83		
14.46	18.01		0.10	5.00	0.10	
			Payment	43.05		
14.46	3.01		0.10	5.00	0.10	
			Delivery	4.04		
6.06	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
6.06	2.01		0.10	20.00	0.10	
			Order Status	4.04		
12.06	2.01		0.10	5.00	0.10	
3.5						
3.5 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.75		
42.10	18.01		0.10	5.00	0.10	
			Payment	43.10		
42.10	3.01		0.10	5.00	0.10	
			Delivery	4.05		
17.60	2.01		0.10	5.00	0.10	
			Stock Level	4.05		
17.60	2.01		0.10	20.00	0.10	
			Order Status	4.05		
35.10	2.01		0.10	5.00	0.10	
1.9						
1.9 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.75		
22.89	18.01		0.10	5.00	0.10	
			Payment	43.10		
22.89	3.01		0.10	5.00	0.10	
			Delivery	4.05		
9.59	2.01		0.10	5.00	0.10	
			Stock Level	4.05		
9.59	2.01		0.10	20.00	0.10	
			Order Status	4.05		
19.09	2.01		0.10	5.00	0.10	
1.1						
1.1 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.83		
13.25	18.01		0.10	5.00	0.10	
			Payment	43.05		
13.25	3.01		0.10	5.00	0.10	
			Delivery	4.04		
5.55	2.01		0.10	5.00	0.10	

			Stock Level	4.04		
5.55	2.01		0.10	20.00	0.10	
			Order Status	4.04		
11.05	2.01		0.10	5.00	0.10	
1.05 better						
1.05 tt better						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.92		
12.65	18.01		0.10	5.00	0.10	
			Payment	43.01		
12.65	3.01		0.10	5.00	0.10	
			Delivery	4.02		
5.30	2.01		0.10	5.00	0.10	
			Stock Level	4.03		
5.30	2.01		0.10	20.00	0.10	
			Order Status	4.02		
10.55	2.01		0.10	5.00	0.10	
1.09						
1.09 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.83		
13.13	18.01		0.10	5.00	0.10	
			Payment	43.05		
13.13	3.01		0.10	5.00	0.10	
			Delivery	4.04		
5.50	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
5.50	2.01		0.10	20.00	0.10	
			Order Status	4.04		
10.95	2.01		0.10	5.00	0.10	
1.08						
1.08 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.83		
13.01	18.01		0.10	5.00	0.10	
			Payment	43.05		
13.01	3.01		0.10	5.00	0.10	
			Delivery	4.04		
5.45	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
5.45	2.01		0.10	20.00	0.10	
			Order Status	4.04		
10.85	2.01		0.10	5.00	0.10	
1.07						
1.07 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	

			New Order	44.83		
12.89	18.01		0.10	5.00	0.10	
			Payment	43.05		
12.89	3.01		0.10	5.00	0.10	
			Delivery	4.04		
5.40	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
5.40	2.01		0.10	20.00	0.10	
			Order Status	4.04		
10.75	2.01		0.10	5.00	0.10	
1.06						
1.06 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.83		
12.77	18.01		0.10	5.00	0.10	
			Payment	43.05		
12.77	3.01		0.10	5.00	0.10	
			Delivery	4.04		
5.35	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
5.35	2.01		0.10	20.00	0.10	
			Order Status	4.04		
10.65	2.01		0.10	5.00	0.10	
1.15						
1.15 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.75		
13.85	18.01		0.10	5.00	0.10	
			Payment	43.10		
13.85	3.01		0.10	5.00	0.10	
			Delivery	4.05		
5.80	2.01		0.10	5.00	0.10	
			Stock Level	4.05		
5.80	2.01		0.10	20.00	0.10	
			Order Status	4.05		
11.55	2.01		0.10	5.00	0.10	
1.25						
1.25 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.83		
15.06	18.01		0.10	5.00	0.10	
			Payment	43.05		
15.06	3.01		0.10	5.00	0.10	
			Delivery	4.04		
6.31	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
6.31	2.01		0.10	20.00	0.10	
			Order Status	4.04		
12.56	2.01		0.10	5.00	0.10	
1.3						

```

1.3 tt
Key RT RT Menu Txn Think
Time Delay Fence Delay Weight Time
15.66 18.01 New Order 0.10 5.00 44.83 0.10
15.66 3.01 Payment 0.10 5.00 43.05 0.10
6.56 2.01 Delivery 0.10 5.00 4.04 0.10
6.56 2.01 Stock Level 0.10 20.00 4.04 0.10
13.06 2.01 Order Status 0.10 5.00 4.04 0.10

1.12
1.12 tt
Key RT RT Menu Txn Think
Time Delay Fence Delay Weight Time
13.49 18.01 New Order 0.10 5.00 44.75 0.10
13.49 3.01 Payment 0.10 5.00 43.10 0.10
5.65 2.01 Delivery 0.10 5.00 4.05 0.10
5.65 2.01 Stock Level 0.10 20.00 4.05 0.10
11.25 2.01 Order Status 0.10 5.00 4.05 0.10

1.18
1.18 tt
Key RT RT Menu Txn Think
Time Delay Fence Delay Weight Time
14.21 18.01 New Order 0.10 5.00 44.75 0.10
14.21 3.01 Payment 0.10 5.00 43.10 0.10
5.95 2.01 Delivery 0.10 5.00 4.05 0.10
5.95 2.01 Stock Level 0.10 20.00 4.05 0.10
11.85 2.01 Order Status 0.10 5.00 4.05 0.10

1.22
1.22 tt
Key RT RT Menu Txn Think
Time Delay Fence Delay Weight Time
14.70 18.01 New Order 0.10 5.00 44.75 0.10
14.70 3.01 Payment 0.10 5.00 43.10 0.10
6.16 2.01 Delivery 0.10 5.00 4.05 0.10

```

```

6.16 2.01 Stock Level 0.10 20.00 4.05 0.10
12.26 2.01 Order Status 0.10 5.00 4.05 0.10

1.28
1.28 tt
Key RT RT Menu Txn Think
Time Delay Fence Delay Weight Time
15.42 18.01 New Order 0.10 5.00 44.75 0.10
15.42 3.01 Payment 0.10 5.00 43.10 0.10
6.46 2.01 Delivery 0.10 5.00 4.05 0.10
6.46 2.01 Stock Level 0.10 20.00 4.05 0.10
12.86 2.01 Order Status 0.10 5.00 4.05 0.10

1.04
1.04 tt
Key RT RT Menu Txn Think
Time Delay Fence Delay Weight Time
12.53 18.01 New Order 0.10 5.00 44.83 0.10
12.53 3.01 Payment 0.10 5.00 43.05 0.10
5.25 2.01 Delivery 0.10 5.00 4.04 0.10
5.25 2.01 Stock Level 0.10 20.00 4.04 0.10
10.45 2.01 Order Status 0.10 5.00 4.04 0.10

1.03
1.03 tt
Key RT RT Menu Txn Think
Time Delay Fence Delay Weight Time
12.41 18.01 New Order 0.10 5.00 44.83 0.10
12.41 3.01 Payment 0.10 5.00 43.05 0.10
5.20 2.01 Delivery 0.10 5.00 4.04 0.10
5.20 2.01 Stock Level 0.10 20.00 4.04 0.10
10.35 2.01 Order Status 0.10 5.00 4.04 0.10

1.02
1.02 tt
Key RT RT Menu Txn Think
Time Delay Fence Delay Weight Time

```

```

12.29 18.01 New Order 0.10 5.00 44.83 0.10
12.29 3.01 Payment 0.10 5.00 43.05 0.10
5.15 2.01 Delivery 0.10 5.00 4.04 0.10
5.15 2.01 Stock Level 0.10 20.00 4.04 0.10
10.25 2.01 Order Status 0.10 5.00 4.04 0.10

1.01
1.01 tt
Key RT RT Menu Txn Think
Time Delay Fence Delay Weight Time
12.17 18.01 New Order 0.10 5.00 44.83 0.10
12.17 3.01 Payment 0.10 5.00 43.05 0.10
5.10 2.01 Delivery 0.10 5.00 4.04 0.10
5.10 2.01 Stock Level 0.10 20.00 4.04 0.10
10.15 2.01 Order Status 0.10 5.00 4.04 0.10

1.001_best
1.001 tt best
Key RT RT Menu Txn Think
Time Delay Fence Delay Weight Time
12.06 18.01 New Order 0.10 5.00 44.91 0.10
12.06 3.01 Payment 0.10 5.00 43.04 0.10
5.06 2.01 Delivery 0.10 5.00 4.01 0.10
5.06 2.01 Stock Level 0.10 20.00 4.02 0.10
10.06 2.01 Order Status 0.10 5.00 4.02 0.10

1.03_better
1.03 tt more aggressive
Key RT RT Menu Txn Think
Time Delay Fence Delay Weight Time
12.41 18.01 New Order 0.10 5.00 44.92 0.10
12.41 3.01 Payment 0.10 5.00 43.01 0.10
5.20 2.01 Delivery 0.10 5.00 4.02 0.10
5.20 2.01 Stock Level 0.10 20.00 4.03 0.10
10.35 2.01 Order Status 0.10 5.00 4.02 0.10

1.005_better

```

1.005 tt more aggressive

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.11	18.01		New Order	44.92	
			0.10	5.00	0.10
12.11	3.01		Payment	43.02	
			0.10	5.00	0.10
5.07	2.01		Delivery	4.02	
			0.10	5.00	0.10
5.07	2.01		Stock Level	4.02	
			0.10	20.00	0.10
10.10	2.01		Order Status	4.02	
			0.10	5.00	0.10

1.02 better
1.02 tt more aggressive

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.29	18.01		New Order	44.92	
			0.10	5.00	0.10
12.29	3.01		Payment	43.01	
			0.10	5.00	0.10
5.15	2.01		Delivery	4.02	
			0.10	5.00	0.10
5.15	2.01		Stock Level	4.03	
			0.10	20.00	0.10
10.25	2.01		Order Status	4.02	
			0.10	5.00	0.10

1.01 best
1.01 tt best

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.17	18.01		New Order	44.90	
			0.10	5.00	0.10
12.17	3.01		Payment	43.05	
			0.10	5.00	0.10
5.10	2.01		Delivery	4.01	
			0.10	5.00	0.10
5.10	2.01		Stock Level	4.03	
			0.10	20.00	0.10
10.15	2.01		Order Status	4.01	
			0.10	5.00	0.10

1.02 best
1.02 tt best

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.29	18.01		New Order	44.96	
			0.00	5.00	0.00
12.29	3.01		Payment	43.00	
			0.00	5.00	0.00
5.15	2.01		Delivery	4.00	
			0.00	5.00	0.00

5.15 2.01 Stock Level 4.03
0.00 20.00 0.00

10.25 2.01 Order Status 4.01
0.00 5.00 0.00

1.03 best
1.03 tt best

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.41	18.01		New Order	44.96	
			0.10	5.00	0.10
12.41	3.01		Payment	43.01	
			0.10	5.00	0.10
5.20	2.01		Delivery	4.01	
			0.10	5.00	0.10
5.20	2.01		Stock Level	4.01	
			0.10	20.00	0.10
10.35	2.01		Order Status	4.01	
			0.10	5.00	0.10

5.5
5.5 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
66.28	18.01		New Order	44.83	
			0.10	5.00	0.10
66.28	3.01		Payment	43.05	
			0.10	5.00	0.10
27.77	2.01		Delivery	4.04	
			0.10	5.00	0.10
27.77	2.01		Stock Level	4.04	
			0.10	20.00	0.10
55.27	2.01		Order Status	4.04	
			0.10	5.00	0.10

6.0
6.0 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
72.30	18.01		New Order	44.83	
			0.10	5.00	0.10
72.30	3.01		Payment	43.05	
			0.10	5.00	0.10
30.30	2.01		Delivery	4.04	
			0.10	5.00	0.10
30.30	2.01		Stock Level	4.04	
			0.10	20.00	0.10
60.30	2.01		Order Status	4.04	
			0.10	5.00	0.10

6.5
6.5 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time

79.53 18.01 New Order 44.83
0.10 5.00 0.10

79.53 3.01 Payment 43.05
0.10 5.00 0.10

33.33 2.01 Delivery 4.04
0.10 5.00 0.10

33.33 2.01 Stock Level 4.04
0.10 20.00 0.10

66.33 2.01 Order Status 4.04
0.10 5.00 0.10

7.0
7.0 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
84.35	18.01		New Order	44.83	
			0.10	5.00	0.10
84.35	3.01		Payment	43.05	
			0.10	5.00	0.10
35.35	2.01		Delivery	4.04	
			0.10	5.00	0.10
35.35	2.01		Stock Level	4.04	
			0.10	20.00	0.10
70.35	2.01		Order Status	4.04	
			0.10	5.00	0.10

7.5
7.5 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
90.38	18.01		New Order	44.83	
			0.10	5.00	0.10
90.38	3.01		Payment	43.05	
			0.10	5.00	0.10
37.88	2.01		Delivery	4.04	
			0.10	5.00	0.10
37.88	2.01		Stock Level	4.04	
			0.10	20.00	0.10
75.38	2.01		Order Status	4.04	
			0.10	5.00	0.10

8.0
8.0 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
96.40	18.01		New Order	44.83	
			0.10	5.00	0.10
96.40	3.01		Payment	43.05	
			0.10	5.00	0.10
40.40	2.01		Delivery	4.04	
			0.10	5.00	0.10
40.40	2.01		Stock Level	4.04	
			0.10	20.00	0.10
80.40	2.01		Order Status	4.04	
			0.10	5.00	0.10

8.5

8.5 tt					
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.83	
102.43	18.01		0.10	5.00	0.10
			Payment	43.05	
192.43	3.01		0.10	5.00	0.10
			Delivery	4.04	
42.92	2.01		0.10	5.00	0.10
			Stock Level	4.04	
42.92	2.01		0.10	20.00	0.10
			Order Status	4.04	
85.42	2.01		0.10	5.00	0.10
			9.0		
			9.0 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.83	
108.45	18.01		0.10	5.00	0.10
			Payment	43.05	
108.45	3.01		0.10	5.00	0.10
			Delivery	4.04	
45.45	2.01		0.10	5.00	0.10
			Stock Level	4.04	
45.45	2.01		0.10	20.00	0.10
			Order Status	4.04	
90.45	2.01		0.10	5.00	0.10
			9.5		
			9.5 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.83	
114.47	18.01		0.10	5.00	0.10
			Payment	43.05	
114.47	3.01		0.10	5.00	0.10
			Delivery	4.04	
47.98	2.01		0.10	5.00	0.10
			Stock Level	4.04	
47.98	2.01		0.10	20.00	0.10
			Order Status	4.04	
95.47	2.01		0.10	5.00	0.10
			10		
			10 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.83	
120.50	18.01		0.10	5.00	0.10
			Payment	43.05	
120.50	3.01		0.10	5.00	0.10
			Delivery	4.04	
50.50	2.01		0.10	5.00	0.10

			Stock Level	4.04	
50.50	2.01		0.10	20.00	0.10
			Order Status	4.04	
100.50	2.01		0.10	5.00	0.10
			1.02 better		
			1.02 more aggressive		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.92	
12.05	18.01		0.10	5.00	0.10
			Payment	43.01	
12.05	3.01		0.10	5.00	0.10
			Delivery	4.02	
5.05	2.01		0.10	5.00	0.10
			Stock Level	4.03	
5.05	2.01		0.10	20.00	0.10
			Order Status	4.02	
10.05	2.01		0.10	5.00	0.10
			1.01 better		
			1.01 more aggressive		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.92	
12.17	18.01		0.10	5.00	0.10
			Payment	43.01	
12.17	3.01		0.10	5.00	0.10
			Delivery	4.02	
5.10	2.01		0.10	5.00	0.10
			Stock Level	4.03	
5.10	2.01		0.10	20.00	0.10
			Order Status	4.02	
10.15	2.01		0.10	5.00	0.10
			1.001 better		
			1.001 more aggressive		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.92	
12.06	18.01		0.10	5.00	0.10
			Payment	43.01	
12.06	3.01		0.10	5.00	0.10
			Delivery	4.02	
5.06	2.01		0.10	5.00	0.10
			Stock Level	4.03	
5.06	2.01		0.10	20.00	0.10
			Order Status	4.02	
10.06	2.01		0.10	5.00	0.10
			FullSpeed		
			1.000 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time

			New Order	44.91	
12.05	18.01		0.10	5.00	0.10
			Payment	43.01	
12.05	3.01		0.10	5.00	0.10
			Delivery	4.02	
5.05	2.01		0.10	5.00	0.10
			Stock Level	4.03	
5.05	2.01		0.10	20.00	0.10
			Order Status	4.03	
10.05	2.01		0.10	5.00	0.10
			1.003 best		
			1.003 best		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.90	
12.09	18.01		0.10	5.00	0.10
			Payment	43.05	
12.09	3.01		0.10	5.00	0.10
			Delivery	4.01	
5.07	2.01		0.10	5.00	0.10
			Stock Level	4.03	
5.07	2.01		0.10	20.00	0.10
			Order Status	4.01	
10.08	2.01		0.10	5.00	0.10

Internet Information Server Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,00,00
"PoolThreadLimit"=dword:000007fe
"ThreadTimeout"=dword:00015180
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infcstrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000842
"Last Help"=dword:00000843
"First Counter"=dword:00000802
"First Help"=dword:00000803
```


Value 2
Name: ErrorControl
Type: REG_DWORD
Data: 0x1

Value 3
Name: Tag
Type: REG_DWORD
Data: 0x102

Value 4
Name: ImagePath
Type: REG_EXPAND_SZ
Data: system32\DRIVERS\hpcqissb.sys

Value 5
Name: DisplayName
Type: REG_SZ
Data: Smart Array Controllers Non-Miniport Bus Driver

Value 6
Name: Group
Type: REG_SZ
Data: port

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpcqissb\Parameters
Class Name: <NO CLASS>
Last Write Time: 6/14/2006 - 11:10 AM
Value 0
Name: CompletionMode
Type: REG_DWORD
Data: 0x1

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpcqissb\Security
Class Name: <NO CLASS>
Last Write Time: 6/14/2006 - 11:10 AM

Value 0
Name: Security
Type: REG_BINARY
Data:
00000000 01 00 14 80 b8 00 00 00 - c4 00 00 00 14
00 00 00Ä.....
00000010 30 00 00 00 02 00 1c 00 - 01 00 00 00 02
80 14 00 0.....
00000020 ff 01 0f 00 01 01 00 00 - 00 00 00 01 00
00 00 00 ý.....
00000030 02 00 88 00 06 00 00 00 - 00 00 14 00 fd
01 02 00ý....
00000040 01 01 00 00 00 00 05 - 12 00 00 00 00
00 18 00
00000050 ff 01 0f 00 01 02 00 00 - 00 00 00 05 20
00 00 00 ý.....
00000060 20 02 00 00 00 14 00 - 8d 01 02 00 01
01 00 00

00000070 00 00 00 05 04 00 00 00 - 00 00 14 00 8d
01 02 00
00000080 01 01 00 00 00 00 05 - 06 00 00 00 00
00 14 00
00000090 00 01 00 00 01 01 00 00 - 00 00 00 05 0b
00 00 00
000000a0 00 00 18 00 fd 01 02 00 - 01 02 00 00 00
00 00 05ý.....
000000b0 20 00 00 00 23 02 00 00 - 01 01 00 00 00
00 00 05 ...#.....
000000c0 12 00 00 00 01 01 00 00 - 00 00 00 05 12
00 00 00

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpcqissb\Enum
Class Name: <NO CLASS>
Last Write Time: 6/20/2006 - 12:32 PM
Value 0
Name: 0
Type: REG_SZ
Data: Root\LEGACY_HPQCISB\0000

Value 1
Name: Count
Type: REG_DWORD
Data: 0x1

Value 2
Name: NextInstance
Type: REG_DWORD
Data: 0x1

QL2300 Registry Parameters

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300
Class Name: <NO CLASS>
Last Write Time: 6/20/2006 - 12:31 PM

Value 0
Name: ErrorControl
Type: REG_DWORD
Data: 0x1

Value 1
Name: Group
Type: REG_SZ
Data: SCSI miniport

Value 2
Name: Start
Type: REG_DWORD
Data: 0

Value 3
Name: Tag
Type: REG_DWORD
Data: 0x28

Value 4
Name: Type
Type: REG_DWORD
Data: 0x1

Value 5
Name: DisplayName
Type: REG_SZ
Data: QLogic Fibre Channel SCSI Miniport Driver (wx64 IP)

Value 6
Name: ImagePath
Type: REG_EXPAND_SZ
Data: system32\DRIVERS\ql2300.sys

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Parameters
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 3:34 PM

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Parameters\Device
Class Name: <NO CLASS>
Last Write Time: 6/12/2006 - 11:39 AM
Value 0
Name: DriverParameter
Type: REG_SZ
Data:

Value 1
Name: BusType
Type: REG_DWORD
Data: 0x6

Value 2
Name: CreateInitiatorLU
Type: REG_DWORD
Data: 0x1

Value 3
Name: MaximumSGList
Type: REG_DWORD
Data: 0xff

Value 4
Name: NumberOfRequests
Type: REG_DWORD
Data: 0x96

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Parameters\PnpInterface
Class Name: <NO CLASS>

Last Write Time: 5/31/2006 - 9:17 AM
 Value 0
 Name: 5
 Type: REG_DWORD
 Data: 0x1

Key Name:
 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Enum
 Class Name: <NO CLASS>
 Last Write Time: 6/20/2006 - 12:31 PM
 Value 0
 Name: 0
 Type: REG_SZ
 Data:
 PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_02\6&1a6a5b31&0&00000010

Value 1
 Name: Count
 Type: REG_DWORD
 Data: 0x6

Value 2
 Name: NextInstance
 Type: REG_DWORD
 Data: 0x6

Value 3
 Name: 1
 Type: REG_SZ
 Data:

PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_02\6&1a6a5b31&0&01000010

Value 4
 Name: 2
 Type: REG_SZ
 Data:

PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_02\4&1636f220&0&0030

Value 5
 Name: 3
 Type: REG_SZ
 Data:

PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_02\4&1636f220&0&0130

Value 6
 Name: 4
 Type: REG_SZ
 Data:

PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_02\4&2a0e5541&0&0020

Value 7
 Name: 5
 Type: REG_SZ
 Data:

PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_02\4&2a0e5541&0&0120

QLDirect Registry Parameters

Key Name:
 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect
 Class Name: <NO CLASS>
 Last Write Time: 6/20/2006 - 12:31 PM
 Value 0
 Name: ErrorControl
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: Type
 Type: REG_DWORD
 Data: 0x1

Value 2
 Name: Start
 Type: REG_DWORD
 Data: 0x2

Value 3
 Name: ImagePath
 Type: REG_EXPAND_SZ
 Data: system32\DRIVERS\qldirect.sys

Key Name:
 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect\Parameters
 Class Name: <NO CLASS>
 Last Write Time: 6/12/2006 - 12:28 PM
 Value 0
 Name: MaxPathsPerDevice
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: SrbListSize
 Type: REG_DWORD
 Data: 0x400

Value 2
 Name: PerCpuData
 Type: REG_DWORD
 Data: 0x1

Key Name:
 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect\Security
 Class Name: <NO CLASS>
 Last Write Time: 6/12/2006 - 12:28 PM
 Value 0
 Name: Security
 Type: REG_BINARY
 Data:

```

00000000 01 00 14 80 b8 00 00 00 - c4 00 00 00 14
00 00 00 .....Ä.....
00000010 30 00 00 00 02 00 1c 00 - 01 00 00 00 02
80 14 00 0.....
00000020 ff 01 0f 00 01 01 00 00 - 00 00 00 01 00
00 00 00 Ÿ.....
00000030 02 00 88 00 06 00 00 00 - 00 00 14 00 fd
01 02 00 .....Ÿ...
00000040 01 01 00 00 00 00 00 05 - 12 00 00 00 00
00 18 00 .....
00000050 ff 01 0f 00 01 02 00 00 - 00 00 00 05 20
00 00 00 Ÿ.....
00000060 20 02 00 00 00 00 14 00 - 8d 01 02 00 01
01 00 00 .....
00000070 00 00 00 05 04 00 00 00 - 00 00 14 00 8d
01 02 00 .....
00000080 01 01 00 00 00 00 05 - 06 00 00 00 00
00 14 00 .....
00000090 00 01 00 00 01 01 00 00 - 00 00 00 05 0b
00 00 00 .....
000000a0 00 00 18 00 fd 01 02 00 - 01 02 00 00 00
00 00 05 ...Ÿ.....
000000b0 20 00 00 00 23 02 00 00 - 01 01 00 00 00
00 00 05 ...#.....
000000c0 12 00 00 00 01 01 00 00 - 00 00 00 05 12
00 00 00 .....
  
```

Key Name:
 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect\Enum
 Class Name: <NO CLASS>
 Last Write Time: 6/20/2006 - 12:31 PM
 Value 0
 Name: 0
 Type: REG_SZ
 Data: Root\LEGACY_QLDIRECT\0000

Value 1
 Name: Count
 Type: REG_DWORD
 Data: 0x2

Value 2
 Name: NextInstance
 Type: REG_DWORD
 Data: 0x2

Value 3
 Name: 1
 Type: REG_SZ
 Data: Root\SCSIADAPTER\0000

HP NC373i Driver Parameters (Port 1)

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0043
Class Name: <NO CLASS>
Last Write Time: 6/15/2006 - 11:33 AM

Value 0
Name: create_pdo_flag
Type: REG_SZ
Data: 4

Value 1
Name: mtu
Type: REG_SZ
Data: 1500

Value 2
Name: req_medium
Type: REG_SZ
Data: 0

Value 3
Name: InfPath
Type: REG_SZ
Data: oem4.inf

Value 4
Name: InfSection
Type: REG_SZ
Data: NC373i_inst_amd64

Value 5
Name: ProviderName
Type: REG_SZ
Data: Hewlett-Packard Company

Value 6
Name: DriverDateData
Type: REG_BINARY
Data: 00 c0 59 88 d6 64 c6 01 -
.ÅY.ÖdE.

Value 7
Name: DriverDate
Type: REG_SZ
Data: 4-21-2006

Value 8
Name: DriverVersion
Type: REG_SZ
Data: 2.6.17.0

Value 9
Name: MatchingDeviceId
Type: REG_SZ
Data: pci\ven_14e4&dev_164c&subsys_7038103c

Value 10
Name: DriverDesc
Type: REG_SZ
Data: HP NC373i Virtual Bus Device

Value 11
Name: target_ips
Type: REG_SZ
Data: 2000

Value 12
Name: optimize_ips
Type: REG_SZ
Data: 0

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0043\ndi
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 3:31 PM

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0043\ndi\params
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 3:31 PM

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0043\ndi\params\mtu
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 3:31 PM

Value 0
Name: paramdesc
Type: REG_SZ
Data: Maximum Transfer Unit

Value 1
Name: default
Type: REG_SZ
Data: 1500

Value 2
Name: type
Type: REG_SZ
Data: dword

Value 3
Name: min
Type: REG_SZ
Data: 1500

Value 4
Name: max
Type: REG_SZ
Data: 9000

Value 5
Name: step

Type: REG_SZ
Data: 500

Value 6
Name: base
Type: REG_SZ
Data: 10

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0043\ndi\params\req_medium
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 3:31 PM

Value 0
Name: paramDesc
Type: REG_SZ
Data: Speed & Duplex

Value 1
Name: default
Type: REG_SZ
Data: 0

Value 2
Name: type
Type: REG_SZ
Data: enum

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0043\ndi\params\req_medium\enum
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 3:31 PM

Value 0
Name: 0
Type: REG_SZ
Data: Auto

Value 1
Name: 65794
Type: REG_SZ
Data: 10 Mb Half

Value 2
Name: 258
Type: REG_SZ
Data: 10 Mb Full

Value 3
Name: 66050
Type: REG_SZ
Data: 100 Mb Half

Value 4
Name: 514
Type: REG_SZ
Data: 100 Mb Full

HP NC373i Driver Parameters (Port 2)

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0044
Class Name: <NO CLASS>
Last Write Time: 6/15/2006 - 11:34 AM

Value 0
Name: create_pdo_flag
Type: REG_SZ
Data: 4

Value 1
Name: mtu
Type: REG_SZ
Data: 1500

Value 2
Name: req_medium
Type: REG_SZ
Data: 0

Value 3
Name: InfPath
Type: REG_SZ
Data: oem4.inf

Value 4
Name: InfSection
Type: REG_SZ
Data: NC373i_inst_amd64

Value 5
Name: ProviderName
Type: REG_SZ
Data: Hewlett-Packard Company

Value 6
Name: DriverDateData
Type: REG_BINARY
Data: 00 c0 59 88 d6 64 c6 01 -
.ÅY.ÖdE.

Value 7
Name: DriverDate
Type: REG_SZ
Data: 4-21-2006

Value 8
Name: DriverVersion
Type: REG_SZ
Data: 2.6.17.0

Value 9
Name: MatchingDeviceId
Type: REG_SZ
Data: pci\ven_14e4&dev_164c&subsys_7038103c

Value 10
Name: DriverDesc
Type: REG_SZ
Data: HP NC373i Virtual Bus Device

Value 11
Name: optimize_ips
Type: REG_SZ
Data: 0

Value 12
Name: target_ips
Type: REG_SZ
Data: 2000

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0044\ndi
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 3:31 PM

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0044\ndi\params
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 3:31 PM

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0044\ndi\params\mtu
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 3:31 PM

Value 0
Name: paramdesc
Type: REG_SZ
Data: Maximum Transfer Unit

Value 1
Name: default
Type: REG_SZ
Data: 1500

Value 2
Name: type
Type: REG_SZ
Data: dword

Value 3
Name: min
Type: REG_SZ
Data: 1500

Value 4
Name: max
Type: REG_SZ
Data: 9000

Value 5
Name: step

Type: REG_SZ
Data: 500

Value 6
Name: base
Type: REG_SZ
Data: 10

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0044\ndi\params\req_medium
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 3:31 PM

Value 0
Name: paramDesc
Type: REG_SZ
Data: Speed & Duplex

Value 1
Name: default
Type: REG_SZ
Data: 0

Value 2
Name: type
Type: REG_SZ
Data: enum

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0044\ndi\params\req_medium\enum
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 3:31 PM

Value 0
Name: 0
Type: REG_SZ
Data: Auto

Value 1
Name: 65794
Type: REG_SZ
Data: 10 Mb Half

Value 2
Name: 258
Type: REG_SZ
Data: 10 Mb Full

Value 3
Name: 66050
Type: REG_SZ
Data: 100 Mb Half

Value 4
Name: 514
Type: REG_SZ
Data: 100 Mb Full

Microsoft SQL Server 2005 Registry Parameters

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90
Class Name: <NO CLASS>
Last Write Time: 6/1/2006 - 11:30 AM
Value 0
  Name: SharedCode
  Type: REG_SZ
  Data: C:\Program Files\Microsoft SQL
Server\90\Shared\

Value 1
  Name: VerSpecificRootDir
  Type: REG_SZ
  Data: C:\Program Files\Microsoft SQL
Server\90\

Value 2
  Name: EnableErrorReporting
  Type: REG_DWORD
  Data: 0

Value 3
  Name: ErrorReportingDir
  Type: REG_SZ
  Data: C:\Program Files\Microsoft SQL
Server\90\Shared>ErrorDumps\

Value 4
  Name: CustomerFeedback
  Type: REG_DWORD
  Data: 0

Value 5
  Name: TimeOfReporting
  Type: REG_DWORD
  Data: 0xf0

Value 6
  Name: CollectorParameters
  Type: REG_SZ
  Data:

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Bootstrap
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 4:41 PM
Value 0
  Name: BootstrapDir
  Type: REG_SZ
  Data: C:\Program Files\Microsoft SQL
Server\90\Setup Bootstrap\

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Bootstrap\MSIRefCount
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 5:01 PM
Value 0
  Name: Uninstall
  Type: REG_SZ
  Data: {11111111-1111-1111-1111-
111111111111}, {79F1B65E-8FC0-4D03-954D-
F9E71C85AEC7}, {73D8B59D-0BFF-4B5B-A031-FAB3AC629E56}

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Bootstrap\Setup
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 4:54 PM
Value 0
  Name: PatchLevel
  Type: REG_SZ
  Data: 9.1.2047

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Cost
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 4:43 PM
Value 0
  Name: SQLSetup
  Type: REG_SZ
  Data: 8388608

Value 1
  Name: MSXML6
  Type: REG_SZ
  Data: 15728640

Value 2
  Name: ToolsSetup
  Type: REG_SZ
  Data: 26214400

Value 3
  Name: PPESku
  Type: REG_SZ
  Data: 157286400

Value 4
  Name: owc11
  Type: REG_SZ
  Data: 4194304

Value 5
  Name: sqlxml4
  Type: REG_SZ
  Data: 8388608

Value 6
  Name: bol

```

```

Type: REG_SZ
Data: 178257920

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Machines
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 5:08 PM
Value 0
  Name: OriginalMachineName
  Type: REG_SZ
  Data: DL380

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration
Class Name: <NO CLASS>
Last Write Time: 6/1/2006 - 11:30 AM

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node0
Class Name: <NO CLASS>
Last Write Time: 6/1/2006 - 11:40 AM
Value 0
  Name: CPUMask
  Type: REG_DWORD
  Data: 0x3

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node1
Class Name: <NO CLASS>
Last Write Time: 6/1/2006 - 11:40 AM
Value 0
  Name: CPUMask
  Type: REG_DWORD
  Data: 0xc

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\ProductID
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 5:02 PM
Value 0
  Name: ProductID77591
  Type: REG_SZ
  Data: 77591-045-4200564-05245

```

```

Value 1
  Name: DigitalProductID77591
  Type: REG_BINARY
  Data:
00000000 a4 00 00 00 03 00 00 00 - 37 37 35 39 31
2d 30 34 0.....77591-04
00000010 35 2d 34 32 30 30 35 36 - 34 2d 30 35 32
34 35 00 5-4200564-05245.
00000020 0a 00 00 00 00 00 00 00 - 00 00 00 00 00
00 00 00 .....

```

```

00000030 00 00 00 00 30 1c 6a 05 - dd 4a b4 eb 29
8b 0f 0b ....0.j.YJ'è)...
00000040 ef 7c 00 00 00 00 00 00 - ed c6 7d 44 45
6d 37 00 i|.....i.}DEm7.
00000050 00 00 00 00 00 00 00 00 - 00 00 00 00 00
00 00 00 .....
00000060 00 00 00 00 00 00 00 00 - 30 35 38 30 30
00 00 00 .....05800...
00000070 00 00 00 00 00 00 00 00 - 9e 62 b9 20 00
80 00 00 .....b¹.....
00000080 00 00 00 00 00 00 00 00 - 00 00 00 00 00
00 00 00 .....
00000090 00 00 00 00 00 00 00 00 - 00 00 00 00 00
00 00 00 .....
2b 3e 4a 2f
+>J/

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Replication
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 5:03 PM
Value 0
Name: <NO NAME>
Type: REG_SZ
Data:

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Replication\ArticleResolver
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 5:03 PM
Value 0
Name: Microsoft SQL Server Additive
Conflict Resolver
Type: REG_SZ
Data: {4B385BCE-190B-46c5-AEAB-
51358A8E1CB6}

```

```

Value 1
Name: Microsoft SQL Server Averaging
Conflict Resolver
Type: REG_SZ
Data: {337754AA-CF6E-4be8-8F13-
F6AC91524EDC}

```

```

Value 2
Name: Microsoft SQL Server DATETIME
(Earlier Wins) Conflict Resolver
Type: REG_SZ
Data: {464CBD74-3177-4593-ADC7-
7F7AC6F29286}

```

```

Value 3
Name: Microsoft SQL Server DATETIME
(Later Wins) Conflict Resolver
Type: REG_SZ
Data: {D604B4B5-686B-4304-9613-
C4F82B527B10}

```

Value 4

```

Name: Microsoft SQL Server Download Only
Conflict Resolver
Type: REG_SZ
Data: {B57463F2-6ACE-4206-B600-
CAB83A0847D2}

```

```

Value 5
Name: Microsoft SQL Server Maximum
Conflict Resolver
Type: REG_SZ
Data: {D604B4B5-686B-4304-9613-
C4F82B527B10}

```

```

Value 6
Name: Microsoft SQL Server Merge Text
Columns Conflict Resolver
Type: REG_SZ
Data: {B451ED26-AB0C-4230-B989-
F2959ECDDF22}

```

```

Value 7
Name: Microsoft SQL Server Minimum
Conflict Resolver
Type: REG_SZ
Data: {464CBD74-3177-4593-ADC7-
7F7AC6F29286}

```

```

Value 8
Name: Microsoft SQL Server Priority
Column Resolver
Type: REG_SZ
Data: {D604B4B5-686B-4304-9613-
C4F82B527B10}

```

```

Value 9
Name: Microsoft SQL Server Subscriber
Always Wins Conflict Resolver
Type: REG_SZ
Data: {FF0EC373-ABAD-44dd-9ECC-
9FB6BB55F54E}

```

```

Value 10
Name: Microsoft SQL Server Upload Only
Conflict Resolver
Type: REG_SZ
Data: {D6A60F76-F121-40e7-88A8-
605BE6CE1EB5}

```

```

Value 11
Name: Microsoft SQLServer Stored
Procedure Resolver
Type: REG_SZ
Data: {87EC4491-1B75-4844-B7CF-
090A1FB84BA6}

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Tools
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 4:43 PM

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Tools\ClientSetup
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 4:43 PM
Value 0
Name: SQLBinRoot
Type: REG_SZ
Data: C:\Program Files\Microsoft SQL
Server\90\Shared\

```

```

Value 1
Name: SQLPath
Type: REG_SZ
Data: C:\Program Files\Microsoft SQL
Server\90\Tools

```

```

Value 2
Name: Path
Type: REG_SZ
Data: C:\Program Files\Microsoft SQL
Server\90\Tools\bin\

```

```

Value 3
Name: checksum
Type: REG_BINARY
Data:
00000000 38 36 33 66 66 66 66 30 65 - 62 38 32 62 30
34 37 62 863fff0eb82b047b
00000010 31 30 35 31 36 38 32 31 - 30 30 63 66 30
36 66 63 1051682100cf06fc
00000020 37 62 32 34 31 66 65 35 - 62 35 62 39 62
63 32 35 7b241fe5b5b9bc25
00000030 30 34 35 64 38 62 35 35 - 35 35 38 65 39
33 66 38 045d8b55558e93f8
00000040 66 65 36 33 63 37 64 66 - 31 33 61 36 33
65 61 39 fe63c7df13a63ea9
00000050 63 33 65 65 37 33 36 37 - 35 36 65 61 37
32 63 39 c3ee736756ea72c9
00000060 64 34 34 66 34 38 33 66 - 30 30 64 39 64
36 30 64 d44f483f00d9d60d
00000070 65 62 35 37 35 37 31 61 - 33 61 31 64 62
61 33 38 eb57571a3aldba38
00000080 39 62 64 39 33 39 36 37 - 36 64 66 62 61
65 38 37 9bd939676dfbae87
00000090 38 33 30 61 32 33 64 37 - 37 35 33 33 62
63 39 38 830a23d77533bc98
61 61 39 66 35 31 66 39 - 65 63 66 37 62 38 00
aa9f51f9ecf7b8.

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Tools\ClientSetup\CurrentVersion
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 5:02 PM
Value 0
Name: Language
Type: REG_DWORD
Data: 0x409

```

```

Value 1
Name: CurrentVersion

```

Type: REG_SZ
Data: 9.00.2047.00

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\Tools\Service Manager
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 4:41 PM
Value 0
Name: Services
Type: REG_MULTI_SZ
Data: ;

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\Tools\Setup
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 5:07 PM
Value 0
Name: Version
Type: REG_SZ
Data: 9.1.2047.00

Value 1
Name: Language
Type: REG_DWORD
Data: 0x409

Value 2
Name: ProductCode
Type: REG_SZ
Data: {73D8B59D-0BFF-4B5B-A031-FAB3AC629E56}

Value 3
Name: FeatureList
Type: REG_SZ
Data: Client_Components=3
SQL_Documentation=3 Connectivity=3 SQL_Tools90=3
SQL_WarehouseDevWorkbench=3 SDK=3 SQLXML=3
Tools_Legacy=3 TOOLS_BC_DEP=3 SQL_BooksOnline=3

Value 4
Name: PatchLevel
Type: REG_SZ
Data: 9.1.2047

Value 5
Name: SQLPath
Type: REG_SZ
Data: C:\Program Files\Microsoft SQL Server\90\Tools\

Value 6
Name: checksum
Type: REG_BINARY
Data: 00000000 38 36 33 66 66 66 30 65 - 62 38 32 62 30
34 37 62 863fff0eb82b047b
00000010 31 30 35 31 36 38 32 31 - 30 30 63 66 30
36 66 63 1051682100cf06fc

00000020 37 62 32 34 31 66 65 35 - 62 35 62 39 62
63 32 35 7b241fe5b5b9bc25
00000030 30 34 35 64 38 62 35 35 - 35 35 38 65 39
33 66 38 045d8b55558e93f8
00000040 66 65 36 33 63 37 64 66 - 31 33 61 36 33
65 61 39 fe63c7df13a63ea9
00000050 63 33 65 65 37 33 36 37 - 35 36 65 61 37
32 63 39 c3ee736756ea72c9
00000060 64 34 34 66 34 38 33 66 - 30 30 64 39 64
36 30 64 d44f483f00d9d60d
00000070 65 62 35 37 35 37 31 61 - 33 61 31 64 62
61 33 38 eb57571a3a1dba38
00000080 39 62 64 39 33 39 36 37 - 36 64 66 62 61
65 38 37 9bd939676dfbae87
00000090 38 33 30 61 32 33 64 37 - 37 35 33 33 62
63 39 38 830a23d77533bc98
61 61 39 66 35 31 66 39 - 65 63 66 37 62 38 00
aa9f51f9ecf7b8.

Value 7
Name: Edition
Type: REG_SZ
Data: Enterprise Edition (64-bit)

Value 8
Name: EditionType
Type: REG_SZ
Data: Enterprise Edition (64-bit)

Value 9
Name: SP
Type: REG_DWORD
Data: 0x1

Value 10
Name: AssociatedHotfixBuild
Type: REG_SZ
Data: 0

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\Tools\Shortcut
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 4:41 PM

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\Tools\Shortcut\SQL Server Error and Usage Reporting
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 4:41 PM
Value 0
Name: 1033
Type: REG_SZ
Data:

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\Uninstall Info
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 4:43 PM

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\Uninstall Info\{73D8B59D-0BFF-4B5B-A031-FAB3AC629E56}
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 5:08 PM
Value 0
Name: Product
Type: REG_SZ
Data: Client_Components

Value 1
Name: SetupHive
Type: REG_SZ
Data: 90\Tools

Value 2
Name: RemoveList
Type: REG_SZ
Data: Client_Components,Connectivity,SQL_Tools90,SQL_WarehouseDevWorkbench,SDK,SQLXML,Tools_Legacy,TOOLS_BC_DEP,SQL_Documentation,SQL_BooksOnline,SQL_DatabaseSamples,SQL_AdventureWorksSamples,SQL_AdventureWorksDWSamples,SQL_AdventureWorksASSamples,SQL_Samples

Value 3
Name: ModifyPath
Type: REG_SZ
Data: "C:\Program Files\Microsoft SQL Server\90\Setup Bootstrap\setup.exe" /resumsetup /productcode {73D8B59D-0BFF-4B5B-A031-FAB3AC629E56} /I {73D8B59D-0BFF-4B5B-A031-FAB3AC629E56}

Value 4
Name: UninstallPath
Type: REG_SZ
Data: "C:\Program Files\Microsoft SQL Server\90\Setup Bootstrap\setup.exe" /resumsetup /productcode {73D8B59D-0BFF-4B5B-A031-FAB3AC629E56} /X {73D8B59D-0BFF-4B5B-A031-FAB3AC629E56}

Value 5
Name: Status
Type: REG_DWORD
Data: 0

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\Uninstall Info\{79F1B65E-8FC0-4D03-954D-F9E71C85AEC7}
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 5:02 PM
Value 0
Name: Product
Type: REG_SZ
Data: SQL_Engine

Value 1
Name: SetupHive
Type: REG_SZ

Data: MSSQL.1

Value 2
Name: RemoveList
Type: REG_SZ
Data: SQL_Engine,SQL_Data_Files,SQL_Replication,SQL_FullText,SQL_SharedTools,SQL_BC_DEP

Value 3
Name: ModifyPath
Type: REG_SZ
Data: "C:\Program Files\Microsoft SQL Server\90\Setup Bootstrap\setup.exe" /resumsetup /productcode {79F1B65E-8FC0-4D03-954D-F9E71C85AEC7} /I {79F1B65E-8FC0-4D03-954D-F9E71C85AEC7}

Value 4
Name: UninstallPath
Type: REG_SZ
Data: "C:\Program Files\Microsoft SQL Server\90\Setup Bootstrap\setup.exe" /resumsetup /productcode {79F1B65E-8FC0-4D03-954D-F9E71C85AEC7} /X {79F1B65E-8FC0-4D03-954D-F9E71C85AEC7}

Value 5
Name: Status
Type: REG_DWORD
Data: 0

Additional Microsoft SQL Server 2005 Registry Parameters

Key Name: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp
Class Name: <NO CLASS>
Last Write Time: 5/31/2006 - 4:41 PM

Value 0
Name: Enabled
Type: REG_DWORD
Data: 0x1

Value 1
Name: ListenOnAllIPs
Type: REG_DWORD
Data: 0x1

Value 2
Name: NoDelay
Type: REG_DWORD
Data: 0

Value 3

Name: KeepAlive
Type: REG_DWORD
Data: 0x7530

Value 4
Name: DisplayName
Type: REG_SZ
Data: TCP/IP

Key Name: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP1
Class Name: <NO CLASS>
Last Write Time: 6/1/2006 - 11:36 AM

Value 0
Name: Enabled
Type: REG_DWORD
Data: 0x1

Value 1
Name: Active
Type: REG_DWORD
Data: 0x1

Value 2
Name: TcpPort
Type: REG_SZ
Data: 2001

Value 3
Name: TcpDynamicPorts
Type: REG_SZ
Data:

Value 4
Name: DisplayName
Type: REG_SZ
Data: Specific IP Address

Value 5
Name: IpAddress
Type: REG_SZ
Data: 130.168.208.25

Key Name: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP2
Class Name: <NO CLASS>
Last Write Time: 6/1/2006 - 11:36 AM

Value 0
Name: Enabled
Type: REG_DWORD
Data: 0x1

Value 1
Name: Active
Type: REG_DWORD
Data: 0x1

Value 2
Name: TcpPort

Type: REG_SZ
Data: 2002

Value 3
Name: TcpDynamicPorts
Type: REG_SZ
Data:

Value 4
Name: DisplayName
Type: REG_SZ
Data: Specific IP Address

Value 5
Name: IpAddress
Type: REG_SZ
Data: 130.120.208.25

Key Name: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP3
Class Name: <NO CLASS>
Last Write Time: 6/1/2006 - 1:42 PM

Value 0
Name: Enabled
Type: REG_DWORD
Data: 0x1

Value 1
Name: Active
Type: REG_DWORD
Data: 0x1

Value 2
Name: TcpPort
Type: REG_SZ
Data: 1433

Value 3
Name: TcpDynamicPorts
Type: REG_SZ
Data:

Value 4
Name: DisplayName
Type: REG_SZ
Data: Specific IP Address

Value 5
Name: IpAddress
Type: REG_SZ
Data: 127.0.0.1

Key Name: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP4
Class Name: <NO CLASS>
Last Write Time: 6/1/2006 - 11:35 AM

Value 0
Name: TcpPort

```

Type:          REG_SZ
Data:          2001[0x1],2002[0x2]

Value 1
Name:          TcpDynamicPorts
Type:          REG_SZ
Data:

Value 2
Name:          DisplayName
Type:          REG_SZ
Data:          Any IP Address

```

System Summary

System Information report written at: 06/17/06 09:39:29

System Name: DL380
[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003 Enterprise x64 Edition
Version	5.2.3790 Service Pack 1 Build 3790
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	DL380
System Manufacturer	HP
System Model	ProLiant DL380 G5
System Type	x64-based PC
Processor EM64T Family	6 Model 15 Stepping 5
GenuineIntel	~3000 Mhz
Processor EM64T Family	6 Model 15 Stepping 5
GenuineIntel	~3000 Mhz
Processor EM64T Family	6 Model 15 Stepping 5
GenuineIntel	~3000 Mhz
BIOS Version/Date	HP P56, 5/30/2006
SMBIOS Version	2.3
Windows Directory	C:\WINDOWS
System Directory	C:\WINDOWS\system32
Boot Device	\Device\HarddiskVolume12
Locale	United States
Hardware Abstraction Layer	Version = "5.2.3790.1830 (srv03_spl_rtm.050324-1447)"
User Name	DL380\Administrator
Time Zone	Central Daylight Time
Total Physical Memory	32,767.11 MB
Available Physical Memory	30.87 GB
Total Virtual Memory	33.08 GB
Available Virtual Memory	32.83 GB
Page File Space	2.00 GB
Page File	C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

```

Resource Device
I/O Port 0x00000000-0x00000CF7 PCI bus
I/O Port 0x00000000-0x00000CF7 Direct memory
access controller

Memory Address 0xFDB00000-0xFDDFFFFFF PCI standard
PCI-to-PCI bridge
Memory Address 0xFDB00000-0xFDDFFFFFF PCI standard
PCI-to-PCI bridge

IRQ 7 Base System Device
IRQ 7 PCI Device

I/O Port 0x00006000-0x00006FFF PCI standard
PCI-to-PCI bridge
I/O Port 0x00006000-0x00006FFF Smart Array
P800 Controller (Non-Miniport)

IRQ 16 PCI standard PCI-to-PCI bridge
IRQ 16 QLogic Fibre Channel Adapter
IRQ 16 Smart Array Controller
IRQ 16 QLogic Fibre Channel Adapter
IRQ 16 QLogic Fibre Channel Adapter
IRQ 16 PCI standard PCI-to-PCI bridge
IRQ 16 HP NC373i Virtual Bus Device
IRQ 16 Standard Universal PCI to USB Host
Controller
IRQ 16 Standard Enhanced PCI to USB Host
Controller

IRQ 17 QLogic Fibre Channel Adapter
IRQ 17 PCI standard PCI-to-PCI bridge
IRQ 17 Smart Array P800 Controller (Non-Miniport)

IRQ 17 QLogic Fibre Channel Adapter
IRQ 17 QLogic Fibre Channel Adapter
IRQ 17 PCI standard PCI-to-PCI bridge
IRQ 17 HP NC373i Virtual Bus Device
IRQ 17 Standard Universal PCI to USB Host
Controller

I/O Port 0x00005000-0x00006FFF PCI standard
PCI-to-PCI bridge
I/O Port 0x00005000-0x00006FFF PCI standard
PCI-to-PCI bridge
I/O Port 0x00005000-0x00006FFF PCI standard
PCI-to-PCI bridge
I/O Port 0x00005000-0x00006FFF QLogic Fibre
Channel Adapter

IRQ 18 PCI standard PCI-to-PCI bridge
IRQ 18 Standard Universal PCI to USB Host
Controller

Memory Address 0xA0000-0xBFFFF PCI bus
Memory Address 0xA0000-0xBFFFF ATI ES1000

Memory Address 0xFA000000-0xFBFFFFFF PCI standard
PCI-to-PCI bridge

```

```

Memory Address 0xFA000000-0xFBFFFFFF PCI standard
PCI-to-PCI bridge
Memory Address 0xFA000000-0xFBFFFFFF HP NC373i
Virtual Bus Device

Memory Address 0xF8000000-0xF9FFFFFF PCI standard
PCI-to-PCI bridge
Memory Address 0xF8000000-0xF9FFFFFF PCI standard
PCI-to-PCI bridge
Memory Address 0xF8000000-0xF9FFFFFF HP NC373i
Virtual Bus Device

I/O Port 0x00007000-0x00007FFF PCI standard
PCI-to-PCI bridge
I/O Port 0x00007000-0x00007FFF QLogic Fibre
Channel Adapter

I/O Port 0x00004000-0x00004FFF PCI standard
PCI-to-PCI bridge
I/O Port 0x00004000-0x00004FFF Smart Array
Controller

I/O Port 0x00008000-0x00008FFF PCI standard
PCI-to-PCI bridge
I/O Port 0x00008000-0x00008FFF QLogic Fibre
Channel Adapter

[DMA]

Resource Device Status
Channel 7 Direct memory access controller OK
Channel 2 Standard floppy disk controller OK

[Forced Hardware]

Device PNP Device ID

[I/O]

Resource Device Status
0x00000000-0x00000CF7 PCI bus OK
0x00000000-0x00000CF7 Direct memory access
controller OK
0x00000D00-0x0000FFFF PCI bus OK
0x00005000-0x00006FFF PCI standard PCI-to-PCI
bridge OK
0x00005000-0x00006FFF PCI standard PCI-to-PCI
bridge OK
0x00005000-0x00006FFF QLogic Fibre Channel
Adapter OK
0x00005400-0x000054FF QLogic Fibre Channel
Adapter OK
0x00006000-0x00006FFF PCI standard PCI-to-PCI
bridge OK
0x00006000-0x00006FFF Smart Array P800
Controller (Non-Miniport) OK
0x00004000-0x00004FFF PCI standard PCI-to-PCI
bridge OK

```

0x00004000-0x00004FFF OK Smart Array Controller
 0x00007000-0x00007FFF OK PCI standard PCI-to-PCI bridge
 0x00007000-0x00007FFF OK QLogic Fibre Channel Adapter
 0x00007400-0x000074FF OK QLogic Fibre Channel Adapter
 0x00008000-0x00008FFF OK PCI standard PCI-to-PCI bridge
 0x00008000-0x00008FFF OK QLogic Fibre Channel Adapter
 0x00008400-0x000084FF OK QLogic Fibre Channel Adapter
 0x00001000-0x0000101F OK Standard Universal PCI to USB Host Controller
 0x00001020-0x0000103F OK Standard Universal PCI to USB Host Controller
 0x00001040-0x0000105F OK Standard Universal PCI to USB Host Controller
 0x00001060-0x0000107F OK Standard Universal PCI to USB Host Controller
 0x00003000-0x000030FF OK ATI ES1000
 0x000003B0-0x000003BB OK ATI ES1000
 0x000003C0-0x000003DF OK ATI ES1000
 0x00002800-0x000028FF OK Base System Device
 0x00003400-0x000034FF OK Base System Device
 0x00003800-0x0000381F OK Standard Universal PCI to USB Host Controller
 0x00000070-0x00000077 OK Motherboard resources
 0x00000408-0x0000040F OK Motherboard resources
 0x000004D0-0x000004D1 OK Motherboard resources
 0x00000020-0x0000003F OK Motherboard resources
 0x000000A0-0x000000BF OK Motherboard resources
 0x00000090-0x0000009F OK Motherboard resources
 0x00000050-0x00000053 OK Motherboard resources
 0x00000700-0x0000071F OK Motherboard resources
 0x00000800-0x0000083F OK Motherboard resources
 0x00000900-0x0000097F OK Motherboard resources
 0x00000010-0x0000001F OK Motherboard resources
 0x00000C80-0x00000C83 OK Motherboard resources
 0x00000CD4-0x00000CD7 OK Motherboard resources
 0x00000F50-0x00000F58 OK Motherboard resources

0x000000F0-0x000000F0 OK Motherboard resources
 0x00000CA0-0x00000CA1 OK Motherboard resources
 0x00000CA4-0x00000CA5 OK Motherboard resources
 0x00000CA2-0x00000CA3 OK System timer
 0x00000040-0x00000043 OK System timer
 0x00000080-0x0000008F OK Direct memory access controller
 0x000000C0-0x000000DF OK Direct memory access controller
 0x00000061-0x00000061 OK System speaker
 0x00000060-0x00000060 OK Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
 0x00000064-0x00000064 OK Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
 0x0000002E-0x0000002F OK Extended IO Bus
 0x0000004E-0x0000004F OK Extended IO Bus
 0x00000620-0x0000065F OK Extended IO Bus
 0x00000680-0x0000069F OK Extended IO Bus
 0x00000600-0x0000061F OK Extended IO Bus
 0x00000660-0x0000067F OK Extended IO Bus
 0x00000300-0x0000030F OK Extended IO Bus
 0x000003F8-0x000003FF OK Communications Port (COM1)
 0x000003F0-0x000003F5 OK Standard floppy disk controller
 0x000003F7-0x000003F7 OK Standard floppy disk controller
 0x00000500-0x0000050F OK Standard Dual Channel PCI IDE Controller
 0x000001F0-0x000001F7 OK Primary IDE Channel
 0x000003F6-0x000003F6 OK Primary IDE Channel
 0x00000170-0x00000177 OK Secondary IDE Channel
 0x00000376-0x00000376 OK Secondary IDE Channel
 [IRQs]
 Resource Device Status
 IRQ 9 Microsoft ACPI-Compliant System OK
 IRQ 16 PCI standard PCI-to-PCI bridge OK
 IRQ 16 QLogic Fibre Channel Adapter OK
 IRQ 16 Smart Array Controller OK
 IRQ 16 QLogic Fibre Channel Adapter OK
 IRQ 16 QLogic Fibre Channel Adapter OK
 IRQ 16 PCI standard PCI-to-PCI bridge OK

IRQ 16 HP NC373i Virtual Bus Device OK
 IRQ 16 Standard Universal PCI to USB Host Controller OK
 IRQ 16 Standard Enhanced PCI to USB Host Controller OK
 IRQ 17 QLogic Fibre Channel Adapter OK
 IRQ 17 PCI standard PCI-to-PCI bridge OK
 IRQ 17 Smart Array P800 Controller (Non-Miniport) OK
 IRQ 17 QLogic Fibre Channel Adapter OK
 IRQ 17 QLogic Fibre Channel Adapter OK
 IRQ 17 PCI standard PCI-to-PCI bridge OK
 IRQ 17 HP NC373i Virtual Bus Device OK
 IRQ 17 Standard Universal PCI to USB Host Controller OK
 IRQ 18 PCI standard PCI-to-PCI bridge OK
 IRQ 18 Standard Universal PCI to USB Host Controller OK
 IRQ 19 Standard Universal PCI to USB Host Controller OK
 IRQ 23 ATI ES1000 OK
 IRQ 7 Base System Device OK
 IRQ 7 PCI Device OK
 IRQ 5 Base System Device OK
 IRQ 22 Standard Universal PCI to USB Host Controller OK
 IRQ 0 System timer OK
 IRQ 1 Standard 101/102-Key or Microsoft Natural PS/2 Keyboard OK
 IRQ 12 PS/2 Compatible Mouse OK
 IRQ 4 Communications Port (COM1) OK
 IRQ 6 Standard floppy disk controller OK
 [Memory]
 Resource Device Status
 0xA0000-0xBFFFF PCI bus OK
 0xA0000-0xBFFFF ATI ES1000 OK
 0xD0000000-0xDFFFFFFF PCI bus OK
 0xF0000000-0xFEBFFFFFFF PCI bus OK
 0xFDA00000-0xFDDFFFFFFF PCI standard PCI-to-PCI bridge OK
 0xFDB00000-0xFDDFFFFFFF PCI standard PCI-to-PCI bridge OK
 0xFDB00000-0xFDDFFFFFFF PCI standard PCI-to-PCI bridge OK
 0xFDB00000-0xFDBF3FFF QLogic Fibre Channel Adapter OK
 0xFDBE0000-0xFDBE3FFF QLogic Fibre Channel Adapter OK
 0xFDC00000-0xFDDFFFFFFF PCI standard PCI-to-PCI bridge OK
 0xFDD00000-0xFDDFFFFFFF Smart Array P800 Controller (Non-Miniport) OK
 0xFDCF0000-0xFDCF0FFF Smart Array P800 Controller (Non-Miniport) OK
 0xFDB00000-0xFD9FFFFFFF PCI standard PCI-to-PCI bridge OK

```

0xFD900000-0xFD9FFFFF Smart Array Controller
OK
0xFD8F0000-0xFD8FFFFF Smart Array Controller
OK
0xFDE00000-0xFDEFFFFF PCI standard PCI-to-PCI
bridge OK
0xFDEF0000-0xFDEF3FFF QLogic Fibre Channel
Adapter OK
0xFDEE0000-0xFDEE3FFF QLogic Fibre Channel
Adapter OK
0xFDF00000-0xFDF3FFF PCI standard PCI-to-PCI
bridge OK
0xFDF00000-0xFDF3FFF QLogic Fibre Channel
Adapter OK
0xFDFE0000-0xFDFE3FFF QLogic Fibre Channel
Adapter OK
0xF8000000-0xF9FFFFF PCI standard PCI-to-PCI
bridge OK
0xF8000000-0xF9FFFFF PCI standard PCI-to-PCI
bridge OK
0xF8000000-0xF9FFFFF HP NC373i Virtual Bus
Device OK
0xFA000000-0xFBFFFFF PCI standard PCI-to-PCI
bridge OK
0xFA000000-0xFBFFFFF PCI standard PCI-to-PCI
bridge OK
0xFA000000-0xFBFFFFF HP NC373i Virtual Bus
Device OK
0xF7DF0000-0xF7DF03FF Standard Enhanced PCI
to USB Host Controller OK
0xD8000000-0xDFFFFFFF ATI ES1000 OK

0xF7F00000-0xF7FFFFFF ATI ES1000 OK

0xF7FE0000-0xF7FE01FF Base System Device OK

0xF7FD0000-0xF7FD07FF Base System Device OK

0xF7FC0000-0xF7FC1FFF Base System Device OK

0xF7F00000-0xF7F7FFFF Base System Device OK

0xF7EF0000-0xF7EF00FF PCI Device OK

0xE0000000-0xEFFFFFFF Motherboard resources
OK
0xFE000000-0xFEFFFFFF Motherboard resources
OK
0xFED00000-0xFED03FFF High precision event
timer OK

[Components]

[Multimedia]

[Audio Codecs]

```

```

CODEC Manufacturer Description
Status File Version Size
Creation Date
c:\windows\system32\msg711.acm Microsoft
Corporation OK
C:\WINDOWS\system32\MSG711.ACM
5.2.3790.1830 (srv03_spl_rtm.050324-1447)
13.50 KB (13,824 bytes) 3/25/2005
6:00 AM
c:\windows\system32\imaadp32.acm Microsoft
Corporation OK
C:\WINDOWS\system32\IMAADP32.ACM
5.2.3790.1830 (srv03_spl_rtm.050324-1447)
24.00 KB (24,576 bytes) 3/25/2005
6:00 AM
c:\windows\system32\msgsm32.acm Microsoft
Corporation OK
C:\WINDOWS\system32\MSGSM32.ACM
5.2.3790.1830 (srv03_spl_rtm.050324-1447)
34.50 KB (35,328 bytes) 3/25/2005
6:00 AM
c:\windows\system32\tssoft32.acm DSP GROUP,
INC. OK
C:\WINDOWS\system32\TSSOFT32.ACM
1.01 13.50 KB (13,824 bytes)
3/25/2005 6:00 AM
c:\windows\system32\msadp32.acm Microsoft
Corporation OK
C:\WINDOWS\system32\MSADP32.ACM
5.2.3790.1830 (srv03_spl_rtm.050324-1447)
23.50 KB (24,064 bytes) 3/25/2005
6:00 AM

[Video Codecs]
CODEC Manufacturer Description
Status File Version Size
Creation Date
c:\windows\system32\mrle32.dll Microsoft
Corporation OK
C:\WINDOWS\system32\MSRLE32.DLL
5.2.3790.1830 (srv03_spl_rtm.050324-1447)
15.50 KB (15,872 bytes) 3/25/2005
6:00 AM
c:\windows\system32\msyuv.dll Microsoft Corporation
OK
C:\WINDOWS\system32\MSYUV.DLL 5.2.3790.1830
(srv03_spl_rtm.050324-1447) 21.00 KB (21,504 bytes)
3/24/2005 11:21 AM
c:\windows\system32\tscopyuv.dll Microsoft
Corporation OK
C:\WINDOWS\system32\TSBYUV.DLL
5.2.3790.1830 (srv03_spl_rtm.050324-1447)
12.50 KB (12,800 bytes) 3/24/2005
11:34 AM
c:\windows\system32\iyuv_32.dll Microsoft
Corporation OK
C:\WINDOWS\system32\IYUV_32.DLL
5.2.3790.1830 (srv03_spl_rtm.050324-1447)
52.50 KB (53,760 bytes) 3/24/2005
11:19 AM
c:\windows\system32\msvidc32.dll Microsoft
Corporation OK

```

```

C:\WINDOWS\system32\MSVIDC32.DLL
5.2.3790.1830 (srv03_spl_rtm.050324-1447)
43.00 KB (44,032 bytes) 3/25/2005
6:00 AM

[CD-ROM]

Item Value

[Sound Device]

Item Value

[Display]

Item Value
Name ATI ES1000
PNP Device ID PCI\VEN_1002&DEV_515E&SUBSYS_31FB103C&REV_0
2\4&2014205D&0&18F0
Adapter Type ATI ES1000 (0x515E), ATI
Technologies Inc. compatible
Adapter Description ATI ES1000
Adapter RAM 32.00 MB (33,554,432 bytes)
Installed Drivers ati2dvag.dll
Driver Version 6.14.10.6583
INF File oem0.inf (ati2mtag_RN50 section)
Color Planes 1
Color Table Entries 4294967296
Resolution 1024 x 768 x 60 hertz
Bits/Pixel 32
Memory Address 0xD8000000-0xDFFFFFFF
I/O Port 0x00003000-0x000030FF
Memory Address 0xF7F00000-0xF7FFFFFF
IRQ Channel IRQ 23
I/O Port 0x000003B0-0x000003BB
I/O Port 0x000003C0-0x000003DF
Memory Address 0xA0000-0xBFFFF
Driver c:\windows\system32\drivers\ati2mtag.sys
(6.14.10.6583, 1.97 MB (2,066,432 bytes), 5/31/2006
3:31 PM)

[Infrared]

Item Value

[Input]

[Keyboard]

Item Value
Description USB Human Interface Device
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID USB\VID_03F0&PID_1027&MI_00\7&2CD6FDA9&0&00
00
Number of Function Keys 12
Driver c:\windows\system32\drivers\hidusb.sys
(5.2.3790.1830 (srv03_spl_rtm.050324-1447), 18.50 KB
(18,944 bytes), 3/25/2005 6:00 AM)

```

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&2AA4AD3D&0
 Number of Function Keys 12
 I/O Port 0x00000060-0x00000060
 I/O Port 0x00000064-0x00000064
 IRQ Channel IRQ 1
 Driver c:\windows\system32\drivers\i8042prt.sys
 (5.2.3790.1830 (srv03_spl_rtm.050324-1447), 91.00 KB
 (93,184 bytes), 3/25/2005 6:00 AM)

[Pointing Device]

Item Value
 Hardware Type USB Human Interface Device
 Number of Buttons 5
 Status OK
 PNP Device ID USB\VID_03F0&PID_1027&MI_01\7&2CD6FDA9&0&00
 01
 Power Management Supported No
 Double Click Threshold 6
 Handedness Right Handed Operation
 Driver c:\windows\system32\drivers\hidusb.sys
 (5.2.3790.1830 (srv03_spl_rtm.050324-1447), 18.50 KB
 (18,944 bytes), 3/25/2005 6:00 AM)

Hardware Type PS/2 Compatible Mouse
 Number of Buttons 5
 Status OK
 PNP Device ID ACPI\PNP0F13\4&2AA4AD3D&0
 Power Management Supported No
 Double Click Threshold 6
 Handedness Right Handed Operation
 IRQ Channel IRQ 12
 Driver c:\windows\system32\drivers\i8042prt.sys
 (5.2.3790.1830 (srv03_spl_rtm.050324-1447), 91.00 KB
 (93,184 bytes), 3/25/2005 6:00 AM)

[Modem]

Item Value

[Network]

[Adapter]

Item Value
 Name [00000001] RAS Async Adapter
 Adapter Type Not Available
 Product Type RAS Async Adapter
 Installed Yes
 PNP Device ID Not Available
 Last Reset 6/16/2006 3:09 PM
 Index 1
 Service Name AsyncMac
 IP Address Not Available
 IP Subnet Not Available

Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000002] WAN Miniport (L2TP)
 Adapter Type Not Available
 Product Type WAN Miniport (L2TP)
 Installed Yes
 PNP Device ID ROOT\MS_L2TPMINIPORT\0000
 Last Reset 6/16/2006 3:09 PM
 Index 2
 Service Name Rasl2tp
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver c:\windows\system32\drivers\rasl2tp.sys
 (5.2.3790.1830 (srv03_spl_rtm.050324-1447), 132.00 KB
 (135,168 bytes), 3/25/2005 6:00 AM)

Name [00000003] WAN Miniport (PPTP)
 Adapter Type Wide Area Network (WAN)
 Product Type WAN Miniport (PPTP)
 Installed Yes
 PNP Device ID ROOT\MS_PPTPMINIPORT\0000
 Last Reset 6/16/2006 3:09 PM
 Index 3
 Service Name PptpMiniport
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 50:50:54:50:30:30
 Driver c:\windows\system32\drivers\raspptp.sys
 (5.2.3790.1830 (srv03_spl_rtm.050324-1447), 117.50 KB
 (120,320 bytes), 3/25/2005 6:00 AM)

Name [00000004] WAN Miniport (PPPOE)
 Adapter Type Wide Area Network (WAN)
 Product Type WAN Miniport (PPPOE)
 Installed Yes
 PNP Device ID ROOT\MS_PPPOEMINIPORT\0000
 Last Reset 6/16/2006 3:09 PM
 Index 4
 Service Name RasPppoe
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 33:50:6F:45:30:30

Driver c:\windows\system32\drivers\raspppoe.sys
 (5.2.3790.1830 (srv03_spl_rtm.050324-1447), 67.50 KB
 (69,120 bytes), 3/25/2005 6:00 AM)

Name [00000005] Direct Parallel
 Adapter Type Not Available
 Product Type Direct Parallel
 Installed Yes
 PNP Device ID ROOT\MS_PTMINIPORT\0000
 Last Reset 6/16/2006 3:09 PM
 Index 5
 Service Name Raspti
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver c:\windows\system32\drivers\raspti.sys
 (5.2.3790.1830 (srv03_spl_rtm.050324-1447), 30.50 KB
 (31,232 bytes), 3/25/2005 6:00 AM)

Name [00000006] WAN Miniport (IP)
 Adapter Type Not Available
 Product Type WAN Miniport (IP)
 Installed Yes
 PNP Device ID ROOT\MS_NDISWANIP\0000
 Last Reset 6/16/2006 3:09 PM
 Index 6
 Service Name NdisWan
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver c:\windows\system32\drivers\ndiswan.sys
 (5.2.3790.1830 (srv03_spl_rtm.050324-1447), 157.50 KB
 (161,280 bytes), 3/25/2005 6:00 AM)

Name [00000007] HP NC373i Multifunction Gigabit
 Server Adapter
 Adapter Type Not Available
 Product Type HP NC373i Multifunction Gigabit
 Server Adapter
 Installed Yes
 PNP Device ID Not Available
 Last Reset 6/16/2006 3:09 PM
 Index 7
 Service Name l2nd
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000008] HP NC373i Multifunction Gigabit
 Server Adapter
 Adapter Type Not Available
 Product Type HP NC373i Multifunction Gigabit
 Server Adapter
 Installed Yes
 PNP Device ID Not Available
 Last Reset 6/16/2006 3:09 PM
 Index 8
 Service Name l2nd
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000009] HP NC373i Multifunction Gigabit
 Server Adapter
 Adapter Type Ethernet 802.3
 Product Type HP NC373i Multifunction Gigabit
 Server Adapter
 Installed Yes
 PNP Device ID B06BDRV\L2ND&PCI_164C14E4&SUBSYS_7038103C&R
 EV_11\6&2826E01F&0&20050300
 Last Reset 6/16/2006 3:09 PM
 Index 9
 Service Name l2nd
 IP Address 130.168.208.25
 IP Subnet 255.255.0.0
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:13:21:0E:12:AA
 Driver c:\windows\system32\drivers\bxnd52a.sys
 (2.6.14.0 built by: WinDDK, 78.00 KB (79,872 bytes),
 5/31/2006 3:31 PM)

Name [00000010] HP NC373i Multifunction Gigabit
 Server Adapter
 Adapter Type Ethernet 802.3
 Product Type HP NC373i Multifunction Gigabit
 Server Adapter
 Installed Yes
 PNP Device ID B06BDRV\L2ND&PCI_164C14E4&SUBSYS_7038103C&R
 EV_11\6&253A0954&0&20050500
 Last Reset 6/16/2006 3:09 PM
 Index 10
 Service Name l2nd
 IP Address 130.120.208.25
 IP Subnet 255.255.0.0
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:13:21:0E:12:A8

Driver c:\windows\system32\drivers\bxnd52a.sys
 (2.6.14.0 built by: WinDDK, 78.00 KB (79,872 bytes),
 5/31/2006 3:31 PM)

[Protocol]

Item	Value
Name	MSAFD Tcpip [TCP/IP]
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD Tcpip [UDP/IP]
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)

Message Oriented	Yes
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	Yes

Name	RSVP UDP Service Provider
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)

Message Oriented	Yes
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	Yes
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	Yes

Name	RSVP TCP Service Provider
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	Yes
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

[WinSock]

Item	Value
File	c:\windows\system32\wsock32.dll
Size	24.50 KB (25,088 bytes)
Version	5.2.3790.1830 (srv03_spl_rtm.050324-1447)

[Ports]

[Serial]

Item	Value
Name	Communications Port (COM1)
Status	OK
PNP Device ID	ACPI\PNP0501\0
Maximum Input Buffer Size	0
Maximum Output Buffer Size	No
Settable Baud Rate	Yes
Settable Data Bits	Yes
Settable Flow Control	Yes
Settable Parity	Yes
Settable Parity Check	Yes
Settable Stop Bits	Yes
Settable RLSD	Yes
Supports RLSD	Yes
Supports 16 Bit Mode	No
Supports Special Characters	No
Baud Rate	9600
Bits/Byte	8
Stop Bits	1
Parity	None
Busy	No
Abort Read/Write on Error	No
Binary Mode Enabled	Yes
Continue Xmit on XOff	No
CTS Outflow Control	No
Discard NULL Bytes	No
DSR Outflow Control	0
DSR Sensitivity	0
DTR Flow Control Type	Enable
EOF Character	0
Error Replace Character	0

```

Error Replacement Enabled      No
Event Character                0
Parity Check Enabled          No
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 Channel                   IRQ 4
I/O Port 0x000003F8-0x000003FF
Driver c:\windows\system32\drivers\serial.sys
(5.2.3790.1830 (srv03_spl_rtm.050324-1447), 118.50 KB
(121,344 bytes), 3/25/2005 6:00 AM)

```

[Parallel]

Item Value

[Storage]

[Drives]

```

Item Value
Drive C:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 33.88 GB (36,381,306,880 bytes)
Free Space 23.98 GB (25,743,155,200 bytes)

```

```

Volume Name
Volume Serial Number 20B9629E

```

```

Drive E:
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

```

```

Drive W:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 663.44 GB (712,358,563,840 bytes)
Free Space 171.35 GB (183,989,882,880 bytes)

```

```

Volume Name Back1
Volume Serial Number E0ABAD04

```

```

Drive X:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 663.44 GB (712,358,563,840 bytes)
Free Space 171.36 GB (183,998,607,360 bytes)

```

```

Volume Name Back2
Volume Serial Number F0B9A7BE

Drive Y:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 663.44 GB (712,358,563,840 bytes)
Free Space 171.36 GB (183,998,607,360 bytes)

```

```

Volume Name Back3
Volume Serial Number 24C94398

```

```

Drive Z:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 663.44 GB (712,358,563,840 bytes)
Free Space 171.36 GB (183,998,574,592 bytes)

```

```

Volume Name Back4
Volume Serial Number 84D996F2

```

[Disks]

```

Item Value
Description \\.\PHYSICALDRIVE0
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 546.68 GB (586,988,881,920 bytes)
Total Cylinders 71,364
Total Sectors 1,146,462,660
Total Tracks 18,197,820
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 546.68 GB (586,989,387,776 bytes)

```

Partition Starting Offset 16,384 bytes

```

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 5
SCSI Target ID 0
Sectors/Track 63
Size 63.47 GB (68,146,444,800 bytes)
Total Cylinders 8,285
Total Sectors 133,098,525

```

```

Total Tracks 2,112,675
Tracks/Cylinder 255
Partition Disk #15, Partition #0
Partition Size 63.47 GB (68,146,412,544 bytes)

```

Partition Starting Offset 32,256 bytes

```

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 5
SCSI Target ID 0
Sectors/Track 63
Size 34.17 GB (36,692,974,080 bytes)
Total Cylinders 4,461
Total Sectors 71,665,965
Total Tracks 1,137,555
Tracks/Cylinder 255
Partition Disk #16, Partition #0
Partition Size 34.17 GB (36,692,941,824 bytes)

```

Partition Starting Offset 32,256 bytes

```

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 5
SCSI Target ID 1
Sectors/Track 63
Size 63.47 GB (68,146,444,800 bytes)
Total Cylinders 8,285
Total Sectors 133,098,525
Total Tracks 2,112,675
Tracks/Cylinder 255
Partition Disk #17, Partition #0
Partition Size 63.47 GB (68,146,412,544 bytes)

```

Partition Starting Offset 32,256 bytes

```

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 5
SCSI Target ID 1
Sectors/Track 63

```

Size 34.17 GB (36,692,974,080 bytes)
 Total Cylinders 4,461
 Total Sectors 71,665,965
 Total Tracks 1,137,555
 Tracks/Cylinder 255
 Partition Disk #18, Partition #0
 Partition Size 34.17 GB (36,692,941,824 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 0
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 5
 SCSI Target ID 1
 Sectors/Track 63
 Size 97.65 GB (104,855,869,440 bytes)
 Total Cylinders 12,748
 Total Sectors 204,796,620
 Total Tracks 3,250,740
 Tracks/Cylinder 255

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 6
 SCSI Target ID 0
 Sectors/Track 63
 Size 63.47 GB (68,146,444,800 bytes)
 Total Cylinders 8,285
 Total Sectors 133,098,525
 Total Tracks 2,112,675
 Tracks/Cylinder 255
 Partition Disk #20, Partition #0
 Partition Size 63.47 GB (68,146,412,544 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 6
 SCSI Target ID 0
 Sectors/Track 63
 Size 34.17 GB (36,692,974,080 bytes)

Total Cylinders 4,461
 Total Sectors 71,665,965
 Total Tracks 1,137,555
 Tracks/Cylinder 255
 Partition Disk #21, Partition #0
 Partition Size 34.17 GB (36,692,941,824 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 6
 SCSI Target ID 0
 Sectors/Track 63
 Size 663.44 GB (712,358,599,680 bytes)
 Total Cylinders 86,606
 Total Sectors 1,391,325,390
 Total Tracks 22,084,530
 Tracks/Cylinder 255
 Partition Disk #22, Partition #0
 Partition Size 663.44 GB (712,358,567,424 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 6
 SCSI Target ID 1
 Sectors/Track 63
 Size 63.47 GB (68,146,444,800 bytes)
 Total Cylinders 8,285
 Total Sectors 133,098,525
 Total Tracks 2,112,675
 Tracks/Cylinder 255
 Partition Disk #23, Partition #0
 Partition Size 63.47 GB (68,146,412,544 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 6

SCSI Target ID 1
 Sectors/Track 63
 Size 34.17 GB (36,692,974,080 bytes)
 Total Cylinders 4,461
 Total Sectors 71,665,965
 Total Tracks 1,137,555
 Tracks/Cylinder 255
 Partition Disk #24, Partition #0
 Partition Size 34.17 GB (36,692,941,824 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 6
 SCSI Target ID 1
 Sectors/Track 63
 Size 663.44 GB (712,358,599,680 bytes)
 Total Cylinders 86,606
 Total Sectors 1,391,325,390
 Total Tracks 22,084,530
 Tracks/Cylinder 255
 Partition Disk #25, Partition #0
 Partition Size 663.44 GB (712,358,567,424 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 3
 SCSI Target ID 0
 Sectors/Track 63
 Size 63.47 GB (68,146,444,800 bytes)
 Total Cylinders 8,285
 Total Sectors 133,098,525
 Total Tracks 2,112,675
 Tracks/Cylinder 255
 Partition Disk #6, Partition #0
 Partition Size 63.47 GB (68,146,412,544 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1


```

SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 3
SCSI Target ID 0
Sectors/Track 63
Size 34.17 GB (36,692,974,080 bytes)
Total Cylinders 4,461
Total Sectors 71,665,965
Total Tracks 1,137,555
Tracks/Cylinder 255
Partition Disk #7, Partition #0
Partition Size 34.17 GB (36,692,941,824 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 3
SCSI Target ID 1
Sectors/Track 63
Size 63.47 GB (68,146,444,800 bytes)
Total Cylinders 8,285
Total Sectors 133,098,525
Total Tracks 2,112,675
Tracks/Cylinder 255
Partition Disk #8, Partition #0
Partition Size 63.47 GB (68,146,412,544 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 3
SCSI Target ID 1
Sectors/Track 63
Size 34.17 GB (36,692,974,080 bytes)
Total Cylinders 4,461
Total Sectors 71,665,965
Total Tracks 1,137,555
Tracks/Cylinder 255
Partition Disk #9, Partition #0
Partition Size 34.17 GB (36,692,941,824 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512

```

```

Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 7
SCSI Target ID 0
Sectors/Track 63
Size 63.47 GB (68,146,444,800 bytes)
Total Cylinders 8,285
Total Sectors 133,098,525
Total Tracks 2,112,675
Tracks/Cylinder 255
Partition Disk #26, Partition #0
Partition Size 63.47 GB (68,146,412,544 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 7
SCSI Target ID 0
Sectors/Track 63
Size 34.17 GB (36,692,974,080 bytes)
Total Cylinders 4,461
Total Sectors 71,665,965
Total Tracks 1,137,555
Tracks/Cylinder 255
Partition Disk #27, Partition #0
Partition Size 34.17 GB (36,692,941,824 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 7
SCSI Target ID 1
Sectors/Track 63
Size 63.47 GB (68,146,444,800 bytes)
Total Cylinders 8,285
Total Sectors 133,098,525
Total Tracks 2,112,675
Tracks/Cylinder 255
Partition Disk #28, Partition #0
Partition Size 63.47 GB (68,146,412,544 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive

```

```

Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 7
SCSI Target ID 1
Sectors/Track 63
Size 34.17 GB (36,692,974,080 bytes)
Total Cylinders 4,461
Total Sectors 71,665,965
Total Tracks 1,137,555
Tracks/Cylinder 255
Partition Disk #29, Partition #0
Partition Size 34.17 GB (36,692,941,824 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 4
SCSI Target ID 0
Sectors/Track 63
Size 63.47 GB (68,146,444,800 bytes)
Total Cylinders 8,285
Total Sectors 133,098,525
Total Tracks 2,112,675
Tracks/Cylinder 255
Partition Disk #10, Partition #0
Partition Size 63.47 GB (68,146,412,544 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 4
SCSI Target ID 0
Sectors/Track 63
Size 34.17 GB (36,692,974,080 bytes)
Total Cylinders 4,461
Total Sectors 71,665,965
Total Tracks 1,137,555
Tracks/Cylinder 255
Partition Disk #11, Partition #0
Partition Size 34.17 GB (36,692,941,824 bytes)

```

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 4
 SCSI Target ID 1
 Sectors/Track 63
 Size 63.47 GB (68,146,444,800 bytes)
 Total Cylinders 8,285
 Total Sectors 133,098,525
 Total Tracks 2,112,675
 Tracks/Cylinder 255
 Partition Disk #12, Partition #0
 Partition Size 63.47 GB (68,146,412,544 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 4
 SCSI Target ID 1
 Sectors/Track 63
 Size 34.17 GB (36,692,974,080 bytes)
 Total Cylinders 4,461
 Total Sectors 71,665,965
 Total Tracks 1,137,555
 Tracks/Cylinder 255
 Partition Disk #13, Partition #0
 Partition Size 34.17 GB (36,692,941,824 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 4
 SCSI Target ID 1
 Sectors/Track 63
 Size 663.44 GB (712,358,599,680 bytes)
 Total Cylinders 86,606
 Total Sectors 1,391,325,390
 Total Tracks 22,084,530
 Tracks/Cylinder 255

Partition Disk #14, Partition #0
 Partition Size 663.44 GB (712,358,567,424 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 2
 SCSI Target ID 0
 Sectors/Track 63
 Size 63.47 GB (68,146,444,800 bytes)
 Total Cylinders 8,285
 Total Sectors 133,098,525
 Total Tracks 2,112,675
 Tracks/Cylinder 255
 Partition Disk #1, Partition #0
 Partition Size 63.47 GB (68,146,412,544 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 2
 SCSI Target ID 0
 Sectors/Track 63
 Size 34.17 GB (36,692,974,080 bytes)
 Total Cylinders 4,461
 Total Sectors 71,665,965
 Total Tracks 1,137,555
 Tracks/Cylinder 255
 Partition Disk #2, Partition #0
 Partition Size 34.17 GB (36,692,941,824 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 2
 SCSI Target ID 0
 Sectors/Track 63
 Size 663.44 GB (712,358,599,680 bytes)
 Total Cylinders 86,606

Total Sectors 1,391,325,390
 Total Tracks 22,084,530
 Tracks/Cylinder 255
 Partition Disk #3, Partition #0
 Partition Size 663.44 GB (712,358,567,424 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 2
 SCSI Target ID 1
 Sectors/Track 63
 Size 63.47 GB (68,146,444,800 bytes)
 Total Cylinders 8,285
 Total Sectors 133,098,525
 Total Tracks 2,112,675
 Tracks/Cylinder 255
 Partition Disk #4, Partition #0
 Partition Size 63.47 GB (68,146,412,544 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 2
 SCSI Target ID 1
 Sectors/Track 63
 Size 34.17 GB (36,692,974,080 bytes)
 Total Cylinders 4,461
 Total Sectors 71,665,965
 Total Tracks 1,137,555
 Tracks/Cylinder 255
 Partition Disk #5, Partition #0
 Partition Size 34.17 GB (36,692,941,824 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP LOGICAL VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 8
 SCSI Target ID 4

Sectors/Track 32
Size 33.89 GB (36,385,505,280 bytes)
Total Cylinders 8,709
Total Sectors 71,065,440
Total Tracks 2,220,795
Tracks/Cylinder 255
Partition Disk #30, Partition #0
Partition Size 33.88 GB (36,381,310,976 bytes)

Partition Starting Offset 16,384 bytes

[SCSI]

Item Value
Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\6&1A6A5B31&0&0000010
I/O Port 0x00005000-0x00006FFF
Memory Address 0xFDBF0000-0xFDBF3FFF
IRQ Channel IRQ 16
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.2.10 Beta 8 (wx64 MM IP), 1.03 MB (1,084,416 bytes), 5/31/2006 3:30 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\6&1A6A5B31&0&01000010
I/O Port 0x00005400-0x000054FF
Memory Address 0xFDBE0000-0xFDBE3FFF
IRQ Channel IRQ 17
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.2.10 Beta 8 (wx64 MM IP), 1.03 MB (1,084,416 bytes), 5/31/2006 3:30 PM)

Name Smart Array P800 Controller (Non-Miniport)

Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_103C&DEV_3230&SUBSYS_3223103C&REV_0
2\6&1225CAAD&0&00080010
Memory Address 0xFDD00000-0xFDD0FFFF
I/O Port 0x00006000-0x00006FFF
Memory Address 0xFDC00000-0xFDC0FFFF
IRQ Channel IRQ 17
Driver c:\windows\system32\drivers\hpqciissb.sys
(5.18.2.64 Build 1 (AMD64) built by: RobertVC, 56.50 KB (57,856 bytes), 6/14/2006 11:09 AM)

Name Smart Array Controller
Manufacturer Compaq
Status OK
PNP Device ID
PCI\VEN_103C&DEV_3230&SUBSYS_3234103C&REV_0
1\4&3028BA6E&0&0018
Memory Address 0xFD900000-0xFD90FFFF
I/O Port 0x00004000-0x00004FFF

Memory Address 0xFD8F0000-0xFD8F0FFF
IRQ Channel IRQ 16
Driver c:\windows\system32\drivers\hpcisss.sys
(5.11.0.64 Build 2 (x86-64) (NT.041115-0204), 30.50 KB (31,232 bytes), 3/25/2005 6:00 AM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\4&2A0E5541&0&0020
I/O Port 0x00007000-0x00007FFF
Memory Address 0xFDEF0000-0xFDEF3FFF
IRQ Channel IRQ 16
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.2.10 Beta 8 (wx64 MM IP), 1.03 MB (1,084,416 bytes), 5/31/2006 3:30 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\4&2A0E5541&0&0120
I/O Port 0x00007400-0x000074FF
Memory Address 0xFDEE0000-0xFDEE3FFF
IRQ Channel IRQ 17
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.2.10 Beta 8 (wx64 MM IP), 1.03 MB (1,084,416 bytes), 5/31/2006 3:30 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\4&1636F220&0&0030
I/O Port 0x00008000-0x00008FFF
Memory Address 0xFDF00000-0xFDF03FFF
IRQ Channel IRQ 16
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.2.10 Beta 8 (wx64 MM IP), 1.03 MB (1,084,416 bytes), 5/31/2006 3:30 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\4&1636F220&0&0130
I/O Port 0x00008400-0x000084FF
Memory Address 0xFDFE0000-0xFDFE3FFF
IRQ Channel IRQ 17
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.2.10 Beta 8 (wx64 MM IP), 1.03 MB (1,084,416 bytes), 5/31/2006 3:30 PM)

Name QLogic Optimizing and Multipath Driver
Manufacturer QLogic
Status Degraded
PNP Device ID ROOT\SCSIADAPTER\0000

Driver c:\windows\system32\drivers\qldirect.sys
(8.01.13 Beta 3 (wx64), 62.00 KB (63,488 bytes), 6/1/2006 7:43 PM)

[IDE]

Item Value
Name Standard Dual Channel PCI IDE Controller
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID
PCI\VEN_8086&DEV_269E&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&F9
I/O Port 0x00000500-0x0000050F
Driver c:\windows\system32\drivers\pciide.sys
(5.2.3790.1830 (srv03_spl_rtm.050324-1447), 6.00 KB (6,144 bytes), 3/25/2005 6:00 AM)

Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID
PCI\VEN_8086&DEV_269E&SUBSYS_31FE103C&REV_0
I/O Port 0x000001F0-0x000001F7
I/O Port 0x000003F6-0x000003F6
Driver c:\windows\system32\drivers\ataapi.sys
(5.2.3790.1830 (srv03_spl_rtm.050324-1447), 145.00 KB (148,480 bytes), 3/25/2005 6:00 AM)

Name Secondary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID
PCI\VEN_8086&DEV_269E&SUBSYS_31FE103C&REV_0
I/O Port 0x00000170-0x00000177
I/O Port 0x00000376-0x00000376
Driver c:\windows\system32\drivers\ataapi.sys
(5.2.3790.1830 (srv03_spl_rtm.050324-1447), 145.00 KB (148,480 bytes), 3/25/2005 6:00 AM)

[Printing]

Name Driver Port Name Server Name

[Problem Devices]

Device	PNP Device ID	Error Code
Base System Device	PCI\VEN_0E11&DEV_B203&SUBSYS_3305103C&REV_0	
	3\4&2014205D&0&20F0	The drivers for this device are not installed.
Base System Device	PCI\VEN_0E11&DEV_B204&SUBSYS_3305103C&REV_0	
	3\4&2014205D&0&22F0	The drivers for this device are not installed.
PCI Device	PCI\VEN_103C&DEV_3302&SUBSYS_3305103C&REV_0	
	0\4&2014205D&0&26F0	The drivers for this device are not installed.

Not Available ACPI\IPI0001\0 The drivers for this device are not installed.

[USB]

Device PNP Device ID
 Standard Universal PCI to USB Host Controller
 PCI\VEN_8086&DEV_2688&SUBSYS_31FE103C&REV_09\3&61AAA01&0&E8
 Standard Universal PCI to USB Host Controller
 PCI\VEN_8086&DEV_2689&SUBSYS_31FE103C&REV_09\3&61AAA01&0&E9
 Standard Universal PCI to USB Host Controller
 PCI\VEN_8086&DEV_268A&SUBSYS_31FE103C&REV_09\3&61AAA01&0&EA
 Standard Universal PCI to USB Host Controller
 PCI\VEN_8086&DEV_268B&SUBSYS_31FE103C&REV_09\3&61AAA01&0&EB
 Standard Enhanced PCI to USB Host Controller
 PCI\VEN_8086&DEV_268C&SUBSYS_31FE103C&REV_09\3&61AAA01&0&EF
 Standard Universal PCI to USB Host Controller
 PCI\VEN_103C&DEV_3300&SUBSYS_3305103C&REV_00\4&2014205D&0&24F0

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started	Start Mode	State	Status	Error Control	Accept Pause
abiosdsk	Abiosdsk	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
acpi	Microsoft ACPI Driver	c:\windows\system32\drivers\acpi.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No
acpiec	ACPIEC	c:\windows\system32\drivers\acpiec.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal	No
adpu160m	adpu160m	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
adpu320	adpu320	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
afd	AFD	c:\windows\system32\drivers\afd.sys	Kernel Driver	Yes	System	Running	OK	Normal	No
aic78u2	aic78u2	Not Available	Kernel Driver	No	Disabled	Stopped	OK		

aic78xx	aic78xx	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
aliide	AliIde	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
amdide	AmdIde	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
arc	arc	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
asynccmac	RAS Asynchronous Media Driver	c:\windows\system32\drivers\asynccmac.sys	Kernel Driver	No	Manual	Stopped	OK	Normal	No
atapi	Standard IDE/ESDI Hard Disk Controller	c:\windows\system32\drivers\atapi.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No
atdisk	Atdisk	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
ati2mtag	ati2mtag	c:\windows\system32\drivers\ati2mtag.sys	Kernel Driver	Yes	Manual	Running	OK	Ignore	No
atmarpc	ATM ARP Client Protocol	c:\windows\system32\drivers\atmarpc.sys	Kernel Driver	No	Manual	Stopped	OK	Normal	No
audstub	Audio Stub Driver	c:\windows\system32\drivers\audstub.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No
b06bdrv	HP Virtual Bus Device	c:\windows\system32\drivers\bxvbda.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No
beep	Beep	c:\windows\system32\drivers\beep.sys	Kernel Driver	Yes	System	Running	OK	Normal	No
cdac15ba	CdaC15BA	c:\windows\system32\drivers\cdac15ba.sys	Kernel Driver	Yes	Auto	Running	OK	Normal	No
cdad10ba	CdaD10BA	c:\windows\system32\drivers\cdad10ba.sys	Kernel Driver	Yes	Auto	Running	OK	Normal	No
cdfs	Cdfs	c:\windows\system32\drivers\cdfs.sys	File System Driver	No	Disabled				

cdrom	CD-ROM Driver	c:\windows\system32\drivers\cdrom.sys	Kernel Driver	No	System	Stopped	OK	Normal	No
changer	Changer	Not Available	Kernel Driver	No	System	Stopped	OK		
clusdisk	Cluster Disk Driver	c:\windows\system32\drivers\clusdisk.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal	No
cmdide	CmdIde	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
cpqccissm	cpqccissm	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
crccdsk	CRC Disk Filter Driver	c:\windows\system32\drivers\crccdsk.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No
dfsdriver	DfsDriver	c:\windows\system32\drivers\dfs.sys	File System Driver	Yes	Boot	Running	OK	Normal	No
disk	Disk Driver	c:\windows\system32\drivers\disk.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No
dmboot	dmboot	c:\windows\system32\drivers\dmboot.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal	No
dmio	Logical Disk Manager Driver	c:\windows\system32\drivers\dmio.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No
dmload	dmload	c:\windows\system32\drivers\dmload.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No
dpti2o	dpti2o	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
elxstor	elxstor	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
fastfat	Fastfat	c:\windows\system32\drivers\fastfat.sys	File System Driver	No	Disabled	Stopped	OK	Normal	No

fdc	Floppy Disk Controller Driver c:\windows\system32\drivers\fdc.sys Kernel Driver Yes Manual Running OK Normal No Yes	i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver c:\windows\system32\drivers\i8042prt.sys Kernel Driver Yes System Running OK Normal No Yes	Kernel Driver Yes Boot Running OK Normal No Yes	
fips	Fips c:\windows\system32\drivers\fips.sys Kernel Driver Yes System Running OK Normal No Yes	iirsp	iirsp Not Available Kernel Driver No Disabled Stopped OK Normal No No	ksthunk	Kernel Streaming WOW64 Thunk Service c:\windows\system32\drivers\ksthunk.sys Kernel Driver Yes Manual Running OK Normal No Yes
flpydisk	Flpydisk c:\windows\system32\drivers\flpydisk.sys Kernel Driver No System Stopped OK Ignore No No	imapi	CD-Burning Filter Driver c:\windows\system32\drivers\imapi.sys Kernel Driver No System Stopped OK Normal No No	l2nd	HP NC370 Multifunction Gigabit Server Adapter c:\windows\system32\drivers\bxnd52a.sys Kernel Driver Yes Manual Running OK Normal No Yes
fltmgr	FltMgr c:\windows\system32\drivers\fltmgr.sys File System Driver Yes Boot Running OK Normal No Yes	intelide	IntelIde Not Available Kernel Driver No Disabled Stopped OK Normal No No	lp6nds35	lp6nds35 Not Available Kernel Driver No Disabled Stopped OK Normal No No
ftdisk	Volume Manager Driver c:\windows\system32\drivers\ftdisk.sys Kernel Driver Yes Boot Running OK Normal No Yes	intelppm	Intel Processor Driver c:\windows\system32\drivers\intelppm.sys Kernel Driver Yes Manual Running OK Normal No Yes	mnmdd	mnmdd c:\windows\system32\drivers\mnmdd.sys Kernel Driver Yes System Running OK Ignore No Yes
gpc	Generic Packet Classifier c:\windows\system32\drivers\msgpc.sys Kernel Driver Yes Manual Running OK Normal No Yes	ip6fw	IPv6 Windows Firewall Driver c:\windows\system32\drivers\ip6fw.sys Kernel Driver No Manual Stopped OK Normal No No	modem	Modem c:\windows\system32\drivers\modem.sys Kernel Driver No Manual Stopped OK Ignore No No
hidusb	Microsoft HID Class Driver c:\windows\system32\drivers\hidusb.sys Kernel Driver Yes Manual Running OK Ignore No Yes	ipfilterdriver	IP Traffic Filter Driver c:\windows\system32\drivers\ipfltdrv.sys Kernel Driver No Manual Stopped OK Normal No No	mouclass	Mouse Class Driver c:\windows\system32\drivers\mouclass.sys Kernel Driver Yes System Running OK Normal No Yes
hpciss	hpciss c:\windows\system32\drivers\hpciss.sys Kernel Driver Yes Boot Running OK Normal No Yes	ipinip	IP in IP Tunnel Driver c:\windows\system32\drivers\ipinip.sys Kernel Driver No Manual Stopped OK Normal No No	mouhid	Mouse HID Driver c:\windows\system32\drivers\mouhid.sys Kernel Driver Yes Manual Running OK Ignore No Yes
hpciss2	HpCISs2 c:\windows\system32\drivers\hpciss2.sys Kernel Driver Yes Boot Running OK Normal No Yes	ipnat	IP Network Address Translator c:\windows\system32\drivers\ipnat.sys Kernel Driver No Manual Stopped OK Normal No No	mountmgr	Mount Point Manager c:\windows\system32\drivers\mountmgr.sys Kernel Driver Yes Boot Running OK Normal No Yes
hpqcissb	Smart Array Controllers Non-Miniport Bus Driver c:\windows\system32\drivers\hpqcissb.sys Kernel Driver Yes Boot Running OK Normal No Yes	ipsec	IPSEC driver c:\windows\system32\drivers\ipsec.sys Kernel Driver Yes System Running OK Normal No Yes	mrraid35x	mrraid35x Not Available Kernel Driver No Disabled Stopped OK Normal No No
hpqcissd	Smart Array Controllers Non-Miniport Disk Driver c:\windows\system32\drivers\hpqcissd.sys Kernel Driver Yes Boot Running OK Normal No Yes	isapnp	PNP ISA/EISA Bus Driver c:\windows\system32\drivers\isapnp.sys Kernel Driver Yes Boot Running OK Critical No Yes	mrxdav	WebDav Client Redirector c:\windows\system32\drivers\mrxdav.sys File System Driver No Manual Stopped OK Normal No No
http	HTTP c:\windows\system32\drivers\http.sys Kernel Driver No Manual Stopped OK Normal No No	kbdclass	Keyboard Class Driver c:\windows\system32\drivers\kbdclass.sys Kernel Driver Yes System Running OK Normal No Yes	mrxsmb	MRXSMB c:\windows\system32\drivers\mrxsmb.sys File System Driver Yes System Running OK Normal No Yes
i2omgmt	i2omgmt Not Available Kernel Driver No System Stopped OK Normal No No	kbdhid	Keyboard HID Driver c:\windows\system32\drivers\kbdhid.sys Kernel Driver Yes System Running OK Ignore No Yes	msfs	Msfs c:\windows\system32\drivers\msfs.sys File System Driver Yes System Running OK Normal No Yes
		ksecdd	KSecDD c:\windows\system32\drivers\ksecdd.sys	mssmbios	Microsoft System Management BIOS Driver c:\windows\system32\drivers\mssmbios.sys Kernel Driver Yes Manual

	Running	OK	Normal	No	Yes		Stopped	OK	Normal	No	No		Running	OK	Normal	No	Yes
mup	Mup c:\windows\system32\drivers\mup.sys File System Driver	Yes	Boot			partmgr	Partition Manager c:\windows\system32\drivers\partmgr.sys Kernel Driver	Yes	Boot			raspppoe	Remote Access PPPOE Driver c:\windows\system32\drivers\raspppoe.sys Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes		Running	OK	Normal	No	Yes		Running	OK	Normal	No	Yes
ndis	NDIS System Driver c:\windows\system32\drivers\ndis.sys Kernel Driver	Yes	Boot			pci	PCI Bus Driver c:\windows\system32\drivers\pci.sys Kernel Driver	Yes	Boot			raspti	Direct Parallel c:\windows\system32\drivers\raspti.sys Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes		Running	OK	Critical	No	Yes		Running	OK	Normal	No	Yes
ndistapi	Remote Access NDIS TAPI Driver c:\windows\system32\drivers\ndistapi.sys Kernel Driver	Yes	Manual			pciide	PCIIde c:\windows\system32\drivers\pciide.sys Kernel Driver	Yes	Boot			rdbss	Rdbss c:\windows\system32\drivers\rdbss.sys File System Driver	Yes	System		
	Running	OK	Normal	No	Yes		Running	OK	Normal	No	Yes		Running	OK	Normal	No	Yes
ndisuio	NDIS Usermode I/O Protocol c:\windows\system32\drivers\ndisuio.sys Kernel Driver	No	Manual			pcmcia	Pcmcia c:\windows\system32\drivers\pcmcia.sys Kernel Driver	No	Disabled			rdpcdd	RDPcDD c:\windows\system32\drivers\rdpcdd.sys Kernel Driver	Yes	System		
	Stopped	OK	Normal	No	No		Stopped	OK	Normal	No	No		Running	OK	Ignore	No	Yes
ndiswan	Remote Access NDIS WAN Driver c:\windows\system32\drivers\ndiswan.sys Kernel Driver	Yes	Manual			pdcomp	PDCOMP Not Available Kernel Driver					rdpdr	Terminal Server Device Redirector Driver c:\windows\system32\drivers\rdpdr.sys Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes		Ignore	No	No				Running	OK	Normal	No	Yes
ndproxy	NDIS Proxy c:\windows\system32\drivers\ndproxy.sys Kernel Driver	Yes	Manual			pdframe	PDFRAME Not Available Kernel Driver					rdpwd	RDPWD c:\windows\system32\drivers\rdpwd.sys Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes		No	Manual	Stopped	OK			Running	OK	Ignore	No	Yes
netbios	NetBIOS Interface c:\windows\system32\drivers\netbios.sys File System Driver	Yes	System			pdreli	PDRELI Not Available Kernel Driver					redbook	Digital CD Audio Playback Filter Driver c:\windows\system32\drivers\redbook.sys Kernel Driver	No	System		
	Running	OK	Normal	No	Yes		Ignore	No	No				Stopped	OK	Normal	No	No
netbt	NetBios over Tcip c:\windows\system32\drivers\netbt.sys Kernel Driver	Yes	System			pdrframe	PDRFRAME Not Available Kernel Driver					secdrv	Security Driver c:\windows\system32\drivers\secdrv.sys Kernel Driver	Yes	Auto		
	Running	OK	Normal	No	Yes		No	Manual	Stopped	OK			Running	OK	Normal	No	Yes
nfrd960	nfrd960 Not Available Kernel Driver					pptpminiport	WAN Miniport (PPTP) c:\windows\system32\drivers\rasppptp.sys Kernel Driver	Yes	Manual			serenum	Serenum Filter Driver c:\windows\system32\drivers\serenum.sys Kernel Driver	Yes	Manual		
	No	Disabled	Stopped	OK			Running	OK	Normal	No	Yes		Running	OK	Normal	No	Yes
npfs	Npfs c:\windows\system32\drivers\npfs.sys File System Driver	Yes	System			ql2300	QLogic Fibre Channel SCSI Miniport Driver (wx64 IP) c:\windows\system32\drivers\ql2300.sys Kernel Driver	Yes	Boot			serial	Serial port driver c:\windows\system32\drivers\serial.sys Kernel Driver	Yes	System		
	Running	OK	Normal	No	Yes		Running	OK	Normal	No	Yes		Running	OK	Ignore	No	Yes
ntfs	Ntfs c:\windows\system32\drivers\ntfs.sys File System Driver	Yes	Disabled			qldirect	qldirect c:\windows\system32\drivers\qldirect.sys Kernel Driver	Yes	Auto			sfloppy	Sfloppy c:\windows\system32\drivers\sfloppy.sys Kernel Driver	No	System		
	Running	OK	Normal	No	Yes		Running	OK	Normal	No	Yes		Stopped	OK	Ignore	No	No
null	Null c:\windows\system32\drivers\null.sys Kernel Driver	Yes	System			rasacd	Remote Access Auto Connection Driver c:\windows\system32\drivers\rasacd.sys Kernel Driver	Yes	System			simbad	Simbad Not Available Kernel Driver				
	Running	OK	Normal	No	Yes		Running	OK	Normal	No	Yes		Normal	No	No	Stopped	OK
parport	Parallel port driver c:\windows\system32\drivers\parport.sys Kernel Driver	No	Manual			rasl2tp	WAN Miniport (L2TP) c:\windows\system32\drivers\rasl2tp.sys Kernel Driver	Yes	Manual			srv	Srv c:\windows\system32\drivers\srv.sys File System Driver	Yes	Manual		

	Running	OK	Normal	No	Yes		Kernel Driver	Yes	Manual			system devices)	machine.inf	Not Available		
swenum	Software Bus Driver						Running	OK	Normal	No	Yes	ROOT\SYSTEM\0001				
	c:\windows\system32\drivers\swenum.sys						usbhub	USB2 Enabled Hub					Plug and Play Software Device Enumerator	No		
	Kernel Driver	Yes	Manual					c:\windows\system32\drivers\usbhub.sys					SYSTEM	5.2.3790.1830	10/1/2002	
	Running	OK	Normal	No	Yes			Kernel Driver	Yes	Manual		(Standard system devices)	machine.inf			
								Running	OK	Normal	No	Not Available	ROOT\SYSTEM\0000			
symc8xx	symc8xx	Not Available		Kernel Driver			usbstor	USB Mass Storage Driver					QLogic Optimizing and Multipath Driver	No		
	No	Disabled	Stopped	OK				c:\windows\system32\drivers\usbstor.sys					SCSIADAPTER	Not Available	Not	
	Normal	No	No					Kernel Driver	Yes	Manual		Available	Not Available	oem6.inf	Not Available	
symmpi	symmpi	Not Available		Kernel Driver				Kernel Driver	No	Manual		Terminal Server Mouse Driver	No	SYSTEM		
	No	Disabled	Stopped	OK				Stopped	OK	Normal	No	5.2.3790.1830	10/1/2002	(Standard		
	Normal	No	No									system devices)	machine.inf	Not Available		
sym_hi	sym_hi	Not Available		Kernel Driver			usbuhci	Microsoft USB Universal Host Controller					ROOT\RDP_MOU\0000			
	No	Disabled	Stopped	OK			Miniport	Driver				Terminal Server Keyboard Driver	No			
	Normal	No	No					c:\windows\system32\drivers\usbuhci.sys					SYSTEM	5.2.3790.1830	10/1/2002	
sym_u3	sym_u3	Not Available		Kernel Driver				Kernel Driver	Yes	Manual		(Standard system devices)	machine.inf			
	No	Disabled	Stopped	OK				Running	OK	Normal	No	Not Available	ROOT\RDP_KBD\0000			
	Normal	No	No									Terminal Server Device Redirector	No			
tcpip	TCP/IP Protocol Driver						vga	vga					SYSTEM	5.2.3790.1830	10/1/2002	
	c:\windows\system32\drivers\tcpip.sys							c:\windows\system32\drivers\vgapnp.sys					(Standard system devices)	machine.inf		
	Kernel Driver	Yes	System					Kernel Driver	No	Manual		Not Available	ROOT\RDPDR\0000			
	Running	OK	Normal	No	Yes			Stopped	OK	Ignore	No	Direct Parallel	No	NET	5.2.3790.1830	
tdpipe	TDPIPE						vgasave	VGA Display Controller.					10/1/2002	Microsoft netrasa.inf	Not	
	c:\windows\system32\drivers\tdpipe.sys							c:\windows\system32\drivers\vga.sys					Available	ROOT\MS_PTMINIPORT\0000		
	Kernel Driver	No	Manual					Kernel Driver	Yes	System		WAN Miniport (PPTP)	No	NET	5.2.3790.1830	
	Stopped	OK	Ignore	No	No			Running	OK	Ignore	No	10/1/2002	Microsoft netrasa.inf	Not		
tdtcp	TDTCP						viaide	ViaIde	Not Available		Kernel Driver	Available	ROOT\MS_PPTMINIPORT\0000			
	c:\windows\system32\drivers\tdtcp.sys							No	Disabled	Stopped	OK	5.2.3790.1830	10/1/2002	Microsoft		
	Kernel Driver	Yes	Manual					Normal	No	No		netrasa.inf	Not Available			
	Running	OK	Ignore	No	Yes		volsnap	Storage volumes					ROOT\MS_PPPOEMINIPORT\0000			
								c:\windows\system32\drivers\volsnap.sys					WAN Miniport (IP)	No	NET	5.2.3790.1830
								Kernel Driver	Yes	Boot		10/1/2002	Microsoft netrasa.inf	Not		
termdd	Terminal Device Driver							Running	OK	Normal	No	Available	ROOT\MS_NDISWANIP\0000			
	c:\windows\system32\drivers\termdd.sys						wanarp	Remote Access IP ARP Driver					WAN Miniport (L2TP)	No	NET	5.2.3790.1830
	Kernel Driver	Yes	System					c:\windows\system32\drivers\wanarp.sys					10/1/2002	Microsoft netrasa.inf	Not	
	Running	OK	Normal	No	Yes			Kernel Driver	Yes	Manual		Available	ROOT\MS_L2TPMINIPORT\0000			
								Running	OK	Normal	No	Video Codecs	No	MEDIA	5.2.3790.1830	
toside	TosIde	Not Available		Kernel Driver				Kernel Driver	Yes	Manual		10/1/2002	(Standard system devices)			
	No	Disabled	Stopped	OK				Running	OK	Normal	No	wave.inf	Not Available			
	Normal	No	No									ROOT\MEDIA\MS_MMVID				
udfs	Udfs						wdica	WDICA	Not Available		Kernel Driver	Legacy Video Capture Devices	No	MEDIA		
	c:\windows\system32\drivers\udfs.sys							No	Manual	Stopped	OK	5.2.3790.1830	10/1/2002	(Standard		
	File System Driver	No	Disabled					Ignore	No	No		system devices)	wave.inf	Not Available		
	Stopped	OK	Normal	No	No		wlbs	Network Load Balancing					ROOT\MEDIA\MS_MMVCD			
								c:\windows\system32\drivers\wlbs.sys					Media Control Devices	No	MEDIA	
ultra	ultra	Not Available		Kernel Driver				Kernel Driver	No	Manual		5.2.3790.1830	10/1/2002	(Standard		
	No	Disabled	Stopped	OK				Stopped	OK	Normal	No	system devices)	wave.inf	Not Available		
	Normal	No	No									ROOT\MEDIA\MS_MMMCI				
update	Microcode Update Driver							[Signed Drivers]					Legacy Audio Drivers	No	MEDIA	
	c:\windows\system32\drivers\update.sys							Device Name	Signed	Device Class		5.2.3790.1830	10/1/2002	(Standard		
	Kernel Driver	Yes	Manual					Driver Version	Driver Date			system devices)	wave.inf	Not Available		
	Running	OK	Normal	No	Yes			Manufacturer	INF Name	Driver Name		ROOT\MEDIA\MS_MMDRV				
usbccgp	Microsoft USB Generic Parent Driver							Device ID				Audio Codecs	No	MEDIA	5.2.3790.1830	
	c:\windows\system32\drivers\usbccgp.sys							Microsoft	System Management	BIOS Driver	No	10/1/2002	(Standard system devices)	machine.inf		
	Kernel Driver	Yes	Manual					SYSTEM	5.2.3790.1830	10/1/2002		Not Available	ROOT\SYSTEM\0002			
	Running	OK	Normal	No	Yes			(Standard system devices)				Remote Access IP ARP Driver	Not Available			
								Not Available	ROOT\SYSTEM\0002			LEGACYDRIVER	Not Available	Not		
usbhci	Microsoft USB 2.0 Enhanced Host Controller							Microcode	Update Device	No	SYSTEM	Available	Not Available	Not Available	Not	
Miniport	Driver							5.2.3790.1830	10/1/2002	(Standard		Available	ROOT\LEGACY_WANARP\0000			
	c:\windows\system32\drivers\usbhci.sys															

volsnap	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_VOLSNAPO\0000			
VGA Display Controller.	Not Available	Not Available	Not
LEGACYDRIVER	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_VGASAVE\0000			
TDTCP	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_TDTCP\0000			
TCP/IP Protocol Driver			
LEGACYDRIVER	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_TCPIP\0000			
Security Driver	Not Available	LEGACYDRIVER	Not
Not Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_SECDRV\0000			
RDPWD	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_RDPWD\0000			
RDP added			
RDP added	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_RDP added\0000			
Remote Access Auto Connection Driver			
LEGACYDRIVER	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_RASACD\0000			
qldirect	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_QLDIRECT\0000			
Partition Manager			
Not Available	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_PARTMGR\0000			
Null	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_NULL\0000			
NetBios over Tcpip			
Not Available	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_NETBT\0000			
NDProxy	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_NDPROXY\0000			
NDIS Usermode I/O Protocol			
LEGACYDRIVER	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_NDISUIO\0000			
Remote Access NDIS TAPI Driver			
LEGACYDRIVER	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_NDIS TAPI\0000			
NDIS System Driver			
Not Available	Not Available	LEGACYDRIVER	Not
Not Available	Not Available	Not Available	Not

Available	Not Available	Not Available	Not
ROOT\LEGACY_NDIS\0000			
mountmgr	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_MOUNTMGR\0000			
mnmdd	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_MNMDD\0000			
ksecdd	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_KSE added\0000			
IPSEC driver			
Not Available	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_IPSEC\0000			
IP Network Address Translator			
LEGACYDRIVER	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_IPNAT\0000			
HpCISSs2	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_HPCISSS2\0000			
Generic Packet Classifier			
LEGACYDRIVER	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_GPC\0000			
Fips	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_FIPS\0000			
dmload	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_DMLOAD\0000			
dmboot			
Not Available	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_DMBOOT\0000			
CRC Disk Filter Driver			
LEGACYDRIVER	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_CRCDISK\0000			
CdaD10BA	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_CDAD10BA\0000			
CdaC15BA	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_CDAC15BA\0000			
Beep	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_BEEP\0000			
AFD			
Not Available	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
ROOT\LEGACY_AFD\0000			

Generic volume	No	VOLUME	5.2.3790.1830
10/1/2002	Microsoft	volume.inf	Not
Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C			
410FFSET7E00LENGTH88B11DC00			
Generic volume	No	VOLUME	5.2.3790.1830
10/1/2002	Microsoft	volume.inf	Not
Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C			
400FFSET7E00LENGTHFDD7BC00			
Generic volume	No	VOLUME	5.2.3790.1830
10/1/2002	Microsoft	volume.inf	Not
Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C			
430FFSET7E00LENGTHFDD7BC00			
Generic volume	No	VOLUME	5.2.3790.1830
10/1/2002	Microsoft	volume.inf	Not
Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C			
2C0FFSET7E00LENGTH88B11DC00			
Generic volume	No	VOLUME	5.2.3790.1830
10/1/2002	Microsoft	volume.inf	Not
Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C			
220FFSET7E00LENGTHFDD7BC00			
Generic volume	No	VOLUME	5.2.3790.1830
10/1/2002	Microsoft	volume.inf	Not
Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C			
560FFSET7E00LENGTH88B11DC00			
Generic volume	No	VOLUME	5.2.3790.1830
10/1/2002	Microsoft	volume.inf	Not
Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C			
490FFSET7E00LENGTHFDD7BC00			
Generic volume	No	VOLUME	5.2.3790.1830
10/1/2002	Microsoft	volume.inf	Not
Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C			
270FFSET7E00LENGTHFDD7BC00			
Generic volume	No	VOLUME	5.2.3790.1830
10/1/2002	Microsoft	volume.inf	Not
Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C			
580FFSET7E00LENGTH88B11DC00			
Generic volume	No	VOLUME	5.2.3790.1830
10/1/2002	Microsoft	volume.inf	Not
Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C			
5B0FFSET7E00LENGTHFDD7BC00			
Generic volume	No	VOLUME	5.2.3790.1830
10/1/2002	Microsoft	volume.inf	Not
Available			

STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
52OFFSET7E00LENGTHA5DBE11E00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
54OFFSET7E00LENGTH88B11DC00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
57OFFSET7E00LENGTHFDDD7BC00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
53OFFSET7E00LENGTHA5DBE11E00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
2E0FFSET7E00LENGTH88B11DC00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
51OFFSET7E00LENGTHFDDD7BC00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC57AC5
7A0FFSET4000LENGTH877EC000
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
5D0FFSET7E00LENGTHA5DBE11E00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
20OFFSET7E00LENGTH88B11DC00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
5A0FFSET7E00LENGTHFDDD7BC00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
5F0FFSET7E00LENGTH88B11DC00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
4B0FFSET7E00LENGTH88B11DC00

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
4E0FFSET7E00LENGTHFDDD7BC00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
4A0FFSET7E00LENGTHA5DBE11E00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
4C0FFSET7E00LENGTH88B11DC00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC4AE4C
4F0FFSET7E00LENGTHFDDD7BC00
Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9DE198
A50FFSET4000LENGTH88AB4B4000
Volume Manager No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ROOT\FTDISK\0000
Logical Disk Manager No SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
ROOT\DMIO\0000
ACPI Fixed Feature Button No SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\FIXEDBUTTON\2&DABA3FF&0
ACPI Thermal Zone No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\THERMALZONE\THM0
Secondary IDE Channel No HDC
5.2.3790.1830 10/1/2002 (Standard IDE
ATA/ATAPI controllers) mshdc.inf Not Available
PCIIDE\IDECHANNEL\4&56E2F28&0&1
Primary IDE Channel No HDC 5.2.3790.1830
10/1/2002 (Standard IDE ATA/ATAPI
controllers) mshdc.inf Not Available
PCIIDE\IDECHANNEL\4&56E2F28&0&0
Standard Dual Channel PCI IDE Controller No
HDC 5.2.3790.1830 10/1/2002
(Standard IDE ATA/ATAPI controllers)
mshdc.inf Not Available
PCI\VEN_8086&DEV_269E&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&F9
Standard floppy disk controller No FDC
5.2.3790.1830 10/1/2002 (Standard
floppy disk controllers) fdc.inf Not Available
ACPI\PNP0700\5&33D3B1FA&0
Communications Port No PORTS 5.2.3790.1830
10/1/2002 (Standard port types)
msports.inf Not Available
ACPI\PNP0501\0

Extended IO Bus No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A06\4&2AA4AD3D&0
PS/2 Compatible Mouse No MOUSE
5.2.3790.1830 10/1/2002 Microsoft
msmouse.inf Not Available
ACPI\PNP0F13\4&2AA4AD3D&0
Standard 101/102-Key or Microsoft Natural PS/2
Keyboard No KEYBOARD 5.2.3790.1830
10/1/2002 (Standard keyboards)
keyboard.inf Not Available
ACPI\PNP0303\4&2AA4AD3D&0
System speaker No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0800\4&2AA4AD3D&0
Direct memory access controller No
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
ACPI\PNP0200\4&2AA4AD3D&0
High precision event timer No SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0103\0
System timer No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0100\4&2AA4AD3D&0
Not Available Not Available Not Available
Not Available Not Available Not Available
Available Not Available Not Available
ACPI\IPI0001\0
Motherboard resources No SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0C02\0
PCI standard ISA bridge No SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_2670&SUBSYS_00000000&REV_0
9\3&61AAA01&0&F8
PCI Device Not Available UNKNOWN
Available Not Available Not Available Not
Available Not Available
PCI\VEN_103C&DEV_3302&SUBSYS_3305103C&REV_0
0\4&2014205D&0&26F0
Generic USB Hub No USB 5.2.3790.1830
10/1/2002 (Generic USB Hub) usb.inf Not
Available USB\VID_03F0&PID_1327\6&18FFBC52&0&2
HID-compliant mouse No MOUSE 5.2.3790.1830
10/1/2002 Microsoft msmouse.inf Not
Available
HID\VID_03F0&PID_1027&MI_01\8&25B103E6&0&00
00
USB Human Interface Device No HIDCLASS
5.2.3790.1830 10/1/2002 (Standard
system devices) input.inf Not Available
USB\VID_03F0&PID_1027&MI_01\7&2CD6FDA9&0&00
01
HID Keyboard Device No KEYBOARD 5.2.3790.1830
10/1/2002 (Standard keyboards)

```

keyboard.inf          Not Available
HID\VID_03F0&PID_1027&MI_00&B&DED77A1&0&000
0
USB Human Interface Device  No      HIDCLASS
5.2.3790.1830          10/1/2002 (Standard
system devices)        input.inf Not Available
USB\VID_03F0&PID_1027&MI_00&7&2CD6FDA9&0&000
00
USB Composite Device    No      USB
5.2.3790.1830          10/1/2002 (Standard USB
Host Controller)      usb.inf Not Available
USB\VID_03F0&PID_1027&6&18FFBC52&0&1
USB Root Hub           No      USB          5.2.3790.1830
10/1/2002 (Standard USB Host Controller)
usbport.inf           Not Available
USB\ROOT_HUB\5&26BC3420&0
Standard Universal PCI to USB Host Controller  No
USB          5.2.3790.1830  10/1/2002
(Standard USB Host Controller)
usbport.inf           Not Available
PCI\VEN_103C&DEV_3300&SUBSYS_3305103C&REV_0
0\4&2014205D&0&24F0
Base System Device    Not Available  UNKNOWN  Not
Available Not Available  Not Available  Not
Available Not Available
PCI\VEN_0E11&DEV_B204&SUBSYS_3305103C&REV_0
3\4&2014205D&0&22F0
Base System Device    Not Available  UNKNOWN  Not
Available Not Available  Not Available  Not
Available Not Available
PCI\VEN_0E11&DEV_B203&SUBSYS_3305103C&REV_0
3\4&2014205D&0&20F0
Default Monitor       No      MONITOR    5.2.3790.1830
10/1/2002 (Standard monitor types)
monitor.inf           Not Available
DISPLAY\DEFAULT_MONITOR\5&E64F3B&0&10000000
&01&03
Plug and Play Monitor No      MONITOR
5.2.3790.1830          10/1/2002 (Standard
monitor types)        monitor.inf Not Available
DISPLAY\AVO0000\5&E64F3B&0&100000081&01&03
ATI ES1000           No      DISPLAY    8.19.4.0
12/6/2005 ATI Technologies Inc.
oem0.inf Not Available
PCI\VEN_1002&DEV_515E&SUBSYS_31FB103C&REV_0
2\4&2014205D&0&18F0
Intel(R) 82801 PCI Bridge - 244E      No
SYSTEM 5.2.3790.1830  10/1/2002
Intel machine.inf     Not Available
PCI\VEN_8086&DEV_244E&SUBSYS_00000000&REV_D
9\3&61AAA01&0&F0
USB Root Hub         No      USB          5.2.3790.1830
10/1/2002 (Standard USB Host Controller)
usbport.inf           Not Available
USB\ROOT_HUB20\4&392538C3&0
Standard Enhanced PCI to USB Host Controller  No
USB          5.2.3790.1830  10/1/2002
(Standard USB Host Controller)
usbport.inf           Not Available
PCI\VEN_8086&DEV_268C&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&EF

```

```

USB Root Hub         No      USB          5.2.3790.1830
10/1/2002 (Standard USB Host Controller)
usbport.inf           Not Available
USB\ROOT_HUB\4&41C0314&0
Standard Universal PCI to USB Host Controller  No
USB          5.2.3790.1830  10/1/2002
(Standard USB Host Controller)
usbport.inf           Not Available
PCI\VEN_8086&DEV_268B&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&EB
USB Root Hub         No      USB          5.2.3790.1830
10/1/2002 (Standard USB Host Controller)
usbport.inf           Not Available
USB\ROOT_HUB\4&A54F890&0
Standard Universal PCI to USB Host Controller  No
USB          5.2.3790.1830  10/1/2002
(Standard USB Host Controller)
usbport.inf           Not Available
PCI\VEN_8086&DEV_268A&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&EA
USB Root Hub         No      USB          5.2.3790.1830
10/1/2002 (Standard USB Host Controller)
usbport.inf           Not Available
USB\ROOT_HUB\4&37897620&0
Standard Universal PCI to USB Host Controller  No
USB          5.2.3790.1830  10/1/2002
(Standard USB Host Controller)
usbport.inf           Not Available
PCI\VEN_8086&DEV_2689&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&E9
USB Root Hub         No      USB          5.2.3790.1830
10/1/2002 (Standard USB Host Controller)
usbport.inf           Not Available
USB\ROOT_HUB\4&7353027&0
Standard Universal PCI to USB Host Controller  No
USB          5.2.3790.1830  10/1/2002
(Standard USB Host Controller)
usbport.inf           Not Available
PCI\VEN_8086&DEV_2688&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&E8
HP NC373i Multifunction Gigabit Server Adapter No
NET          2.6.14.0  4/3/2006 Hewlett-
Packard Company      oem1.inf Not Available
B06BDRV\L2ND&PCI_164C14E4&SUBSYS_7038103C&R
EV_11\6&253A0954&0&20050500
HP NC373i Virtual Bus Device No      SYSTEM
2.6.17.0  4/21/2006 Hewlett-Packard Company
oem4.inf Not Available
PCI\VEN_14E4&DEV_164C&SUBSYS_7038103C&REV_1
1\5&DE7916A&0&0000E1
PCI standard PCI-to-PCI bridge          No
SYSTEM 5.2.3790.1830  10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_1166&DEV_0103&SUBSYS_00000000&REV_C
2\4&110C88BD&0&00E1
PCI standard PCI-to-PCI bridge          No
SYSTEM 5.2.3790.1830  10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_2692&SUBSYS_00000000&REV_0
9\3&61AAA01&0&E1

```

```

HP NC373i Multifunction Gigabit Server Adapter No
NET          2.6.14.0  4/3/2006 Hewlett-
Packard Company      oem1.inf Not Available
B06BDRV\L2ND&PCI_164C14E4&SUBSYS_7038103C&R
EV_11\6&2826E01F&0&20050300
HP NC373i Virtual Bus Device No      SYSTEM
2.6.17.0  4/21/2006 Hewlett-Packard Company
oem4.inf Not Available
PCI\VEN_14E4&DEV_164C&SUBSYS_7038103C&REV_1
1\5&2EADD4B0&0&0000E0
PCI standard PCI-to-PCI bridge          No
SYSTEM 5.2.3790.1830  10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_1166&DEV_0103&SUBSYS_00000000&REV_C
2\4&187919FE&0&00E0
PCI standard PCI-to-PCI bridge          No
SYSTEM 5.2.3790.1830  10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_2690&SUBSYS_00000000&REV_0
9\3&61AAA01&0&E0
PCI standard host CPU bridge No      SYSTEM
5.2.3790.1830  10/1/2002 (Standard
system devices)        machine.inf Not Available
PCI\VEN_8086&DEV_25F6&SUBSYS_00000000&REV_9
3\3&61AAA01&0&B0
PCI standard host CPU bridge No      SYSTEM
5.2.3790.1830  10/1/2002 (Standard
system devices)        machine.inf Not Available
PCI\VEN_8086&DEV_25F5&SUBSYS_00000000&REV_9
3\3&61AAA01&0&A8
PCI standard host CPU bridge No      SYSTEM
5.2.3790.1830  10/1/2002 (Standard
system devices)        machine.inf Not Available
PCI\VEN_8086&DEV_25F3&SUBSYS_00000000&REV_9
3\3&61AAA01&0&98
PCI standard host CPU bridge No      SYSTEM
5.2.3790.1830  10/1/2002 (Standard
system devices)        machine.inf Not Available
PCI\VEN_8086&DEV_25F1&SUBSYS_00000000&REV_9
3\3&61AAA01&0&88
PCI standard host CPU bridge No      SYSTEM
5.2.3790.1830  10/1/2002 (Standard
system devices)        machine.inf Not Available
PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&REV_9
3\3&61AAA01&0&82
PCI standard host CPU bridge No      SYSTEM
5.2.3790.1830  10/1/2002 (Standard
system devices)        machine.inf Not Available
PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&REV_9
3\3&61AAA01&0&80
PCI standard PCI-to-PCI bridge          No
SYSTEM 5.2.3790.1830  10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25E7&SUBSYS_00000000&REV_9
3\3&61AAA01&0&38

```

```

Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&2BDAED3D&0&012
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&2BDAED3D&0&011
HP MSA1000      No      SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsidev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\5&2BDAED3D&0&010
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&2BDAED3D&0&002
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&2BDAED3D&0&001
HP MSA1000      No      SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsidev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\5&2BDAED3D&0&000
QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.2.10 1/31/2006 QLogic oem5.inf Not
Available
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\4&1636F220&0&0130
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&1341C593&0&013
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&1341C593&0&012
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&1341C593&0&011
HP MSA1000      No      SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsidev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\5&1341C593&0&010
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&1341C593&0&002
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available

```

```

SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&1341C593&0&001
HP MSA1000      No      SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsidev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\5&1341C593&0&000
QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.2.10 1/31/2006 QLogic oem5.inf Not
Available
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\4&1636F220&0&0030
PCI standard PCI-to-PCI bridge No
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25F9&SUBSYS_00000000&REV_9
3\3&61AA01&0&30
PCI standard PCI-to-PCI bridge No
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25E5&SUBSYS_00000000&REV_9
3\3&61AA01&0&28
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&3B968143&0&012
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&3B968143&0&011
HP MSA1000      No      SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsidev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\5&3B968143&0&010
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&3B968143&0&002
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&3B968143&0&001
HP MSA1000      No      SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsidev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\5&3B968143&0&000
QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.2.10 1/31/2006 QLogic oem5.inf Not
Available
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\4&2A0E5541&0&0120
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available

```

```

SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&1894DEE6&0&013
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&1894DEE6&0&012
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&1894DEE6&0&011
HP MSA1000      No      SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsidev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\5&1894DEE6&0&010
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&1894DEE6&0&003
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&1894DEE6&0&002
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\5&1894DEE6&0&001
HP MSA1000      No      SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsidev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\5&1894DEE6&0&000
QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.2.10 1/31/2006 QLogic oem5.inf Not
Available
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\4&2A0E5541&0&0020
PCI standard PCI-to-PCI bridge No
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25F8&SUBSYS_00000000&REV_9
3\3&61AA01&0&20
Disk drive      No      DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_HP&PROD_LOGICAL_VOLUME&REV_1.
16\5&2C5E9AB1&0&040
HP Virtual LUN No SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsidev.inf Not
Available
SCSI\OTHER&VEN_COMPAQ&PROD_SCSI_COMMUNICATE
&REV_CISS\5&2C5E9AB1&0&000
Smart Array Controller No SCSIADAPTER
5.2.3790.1830 10/1/2002 Compaq
pnpscsi.inf Not Available
PCI\VEN_103C&DEV_3230&SUBSYS_3234103C&REV_0
1\4&3028BA6E&0&0018

```

```

PCI standard PCI-to-PCI bridge No
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25E3&SUBSYS_00000000&REV_9
3\3&61AAA01&0&18
PCI standard PCI-to-PCI bridge No
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_350C&SUBSYS_00000000&REV_0
1\4&276E5639&0&0310
PCI standard PCI-to-PCI bridge No
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3518&SUBSYS_00000000&REV_0
1\5&36FA8587&0&100010
Smart Array Logical Volume No DISKDRIVE
5.12.2.64 1/23/2005 Hewlett-Packard
oem12.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2
47B7B6&0&0000004000000000
Smart Array P800 Controller (Non-Miniport) No
SCSIADAPTER 5.18.2.64 1/23/2006
Hewlett-Packard oem11.inf Not Available
PCI\VEN_103C&DEV_3230&SUBSYS_3223103C&REV_0
2\6&1225CAD&0&00080010
PCI standard PCI-to-PCI bridge No
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3514&SUBSYS_00000000&REV_0
1\5&36FA8587&0&080010
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&1BF70A9F&0&013
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&1BF70A9F&0&012
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&1BF70A9F&0&011
HP MSA1000 No SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsudev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\7&1BF70A9F&0&010
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&1BF70A9F&0&002
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&1BF70A9F&0&001
HP MSA1000 No SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsudev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\7&1BF70A9F&0&010
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&1BF70A9F&0&002
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available

```

```

SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&1BF70A9F&0&001
HP MSA1000 No SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsudev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\7&1BF70A9F&0&000
QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.2.10 1/31/2006 QLogic oem5.inf Not
Available
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\6&1A6A5B31&0&01000010
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&34EBE3D4&0&012
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&34EBE3D4&0&011
HP MSA1000 No SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsudev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\7&34EBE3D4&0&010
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&34EBE3D4&0&003
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&34EBE3D4&0&002
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&RE
V_4.48\7&34EBE3D4&0&001
HP MSA1000 No SYSTEM 5.2.3790.1830
10/1/2002 Compaq scsudev.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.48
\7&34EBE3D4&0&000
QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.2.10 1/31/2006 QLogic oem5.inf Not
Available
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\6&1A6A5B31&0&00000010
PCI standard PCI-to-PCI bridge No
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3510&SUBSYS_00000000&REV_0
1\5&36FA8587&0&000010
PCI standard PCI-to-PCI bridge No
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available

```

```

PCI\VEN_8086&DEV_3500&SUBSYS_00000000&REV_0
1\4&276E5639&0&0010
PCI standard PCI-to-PCI bridge No
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25E2&SUBSYS_00000000&REV_9
3\3&61AAA01&0&10
PCI standard host CPU bridge No SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25D8&SUBSYS_00000000&REV_9
3\3&61AAA01&0&00
PCI bus No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A03\2&DABA3FF&0
Intel Processor No PROCESSOR 5.2.3790.1830
10/1/2002 Intel cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_15\_7
Intel Processor No PROCESSOR 5.2.3790.1830
10/1/2002 Intel cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_15\_6
Intel Processor No PROCESSOR 5.2.3790.1830
10/1/2002 Intel cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_15\_1
Intel Processor No PROCESSOR 5.2.3790.1830
10/1/2002 Intel cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_15\_0
Microsoft ACPI-Compliant System No
SYSTEM 5.2.3790.1830 10/1/2002
Microsoft acpi.inf Not Available
ACPI_HAL\PNP0C08\0
ACPI Multiprocessor x64-based PC No
COMPUTER 5.2.3790.1830 10/1/2002
(Standard computers) hal.inf Not
Available ROOT\ACPI_HAL\0000
Not Available Not Available Not Available
Not Available Not Available Not Available
Available Not Available Not Available
HTREE\ROOT\0

[Environment Variables]

Variable Value User Name
ClusterLog C:\WINDOWS\Cluster\cluster.log
<SYSTEM>
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
FP_NO_HOST_CHECK NO <SYSTEM>
lib C:\Program Files\SQLXML 4.0\bin\
<SYSTEM>
NUMBER_OF_PROCESSORS 4 <SYSTEM>
OS Windows_NT <SYSTEM>
Path
%SystemRoot%\system32;%SystemRoot%\SystemR
oot%\System32\Wbem;C:\Program Files (x86)\Microsoft
SQL Server\80\Tools\Binn\;C:\Program Files\Microsoft
SQL Server\90\Tools\Binn\;C:\Program Files
(x86)\Microsoft SQL Server\90\Tools\Binn\;C:\Program

```

```

Files (x86)\Microsoft SQL
Server\90\DTSD\Binn\;C:\Program Files (x86)\Microsoft
SQL
Server\90\Tools\Binn\VSShell\Common7\IDE\;C:\Program
Files (x86)\Microsoft Visual Studio
8\Common7\IDE\PrivateAssemblies\;C:\Program Files
(x86)\Compaq\hpadu\Bin <SYSTEM>
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF
.WSH <SYSTEM>
PROCESSOR_ARCHITECTURE AMD64 <SYSTEM>
PROCESSOR_IDENTIFIER EM64T Family 6 Model 15
Stepping 5, GenuineIntel <SYSTEM>
PROCESSOR_LEVEL 6 <SYSTEM>
PROCESSOR_REVISION 0f05 <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
windir %SystemRoot% <SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE
TEMP %USERPROFILE%\Local Settings\Temp
DL380\Administrator
TMP %USERPROFILE%\Local Settings\Temp
DL380\Administrator

[Print Jobs]

Document Size Owner Notify Status
Time Submitted Start Time
Until Time Elapsed Time
Pages Printed Job ID Priority
Parameters Driver Print
Processor Host Print Queue Data Type Name

[Network Connections]

Local Name Remote Name Type
Status User Name

[Running Tasks]

Name Path Process ID Priority Min
Working Set Max Working Set Start Time
Version Size File Date
system idle process Not Available 0 0
Not Available Not Available Not
Available Not Available Not Available Not
Available
system Not Available 4 8 0
1413120 Not Available Not Available
Not Available Not Available
smss.exe Not Available 792 11
204800 1413120 6/16/2006 3:10 PM Not
Available Not Available Not Available
csrss.exe Not Available 920 13 Not
Available Not Available 6/16/2006 3:10 PM Not
Available Not Available Not Available

```

```

winlogon.exe c:\windows\system32\winlogon.exe
996 13 204800 1413120
6/16/2006 3:10 PM 5.2.3790.1830
(srv03_spl_rtm.050324-1447) 901.00 KB (922,624
bytes) 3/25/2005 6:00 AM
services.exe c:\windows\system32\services.exe
216 9 204800 1413120
6/16/2006 3:10 PM 5.2.3790.1830
(srv03_spl_rtm.050324-1447) 216.50 KB (221,696
bytes) 3/25/2005 6:00 AM
lsass.exe c:\windows\system32\lsass.exe 232 9
204800 1413120 6/16/2006 3:10 PM
5.2.3790.1830 (srv03_spl_rtm.050324-1447)
14.00 KB (14,336 bytes) 3/25/2005
6:00 AM
svchost.exe c:\windows\system32\svchost.exe
480 8 204800 1413120
6/16/2006 3:10 PM 5.2.3790.1830
(srv03_spl_rtm.050324-1447) 24.50 KB (25,088 bytes)
3/25/2005 6:00 AM
svchost.exe Not Available Not Available 588 8
Not Available Not Available Not
6/16/2006 3:10 PM Not Available Not
Available Not Available
svchost.exe c:\windows\system32\svchost.exe
700 8 204800 1413120
6/16/2006 3:10 PM 5.2.3790.1830
(srv03_spl_rtm.050324-1447) 24.50 KB (25,088 bytes)
3/25/2005 6:00 AM
msdtc.exe Not Available 1112 8 Not
Available Not Available 6/16/2006 3:10 PM Not
Available Not Available Not Available
svchost.exe c:\windows\system32\svchost.exe
1252 8 204800 1413120
6/16/2006 3:10 PM 5.2.3790.1830
(srv03_spl_rtm.050324-1447) 24.50 KB (25,088 bytes)
3/25/2005 6:00 AM
svchost.exe c:\windows\system32\svchost.exe
1712 8 204800 1413120
6/16/2006 3:10 PM 5.2.3790.1830
(srv03_spl_rtm.050324-1447) 24.50 KB (25,088 bytes)
3/25/2005 6:00 AM
explorer.exe c:\windows\explorer.exe
2016 8 204800 1413120
6/16/2006 3:10 PM 6.00.3790.1830
(srv03_spl_rtm.050324-1447) 1.30 MB (1,364,480
bytes) 3/25/2005 6:00 AM
wmiprvse.exe Not Available 400 8
Not Available Not Available Not
6/16/2006 3:11 PM Not Available Not
Available Not Available
csrss.exe Not Available 912 13 Not
Available Not Available 6/16/2006 6:36 PM Not
Available Not Available
winlogon.exe c:\windows\system32\winlogon.exe
960 13 204800 1413120
6/16/2006 6:36 PM 5.2.3790.1830
(srv03_spl_rtm.050324-1447) 901.00 KB (922,624
bytes) 3/25/2005 6:00 AM
rdpclip.exe c:\windows\system32\rdpclip.exe
560 8 204800 1413120
6/16/2006 6:36 PM 5.2.3790.1830

```

```

(srv03_spl_rtm.050324-1447) 99.00 KB (101,376
bytes) 5/31/2006 2:49 PM
explorer.exe c:\windows\explorer.exe
1340 8 204800 1413120
6/16/2006 6:36 PM 6.00.3790.1830
(srv03_spl_rtm.050324-1447) 1.30 MB (1,364,480
bytes) 3/25/2005 6:00 AM
cmd.exe c:\windows\system32\cmd.exe 1828 8
204800 1413120 6/17/2006 9:32 AM
5.2.3790.1830 (srv03_spl_rtm.050324-1447)
538.50 KB (551,424 bytes) 3/25/2005
6:00 AM
helpctr.exe c:\windows\pchealth\helpctr\binaries\helpct
r.exe 708 8 204800 1413120
6/17/2006 9:37 AM 5.2.3790.1830
(srv03_spl_rtm.050324-1447) 1.30 MB (1,363,456
bytes) 5/31/2006 2:51 PM
helpsvc.exe c:\windows\pchealth\helpctr\binaries\helpsv
c.exe 256 8 204800 1413120
6/17/2006 9:37 AM 5.2.3790.1830
(srv03_spl_rtm.050324-1447) 1.52 MB (1,591,296
bytes) 5/31/2006 2:51 PM
wmiprvse.exe Not Available 512 8
Not Available Not Available
6/17/2006 9:37 AM Not Available Not
Available Not Available

[Loaded Modules]

Name Version Size File Date Manufacturer
Path
winlogon 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
901.00 KB (922,624 bytes) 3/25/2005
Microsoft Corporation
c:\windows\system32\winlogon.exe
ntdll 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
1.20 MB (1,257,472 bytes) 3/25/2005
Microsoft Corporation
c:\windows\system32\ntdll.dll
kernel32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
1.43 MB (1,500,160 bytes) 3/25/2005
Microsoft Corporation
c:\windows\system32\kernel32.dll
advapi32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
1.00 MB (1,051,136 bytes) 3/25/2005
Microsoft Corporation
c:\windows\system32\advapi32.dll
rpcrt4 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
1.63 MB (1,714,176 bytes) 3/25/2005
Microsoft Corporation
c:\windows\system32\rpcrt4.dll
crypt32 5.131.3790.1830 (srv03_spl_rtm.050324-1447)
1.36 MB (1,428,992 bytes) 3/25/2005
Microsoft Corporation
c:\windows\system32\crypt32.dll
msasn1 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
152.50 KB (156,160 bytes) 3/25/2005
Microsoft Corporation
c:\windows\system32\msasn1.dll
msvcrt 7.0.3790.1830 (srv03_spl_rtm.050324-1447)
508.00 KB (520,192 bytes) 3/25/2005

```

6:00 AM Microsoft Corporation
 c:\windows\system32\msvcr7.dll
 user32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 1.04 MB (1,085,952 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\user32.dll
 gdi32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 592.00 KB (606,208 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\gdi32.dll
 nddeapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 25.00 KB (25,600 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\nddeapi.dll
 profmap 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 36.00 KB (36,864 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\profmap.dll
 netapi32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 589.00 KB (603,136 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\netapi32.dll
 userenv 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 1.02 MB (1,069,056 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\userenv.dll
 psapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 29.00 KB (29,696 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\psapi.dll
 regapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 108.50 KB (111,104 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\regapi.dll
 secur32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 120.00 KB (122,880 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\secur32.dll
 setupapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 1.45 MB (1,523,200 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\setupapi.dll
 version 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 28.00 KB (28,672 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\version.dll
 winsta 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 89.00 KB (91,136 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\winsta.dll
 ws2_32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 176.50 KB (180,736 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\ws2_32.dll
 ws2help 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 30.50 KB (31,232 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\ws2help.dll
 msgina 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 1.14 MB (1,193,472 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\msgina.dll

shsvcs 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 193.50 KB (198,144 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\shsvcs.dll
 shlwapi 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 606.50 KB (621,056 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\shlwapi.dll
 sfc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 6.00 KB (6,144 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\sfc.dll
 sfc_os 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 183.50 KB (187,904 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\sfc_os.dll
 wintrust 5.131.3790.1830 (srv03_spl_rtm.050324-1447)
 297.50 KB (304,640 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\wintrust.dll
 imagehlp 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 57.50 KB (58,880 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\imagehlp.dll
 ole32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 2.43 MB (2,543,616 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\ole32.dll
 comctl32 6.0 (srv03_spl_rtm.050324-1447)
 1.51 MB (1,584,128 bytes) 5/31/2006
 9:14 AM Microsoft Corporation
 c:\windows\winsxs\amd64_microsoft.windows.c
 ommon-controls_6595b64144ccf1df_6.0.3790.1830_x-
 ww_aced72af\comctl32.dll
 winscard 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 230.00 KB (235,520 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\winscard.dll
 wtsapi32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 29.00 KB (29,696 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\wtsapi32.dll
 winmm 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 303.50 KB (310,784 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\winmm.dll
 sxs 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 1.91 MB (2,003,968 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\sxs.dll
 shell32 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 10.01 MB (10,492,416 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\shell32.dll
 rsaenh 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 241.96 KB (247,768 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\rsaenh.dll
 widap32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 390.00 KB (399,360 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\widap32.dll

cscd.dll 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 151.50 KB (155,136 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\cscd.dll
 dimntfy 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 28.00 KB (28,672 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\dimntfy.dll
 wlnotify 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 148.00 KB (151,552 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\wlnotify.dll
 mpr 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 115.00 KB (117,760 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\mpr.dll
 oleaut32 5.2.3790.1830 1.06 MB (1,116,160
 bytes) 3/25/2005 6:00 AM Microsoft Corporation
 c:\windows\system32\oleaut32.dll
 winspool 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 247.00 KB (252,928 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\winspool.drv
 comctl32 5.82 (srv03_spl_rtm.050324-1447)
 934.50 KB (956,928 bytes) 5/31/2006
 9:14 AM Microsoft Corporation
 c:\windows\winsxs\amd64_microsoft.windows.c
 ommon-controls_6595b64144ccf1df_5.82.3790.1830_x-
 ww_4d792d2a\comctl32.dll
 uxtheme 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 494.50 KB (506,368 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\uxtheme.dll
 samlib 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 69.00 KB (70,656 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\samlib.dll
 clbcatq 2001.12.4720.1830 (srv03_spl_rtm.050324-
 1447) 865.00 KB (885,760 bytes) 5/31/2006
 Microsoft Corporation
 c:\windows\system32\clbcatq.dll
 comres 2001.12.4720.1830 (srv03_spl_rtm.050324-
 1447) 779.50 KB (798,208 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\comres.dll
 xpsp2res 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 2.77 MB (2,899,456 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\xpsp2res.dll
 cscui 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 441.00 KB (451,584 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\cscui.dll
 ntmarta 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 222.50 KB (227,840 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\ntmarta.dll
 wbemprox 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 38.00 KB (38,912 bytes) 5/31/2006
 Microsoft Corporation
 c:\windows\system32\wbem\wbemprox.dll
 wbemcomn 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 524.00 KB (536,576 bytes) 3/25/2005

6:00 AM Microsoft Corporation
c:\windows\system32\wbem\wbemcomn.dll
wbemsvc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
58.00 KB (59,392 bytes) 5/31/2006
2:49 PM Microsoft Corporation
c:\windows\system32\wbem\wbemsvc.dll
fastprox 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
866.50 KB (887,296 bytes) 5/31/2006
2:49 PM Microsoft Corporation
c:\windows\system32\wbem\fastprox.dll
msvcp60 7.0.3790.1830 (srv03_spl_rtm.050324-1447)
919.50 KB (941,568 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\msvcp60.dll
ntdsapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
127.50 KB (130,560 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\ntdsapi.dll
dnsapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
297.50 KB (304,640 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\dnsapi.dll
services 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
216.50 KB (221,696 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\services.exe
ncobjapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
80.00 KB (81,920 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\ncobjapi.dll
scesrv 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
594.50 KB (608,768 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\scesrv.dll
authz 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
167.00 KB (171,008 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\authz.dll
umpnpgmr 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
205.00 KB (209,920 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\umpnpgmr.dll
eventlog 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
127.00 KB (130,048 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\eventlog.dll
lsass 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
14.00 KB (14,336 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\lsass.exe
lsaasrv 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
1.50 MB (1,568,256 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\lsaasrv.dll
samsvr 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
1.01 MB (1,059,328 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\samsvr.dll
cryptdll 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
47.00 KB (48,128 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\cryptdll.dll

msprivs 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
47.50 KB (48,640 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\msprivs.dll
kerberos 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
698.00 KB (714,752 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\kerberos.dll
msvl_0 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
253.00 KB (259,072 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\msvl_0.dll
iphlpapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
177.00 KB (181,248 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\iphlpapi.dll
netlogon 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
666.00 KB (681,984 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\netlogon.dll
w32time 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
400.50 KB (410,112 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\w32time.dll
schannel 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
248.00 KB (253,952 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\schannel.dll
wdigest 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
130.50 KB (133,632 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\wdigest.dll
rassfm 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
36.00 KB (36,864 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\rassfm.dll
kdcsvc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
409.00 KB (418,816 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\kdcsvc.dll
ntdsa 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
2.81 MB (2,948,096 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\ntdsa.dll
esent 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
2.26 MB (2,366,976 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\esent.dll
ntdsatq 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
51.00 KB (52,224 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\ntdsatq.dll
mswsock 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
478.00 KB (489,472 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\mswsock.dll
scecli 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
308.00 KB (315,392 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\scecli.dll
ws03res 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
794.00 KB (813,056 bytes) 3/25/2005

6:00 AM Microsoft Corporation
c:\windows\system32\ws03res.dll
hnetcfg 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
561.00 KB (574,464 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\hnetcfg.dll
wshtccpip 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
29.00 KB (29,696 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\wshtccpip.dll
pstorsvc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
36.00 KB (36,864 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\pstorsvc.dll
psbase 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
124.00 KB (126,976 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\psbase.dll
dssenh 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
226.96 KB (232,408 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\dssenh.dll
svchost 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
24.50 KB (25,088 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\svchost.exe
rpcss 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
672.00 KB (688,128 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\rpcss.dll
wkssvc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
221.00 KB (226,304 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\wkssvc.dll
wiarpc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
57.00 KB (58,368 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\wiarpc.dll
es 2001.12.4720.1830 (srv03_spl_rtm.050324-1447)
357.00 KB (365,568 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\es.dll
srvsvc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
156.50 KB (160,256 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\srvsvc.dll
wmisvc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
227.00 KB (232,448 bytes) 5/31/2006
2:49 PM Microsoft Corporation
c:\windows\system32\wbem\wmisvc.dll
vssapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
1.26 MB (1,320,960 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\vssapi.dll
atl 3.05.2284.96.50 KB (98,816 bytes)
3/25/2005 6:00 AM Microsoft Corporation
c:\windows\system32\atl.dll
sens 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
63.50 KB (65,024 bytes) 3/25/2005
6:00 AM Microsoft Corporation
c:\windows\system32\sens.dll
browser 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
125.50 KB (128,512 bytes) 3/25/2005

6:00 AM Microsoft Corporation
 c:\windows\system32\browser.dll
 wbemcore 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 1.24 MB (1,299,968 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\wbem\wbemcore.dll
 esscli 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 626.50 KB (641,536 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\wbem\esscli.dll
 comsvcs 2001.12.4720.1830 (srv03_spl_rtm.050324-
 1447) 2.06 MB (2,156,544 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\comsvcs.dll
 wmiutils 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 171.00 KB (175,104 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\wbem\wmiutils.dll
 repdrvfs 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 353.50 KB (361,984 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\wbem\repdrvfs.dll
 wmiprvsd 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 743.00 KB (760,832 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\wbem\wmiprvsd.dll
 wbemess 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 532.50 KB (545,280 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\wbem\wbemess.dll
 nettrap 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 26.00 KB (26,624 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\nettrap.dll
 netman 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 457.00 KB (467,968 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\netman.dll
 mprapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 154.50 KB (158,208 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\mprapi.dll
 activeds 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 348.50 KB (356,864 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\activeds.dll
 adslfdc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 240.50 KB (246,272 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\adslfdc.dll
 credui 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 202.00 KB (206,848 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\credui.dll
 rtutils 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 66.00 KB (67,584 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\rtutils.dll
 netshell 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 2.32 MB (2,437,120 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\netshell.dll

clusapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 127.00 KB (130,048 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\clusapi.dll
 rasapi32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 410.00 KB (419,840 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\rasapi32.dll
 rasman 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 95.50 KB (97,792 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\rasman.dll
 tapi32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 332.50 KB (340,480 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\tapi32.dll
 wininet 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 1.13 MB (1,186,304 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\wininet.dll
 wzcsapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 49.00 KB (50,176 bytes) 3/24/2005
 11:35 AM Microsoft Corporation
 c:\windows\system32\wzcsapi.dll
 wzcsvc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 492.00 KB (503,808 bytes) 3/24/2005
 11:35 AM Microsoft Corporation
 c:\windows\system32\wzcsvc.dll
 wmi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 5.50 KB (5,632 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\wmi.dll
 dhcpcsvc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 219.00 KB (224,256 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\dhcpcsvc.dll
 rasdlg 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 859.50 KB (880,128 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\rasdlg.dll
 ncprov 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 73.00 KB (74,752 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\wbem\ncprov.dll
 pchsvc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 76.00 KB (77,824 bytes) 5/31/2006
 2:51 PM Microsoft Corporation
 c:\windows\pchealth\helpctr\binaries\pchsvc
 .dll
 wbemcons 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 65.50 KB (67,072 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\wbem\wbemcons.dll
 ersvc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 31.00 KB (31,744 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\ersvc.dll
 termsrv 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 354.50 KB (363,008 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\termsrv.dll
 icaapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 27.50 KB (28,160 bytes) 5/31/2006

2:49 PM Microsoft Corporation
 c:\windows\system32\icaapi.dll
 mstlsapi 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 187.00 KB (191,488 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\mstlsapi.dll
 rdpswx 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 170.13 KB (174,216 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\rdpswx.dll
 explorer 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 1.30 MB (1,364,480 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\explorer.exe
 browseui 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 1.53 MB (1,601,536 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\browseui.dll
 shdocvw 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 2.30 MB (2,416,128 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\shdocvw.dll
 cryptui 5.131.3790.1830 (srv03_spl_rtm.050324-1447)
 705.50 KB (722,432 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\cryptui.dll
 apphelp 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 241.00 KB (246,784 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\apphelp.dll
 themeui 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 530.50 KB (543,232 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\themeui.dll
 msimg32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 6.50 KB (6,656 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\msimg32.dll
 actxprxy 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 220.50 KB (225,792 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\actxprxy.dll
 linkinfo 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 30.00 KB (30,720 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\linkinfo.dll
 ntshrui 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 184.00 KB (188,416 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\ntshrui.dll
 msi 3.1.4000.1830 4.27 MB (4,476,416
 bytes) 3/25/2005 6:00 AM Microsoft Corporation
 c:\windows\system32\msi.dll
 webcheck 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 439.00 KB (449,536 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\webcheck.dll
 wsock32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 24.50 KB (25,088 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\wsock32.dll
 stobject 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 142.50 KB (145,920 bytes) 3/25/2005

6:00 AM Microsoft Corporation
 c:\windows\system32\stobject.dll
 batmeter 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 41.50 KB (42,496 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\batmeter.dll
 powrprof 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 32.50 KB (33,280 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\powrprof.dll
 urlmon 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 1.02 MB (1,074,176 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32?urlmon.dll
 drprov 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 24.00 KB (24,576 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\drprov.dll
 ntlanman 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 71.50 KB (73,216 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\ntlanman.dll
 netui0 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 130.00 KB (133,120 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\netui0.dll
 netui1 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 338.50 KB (346,624 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\netui1.dll
 davclnt 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 38.00 KB (38,912 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\davclnt.dll
 rdpsnd 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 25.00 KB (25,600 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\rdpsnd.dll
 scredir 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 38.50 KB (39,424 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\scredir.dll
 msacm32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 31.00 KB (31,744 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\msacm32.drv
 msacm32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 112.00 KB (114,688 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\msacm32.dll
 imaadp32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 24.00 KB (24,576 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\imaadp32.acm
 msadp32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 23.50 KB (24,064 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\msadp32.acm
 msg711 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 13.50 KB (13,824 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\msg711.acm

msgsm32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 34.50 KB (35,328 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\msgsm32.acm
 tssoft32 1.01 13.50 KB (13,824 bytes)
 3/25/2005 6:00 AM DSP GROUP, INC.
 c:\windows\system32\tssoft32.acm
 tsd32 1.03 24.50 KB (25,088 bytes)
 3/25/2005 6:00 AM DSP GROUP, INC.
 c:\windows\system32\tsd32.dll
 rdpcclip 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 99.00 KB (101,376 bytes) 5/31/2006
 2:49 PM Microsoft Corporation
 c:\windows\system32\rdpcclip.exe
 browselc 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 63.00 KB (64,512 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\browselc.dll
 shdoclc 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 589.50 KB (603,648 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\shdoclc.dll
 mlang 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 686.00 KB (702,464 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\mlang.dll
 mydocs 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 101.00 KB (103,424 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\mydocs.dll
 mprui 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 67.50 KB (69,120 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\mprui.dll
 netui2 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 542.00 KB (555,008 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\netui2.dll
 comdlg32 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 446.50 KB (457,216 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\comdlg32.dll
 netmsg 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 179.00 KB (183,296 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\netmsg.dll
 netplwiz 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 938.50 KB (961,024 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\netplwiz.dll
 cmd 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 538.50 KB (551,424 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\cmd.exe
 helpctr 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 1.30 MB (1,363,456 bytes) 5/31/2006
 2:51 PM Microsoft Corporation
 c:\windows\pchealth\helpctr\binaries\helpct
 r.exe
 hcappres 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 7.50 KB (7,680 bytes) 5/31/2006
 2:51 PM Microsoft Corporation

c:\windows\pchealth\helpctr\binaries\hcappre
 es.dll
 itss 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 208.00 KB (212,992 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\itss.dll
 msxml3 8.70.1104.0 2.04 MB (2,141,184
 bytes) 3/25/2005 6:00 AM Microsoft Corporation
 c:\windows\system32\msxml3.dll
 pchshell 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 155.00 KB (158,720 bytes) 5/31/2006
 2:51 PM Microsoft Corporation
 c:\windows\pchealth\helpctr\binaries\pchshe
 ll.dll
 mshtml 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 5.65 MB (5,928,448 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\mshtml.dll
 msls31 3.10.349.0 357.00 KB (365,568
 bytes) 3/25/2005 6:00 AM Microsoft Corporation
 c:\windows\system32\msls31.dll
 msimtf 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 380.50 KB (389,632 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\msimtf.dll
 msctf 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 617.50 KB (632,320 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\msctf.dll
 jscript 5.6.0.8827 974.50 KB (997,888
 bytes) 3/25/2005 6:00 AM Microsoft Corporation
 c:\windows\system32\jscript.dll
 imm32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 208.00 KB (212,992 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\imm32.dll
 mshtml 6.00.3790.1830 (srv03_spl_rtm.050324-1447)
 905.50 KB (927,232 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\mshtml.dll
 vbscript 5.6.0.8827 646.50 KB (662,016
 bytes) 3/25/2005 6:00 AM Microsoft Corporation
 c:\windows\system32\vbscript.dll
 msinfo 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 636.00 KB (651,264 bytes) 5/31/2006
 2:51 PM Microsoft Corporation
 c:\windows\pchealth\helpctr\binaries\msinfo
 .dll
 mfc42u 6.50.9146.0 1.39 MB (1,462,272
 bytes) 3/25/2005 6:00 AM Microsoft Corporation
 c:\windows\system32\mfc42u.dll
 riched32 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 7.00 KB (7,168 bytes) 3/25/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\riched32.dll
 riched20 5.31.23.1224 1.10 MB (1,157,120
 bytes) 3/25/2005 6:00 AM Microsoft Corporation
 c:\windows\system32\riched20.dll
 helpsvc 5.2.3790.1830 (srv03_spl_rtm.050324-1447)
 1.52 MB (1,591,296 bytes) 5/31/2006
 2:51 PM Microsoft Corporation
 c:\windows\pchealth\helpctr\binaries\helpsv
 c.exe

```

[Services]
Display Name      Name      State      Start Mode
Service Type     Path      Error Control
Start Name       Tag ID
Application Experience Lookup Service AeLookupSvc
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Alerter Alerter Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Application Layer Gateway Service ALG
Stopped Manual Own Process
c:\windows\system32\alg.exe Normal NT
AUTHORITY\LocalService 0
Application Management AppMgmt Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
ASP.NET State Service aspnet_state
Stopped Manual Own Process
c:\windows\microsoft.net\framework64\v2.0.5
0727\aspnet_state.exe Normal NT
AUTHORITY\NetworkService 0
Windows Audio AudioSrv Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Background Intelligent Transfer Service BITS
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Computer Browser Browser Running Auto
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Indexing Service CiSvc Stopped Disabled
Share Process
c:\windows\system32\cisvc.exe Normal
LocalSystem 0
ClipBook ClipSrv Stopped Disabled Own Process
c:\windows\system32\clipsrv.exe
Normal LocalSystem 0
.NET Runtime Optimization Service v2.0.50727_X86
clr_optimization_v2.0.50727_32
Stopped Manual Own Process
c:\windows\microsoft.net\framework\v2.0.507
27\mscorsvw.exe Ignore LocalSystem 0
.NET Runtime Optimization Service v2.0.50727_x64
clr_optimization_v2.0.50727_64
Stopped Manual Own Process
c:\windows\microsoft.net\framework64\v2.0.5
0727\mscorsvw.exe Ignore LocalSystem 0
COM+ System Application COMSysApp Stopped
Manual Own Process
c:\windows\system32\dlhhost.exe
/processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}
Normal LocalSystem 0

```

```

Cryptographic Services CryptSvc Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
DCOM Server Process Launcher DcomLaunch
Running Auto Share Process
c:\windows\system32\svchost.exe -k
dcomlaunch Normal LocalSystem 0
Distributed File System Dfs Stopped
Manual Own Process
c:\windows\system32\dfssvc.exe
Normal LocalSystem 0
DHCP Client Dhcp Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k
networkservice Normal NT
AUTHORITY\NetworkService 0
Logical Disk Manager Administrative Service
dmadmin Stopped Manual Share Process
c:\windows\system32\dmadmin.exe /com
Normal LocalSystem 0
Logical Disk Manager dmserver Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
DNS Client Dnscache Stopped Auto
Share Process
c:\windows\system32\svchost.exe -k
networkservice Normal NT
AUTHORITY\NetworkService 0
Error Reporting Service ERSvc Running
Auto Share Process
c:\windows\system32\svchost.exe -k winerr
Ignore LocalSystem 0
Event Log Eventlog Running Auto Share Process
c:\windows\system32\services.exe
Normal LocalSystem 0
COM+ Event System EventSystem Running
Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Help and Support helpsvc Running Manual
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Human Interface Device Access HidServ Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
HTTP SSL HTTPFilter Stopped Manual
Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
IAS Jet Database Access IASJet Stopped
Manual Share Process
c:\windows\syswow64\svchost.exe -k iasjet
Normal LocalSystem 0
IMAPI CD-Burning COM Service ImapiService
Stopped Disabled Own Process
c:\windows\system32\imapi.exe Normal
LocalSystem 0

```

```

Intersite Messaging IsmServ Stopped Disabled Own
Process c:\windows\system32\ismserv.exe
Normal LocalSystem 0
Kerberos Key Distribution Center kdc
Stopped Disabled Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Server lanmanserver Running Auto
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Workstation lanmanworkstation Running
Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
License Logging LicenseService Stopped
Disabled Own Process
c:\windows\system32\llsdrv.exe
Normal NT AUTHORITY\NetworkService 0
TCP/IP NetBIOS Helper LmHosts Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Messenger Messenger Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
NetMeeting Remote Desktop Sharing mnmsrvc
Stopped Disabled Own Process
c:\windows\system32\mnmsrvc.exe
Normal LocalSystem 0
Distributed Transaction Coordinator MSDTC
Running Auto Own Process
c:\windows\system32\msdtc.exe Normal NT
AUTHORITY\NetworkService 0
SQL Server FullText Search (MSSQLSERVER)
msftesql Stopped Disabled Own Process
"c:\program files\microsoft sql
server\mssql.1\mssql\binn\msftesql.exe" -s:mssql.1 -
f:mssqlserver Normal LocalSystem 0
Windows Installer MSIServer Stopped Manual
Share Process
c:\windows\system32\msiexec.exe /v
Normal LocalSystem 0
SQL Server (MSSQLSERVER) MSSQLSERVER
Stopped Manual Own Process
"c:\program files\microsoft sql
server\mssql.1\mssql\binn\sqlservr.exe" -smssqlserver
Normal LocalSystem 0
SQL Server Active Directory Helper
MSSQLServerADHelper Stopped Disabled Own
Process "c:\program files\microsoft sql
server\90\shared\sqladhlp90.exe"
Normal NT
AUTHORITY\NetworkService 0
Network DDE NetDDE Stopped Disabled
Share Process
c:\windows\system32\netdde.exe
Normal LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped
Disabled Share Process

```

```

c:\windows\system32\netdde.exe
Normal LocalSystem 0
Net Logon NetLogon Stopped Manual Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Network Connections Netman Running Manual
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Network Location Awareness (NLA) Nla
Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
File Replication NtFrfs Stopped Manual Own
Process c:\windows\system32\ntfrfs.exe Ignore
LocalSystem 0
NT LM Security Support Provider NtLmSsp
Stopped Manual Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Removable Storage NtmsSvc Stopped Manual
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Office Source Engine ose Stopped
Manual Own Process "c:\program
files (x86)\common files\microsoft shared\source
engine\ose.exe" Normal LocalSystem 0
Plug and Play PlugPlay Running Auto
Share Process
c:\windows\system32\services.exe
Normal LocalSystem 0
IPSEC Services PolicyAgent Stopped
Disabled Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Protected Storage ProtectedStorage Running
Auto Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Remote Access Auto Connection Manager RasAuto
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Remote Access Connection Manager RasMan
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Remote Desktop Help Session Manager RDSessMgr
Stopped Manual Own Process
c:\windows\system32\rsessmgr.exe
Normal LocalSystem 0
Routing and Remote Access RemoteAccess
Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Remote Registry RemoteRegistry Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -k regsvc
Normal NT AUTHORITY\LocalService 0

```

```

Remote Procedure Call (RPC) Locator RpcLocator
Stopped Manual Own Process
c:\windows\system32\locator.exe
Normal NT AUTHORITY\NetworkService 0
Remote Procedure Call (RPC) RpcSs Running
Auto Share Process
c:\windows\system32\svchost.exe -k rpcss
Normal NT AUTHORITY\NetworkService 0
Resultant Set of Policy Provider RSOPProv
Stopped Manual Share Process
c:\windows\system32\rsopprov.exe
Normal LocalSystem 0
Special Administration Console Helper sacsvr
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Security Accounts Manager SamSs Running
Auto Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Smart Card SCardSvr Stopped Manual
Share Process
c:\windows\system32\scardsvr.exe
Ignore NT AUTHORITY\LocalService 0
Task Scheduler Schedule Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Secondary Logon seclogon Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Ignore LocalSystem 0
System Event Notification SENS Stopped
Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Windows Firewall/Internet Connection Sharing (ICS)
SharedAccess Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Shell Hardware Detection ShellHWDetection
Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Ignore LocalSystem 0
Print Spooler Spooler Stopped Disabled Own
Process c:\windows\system32\spoolsv.exe
Normal LocalSystem 0
SQL Server Browser SQLBrowser Stopped
Disabled Own Process "c:\program
files (x86)\microsoft sql
server\90\shared\sqlbrowser.exe"
Normal
LocalSystem 0
SQL Server Agent (MSSQLSERVER)
SQLSERVERAGENT Stopped Manual Own
Process "c:\program files\microsoft sql
server\mssql.1\mssql\bin\sqlagent90.exe" -i
mssqlserver Normal LocalSystem 0

```

```

SQL Server VSS Writer SQLWriter Stopped
Manual Own Process "c:\program
files\microsoft sql server\90\shared\sqlwriter.exe"
Normal LocalSystem 0
Windows Image Acquisition (WIA) stisvc
Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k imgsvc
Normal NT AUTHORITY\LocalService 0
Microsoft Software Shadow Copy Provider swprv
Stopped Manual Own Process
c:\windows\system32\svchost.exe -k swprv
Normal LocalSystem 0
Performance Logs and Alerts SysmonLog Stopped
Auto Own Process
c:\windows\system32\smlogsvc.exe
Normal NT Authority\NetworkService 0
Telephony TapiSrv Stopped Manual Share Process
c:\windows\system32\svchost.exe -k tapisrv
Normal LocalSystem 0
Terminal Services TermService Running
Manual Share Process
c:\windows\system32\svchost.exe -k termvcs
Normal LocalSystem 0
Themes Themes Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Telnet TlntSvr Stopped Disabled Own Process
c:\windows\system32\tlntsvr.exe
Normal NT AUTHORITY\LocalService 0
Distributed Link Tracking Server TrkSvr
Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Distributed Link Tracking Client TrkWks
Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Terminal Services Session Directory Tssdis
Stopped Disabled Own Process
c:\windows\system32\tssdis.exe
Normal LocalSystem 0
Windows User Mode Driver Framework UMWdf
Stopped Manual Own Process
c:\windows\system32\wdftmgr.exe
Normal NT AUTHORITY\LocalService 0
Uninterruptible Power Supply UPS Stopped
Manual Own Process
c:\windows\system32\ups.exe Normal NT
AUTHORITY\LocalService 0
Virtual Disk Service vds Stopped
Manual Own Process
c:\windows\system32\vds.exe Normal
LocalSystem 0
Volume Shadow Copy VSS Stopped Manual Own
Process c:\windows\system32\vssvc.exe Normal
LocalSystem 0
Windows Time W32Time Stopped Auto
Share Process
c:\windows\system32\svchost.exe -k

```

```

localservice Normal NT
AUTHORITY\LocalService 0
WebClient WebClient Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
WinHTTP Web Proxy Auto-Discovery Service
WinHttpAutoProxySvc Stopped Manual
Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Windows Management Instrumentation winmgmt
Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Ignore LocalSystem 0
Portable Media Serial Number Service WmdmPmSN
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Windows Management Instrumentation Driver Extensions
Wmi Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
WMI Performance Adapter WmiApSrv Stopped
Manual Own Process
c:\windows\system32\wbem\wmiapsrv.exe
Normal LocalSystem 0
Automatic Updates wuauerv Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Wireless Configuration WZCSVC Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Network Provisioning Service xmlprov Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
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
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\System Tools All
Users:Accessories\System Tools All Users
Administrative Tools All
Users:Administrative Tools All Users
HP SAN Utilities All Users:HP SAN Utilities All
Users
HP System Tools All Users:HP System Tools All
Users
HP System Tools\HP Array Configuration Utility All
Users:HP System Tools\HP Array Configuration Utility
All Users
HP System Tools\HP Array Configuration Utility CLI
All Users:HP System Tools\HP Array
Configuration Utility CLI All Users
HP System Tools\HP Array Diagnostic Utility All
Users:HP System Tools\HP Array Diagnostic Utility All
Users
Microsoft SQL Server 2005 All Users:Microsoft SQL
Server 2005 All Users
Microsoft SQL Server 2005\Analysis Services All
Users:Microsoft SQL Server 2005\Analysis Services All
Users
Microsoft SQL Server 2005\Configuration Tools All
Users:Microsoft SQL Server 2005\Configuration Tools
All Users
Microsoft SQL Server 2005\Documentation and Tutorials
All Users:Microsoft SQL Server
2005\Documentation and Tutorials All Users
Microsoft SQL Server 2005\Documentation and
Tutorials\Tutorials All Users:Microsoft SQL Server
2005\Documentation and Tutorials\Tutorials All
Users
Microsoft SQL Server 2005\Performance Tools All
Users:Microsoft SQL Server 2005\Performance Tools All
Users
Microsoft Visual Studio 2005 All Users:Microsoft
Visual Studio 2005 All Users
Microsoft Visual Studio 2005\Visual Studio Tools All
Users:Microsoft Visual Studio 2005\Visual Studio
Tools All Users
Startup All Users:Startup All Users
Accessories NT AUTHORITY\SYSTEM:Accessories
NT AUTHORITY\SYSTEM
Accessories\Accessibility NT
AUTHORITY\SYSTEM:Accessories\Accessibility NT
AUTHORITY\SYSTEM
Accessories\Entertainment NT
AUTHORITY\SYSTEM:Accessories\Entertainment NT
AUTHORITY\SYSTEM
Startup NT AUTHORITY\SYSTEM:Startup NT
AUTHORITY\SYSTEM
Accessories DL380\Administrator:Accessories
DL380\Administrator
Accessories\Accessibility
DL380\Administrator:Accessories\Accessibili
ty
DL380\Administrator
Accessories\Entertainment
DL380\Administrator:Accessories\Entertainme
nt
DL380\Administrator
Administrative Tools
DL380\Administrator:Administrative Tools
DL380\Administrator

```

```

HP SAN Utilities DL380\Administrator:HP SAN
Utilities DL380\Administrator
Startup DL380\Administrator:Startup
DL380\Administrator

[Startup Programs]

Program Command User Name Location
desktop desktop.ini NT AUTHORITY\SYSTEM
Startup
desktop desktop.ini DL380\Administrator
Startup
desktop desktop.ini .DEFAULT Startup
desktop desktop.ini All Users Common
Startup

[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
WordPad Document "%programfiles%\windows
nt\accessories\wordpad.exe"
Bitmap Image mspaint.exe

[Windows Error Reporting]

Time Type Details

[Internet Settings]

[Internet Explorer]

[ Following are sub-categories of this main category
]
[Summary]

Item Value
Version 6.0.3790.1830
Build 63790.1830
Application Path C:\Program Files\Internet
Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 128-bit
Content Advisor Disabled
IEAK Install No

[File Versions]

File Version Size Date Path
Company
actxprxy.dll 6.0.3790.1830 221 KB
3/25/2005 7:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation

```

advpack.dll	6.0.3790.1830	146 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
asctrls.ocx	6.0.3790.1830	147 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
browsecl.dll	6.0.3790.1830	63 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
browseui.dll	6.0.3790.1830	1,564 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
cdfview.dll	6.0.3790.1830	216 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
comctl32.dll	5.82.3790.1830	935 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
dxtrans.dll	6.3.3790.1830	320 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
dxtmsft.dll	6.3.3790.1830	549 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
iecont.dll	<File Missing>	Not Available	
	Not Available	Not Available	Not Available
iecontl.dll	<File Missing>	Not Available	
	Not Available	Not Available	Not Available
iedkcs32.dll	16.0.3790.1830	417 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
iepeers.dll	6.0.3790.1830	361 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
iesetup.dll	6.0.3790.1830	71 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
ieuinit.inf	Not Available	24 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Not Available
ieexplore.exe	6.0.3790.1830	94 KB	
	3/25/2005 7:00:00 AM		C:\Program Files\Internet Explorer Microsoft Corporation
imgutil.dll	6.0.3790.1830	61 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
inetcpl.cpl	6.0.3790.1830	428 KB	
	3/25/2005 7:00:00 AM		

	C:\WINDOWS\system32 Microsoft Corporation		
inetcpl.dll	6.0.3790.1830	110 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
inseng.dll	6.0.3790.1830	147 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
mlang.dll	6.0.3790.1830	686 KB	3/25/2005 7:00:00 AM
	Corporation		C:\WINDOWS\system32 Microsoft Corporation
msencode.dll	<File Missing>	Not Available	
	Not Available	Not Available	Not Available
mshta.exe	6.0.3790.1830	38 KB	3/25/2005 7:00:00 AM
	Corporation		C:\WINDOWS\system32 Microsoft Corporation
mshtml.dll	6.0.3790.1830	5,790 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
mshtml.tlb	6.0.3790.1830	1,320 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
mshtml.dll	6.0.3790.1830	906 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
mshtmlr.dll	6.0.3790.1830	56 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
msident.dll	6.0.3790.1830	69 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
msidntld.dll	6.0.3790.1830	16 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
msieftp.dll	6.0.3790.1830	369 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
msrating.dll	6.0.3790.1830	240 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
mstime.dll	6.0.3790.1830	878 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
occache.dll	6.0.3790.1830	126 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
proctexe.ocx	<File Missing>	Not Available	
	Not Available	Not Available	Not Available

sendmail.dll	6.0.3790.1830	64 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
shdoclc.dll	6.0.3790.1830	590 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
shdocvw.dll	6.0.3790.1830	2,360 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
shfolder.dll	6.0.3790.1830	34 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
shlwapi.dll	6.0.3790.1830	607 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
tdc.ocx	1.3.0.3130	91 KB	3/25/2005 7:00:00 AM
	Corporation		C:\WINDOWS\system32 Microsoft Corporation
url.dll	6.0.3790.1830	40 KB	3/25/2005 7:00:00 AM
	Corporation		C:\WINDOWS\system32 Microsoft Corporation
urlmon.dll	6.0.3790.1830	1,049 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
webcheck.dll	6.0.3790.1830	439 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
wininet.dll	6.0.3790.1830	1,159 KB	
	3/25/2005 7:00:00 AM		C:\WINDOWS\system32 Microsoft Corporation
[Connectivity]			
Item	Value		
Connection Preference	Never dial		
LAN Settings			
AutoConfigProxy	wininet.dll		
AutoProxyDetectMode	Disabled		
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\Administrator\Local Settings\Temporary Internet Files
 Total Disk Space Not Available
 Available Disk Space Not Available
 Maximum Cache Size Not Available
 Available Cache Size Not Available

[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
No personal certificate information available				

[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
My Computer	Custom
Local intranet	Custom
Trusted sites	Custom
Internet High	
Restricted sites	Custom

Client Summary

System Information report written at: 06/20/2006 10:03:18 AM
 [System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2195 Service Pack 4 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	CL31
System Manufacturer	HP
System Model	ProLiant DL360 G4
System Type	X86-based PC
Processor x86 Family 15 Model 4 Stepping 1	
GenuineIntel ~3600 Mhz	
Processor x86 Family 15 Model 4 Stepping 1	
GenuineIntel ~3600 Mhz	
Processor x86 Family 15 Model 4 Stepping 1	
GenuineIntel ~3600 Mhz	
Processor x86 Family 15 Model 4 Stepping 1	
GenuineIntel ~3600 Mhz	
BIOS Version	08/16/05
Windows Directory	C:\WINNT
System Directory	C:\WINNT\system32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	CL31\Administrator
Time Zone	Central Daylight Time
Total Physical Memory	1,048,056 KB
Available Physical Memory	788,236 KB
Total Virtual Memory	2,782,676 KB
Available Virtual Memory	2,355,356 KB
Page File Space	1,734,620 KB
Page File	C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
IRQ 16	PCI standard PCI-to-PCI bridge
IRQ 16	PCI standard PCI-to-PCI bridge
IRQ 16	PCI standard PCI-to-PCI bridge
IRQ 16	Standard Universal PCI to USB Host Controller
IRQ 5	Universal Serial Bus (USB) Controller
IRQ 5	Base System Device
IRQ 5	Base System Device

[DMA]

Channel	Device	Status
7	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
0x0000-0x0CF7	PCI bus	OK

0x0000-0x0CF7	Direct memory access controller	OK
0x0D00-0xFFFF	PCI bus	OK
0x4000-0x4FFF	PCI standard PCI-to-PCI bridge	OK
0x4000-0x4FFF	Smart Array 6i	OK
0x2000-0x201F	Standard Universal PCI to USB Host Controller	OK
0x2020-0x203F	Standard Universal PCI to USB Host Controller	OK
0x3000-0x30FF	ATI Technologies Inc. RAGE XL PCI	OK
0x03B0-0x03BB	ATI Technologies Inc. RAGE XL PCI	OK
0x03C0-0x03DF	ATI Technologies Inc. RAGE XL PCI	OK
0x1800-0x18FF	Base System Device	OK
0x3400-0x34FF	Base System Device	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x0274-0x0277	ISAPNP Read Data Port	OK
0x0070-0x0077	Motherboard resources	OK
0x0408-0x040F	Motherboard resources	OK
0x04D0-0x04D1	Motherboard resources	OK
0x0020-0x003F	Motherboard resources	OK
0x00A0-0x00BF	Motherboard resources	OK
0x0090-0x009F	Motherboard resources	OK
0x0050-0x0053	Motherboard resources	OK
0x0700-0x071F	Motherboard resources	OK
0x0800-0x083F	Motherboard resources	OK
0x0900-0x097F	Motherboard resources	OK
0x0010-0x001F	Motherboard resources	OK
0x0C80-0x0C83	Motherboard resources	OK
0x0CD4-0x0CD7	Motherboard resources	OK
0x0F50-0x0F58	Motherboard resources	OK
0x02F8-0x02FF	Motherboard resources	OK
0x0040-0x0043	System timer	OK
0x0080-0x008F	Direct memory access controller	OK
0x00C0-0x00DF	Direct memory access controller	OK
0x0061-0x0061	System speaker	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
0x002E-0x002F	Extended IO Bus	OK
0x004E-0x004F	Extended IO Bus	OK
0x0220-0x025F	Extended IO Bus	OK
0x0280-0x029F	Extended IO Bus	OK
0x03F8-0x03FF	Communications Port (COM1)	OK
0x03F2-0x03F5	Standard floppy disk controller	OK
0x03F7-0x03F7	Standard floppy disk controller	OK
0x0500-0x050F	Standard Dual Channel PCI IDE Controller	OK
0x01F0-0x01F7	Primary IDE Channel	OK
0x03F6-0x03F6	Primary IDE Channel	OK
0x0170-0x0177	Secondary IDE Channel	OK
0x0376-0x0376	Secondary IDE Channel	OK

[IRQs]

```

IRQ Number      Device
9               Microsoft ACPI-Compliant System
16              PCI standard PCI-to-PCI bridge
16              PCI standard PCI-to-PCI bridge
16              PCI standard PCI-to-PCI bridge
16              Standard Universal PCI to USB Host
Controller
24              Smart Array 6i
25              HP NC7782 Gigabit Server Adapter
26              HP NC7782 Gigabit Server Adapter #2
19              Standard Universal PCI to USB Host
Controller
5               Universal Serial Bus (USB) Controller
5               Base System Device
5               Base System Device
1               Standard 101/102-Key or Microsoft Natural
PS/2 Keyboard
12              PS/2 Compatible Mouse
4               Communications Port (COM1)
6               Standard floppy disk controller
14              Primary IDE Channel

[Memory]

Range  Device  Status
0xA0000-0xBFFFF PCI bus OK
0xA0000-0xBFFFF ATI Technologies Inc. RAGE XL PCI
OK
0x40000000-0xFEFFFFFF PCI bus OK
0xFDF00000-0xFDFFFFFFF PCI standard PCI-to-PCI
bridge OK
0xFDF00000-0xFDF1FFFF Smart Array 6i OK
0xFDF80000-0xFDEBFFFF Smart Array 6i OK
0xFDF70000-0xFDF7FFFF HP NC7782 Gigabit
Server Adapter OK
0xFDF60000-0xFDF6FFFF HP NC7782 Gigabit
Server Adapter #2 OK
0xFBEBF0000-0xFBEBF000F Intel(R) 6300ESB
Watchdog Timer - 25AB OK
0xFBEE0000-0xFBEE03FF Universal Serial Bus
(USB) Controller OK
0xFC000000-0xFCFFFFFFF ATI Technologies Inc.
RAGE XL PCI OK
0xFBFF0000-0xFBFF0FFFF ATI Technologies Inc.
RAGE XL PCI OK
0xFBFE0000-0xFBFE01FF Base System Device OK
0xFBFD0000-0xFBFD07FF Base System Device OK
0xFBFC0000-0xFBFC1FFF Base System Device OK
0xFBFB00000-0xFBFB7FFFF Base System Device OK
0xE0000000-0xEFFFFFFF Motherboard resources
OK
0xFBFBFC00-0xFBFBFFFF Standard Dual Channel
PCI IDE Controller 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\iac25_32.ax Intel Corporation
Indeo® audio software OK
C:\WINNT\system32\IAC25_32.AX 2.05.53
195.00 KB (199,680 bytes) 12/7/1999
7:00:00 AM
c:\winnt\system32\lhacm.acm Microsoft Corporation
OK
C:\WINNT\system32\LHACM.ACM 4.4.3385
33.27 KB (34,064 bytes) 9/13/2002
5:46:04 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 7:00:00 AM
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
7:00:00 AM
c:\winnt\system32\imaadp32.acm Microsoft
Corporation OK
C:\WINNT\system32\IMAADP32.ACM
5.00.2195.6612 16.27 KB (16,656 bytes)
8/16/2005 1:51:42 PM
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
7:00:00 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
7:00:00 AM
c:\winnt\system32\msg723.acm Microsoft Corporation
OK
C:\WINNT\system32\MSG723.ACM 4.4.3385
106.77 KB (109,328 bytes) 9/13/2002
5:46:03 PM

[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 7:00:00 AM
c:\winnt\system32\msh263.driv Microsoft Corporation
OK
C:\WINNT\system32\MSH263.DRV 4.4.3385

```

```

252.27 KB (258,320 bytes) 9/13/2002
5:45:39 PM
c:\winnt\system32\msh261.driv Microsoft Corporation
OK
C:\WINNT\system32\MSH261.DRV 4.4.3385
163.77 KB (167,696 bytes) 9/13/2002
5:46:04 PM
c:\winnt\system32\msrle32.dll Microsoft Corporation
OK
C:\WINNT\system32\MSRLE32.DLL
5.00.2195.6612 10.77 KB (11,024 bytes)
8/16/2005 1:51:49 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 7:00:00 AM
c:\winnt\system32\iccvld.dll Radius Inc.
OK C:\WINNT\system32\ICCVLD.DLL
1.10.0.6 108.00 KB (110,592 bytes)
12/7/1999 7:00:00 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
7:00:00 AM

[CD-ROM]

Item Value
Drive D:
Description CD-ROM Drive
Media Loaded False
Media Type CD-ROM
Name COMPAQ CD-ROM SN-124
Manufacturer (Standard CD-ROM drives)
Status OK
Transfer Rate Not Available
SCSI Target ID 0
PNP Device ID IDE\CDROMCOMPAQ_CD-ROM_SN-
124_____N104_____5&180B77CF&0&0.0.0

[Sound Device]

Item Value
No sound devices

[Display]

Item Value
Name ATI Technologies Inc. RAGE XL PCI
PNP Device ID PCI\VEN_1002&DEV_4752&SUBSYS_001E0E11&REV_2
7\4&2183A681&0&18F0
Adapter Type ATI RAGE XL PCI, ATI Technologies
Inc. compatible
Adapter Description ATI Technologies Inc. RAGE XL PCI
Adapter RAM 8.00 MB (8,388,608 bytes)
Installed Drivers atidrab.dll
Driver Version 5.00.2179.1
INF File display.inf (atirage3 section)
Color Planes 1

```

Color Table Entries 65536
Resolution 640 x 480 x 60 hertz
Bits/Pixel 16

[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&1F443D2A&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&1F443D2A&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] RAS Async Adapter
Adapter Type Not Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 6/19/2006 1:48:20 PM
Index 0
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 [00000001] WAN Miniport (L2TP)
Adapter Type Not Available
Product Name WAN Miniport (L2TP)
Installed True
PNP Device ID ROOT\MS_L2TPMINIPOINT\0000
Last Reset 6/19/2006 1:48:20 PM
Index 1
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
(52112, 5.00.2195.6655)

Name [00000002] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Name WAN Miniport (PPTP)
Installed True
PNP Device ID ROOT\MS_PPTPMINIPOINT\0000
Last Reset 6/19/2006 1:48:20 PM
Index 2
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
(48464, 5.00.2195.6711)

Name [00000003] Direct Parallel
Adapter Type Not Available
Product Name Direct Parallel
Installed True
PNP Device ID ROOT\MS_PTMINIPOINT\0000
Last Reset 6/19/2006 1:48:20 PM
Index 3
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 [00000004] WAN Miniport (IP)
Adapter Type Not Available
Product Name WAN Miniport (IP)
Installed True
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 6/19/2006 1:48:20 PM
Index 4
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
(93360, 5.00.2195.6699)

Name [00000005] Compaq NC7780 Gigabit Server
Adapter
Adapter Type Not Available
Product Name Compaq NC7780 Gigabit Server
Adapter
Installed True
PNP Device ID Not Available
Last Reset 6/19/2006 1:48:20 PM
Index 5
Service Name q57w2k
IP Address 130.168.40.31
IP Subnet 255.255.0.0
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server 130.168.253.2
DHCP Lease Expires 9/16/2002 7:03:07 PM
DHCP Lease Obtained 9/15/2002 7:03:07 PM
MAC Address 00:15:60:0C:B0:F3
Service Name Not Available

Name [00000006] Compaq NC7780 Gigabit Server
Adapter
Adapter Type Not Available
Product Name Compaq NC7780 Gigabit Server
Adapter
Installed True
PNP Device ID Not Available
Last Reset 6/19/2006 1:48:20 PM
Index 6
Service Name q57w2k
IP Address 130.168.40.31
IP Subnet 255.255.0.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:15:60:0C:B0:F3
Service Name Not Available

Name [00000007] Compaq NC3123 Fast Ethernet NIC
 Adapter Type Not Available
 Product Name Compaq NC3123 Fast Ethernet NIC
 Installed True
 PNP Device ID Not Available
 Last Reset 6/19/2006 1:48:20 PM
 Index 7
 Service Name N100
 IP Address 130.168.40.31
 IP Subnet 255.255.0.0
 Default IP Gateway Not Available
 DHCP Enabled True
 DHCP Server 130.168.253.2
 DHCP Lease Expires 9/16/2002 3:58:55 PM
 DHCP Lease Obtained 9/15/2002 3:58:55 PM
 MAC Address 00:15:60:0C:B0:F3
 Service Name Not Available

Name [00000008] Compaq NC7781 Gigabit Server
 Adapter
 Adapter Type Not Available
 Product Name Compaq NC7781 Gigabit Server
 Adapter
 Installed True
 PNP Device ID Not Available
 Last Reset 6/19/2006 1:48:20 PM
 Index 8
 Service Name q57w2k
 IP Address 130.168.40.31
 IP Subnet 255.255.0.0
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:15:60:0C:B0:F3
 Service Name Not Available

Name [00000009] Compaq NC7781 Gigabit Server
 Adapter
 Adapter Type Not Available
 Product Name Compaq NC7781 Gigabit Server
 Adapter
 Installed True
 PNP Device ID Not Available
 Last Reset 6/19/2006 1:48:20 PM
 Index 9
 Service Name q57w2k
 IP Address 130.168.40.31
 IP Subnet 255.255.0.0
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:15:60:0C:B0:F3
 Service Name Not Available

Name [00000010] HP NC7782 Gigabit Server Adapter
 Adapter Type Ethernet 802.3
 Product Name HP NC7782 Gigabit Server Adapter
 Installed True

PNP Device ID
 PCI\VEN_14E4&DEV_1648&SUBSYS_00D00E11&REV_1
 0\4&19638ECB&0&10E0
 Last Reset 6/19/2006 1:48:20 PM
 Index 10
 Service Name q57w2k
 IP Address 130.172.11.31
 IP Subnet 255.255.0.0
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:15:60:0C:B0:F4
 Service Name q57w2k
 IRQ Number 25
 Driver c:\winnt\system32\drivers\q57w2k.sys
 (146436, 8.52.0.0)

Name [00000011] HP NC7782 Gigabit Server Adapter
 Adapter Type Ethernet 802.3
 Product Name HP NC7782 Gigabit Server Adapter
 Installed True
 PNP Device ID
 PCI\VEN_14E4&DEV_1648&SUBSYS_00D00E11&REV_1
 0\4&19638ECB&0&11E0
 Last Reset 6/19/2006 1:48:20 PM
 Index 11
 Service Name q57w2k
 IP Address 130.168.40.31
 IP Subnet 255.255.0.0
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:15:60:0C:B0:F3
 Service Name q57w2k
 IRQ Number 26
 Driver c:\winnt\system32\drivers\q57w2k.sys
 (146436, 8.52.0.0)

[Protocol]

Item	Value
Name	MSAFD Tcpip [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 Tcpip [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_Tcpip_{37E5A54E-FF18-486C-B3AD-E80449420A01}}	SETPACKET 8
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_{37E5A54E-FF18-486C-B3AD-E80449420A01}] DATAGRAM 8
 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_{4D85C014-5E76-48CF-93EA-317E0F725486}] SEQPACKET 7
 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_{4D85C014-5E76-48CF-93EA-317E0F725486}] DATAGRAM 7
 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_{2D8AA674-9F13-43EE-9055-F9ECADD87F7F}] SEQPACKET 6
 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_{2D8AA674-9F13-43EE-9055-F9ECADD87F7F}] DATAGRAM 6
 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_{EFD5741D-3A14-456C-98EB-17ABC580A075}] SEQPACKET 5
 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_{EFD5741D-3A14-456C-98EB-17ABC580A075}] DATAGRAM 5
 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_{4249431A-469E-4735-A292-01AA526741FC}] SEQPACKET 4
 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_{4249431A-469E-4735-A292-01AA526741FC}] DATAGRAM 4
 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_{3B09DDB7-7EB8-4941-8121-52DC6359F5A6}] SEQPACKE 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_{3B09DDB7-7EB8-4941-8121-52DC6359F5A6}] 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_{684FA660-D082-4A8C-AC8C-C9D449B21686}] SEQPACKE 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_{684FA660-D082-4A8C-AC8C-C9D449B21686}] 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_{D90E04F2-3AD9-4F98-9464-751E106D7E6A}] SEQPACKE 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_{D90E04F2-3AD9-4F98-9464-751E106D7E6A}] 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_{3F1BA297-E685-416B-82D7-70E771CC8745}] SEQPACKE 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_{3F1BA297-E685-416B-82D7-70E771CC8745}] 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	5.00.2195.6603
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\0
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
(62736, 5.00.2195.6655)

[Parallel]

Item Value
No parallel port information

[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 33.91 GB (36,410,556,416 bytes)
Free Space 30.63 GB (32,887,308,288 bytes)

```

```

Volume Name
Volume Serial Number C8B488FA
Partition Disk #0, Partition #0
Partition Size 33.91 GB (36,410,556,416 bytes)
Starting Offset 16384 bytes
Drive Description Disk drive
Drive Manufacturer (Standard disk drives)
Drive Model HP LOGICAL VOLUME SCSI Disk
Device
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSI Bus 0
Drive SCSI LogicalUnit 0
Drive SCSI Port 2
Drive SCSI Target ID 4
Drive SectorsPerTrack 32
Drive Size 36414750720 bytes
Drive TotalCylinders 8716
Drive TotalSectors 71122560
Drive TotalTracks 2222580
Drive TracksPerCylinder 255

[SCSI]

Item Value
Name Smart Array 6i
Caption Smart Array 6i
Driver cpqcissm
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_40910E11&REV_0
1\4&19638ECB&0&08E0
Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_40910E11&REV_0
1\4&19638ECB&0&08E0
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 24
I/O Port 0x4000-0x4FFF
Driver c:\winnt\system32\drivers\cpqcissm.sys
(16512, 5.64.0.32 Build 7 (x86))

[Printing]

Name Port Name Server Name
No printing information

[Problem Devices]

Device PNP Device ID Error Code
Universal Serial Bus (USB) Controller
PCI\VEN_8086&DEV_25AD&SUBSYS_32010E11&REV_0
2\3&61AAA01&0&EF 28
Base System Device
PCI\VEN_0E11&DEV_B203&SUBSYS_B2060E11&REV_0
1\4&2183A681&0&22F0 28

```

```

Base System Device
PCI\VEN_0E11&DEV_B204&SUBSYS_B2060E11&REV_0
1\4&2183A681&0&22F0 28

[USB]

Device PNP Device ID
Standard Universal PCI to USB Host Controller
PCI\VEN_8086&DEV_25A9&SUBSYS_32010E11&REV_0
2\3&61AAA01&0&E8
USB Root Hub USB\ROOT_HUB\4&312B1C17&0
Standard Universal PCI to USB Host Controller
PCI\VEN_8086&DEV_25AA&SUBSYS_32010E11&REV_0
2\3&61AAA01&0&E9
USB Root Hub USB\ROOT_HUB\4&24B43ADC&0

[Software Environment]

[ Following are sub-categories of this main category
]

[Drivers]

Name Description File Type
Started Start Mode State
Status Error Control Accept Pause
Accept Stop
abiosdsk Abiosdsk Not Available Kernel Driver
False Disabled Stopped OK
Ignore False False
abp480n5 abp480n5 Not Available Kernel Driver
False Disabled Stopped OK
Normal 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 Not Available Kernel Driver
False Disabled Stopped OK
Normal False False
afd AFD Networking Support Environment
c:\winnt\system32\drivers\afd.sys
Kernel Driver True Auto
Running OK Normal False
True
ahal54x Ahal54x 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 Not Available Kernel Driver
False Disabled Stopped OK
Normal False False

```

```

alkernel  Altiris Kernel Driver
c:\winnt\system32\drivers\alkernel.sys
Kernel Driver      True   Manual
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           Disabled  Stopped          OK
Normal          False    False

asc3550   asc3550   Not Available     Kernel Driver
False           Disabled  Stopped          OK
Normal          False    False

asyncmac  RAS Asynchronous Media Driver
c:\winnt\system32\drivers\asyncmac.sys
Kernel Driver      False  Manual
Stopped           OK     Normal  False
False

atapi     Standard IDE/ESDI Hard Disk Controller
c:\winnt\system32\drivers\atapi.sys
Kernel Driver      True   Boot
Running           OK     Normal  False
True

atdisk    Atdisk     Not Available     Kernel Driver
False           Disabled  Stopped          OK
Ignore          False    False

atirage3  atirage3
c:\winnt\system32\drivers\atimpab.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
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

cpqarray  Cpqarray   Not Available     Kernel Driver
False           Disabled  Stopped          OK
Normal          False    False

cpqarray2 cpqarray2  Not Available     Kernel Driver
False           Disabled  Stopped          OK
Normal          False    False

cpqcissm  cpqcissm
c:\winnt\system32\drivers\cpqcissm.sys
Kernel Driver      True   Boot
Running           OK     Normal  False
True

cpqfcalm  cpqfcalm  Not Available     Kernel Driver
False           Disabled  Stopped          OK
Normal          False    False

cpqfws2e  cpqfws2e  Not Available     Kernel Driver
False           Disabled  Stopped          OK
Normal          False    False

dac960nt  dac960nt  Not Available     Kernel Driver
False           Disabled  Stopped          OK
Normal          False    False

deckzpsx  deckzpsx  Not Available     Kernel Driver
False           Disabled  Stopped          OK
Normal          False    False

dfsdriver DfsDriver c:\winnt\system32\drivers\dfs.sys
File System Driver True   Boot
Running           OK     Normal  False
True

disk      Disk Driver
c:\winnt\system32\drivers\disk.sys
Kernel Driver      True   Boot
Running           OK     Normal  False
True

diskperf  Diskperf
c:\winnt\system32\drivers\diskperf.sys
Kernel Driver      True   Boot
Running           OK     Normal  False
True

dmboot    dmboot
c:\winnt\system32\drivers\dmboot.sys
Kernel Driver      False  Disabled
Stopped           OK     Normal  False
False

dmio      Logical Disk Manager Driver
c:\winnt\system32\drivers\dmio.sys
Kernel Driver      True   Boot
Running           OK     Normal  False
True

```

```

dmload    dmload
c:\winnt\system32\drivers\dmload.sys
Kernel Driver      True   Boot
Running           OK     Normal  False
True

efs       EFS        c:\winnt\system32\drivers\efs.sys
File System Driver True   Disabled
Running           OK     Normal  False
True

fastfat   Fastfat
c:\winnt\system32\drivers\fastfat.sys
File System Driver True   Disabled
Running           OK     Normal  False
True

fd16_700  Fd16_700  Not Available     Kernel Driver
False           Disabled  Stopped          OK
Normal          False    False

fdc        Floppy Disk Controller Driver
c:\winnt\system32\drivers\fdc.sys
Kernel Driver      True   Manual
Running           OK     Normal  False
True

fips      Fips
c:\winnt\system32\drivers\fips.sys
Kernel Driver      True   Auto
Running           OK     Normal  False
True

fireport  fireport   Not Available     Kernel Driver
False           Disabled  Stopped          OK
Normal          False    False

flashpnt  flashpnt  Not Available     Kernel Driver
False           Disabled  Stopped          OK
Normal          False    False

flpydisk  Floppy Disk Driver
c:\winnt\system32\drivers\flpydisk.sys
Kernel Driver      True   Manual
Running           OK     Normal  False
True

ftdisk    Volume Manager Driver
c:\winnt\system32\drivers\ftdisk.sys
Kernel Driver      True   Boot
Running           OK     Normal  False
True

gpc       Generic Packet Classifier
c:\winnt\system32\drivers\msgpc.sys
Kernel Driver      True   Manual
Running           OK     Normal  False
True

hidusb    Microsoft HID Class Driver
c:\winnt\system32\drivers\hidusb.sys
Kernel Driver      False  Auto
Stopped           OK     Ignore   False
False

i8042prt  i8042 Keyboard and PS/2 Mouse Port Driver
c:\winnt\system32\drivers\i8042prt.sys
Kernel Driver      True   System
Running           OK     Normal  False
True

ini910u   ini910u   Not Available     Kernel Driver
False           Disabled  Stopped          OK
Normal          False    False

```

```

intelide IntelIde Not Available Kernel Driver
False Disabled Stopped OK
Normal False False
ipfilterdriver IP Traffic Filter Driver
c:\winnt\system32\drivers\ipfltdrv.sys
Kernel Driver False Manual
Stopped OK Normal False
False
ipinip IP in IP Tunnel Driver
c:\winnt\system32\drivers\ipinip.sys
Kernel Driver False Manual
Stopped OK Normal False
False
ipnat IP Network Address Translator
c:\winnt\system32\drivers\ipnat.sys
Kernel Driver False Manual
Stopped OK Normal False
False
ipsec IPSEC driver
c:\winnt\system32\drivers\ipsec.sys
Kernel Driver True Manual
Running OK Normal False
True
ipsraidn ipsraidn Not Available Kernel Driver
False Disabled Stopped OK
Normal False False
irenum IR Enumerator Service
c:\winnt\system32\drivers\irenum.sys
Kernel Driver False Manual
Stopped OK Normal False
False
isapnp PnP ISA/EISA Bus Driver
c:\winnt\system32\drivers\isapnp.sys
Kernel Driver True Boot
Running OK Critical False
True
kbdclass Keyboard Class Driver
c:\winnt\system32\drivers\kbdclass.sys
Kernel Driver True System
Running OK Normal False
True
ksecdd KSecDD
c:\winnt\system32\drivers\ksecdd.sys
Kernel Driver True Boot
Running OK Normal False
True
lbrtfdc lbrtfdc Not Available Kernel Driver
False System Stopped OK
Ignore False False
lp6nds35 lp6nds35 Not Available Kernel Driver
False Disabled Stopped OK
Normal False False
mmnmd mnmdd
c:\winnt\system32\drivers\mmnmd.sys
Kernel Driver True System
Running OK Ignore False
True
modem Modem
c:\winnt\system32\drivers\modem.sys
Kernel Driver False Manual
Stopped OK Ignore False
False

```

```

mouclass Mouse Class Driver
c:\winnt\system32\drivers\mouclass.sys
Kernel Driver True System
Running OK Normal False
True
mouhid Mouse HID Driver
c:\winnt\system32\drivers\mouhid.sys
Kernel Driver False Manual
Stopped OK Ignore False
False
mountmgr MountMgr
c:\winnt\system32\drivers\mountmgr.sys
Kernel Driver True Boot
Running OK Normal False
True
mraid35x mraid35x Not Available Kernel Driver
False Disabled Stopped OK
Normal False False
mrxsmb MRXSMB
c:\winnt\system32\drivers\mrxsmb.sys
File System Driver True System
Running OK Normal False
True
msfs Msfs
c:\winnt\system32\drivers\msfs.sys
File System Driver True System
Running OK Normal False
True
mksksrv Microsoft Streaming Service Proxy
c:\winnt\system32\drivers\mksksrv.sys
Kernel Driver False Manual
Stopped OK Normal False
False
mspclock Microsoft Streaming Clock Proxy
c:\winnt\system32\drivers\mspclock.sys
Kernel Driver False Manual
Stopped OK Normal False
False
mspqm Microsoft Streaming Quality Manager Proxy
c:\winnt\system32\drivers\mspqm.sys
Kernel Driver False Manual
Stopped OK Normal False
False
mup Mup c:\winnt\system32\drivers\mup.sys
File System Driver True Boot
Running OK Normal False
True
n100 Compaq Ethernet or Fast Ethernet NIC NT
Driver
c:\winnt\system32\drivers\n100nt5.sys
Kernel Driver False Manual
Stopped OK Normal False
False
ncrc710 Ncrc710 Not Available Kernel Driver
False Disabled Stopped OK
Normal False False
ndis NDIS System Driver
c:\winnt\system32\drivers\ndis.sys
Kernel Driver True Boot
Running OK Normal False
True
ndistapi Remote Access NDIS TAPI Driver
c:\winnt\system32\drivers\ndistapi.sys
Kernel Driver True Manual

```

```

Running OK Normal False
True
ndisuiio NDIS Usermode I/O Protocol
c:\winnt\system32\drivers\ndisuiio.sys
Kernel Driver False Manual
Stopped OK Normal False
False
ndiswan Remote Access NDIS WAN Driver
c:\winnt\system32\drivers\ndiswan.sys
Kernel Driver True Manual
Running OK Normal False
True
ndproxy NDIS Proxy
c:\winnt\system32\drivers\ndproxy.sys
Kernel Driver True Manual
Running OK Normal False
True
netbios NetBIOS Interface
c:\winnt\system32\drivers\netbios.sys
File System Driver True System
Running OK Normal False
True
netbt NetBios over Tcpip
c:\winnt\system32\drivers\netbt.sys
Kernel Driver True System
Running OK Normal False
True
netdetect NetDetect
c:\winnt\system32\drivers\netdect.sys
Kernel Driver False Manual
Stopped OK Normal False
False
npfs Npfs
c:\winnt\system32\drivers\npfs.sys
File System Driver True System
Running OK Normal False
True
ntfs Ntfs
c:\winnt\system32\drivers\ntfs.sys
File System Driver True Disabled
Running OK Normal False
True
null Null
c:\winnt\system32\drivers\null.sys
Kernel Driver True System
Running OK Normal False
True
nwlnkflt IPX Traffic Filter Driver
c:\winnt\system32\drivers\nwlnkflt.sys
Kernel Driver False Manual
Stopped OK Normal False
False
nwlnkfwf IPX Traffic Forwarder Driver
c:\winnt\system32\drivers\nwlnkfwf.sys
Kernel Driver False Manual
Stopped OK Normal False
False
openhci Microsoft USB Open Host Controller Driver
c:\winnt\system32\drivers\openhci.sys
Kernel Driver False Manual
Stopped OK Normal False
False

```

paralle1	Parallel c:\winnt\system32\drivers\paralle1.sys Kernel Driver False Auto Stopped OK Ignore False False	Running OK Normal False True				Running OK Ignore False True
parport	Parport c:\winnt\system32\drivers\parport.sys Kernel Driver False Auto Stopped OK Ignore False False	q11080 q11080 Not Available Kernel Driver False Disabled Stopped OK Normal False False				sfloppy c:\winnt\system32\drivers\sfloppy.sys Kernel Driver False System Stopped OK Ignore False False
partmgr	PartMgr c:\winnt\system32\drivers\partmgr.sys Kernel Driver True Boot Running OK Normal False True	q110wnt q110wnt Not Available Kernel Driver False Disabled Stopped OK Normal False False				sglfb sglfb Not Available Kernel Driver False System Stopped OK Normal False False
parvdm	ParVdm c:\winnt\system32\drivers\parvdm.sys Kernel Driver False Auto Stopped OK Ignore False False	q11240 q11240 Not Available Kernel Driver False Disabled Stopped OK Normal False False				simbad Simbad Not Available Kernel Driver False Disabled Stopped OK Normal False False
pci	PCI Bus Driver c:\winnt\system32\drivers\pci.sys Kernel Driver True Boot Running OK Critical False True	q12100 q12100 Not Available Kernel Driver False Disabled Stopped OK Normal False False				sparrow Sparrow Not Available Kernel Driver False Disabled Stopped OK Normal False False
pcidump	PCIDump Not Available Kernel Driver False System Stopped OK Ignore False False	rasacd Remote Access Auto Connection Driver c:\winnt\system32\drivers\rasacd.sys Kernel Driver True System Running OK Normal False True				spud Special Purpose Utility Driver c:\winnt\system32\drivers\spud.sys Kernel Driver True Manual Running OK Normal False True
pciide	PCIide c:\winnt\system32\drivers\pciide.sys Kernel Driver True Boot Running OK Normal False True	rasl2tp WAN Miniport (L2TP) c:\winnt\system32\drivers\rasl2tp.sys Kernel Driver True Manual Running OK Normal False True				srv Srv c:\winnt\system32\drivers\srv.sys File System Driver True Manual Running OK Normal False True
pcmcia	Pcmcia c:\winnt\system32\drivers\pcmcia.sys Kernel Driver False Disabled Stopped OK Normal False False	raspti Direct Parallel c:\winnt\system32\drivers\raspti.sys Kernel 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
pdcomp	PDCOMP Not Available Kernel Driver False Manual Stopped OK Ignore False False	rca Microsoft Streaming Network Raw Channel c:\winnt\system32\drivers\rca.sys Kernel Driver False Manual Stopped OK Normal False False				symc810 symc810 Not Available Kernel Driver False Disabled Stopped OK Normal False False
pdframe	PDFRAME Not Available Kernel Driver False Manual Stopped OK Ignore False False	rdbss Rdbss c:\winnt\system32\drivers\rdbss.sys File System Driver True System Running OK Normal False True				symc8xx symc8xx Not Available Kernel Driver False Disabled Stopped OK Normal False False
pdreli	PDRELI Not Available Kernel Driver False Manual Stopped OK Ignore False False	rdpdr Terminal Server Device Redirector Driver c:\winnt\system32\drivers\rdpdr.sys Kernel Driver True Manual Running OK Normal False True				sym_hi sym_hi Not Available Kernel Driver False Disabled Stopped OK Normal False False
pdrframe	PDRFRAME Not Available Kernel Driver False Manual Stopped OK Ignore False False	rdpwd RDPWD c:\winnt\system32\drivers\rdpwd.sys Kernel Driver True Manual Running OK Ignore False True				tcpip TCP/IP Protocol Driver c:\winnt\system32\drivers\tcpip.sys Kernel Driver True System Running OK Normal False True
pptpminiport	WAN Miniport (PPTP) c:\winnt\system32\drivers\rasppptp.sys Kernel Driver True Manual Running OK Normal False True	redbook Digital CD Audio Playback Filter Driver c:\winnt\system32\drivers\redbook.sys Kernel Driver False System Stopped OK Normal False False				tdasync TDASYNC c:\winnt\system32\drivers\tdasync.sys Kernel Driver False Manual Stopped OK Ignore False False
ptilink	Direct Parallel Link Driver c:\winnt\system32\drivers\ptilink.sys Kernel Driver True Manual Running OK Normal False True	serenum Serenum Filter Driver c:\winnt\system32\drivers\serenum.sys Kernel Driver True Manual Running OK Normal False True				tdnetb TDNETB c:\winnt\system32\drivers\tdnetb.sys Kernel Driver False Manual Stopped OK Ignore False False
q57w2k	HP NC7782 Gigabit Server Adapter c:\winnt\system32\drivers\q57w2k.sys Kernel Driver True Manual	serial Serial port driver c:\winnt\system32\drivers\serial.sys Kernel Driver True System				tdpipe TDPIPE c:\winnt\system32\drivers\tdpipe.sys Kernel Driver False Manual

```

Stopped OK Ignore False
False
tdspcx TDSXP
c:\winnt\system32\drivers\tdspcx.sys
Kernel Driver False Manual
Stopped OK Ignore False
False
tdtcp TDTCP
c:\winnt\system32\drivers\tdtcp.sys
Kernel Driver True Manual
Running OK Ignore False
True
termdd Terminal Device Driver
c:\winnt\system32\drivers\termdd.sys
Kernel Driver True Auto
Running OK Normal False
True
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
uhcd Microsoft USB Universal Host Controller
Driver c:\winnt\system32\drivers\uhcd.sys
Kernel Driver True Manual
Running OK Normal False
True
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>
Os2LibPath %SystemRoot%\system32\os2\dll;
<SYSTEM>

```

```

Path
%SystemRoot%\system32;%SystemRoot%;%SystemR
oot%\System32\Wbem;C:\Program Files\Microsoft SQL
Server\90\Tools\bin\ <SYSTEM>
windir %SystemRoot% <SYSTEM>
OS Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_LEVEL 15 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 15 Model 4
Stepping 1, GenuineIntel <SYSTEM>
PROCESSOR_REVISION 0401 <SYSTEM>
NUMBER_OF_PROCESSORS 4 <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF
;.WSH <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
lib C:\Program Files\SQLXML 4.0\bin\
<SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp
CL31\Administrator
TMP %USERPROFILE%\Local Settings\Temp
CL31\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
Parameters Driver Name
Print Processor Host Print Queue
Data Type Name

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 Start Time
Version Size File Date
system idle process Not Available 0 0
Not Available Not Available Not
Available Unknown Unknown
system Not Available 8 8 0
1413120 Not Available Unknown
Unknown Unknown

```

```

smss.exe c:\winnt\system32\smss.exe 192 11
204800 1413120 6/19/2006 6:48:22 PM
5.00.2195.6601 44.77 KB (45,840 bytes)
12/7/1999 7:00:00 AM
csrss.exe Not Available 216 13 Not
Available Not Available 6/19/2006 6:48:24 PM
Unknown Unknown Unknown
winlogon.exe c:\winnt\system32\winlogon.exe
212 13 204800 1413120
6/19/2006 6:48:25 PM
5.00.2195.6714 176.77 KB (181,008
bytes) 8/16/2005 1:52:00 PM
services.exe c:\winnt\system32\services.exe
268 9 204800 1413120
6/19/2006 6:48:25 PM
5.00.2195.6700 87.27 KB (89,360 bytes)
12/7/1999 7:00:00 AM
lsass.exe c:\winnt\system32\lsass.exe 280 9
204800 1413120 6/19/2006 6:48:25 PM
5.00.2195.6695 32.77 KB (33,552 bytes)
12/7/1999 7:00:00 AM
termsrv.exe c:\winnt\system32\termsrv.exe 388
10 204800 1413120 6/19/2006
6:48:25 PM 5.00.2195.6696 139.27 KB
(142,608 bytes) 8/16/2005 1:51:58 PM
aclnt.exe c:\program
files\altiris\aclnt\aclnt.exe 460 8
204800 1413120 6/19/2006 6:48:26 PM
6.1.401 4.63 MB (4,857,932 bytes)
6/5/2003 1:55:46 PM
regsvcs.exe c:\winnt\system32\regsvcs.exe 488
8 204800 1413120 6/19/2006
6:48:26 PM 5.00.2195.6701 66.77 KB
(68,368 bytes) 8/16/2005 1:51:54 PM
rsys.exe c:\benchcraft\rsys.exe 512 8
204800 1413120 6/19/2006 6:48:27 PM
Not Available 32.00 KB (32,768 bytes)
9/13/2002 6:30:57 PM
svchost.exe c:\winnt\system32\svchost.exe 520
8 204800 1413120 6/19/2006
6:48:27 PM 5.00.2134.1 7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM
svchost.exe c:\winnt\system32\svchost.exe 556
8 204800 1413120 6/19/2006
6:48:27 PM 5.00.2134.1 7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM
svchost.exe c:\winnt\system32\svchost.exe 664
8 204800 1413120 6/19/2006
6:48:33 PM 5.00.2134.1 7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM
mstask.exe c:\winnt\system32\mstask.exe 696
8 204800 1413120 6/19/2006
6:48:33 PM 4.71.2195.6704 116.77 KB
(119,568 bytes) 8/16/2005 1:51:49 PM
winmgmt.exe c:\winnt\system32\wbem\winmgmt.exe 744
8 204800 1413120 6/19/2006
6:48:33 PM 1.50.1085.0100 192.10 KB
(196,706 bytes) 8/16/2005 1:52:03 PM
inetinfo.exe c:\winnt\system32\inetrv\inetinfo.exe 760
8 204800 1413120 6/19/2006

```



```

6:48:34 PM      5.00.0984 14.27 KB (14,608 bytes)
8/16/2005 1:52:12 PM
dfssvc.exe      c:\winnt\system32\dfssvc.exe 884
8      204800 1413120 6/19/2006
6:48:36 PM      5.00.2195.6664 88.77 KB
(90,896 bytes) 8/16/2005 1:51:38 PM
svchost.exe    c:\winnt\system32\svchost.exe
1092      8      204800 1413120
6/19/2006 6:48:49 PM      5.00.2134.1
7.77 KB (7,952 bytes) 12/7/1999
7:00:00 AM
logon.scr      c:\winnt\system32\logon.scr 652 4
204800 1413120 6/19/2006 7:03:36 PM
5.00.2195.6601 127.77 KB (130,832
bytes) 8/16/2005 1:51:44 PM
dllhost.exe    Not Available Not Available 892 8
Not Available Not Available
6/19/2006 7:36:01 PM      Unknown
Unknown
csrss.exe      Not Available 848 13 Not
Available Not Available 6/20/2006 8:57:17 AM
Unknown Unknown Unknown
winlogon.exe   c:\winnt\system32\winlogon.exe
1120 13 204800 1413120
6/20/2006 8:57:17 AM
5.00.2195.6714 176.77 KB (181,008
bytes) 8/16/2005 1:52:00 PM
rdpclip.exe   c:\winnt\system32\rdpclip.exe
2100 8 204800 1413120
6/20/2006 8:57:20 AM      5.00.2174.1
39.77 KB (40,720 bytes) 9/13/2002
5:45:10 PM
explorer.exe   c:\winnt\explorer.exe
1180 8 204800 1413120
6/20/2006 8:57:20 AM
5.00.3700.6690 237.77 KB (243,472
bytes) 8/16/2005 1:52:01 PM
aclntusr.exe  c:\program
files\altiris\aclnt\aclntusr.exe 3756 8
204800 1413120 6/20/2006 8:57:21 AM
6, 1, 401 180.00 KB (184,320 bytes)
6/5/2003 1:55:47 PM
tardis.exe     c:\program files\tardis 2000
v1.4\tardis.exe 1648 8 204800
1413120 6/20/2006 8:57:21 AM 5,
0, 1, 4 308.00 KB (315,392 bytes) 9/13/2002
6:21:25 PM
mmc.exe        c:\winnt\system32\mmc.exe 6380 8
204800 1413120 6/20/2006 10:02:25 AM
5.00.2195.6601 589.27 KB (603,408
bytes) 8/16/2005 1:51:45 PM
rsvp.exe      c:\winnt\system32\rsvp.exe 2872 8
204800 1413120 6/20/2006 10:03:07 AM
5.00.2195.6663 172.77 KB (176,912
bytes) 8/16/2005 1:51:55 PM

[Loaded Modules]

Name      Version      Size      File Date      Manufacturer
Path
traffic.dll 5.00.2195.6613 30.77 KB
(31,504 bytes) 8/16/2005 1:51:58 PM

```

```

Microsoft Corporation
c:\winnt\system32\traffic.dll
rsvp.exe      5.00.2195.6663 172.77 KB (176,912
bytes) 8/16/2005 1:51:55 PM Microsoft
Corporation c:\winnt\system32\rsvp.exe
wbemprox.dll  1.50.1085.0100 40.10 KB
(41,061 bytes) 8/16/2005 1:52:03 PM
Microsoft Corporation
c:\winnt\system32\wbem\wbemprox.dll
mlang.dll     5.00.3700.6655 510.77 KB (523,024
bytes) 8/16/2005 1:51:45 PM Microsoft
Corporation c:\winnt\system32\mlang.dll
cabinet.dll  5.00.2147.1 54.77 KB
(56,080 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\cabinet.dll
msinfo32.dll  5.00.2195.6601 312.27 KB
(319,760 bytes) 8/16/2005 1:52:05 PM
Microsoft Corporation c:\program
files\common files\microsoft
shared\msinfo\msinfo32.dll
mmcmdmgr.dll  5.00.2195.6601 816.27 KB
(835,856 bytes) 8/16/2005 1:51:45 PM
Microsoft Corporation
c:\winnt\system32\mmcmdmgr.dll
msvcps50.dll  5.00.7051 552.50 KB (565,760
bytes) 12/7/1999 7:00:00 AM Microsoft
Corporation c:\winnt\system32\msvcps50.dll
mmc.exe       5.00.2195.6601 589.27 KB (603,408
bytes) 8/16/2005 1:51:45 PM Microsoft
Corporation c:\winnt\system32\mmc.exe
rapilib.dll   5.00.2195.6601 24.77 KB
(25,360 bytes) 8/16/2005 1:51:54 PM
Microsoft Corporation
c:\winnt\system32\rapilib.dll
rsvpsp.dll    5.00.2195.6611 75.27 KB
(77,072 bytes) 8/16/2005 1:51:55 PM
Microsoft Corporation
c:\winnt\system32\rsvpsp.dll
tardis.exe    5, 0, 1, 4 308.00 KB
(315,392 bytes) 9/13/2002 6:21:25 PM
H.C.Mingham-Smith Ltd. c:\program
files\tardis 2000 v1.4\tardis.exe
aclntusr.exe  6, 1, 401 180.00 KB (184,320
bytes) 6/5/2003 1:55:47 PM c:\program
files\altiris\aclnt\aclntusr.exe
faxshell.dll  5.00.2134.1 8.27 KB
(8,464 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\faxshell.dll
msacm32.dll   5.00.2134.1 65.27 KB
(66,832 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\msacm32.dll
avifil32.dll  5.00.2195.6612 76.77 KB
(78,608 bytes) 8/16/2005 1:51:34 PM
Microsoft Corporation
c:\winnt\system32\avifil32.dll
msvfw32.dll   5.00.2195.6612 113.77 KB
(116,496 bytes) 8/16/2005 1:51:49 PM
Microsoft Corporation
c:\winnt\system32\msvfw32.dll

```

```

docprop2.dll  5.00.2178.1 297.77 KB
(304,912 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\docprop2.dll
shdoclc.dll  5.00.3700.6668 324.50 KB
(332,288 bytes) 8/16/2005 1:51:56 PM
Microsoft Corporation
c:\winnt\system32\shdoclc.dll
diskcopy.dll 5.00.2195.6601 15.77 KB
(16,144 bytes) 8/16/2005 1:51:38 PM
Microsoft Corporation
c:\winnt\system32\diskcopy.dll
urlmon.dll   5.00.3700.6705 442.77 KB
(453,392 bytes) 8/16/2005 1:51:58 PM
Microsoft Corporation
c:\winnt\system32\urlmon.dll
browselc.dll 5.00.3700.6661 34.50 KB
(35,328 bytes) 8/16/2005 1:51:34 PM
Microsoft Corporation
c:\winnt\system32\browselc.dll
linkinfo.dll 5.00.2134.1 15.77 KB
(16,144 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\linkinfo.dll
powrprof.dll 5.00.3502.6601 13.27 KB
(13,584 bytes) 8/16/2005 1:51:53 PM
Microsoft Corporation
c:\winnt\system32\powrprof.dll
batmeter.dll 5.00.3502.6601 20.27 KB
(20,752 bytes) 8/16/2005 1:51:34 PM
Microsoft Corporation
c:\winnt\system32\batmeter.dll
stobject.dll  5.00.2195.6601 79.27 KB
(81,168 bytes) 8/16/2005 1:51:57 PM
Microsoft Corporation
c:\winnt\system32\stobject.dll
msi.dll       3.1.4000.2435 2.76 MB (2,890,240
bytes) 8/16/2005 1:51:47 PM Microsoft
Corporation c:\winnt\system32\msi.dll
webcheck.dll  5.00.3502.6601 251.77 KB
(257,808 bytes) 8/16/2005 1:51:59 PM
Microsoft Corporation
c:\winnt\system32\webcheck.dll
ntshrui.dll  5.00.2134.1 46.77 KB
(47,888 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\ntshrui.dll
mydocs.dll   5.00.3502.6601 55.77 KB
(57,104 bytes) 8/16/2005 1:51:50 PM
Microsoft Corporation
c:\winnt\system32\mydocs.dll
browseui.dll 5.00.3700.6661 789.27 KB
(808,208 bytes) 8/16/2005 1:51:34 PM
Microsoft Corporation
c:\winnt\system32\browseui.dll
shdocvw.dll  5.00.3700.6668 1.06 MB
(1,107,728 bytes) 8/16/2005 1:51:56 PM
Microsoft Corporation
c:\winnt\system32\shdocvw.dll
explorer.exe  5.00.3700.6690 237.77 KB
(243,472 bytes) 8/16/2005 1:52:01 PM
Microsoft Corporation
c:\winnt\explorer.exe

```

```

rdpclip.exe      5.00.2174.1      39.77 KB
(40,720 bytes)   9/13/2002 5:45:10 PM
Microsoft Corporation
c:\winnt\system32\rdpclip.exe
wzcsapi.dll     5.00.2195.6604  29.27 KB
(29,968 bytes)  8/16/2005 1:52:10 PM
Microsoft Corporation
c:\winnt\system32\wzcsapi.dll
wzcdlg.dll     5.00.2195.6604  51.27 KB
(52,496 bytes)  8/16/2005 1:52:10 PM
Microsoft Corporation
c:\winnt\system32\wzcdlg.dll
cscui.dll      5.00.2195.6705  237.27 KB (242,960
bytes) 8/16/2005 1:51:37 PM
Microsoft Corporation
c:\winnt\system32\cscui.dll
logon.scr     5.00.2195.6601  127.77 KB (130,832
bytes) 8/16/2005 1:51:44 PM
Microsoft Corporation
c:\winnt\system32\logon.scr
h323.tsp     5.00.2195.6699  248.77 KB (254,736
bytes) 8/16/2005 1:51:40 PM
Microsoft Corporation
c:\winnt\system32\h323.tsp
ipconf.tsp    5.00.2143.1     10.77 KB
(11,024 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\ipconf.tsp
ndptsp.tsp    5.00.2143.1     38.27 KB
(39,184 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\ndptsp.tsp
kmdbsp.tsp    5.00.2150.1     17.77 KB
(18,192 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\kmdbsp.tsp
uniplat.dll   5.00.2195.6601  14.27 KB
(14,608 bytes) 8/16/2005 1:51:58 PM
Microsoft Corporation
c:\winnt\system32\uniplat.dll
unimdm.tsp    5.00.2195.6601  199.27 KB
(204,048 bytes) 8/16/2005 1:51:58 PM
Microsoft Corporation
c:\winnt\system32\unimdm.tsp
tapisrv.dll   5.00.2195.6666  169.27 KB
(173,328 bytes) 8/16/2005 1:51:58 PM
Microsoft Corporation
c:\winnt\system32\tapisrv.dll
dfssvc.exe    5.00.2195.6664  88.77 KB
(90,896 bytes) 8/16/2005 1:51:38 PM
Microsoft Corporation
c:\winnt\system32\dfssvc.exe
dbnetlib.dll  2000.080.0194.00 84.06 KB
(86,082 bytes) 9/13/2002 6:19:43 PM
Microsoft Corporation
c:\winnt\system32\dbnetlib.dll
odbc32.dll    3.520.6526.0    100.27 KB
(102,672 bytes) 9/13/2002 6:19:39 PM
Microsoft Corporation
c:\winnt\system32\odbc32.dll
sqlsrv32.rll  2000.080.0194.00 88.00 KB
(90,112 bytes) 9/13/2002 6:19:44 PM
Microsoft Corporation
c:\winnt\system32\sqlsrv32.rll

```

```

mtxdm.dll 2000.2.3504.0 22.77 KB (23,312 bytes)
8/16/2005 1:51:50 PM
Microsoft Corporation
c:\winnt\system32\mtxdm.dll
sqlunirl.dll 2000.080.0194.00 176.06 KB
(180,290 bytes) 8/6/2000 1:51:56 AM
Microsoft Corporation
c:\winnt\system32\sqlunirl.dll
sqlsrv32.dll 2000.080.0194.00 460.08 KB
(471,119 bytes) 9/13/2002 6:19:44 PM
Microsoft Corporation
c:\winnt\system32\sqlsrv32.dll
tpcc_com_all.dll 1, 0, 0, 1 104.00 KB
(106,496 bytes) 9/13/2002 6:29:44 PM
Microsoft Corporation
c:\inetpub\wwwroot\tpcc_com_all.dll
tpcc_odbc.dll Not Available 20.00 KB
(20,480 bytes) 6/1/2005 11:49:37 PM
Not Available
c:\inetpub\wwwroot\tpcc_odbc.dll
msvcr70.dll 7.00.9466.0 336.00 KB
(344,064 bytes) 6/1/2005 11:48:32 PM
Microsoft Corporation
c:\winnt\system32\msvcr70.dll
mfc42.dll 6.00.9586.0 992.05 KB (1,015,859
bytes) 8/16/2005 1:51:44 PM
Microsoft Corporation
c:\winnt\system32\mfc42.dll
wam.dll 5.00.0984 70.77 KB (72,464 bytes)
8/16/2005 1:52:13 PM
Microsoft Corporation
c:\winnt\system32\wam.dll
odbcint.dll 3.520.6526.0 88.00 KB
(90,112 bytes) 9/13/2002 6:19:39 PM
Microsoft Corporation
c:\winnt\system32\odbcint.dll
odbc32.dll 3.520.6526.0 216.27 KB
(221,456 bytes) 9/13/2002 6:19:39 PM
Microsoft Corporation
c:\winnt\system32\odbc32.dll
mtxoci.dll 2000.2.3504.0 103.27 KB
(105,744 bytes) 8/16/2005 1:51:50 PM
Microsoft Corporation
c:\winnt\system32\mtxoci.dll
iislog.dll 5.00.0984 75.27 KB (77,072 bytes)
8/16/2005 1:52:12 PM
Microsoft Corporation
c:\winnt\system32\iislog.dll
inetsloc.dll 5.00.0984 20.27 KB (20,752 bytes)
8/16/2005 1:51:42 PM
Microsoft Corporation
c:\winnt\system32\inetsloc.dll
isatq.dll 5.00.0984 61.27 KB (62,736 bytes)
8/16/2005 1:52:12 PM
Microsoft Corporation
c:\winnt\system32\isatq.dll
security.dll 5.00.2154.1 5.77 KB
(5,904 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\security.dll
svcxext.dll 5.00.0984 39.77 KB (40,720 bytes)
8/16/2005 1:52:13 PM
Microsoft Corporation
c:\winnt\system32\svcxext.dll
admexs.dll 5.00.0984 27.77 KB (28,432 bytes)
8/16/2005 1:52:11 PM
Microsoft Corporation
c:\winnt\system32\admexs.dll
wamreg.dll 5.00.0984 45.77 KB (46,864 bytes)
8/16/2005 1:52:13 PM
Microsoft Corporation

```

```

Corporation
c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll 5.00.0984 68.77 KB (70,416 bytes)
8/16/2005 1:52:13 PM
Microsoft Corporation
c:\winnt\system32\inetsrv\metadata.dll
iismap.dll 5.00.0984 56.27 KB (57,616 bytes)
8/16/2005 1:51:42 PM
Microsoft Corporation
c:\winnt\system32\iismap.dll
nsepm.dll 5.00.0984 43.27 KB (44,304 bytes)
8/16/2005 1:52:13 PM
Microsoft Corporation
c:\winnt\system32\inetsrv\nsepm.dll
admwprox.dll 5.00.0984 31.77 KB (32,528 bytes)
9/13/2002 5:45:33 PM
Microsoft Corporation
c:\winnt\system32\admwprox.dll
coadmin.dll 5.00.0984 39.77 KB (40,720 bytes)
8/16/2005 1:52:12 PM
Microsoft Corporation
c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll 5.00.0984 15.77 KB (16,144 bytes)
8/16/2005 1:52:12 PM
Microsoft Corporation
c:\winnt\system32\inetsrv\iisadmin.dll
rpcpref.dll 5.00.0984 4.27 KB (4,368 bytes)
8/16/2005 1:52:13 PM
Microsoft Corporation
c:\winnt\system32\inetsrv\rpcpref.dll
iisrtl.dll 5.00.0984 121.27 KB (124,176
bytes) 8/16/2005 1:51:42 PM
Microsoft Corporation
c:\winnt\system32\iisrtl.dll
inetinfo.exe 5.00.0984 14.27 KB (14,608 bytes)
8/16/2005 1:52:12 PM
Microsoft Corporation
c:\winnt\system32\inetsrv\inetinfo.exe
netui1.dll 5.00.2134.1 210.27 KB
(215,312 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\netui1.dll
netui0.dll 5.00.2195.6601 70.27 KB
(71,952 bytes) 8/16/2005 1:51:51 PM
Microsoft Corporation
c:\winnt\system32\netui0.dll
ntlanman.dll 5.00.2195.6601 35.27 KB
(36,112 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\ntlanman.dll
wshnetbs.dll 5.00.2134.1 7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\wshnetbs.dll
provthrd.dll 1.50.1085.0000 68.07 KB
(69,708 bytes) 9/13/2002 5:45:53 PM
Microsoft Corporation
c:\winnt\system32\wbem\provthrd.dll
ntevt.dll 1.50.1085.0072 192.06 KB (196,671
bytes) 8/16/2005 1:52:03 PM
Microsoft Corporation
c:\winnt\system32\wbem\ntevt.dll
perfos.dll 5.00.2155.1 21.27 KB
(21,776 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\perfos.dll

```

```

framedyn.dll      1.50.1085.0076      164.07 KB
(168,009 bytes)   8/16/2005 1:52:02 PM
Microsoft Corporation
c:\winnt\system32\wbem\framedyn.dll
cimwin32.dll     1.50.1085.0103      1.04 MB
(1,089,637 bytes) 8/16/2005 1:52:02 PM
Microsoft Corporation
c:\winnt\system32\wbem\cimwin32.dll
wbemsvc.dll     1.50.1085.0007      40.07 KB
(41,036 bytes)    8/16/2005 1:52:03 PM
Microsoft Corporation
c:\winnt\system32\wbem\wbemsvc.dll
wbemess.dll     1.50.1085.0100      364.09 KB
(372,825 bytes)   8/16/2005 1:52:03 PM
Microsoft Corporation
c:\winnt\system32\wbem\wbemess.dll
fastprox.dll    1.50.1085.0100      152.10 KB
(155,749 bytes)   8/16/2005 1:52:02 PM
Microsoft Corporation
c:\winnt\system32\wbem\fastprox.dll
wbemcore.dll    1.50.1085.0100      632.09 KB
(647,257 bytes)   8/16/2005 1:52:03 PM
Microsoft Corporation
c:\winnt\system32\wbem\wbemcore.dll
wbemcomn.dll    1.50.1085.0100      692.09 KB
(708,696 bytes)   8/16/2005 1:52:03 PM
Microsoft Corporation
c:\winnt\system32\wbem\wbemcomn.dll
winmgmt.exe     1.50.1085.0100      192.10 KB
(196,706 bytes)   8/16/2005 1:52:03 PM
Microsoft Corporation
c:\winnt\system32\wbem\winmgmt.exe
msidle.dll      5.00.2920.0000      6.27 KB
(6,416 bytes)     12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\msidle.dll
mstask.exe      4.71.2195.6704      116.77 KB
(119,568 bytes)   8/16/2005 1:51:49 PM
Microsoft Corporation
c:\winnt\system32\mstask.exe
wmi.dll          5.00.2191.1          6.27 KB (6,416 bytes)
12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\wmi.dll
netshell.dll    5.00.2195.6604      466.27 KB
(477,456 bytes)   8/16/2005 1:51:51 PM
Microsoft Corporation
c:\winnt\system32\netshell.dll
netman.dll      5.00.2195.6660      93.27 KB
(95,504 bytes)    8/16/2005 1:51:51 PM
Microsoft Corporation
c:\winnt\system32\netman.dll
resutils.dll    5.00.2195.6702      39.77 KB
(40,720 bytes)    8/16/2005 1:51:54 PM
Microsoft Corporation
c:\winnt\system32\resutils.dll
clusapi.dll     5.00.2195.6683      54.27 KB
(55,568 bytes)    8/16/2005 1:51:35 PM
Microsoft Corporation
c:\winnt\system32\clusapi.dll
mtxclu.dll      2000.2.3504.0        51.27 KB
(52,496 bytes)    8/16/2005 1:51:50 PM
Microsoft Corporation
c:\winnt\system32\mtxclu.dll

```

```

msdtcprx.dll    2000.2.3504.0        690.77 KB
(707,344 bytes)   8/16/2005 1:51:46 PM
Microsoft Corporation
c:\winnt\system32\msdtcprx.dll
comsvcs.dll     2000.2.3504.0        1.38 MB
(1,448,208 bytes) 8/16/2005 1:51:36 PM
Microsoft Corporation
c:\winnt\system32\comsvcs.dll
ntmsdba.dll     5.00.2195.6655      169.27 KB
(173,328 bytes)   8/16/2005 1:51:52 PM
Microsoft Corporation
c:\winnt\system32\ntmsdba.dll
ipbootp.dll     5.00.2168.1          33.77 KB
(34,576 bytes)    12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\ipbootp.dll
cryptui.dll     5.131.2195.6628     433.27 KB
(443,664 bytes)   8/16/2005 1:51:37 PM
Microsoft Corporation
c:\winnt\system32\cryptui.dll
rastls.dll      5.00.2195.6680      98.27 KB
(100,624 bytes)   8/16/2005 1:51:54 PM
Microsoft Corporation
c:\winnt\system32\rastls.dll
raschap.dll     5.00.2195.6663      59.27 KB
(60,688 bytes)    8/16/2005 1:51:54 PM
Microsoft Corporation
c:\winnt\system32\raschap.dll
rasppp.dll      5.00.2195.6626      194.27 KB
(198,928 bytes)   8/16/2005 1:51:54 PM
Microsoft Corporation
c:\winnt\system32\rasppp.dll
rastapi.dll     5.00.2195.6604      52.77 KB
(54,032 bytes)    12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\rastapi.dll
rasdlg.dll      5.00.2195.6625      516.77 KB
(529,168 bytes)   12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\rasdlg.dll
netcfgx.dll     5.00.2195.6604      534.77 KB
(547,600 bytes)   8/16/2005 1:51:50 PM
Microsoft Corporation
c:\winnt\system32\netcfgx.dll
rasmans.dll     5.00.2195.6696      149.77 KB
(153,360 bytes)   8/16/2005 1:51:54 PM
Microsoft Corporation
c:\winnt\system32\rasmans.dll
sens.dll        5.00.2195.6627      37.27 KB (38,160 bytes)
8/16/2005 1:51:55 PM
Microsoft Corporation
c:\winnt\system32\sens.dll
ntmssvc.dll     5.00.2195.6655      391.77 KB
(401,168 bytes)   8/16/2005 1:51:52 PM
Microsoft Corporation
c:\winnt\system32\ntmssvc.dll
sensapi.dll     5.00.2195.6627      7.27 KB
(7,440 bytes)     8/16/2005 1:51:55 PM
Microsoft Corporation
c:\winnt\system32\sensapi.dll
winhttp.dll     5.1.2600.1188 (xpsp2.030318-2132)
303.50 KB (310,784 bytes) 8/16/2005
1:52:11 PM
Microsoft Corporation
c:\winnt\system32\winhttp.dll

```

```

txfaux.dll      2000.2.3504.0        388.27 KB
(397,584 bytes)   8/16/2005 1:51:58 PM
Microsoft Corporation
c:\winnt\system32\txfaux.dll
es.dll          2000.2.3504.0        227.77 KB (233,232 bytes)
8/16/2005 1:51:39 PM
Microsoft Corporation
c:\winnt\system32\es.dll
util.dll       5.00.2195.6701      25.77 KB
(26,384 bytes)    8/16/2005 1:51:59 PM
Microsoft Corporation
c:\winnt\system32\util.dll
wtsapi32.dll    5.00.2134.1          14.27 KB
(14,608 bytes)    12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\wtsapi32.dll
advpack.dll     5.00.3502.6601      86.77 KB
(88,848 bytes)    8/16/2005 1:51:33 PM
Microsoft Corporation
c:\winnt\system32\advpack.dll
wuaueng.dll     5.4.3630.2554 built by: lab04_n
188.00 KB (192,512 bytes) 8/16/2005
1:52:11 PM
Microsoft Corporation
c:\winnt\system32\wuaueng.dll
wuauerv.dll     5.4.3630.2554 built by: lab04_n
9.00 KB (9,216 bytes) 8/16/2005
1:52:11 PM
Microsoft Corporation
c:\winnt\system32\wuauerv.dll
clbcatq.dll     2000.2.3504.0        498.27 KB
(510,224 bytes)   8/16/2005 1:51:35 PM
Microsoft Corporation
c:\winnt\system32\clbcatq.dll
rpcss.dll       5.00.2195.6702      233.77 KB (239,376 bytes)
8/16/2005 1:51:55 PM
Microsoft Corporation
c:\winnt\system32\rpcss.dll
svchost.exe    5.00.2134.1          7.77 KB
(7,952 bytes)     12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\svchost.exe
rsys.exe        Not Available        32.00 KB (32,768 bytes)
9/13/2002 6:30:57 PM
Not Available
c:\benchcraft\rsys.exe
regsvcs.exe     5.00.2195.6701      66.77 KB
(68,368 bytes)    8/16/2005 1:51:54 PM
Microsoft Corporation
c:\winnt\system32\regsvcs.exe
ntmarta.dll     5.00.2195.6666      100.27 KB
(102,672 bytes)   8/16/2005 1:51:52 PM
Microsoft Corporation
c:\winnt\system32\ntmarta.dll
psapi.dll       5.00.2134.1          28.27 KB (28,944 bytes)
12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\psapi.dll
riched20.dll    5.30.23.1215         421.77 KB
(431,888 bytes)   8/16/2005 1:51:54 PM
Microsoft Corporation
c:\winnt\system32\riched20.dll
riched32.dll    5.00.2134.1          3.77 KB
(3,856 bytes)     12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\riched32.dll
comdlg32.dll    5.00.3700.6693      235.77 KB
(241,424 bytes)   12/7/1999 7:00:00 AM

```

```

Microsoft Corporation
c:\winnt\system32\comdlg32.dll
aclient.exe 6.1.401 4.63 MB (4,857,932
bytes) 6/5/2003 1:55:46 PM Altiris, Inc.
c:\program
files\altiris\aclient\aclient.exe
rdpwsx.dll 5.00.2195.6697 97.90 KB
(100,248 bytes) 8/16/2005 1:51:54 PM
Microsoft Corporation
c:\winnt\system32\rdpwsx.dll
mstlsapi.dll 5.00.2195.6659 25.77 KB
(26,384 bytes) 8/16/2005 1:51:49 PM
Microsoft Corporation
c:\winnt\system32\mstlsapi.dll
icaapi.dll 5.00.2195.6654 122.77 KB
(125,712 bytes) 8/16/2005 1:51:41 PM
Microsoft Corporation
c:\winnt\system32\icaapi.dll
regapi.dll 5.00.2195.6602 35.27 KB
(36,112 bytes) 8/16/2005 1:51:54 PM
Microsoft Corporation
c:\winnt\system32\regapi.dll
termsrv.exe 5.00.2195.6696 139.27 KB
(142,608 bytes) 8/16/2005 1:51:58 PM
Microsoft Corporation
c:\winnt\system32\termsrv.exe
iissuba.dll 5.00.0984 9.77 KB (10,000 bytes)
12/7/1999 7:00:00 AM Microsoft
Corporation c:\winnt\system32\iissuba.dll
dsenh.dll 5.00.2195.6612 143.77 KB
(147,216 bytes) 8/16/2005 1:52:09 PM
Microsoft Corporation
c:\winnt\system32\dsenh.dll
oakley.dll 5.00.2195.6662 435.77 KB
(446,224 bytes) 8/16/2005 1:51:52 PM
Microsoft Corporation
c:\winnt\system32\oakley.dll
mfc42u.dll 6.00.9586.0 988.05 KB
(1,011,764 bytes) 8/16/2005 1:51:44 PM
Microsoft Corporation
c:\winnt\system32\mfc42u.dll
polagent.dll 5.00.2195.6655 109.27 KB
(111,888 bytes) 8/16/2005 1:51:53 PM
Microsoft Corporation
c:\winnt\system32\polagent.dll
scecli.dll 5.00.2195.6704 111.77 KB
(114,448 bytes) 8/16/2005 1:51:55 PM
Microsoft Corporation
c:\winnt\system32\scecli.dll
mswsock.dll 5.00.2195.6603 62.77 KB
(64,272 bytes) 8/16/2005 1:51:50 PM
Microsoft Corporation
c:\winnt\system32\mswsock.dll
ntdsatq.dll 5.00.2195.6620 31.27 KB
(32,016 bytes) 8/16/2005 1:51:51 PM
Microsoft Corporation
c:\winnt\system32\ntdsatq.dll
ntdsa.dll 5.00.2195.6697 1016.27 KB (1,040,656
bytes) 8/16/2005 1:51:51 PM Microsoft
Corporation c:\winnt\system32\ntdsa.dll
kdcsvc.dll 5.00.2195.6627 144.77 KB
(148,240 bytes) 8/16/2005 1:51:44 PM

```

```

Microsoft Corporation
c:\winnt\system32\kdcsvc.dll
sfmapi.dll 5.00.2134.1 38.77 KB
(39,696 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\sfmapi.dll
rassfm.dll 5.00.2195.6604 21.27 KB
(21,776 bytes) 8/16/2005 1:51:54 PM
Microsoft Corporation
c:\winnt\system32\rassfm.dll
rsabase.dll 5.00.2195.6619 129.27 KB
(132,368 bytes) 6/19/2003 12:05:04 PM
Microsoft Corporation
c:\winnt\system32\rsabase.dll
schannel.dll 5.00.2195.6705 144.27 KB
(147,728 bytes) 5/4/2001 12:05:02 PM
Microsoft Corporation
c:\winnt\system32\schannel.dll
netlogon.dll 5.00.2195.6695 363.27 KB
(371,984 bytes) 8/16/2005 1:51:50 PM
Microsoft Corporation
c:\winnt\system32\netlogon.dll
kerberos.dll 5.00.2195.6666 207.77 KB
(212,752 bytes) 8/16/2005 1:51:44 PM
Microsoft Corporation
c:\winnt\system32\kerberos.dll
msprivs.dll 5.00.2195.6695 46.00 KB
(47,104 bytes) 8/16/2005 1:51:48 PM
Microsoft Corporation
c:\winnt\system32\msprivs.dll
samsrv.dll 5.00.2195.6697 380.77 KB
(389,904 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\samsrv.dll
lsasrv.dll 5.00.2195.6695 506.77 KB
(518,928 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\lsasrv.dll
lsass.exe 5.00.2195.6695 32.77 KB (33,552 bytes)
12/7/1999 7:00:00 AM Microsoft
Corporation c:\winnt\system32\lsass.exe
esent.dll 6.1.3940.31 1.08 MB (1,135,376
bytes) 8/16/2005 1:51:39 PM Microsoft
Corporation c:\winnt\system32\esent.dll
ntlsapi.dll 5.00.2195.6601 6.77 KB
(6,928 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\ntlsapi.dll
xactsrv.dll 5.00.2195.6662 90.27 KB
(92,432 bytes) 8/16/2005 1:52:00 PM
Microsoft Corporation
c:\winnt\system32\xactsrv.dll
wmicore.dll 5.00.2195.6611 72.77 KB
(74,512 bytes) 8/16/2005 1:52:00 PM
Microsoft Corporation
c:\winnt\system32\wmicore.dll
browser.dll 5.00.2195.6693 67.27 KB
(68,880 bytes) 8/16/2005 1:51:34 PM
Microsoft Corporation
c:\winnt\system32\browser.dll
trkwks.dll 5.00.2195.6623 88.27 KB
(90,384 bytes) 8/16/2005 1:51:58 PM

```

```

Microsoft Corporation
c:\winnt\system32\trkwks.dll
psbase.dll 5.00.2195.6661 112.77 KB
(115,472 bytes) 8/16/2005 1:51:53 PM
Microsoft Corporation
c:\winnt\system32\psbase.dll
cryptsvc.dll 5.00.2195.6661 74.27 KB
(76,048 bytes) 8/16/2005 1:51:37 PM
Microsoft Corporation
c:\winnt\system32\cryptsvc.dll
cfgmgr32.dll 5.00.2134.1 16.77 KB
(17,168 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\cfgmgr32.dll
dmsrvr.dll 2195.6605.297.3 11.77 KB
(12,048 bytes) 8/16/2005 1:51:38 PM
VERITAS Software Corp.
c:\winnt\system32\dmsrvr.dll
seclogon.dll 5.00.2195.6707 16.77 KB
(17,168 bytes) 8/16/2005 1:51:55 PM
Microsoft Corporation
c:\winnt\system32\seclogon.dll
cryptdll.dll 5.00.2195.6607 43.27 KB
(44,304 bytes) 8/16/2005 1:51:37 PM
Microsoft Corporation
c:\winnt\system32\cryptdll.dll
wkssvc.dll 5.00.2195.6692 95.77 KB
(98,064 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\wkssvc.dll
srvsvc.dll 5.00.2195.6697 81.77 KB
(83,728 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\srvsvc.dll
eventlog.dll 5.00.2195.6716 46.77 KB
(47,888 bytes) 8/16/2005 1:51:40 PM
Microsoft Corporation
c:\winnt\system32\eventlog.dll
scesrv.dll 5.00.2195.6704 248.77 KB
(254,736 bytes) 8/16/2005 1:51:55 PM
Microsoft Corporation
c:\winnt\system32\scesrv.dll
umpnpgm.dll 5.00.2182.1 86.27 KB
(88,336 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\umpnpgm.dll
services.exe 5.00.2195.6700 87.27 KB
(89,360 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\services.exe
wininet.dll 5.00.3700.6713 455.77 KB
(466,704 bytes) 8/16/2005 1:52:00 PM
Microsoft Corporation
c:\winnt\system32\wininet.dll
cryptnet.dll 5.131.2195.6601 42.27 KB
(43,280 bytes) 8/16/2005 1:51:37 PM
Microsoft Corporation
c:\winnt\system32\cryptnet.dll
msv1_0.dll 5.00.2195.6680 114.77 KB
(117,520 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\msv1_0.dll

```

```

ntdsapi.dll 5.00.2195.6666 56.27 KB
(57,616 bytes) 8/16/2005 1:51:51 PM
Microsoft Corporation
c:\winnt\system32\ntdsapi.dll
rasadhlp.dll 5.00.2168.1 7.27 KB
(7,440 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\rasadhlp.dll
winrnr.dll 5.00.2160.1 18.77 KB
(19,216 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\winrnr.dll
rnr20.dll 5.00.2195.6603 35.77 KB (36,624 bytes)
8/16/2005 1:51:55 PM Microsoft
Corporation
c:\winnt\system32\rnr20.dll
dhcpcsvc.dll 5.00.2195.6685 90.77 KB
(92,944 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\dhcpcsvc.dll
tapi32.dll 5.00.2195.6664 123.77 KB
(126,736 bytes) 8/16/2005 1:51:57 PM
Microsoft Corporation
c:\winnt\system32\tapi32.dll
rasman.dll 5.00.2195.6604 54.77 KB
(56,080 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\rasman.dll
rasapi32.dll 5.00.2195.6625 192.77 KB
(197,392 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\rasapi32.dll
rtutils.dll 5.00.2168.1 43.77 KB
(44,816 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\rtutils.dll
adsldpc.dll 5.00.2195.6701 130.77 KB
(133,904 bytes) 8/16/2005 1:51:33 PM
Microsoft Corporation
c:\winnt\system32\adsldpc.dll
activeds.dll 5.00.2195.6601 177.77 KB
(182,032 bytes) 8/16/2005 1:51:31 PM
Microsoft Corporation
c:\winnt\system32\activeds.dll
oleaut32.dll 2.40.4522 612.27 KB (626,960
bytes) 12/7/1999 7:00:00 AM Microsoft
Corporation
c:\winnt\system32\oleaut32.dll
mprapi.dll 5.00.2181.1 79.27 KB
(81,168 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\mprapi.dll
icmp.dll 5.00.2134.1 7.27 KB (7,440 bytes)
12/7/1999 7:00:00 AM Microsoft
Corporation
c:\winnt\system32\icmp.dll
iphlpapi.dll 5.00.2195.6602 68.27 KB
(69,904 bytes) 8/16/2005 1:51:43 PM
Microsoft Corporation
c:\winnt\system32\iphlpapi.dll
wshtcpip.dll 5.00.2195.6601 17.27 KB
(17,680 bytes) 8/16/2005 1:52:00 PM
Microsoft Corporation
c:\winnt\system32\wshtcpip.dll

```

```

msafd.dll 5.00.2195.6602 106.27 KB (108,816
bytes) 8/16/2005 1:51:46 PM Microsoft
Corporation
c:\winnt\system32\msafd.dll
mpr.dll 5.00.2195.6611 53.77 KB (55,056 bytes)
8/16/2005 1:51:45 PM Microsoft
Corporation
c:\winnt\system32\mpr.dll
winspool.drv 5.00.2195.6659 111.27 KB
(113,936 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\winspool.drv
winscard.dll 5.00.2195.6609 77.27 KB
(79,120 bytes) 8/16/2005 1:52:00 PM
Microsoft Corporation
c:\winnt\system32\winscard.dll
atl.dll 3.00.9435 73.06 KB (74,810 bytes)
8/16/2005 1:51:33 PM Microsoft
Corporation
c:\winnt\system32\atl.dll
certcli.dll 5.00.2195.6619 132.27 KB
(135,440 bytes) 8/16/2005 1:51:35 PM
Microsoft Corporation
c:\winnt\system32\certcli.dll
wlnotify.dll 5.00.2195.6706 56.27 KB
(57,616 bytes) 8/16/2005 1:52:00 PM
Microsoft Corporation
c:\winnt\system32\wlnotify.dll
cscdll.dll 5.00.2195.6713 98.77 KB
(101,136 bytes) 8/16/2005 1:51:37 PM
Microsoft Corporation
c:\winnt\system32\cscdll.dll
lz32.dll 5.00.2195.6611 9.77 KB (10,000 bytes)
8/16/2005 1:51:44 PM Microsoft
Corporation
c:\winnt\system32\lz32.dll
version.dll 5.00.2195.6623 15.77 KB
(16,144 bytes) 8/16/2005 1:51:59 PM
Microsoft Corporation
c:\winnt\system32\version.dll
rsaenh.dll 5.00.2195.6611 131.77 KB
(134,928 bytes) 8/16/2005 1:52:09 PM
Microsoft Corporation
c:\winnt\system32\rsaenh.dll
mscat32.dll 5.131.2134.1 7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\mscat32.dll
ole32.dll 5.00.2195.6692 972.77 KB (996,112
bytes) 8/16/2005 1:51:52 PM Microsoft
Corporation
c:\winnt\system32\ole32.dll
imagehlp.dll 5.00.2195.6613 125.77 KB
(128,784 bytes) 5/4/2001 12:05:02 PM
Microsoft Corporation
c:\winnt\system32\imagehlp.dll
msasn1.dll 5.00.2195.6666 51.77 KB
(53,008 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\msasn1.dll
crypt32.dll 5.131.2195.6661 468.27 KB
(479,504 bytes) 8/16/2005 1:51:37 PM
Microsoft Corporation
c:\winnt\system32\crypt32.dll
wintrust.dll 5.131.2195.6624 162.27 KB
(166,160 bytes) 8/16/2005 1:52:00 PM
Microsoft Corporation
c:\winnt\system32\wintrust.dll

```

```

shlwapi.dll 5.00.3502.6601 282.77 KB
(289,552 bytes) 8/16/2005 1:51:56 PM
Microsoft Corporation
c:\winnt\system32\shlwapi.dll
shell32.dll 5.00.3700.6705 2.27 MB
(2,383,632 bytes) 8/16/2005 1:51:56 PM
Microsoft Corporation
c:\winnt\system32\shell32.dll
msgina.dll 5.00.2195.6669 326.27 KB
(334,096 bytes) 8/16/2005 1:51:46 PM
Microsoft Corporation
c:\winnt\system32\msgina.dll
comctl32.dll 5.81 537.77 KB (550,672
bytes) 12/7/1999 7:00:00 AM Microsoft
Corporation
c:\winnt\system32\comctl32.dll
setupapi.dll 5.00.2195.6622 556.77 KB
(570,128 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\setupapi.dll
winmm.dll 5.00.2161.1 184.77 KB (189,200
bytes) 12/7/1999 7:00:00 AM Microsoft
Corporation
c:\winnt\system32\winmm.dll
winsta.dll 5.00.2195.6701 38.27 KB
(39,184 bytes) 8/16/2005 1:52:00 PM
Microsoft Corporation
c:\winnt\system32\winsta.dll
wsock32.dll 5.00.2195.6603 21.27 KB
(21,776 bytes) 8/16/2005 1:52:00 PM
Microsoft Corporation
c:\winnt\system32\wsock32.dll
dnsapi.dll 5.00.2195.6680 131.77 KB
(134,928 bytes) 8/16/2005 1:51:38 PM
Microsoft Corporation
c:\winnt\system32\dnsapi.dll
wldap32.dll 5.00.2195.6666 158.27 KB
(162,064 bytes) 8/16/2005 1:52:00 PM
Microsoft Corporation
c:\winnt\system32\wldap32.dll
ws2help.dll 5.00.2134.1 17.77 KB
(18,192 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\ws2help.dll
ws2_32.dll 5.00.2195.6601 68.27 KB
(69,904 bytes) 8/16/2005 1:52:00 PM
Microsoft Corporation
c:\winnt\system32\ws2_32.dll
samlib.dll 5.00.2195.6666 48.77 KB
(49,936 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\samlib.dll
netrap.dll 5.00.2134.1 11.27 KB
(11,536 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\netrap.dll
netapi32.dll 5.00.2195.6601 304.27 KB
(311,568 bytes) 8/16/2005 1:51:50 PM
Microsoft Corporation
c:\winnt\system32\netapi32.dll
profmap.dll 5.00.2195.6610 29.27 KB
(29,968 bytes) 8/16/2005 1:51:53 PM
Microsoft Corporation
c:\winnt\system32\profmap.dll

```

```

secur32.dll      5.00.2195.6695      47.77 KB
(48,912 bytes)   8/16/2005 1:51:55 PM
Microsoft Corporation
c:\winnt\system32\secur32.dll
sfc.dll          5.00.2195.6673      92.80 KB (95,024 bytes)
8/16/2005 1:51:55 PM      Microsoft
Corporation      c:\winnt\system32\sfc.dll
nddeapi.dll     5.00.2195.6661      15.77 KB
(16,144 bytes)    8/16/2005 1:51:50 PM      Microsoft
Corporation      c:\winnt\system32\nddeapi.dll
userenv.dll     5.00.2195.6711      380.77 KB
(389,904 bytes)  8/16/2005 1:51:59 PM      Microsoft
Corporation      c:\winnt\system32\userenv.dll
user32.dll      5.00.2195.6688      393.77 KB
(403,216 bytes)  8/16/2005 1:51:59 PM      Microsoft
Corporation      c:\winnt\system32\user32.dll
gdi32.dll       5.00.2195.6660      228.27 KB (233,744
bytes)            8/16/2005 1:51:40 PM      Microsoft
Corporation      c:\winnt\system32\gdi32.dll
rpcrt4.dll     5.00.2195.6701      443.77 KB
(454,416 bytes)  8/16/2005 1:51:55 PM      Microsoft
Corporation      c:\winnt\system32\rpcrt4.dll
advapi32.dll   5.00.2195.6710      378.27 KB
(387,344 bytes)  12/7/1999 7:00:00 AM      Microsoft
Corporation      c:\winnt\system32\advapi32.dll
kernel32.dll   5.00.2195.6688      725.77 KB
(743,184 bytes)  12/7/1999 7:00:00 AM      Microsoft
Corporation      c:\winnt\system32\kernel32.dll
msvcrt.dll    6.10.9844.0         280.05 KB
(286,773 bytes)  6/19/2003 12:05:04 PM      Microsoft
Corporation      c:\winnt\system32\msvcrt.dll
winlogon.exe   5.00.2195.6714      176.77 KB
(181,008 bytes)  8/16/2005 1:52:00 PM      Microsoft
Corporation      c:\winnt\system32\winlogon.exe
sfcfiles.dll   5.00.2195.6717      948.27 KB
(971,024 bytes)  8/16/2005 1:51:55 PM      Microsoft
Corporation      c:\winnt\system32\sfcfiles.dll
ntdll.dll     5.00.2195.6685      480.27 KB (491,792
bytes)            5/4/2001 12:05:02 PM      Microsoft
Corporation      c:\winnt\system32\ntdll.dll
smss.exe      5.00.2195.6601      44.77 KB (45,840 bytes)
12/7/1999 7:00:00 AM      Microsoft
Corporation      c:\winnt\system32\smss.exe

[Services]

Display Name      Name      State      Start Mode
Service Type      Path      Error Control
Start Name      Tag ID
Altiris Client Service
Auto      Own Process      c:\program
files\altiris\aclient\aclient.exe -service
Normal      LocalSystem      0

```

```

Alerter      Alerter      Stopped      Manual      Share Process
c:\winnt\system32\services.exe
Normal      LocalSystem      0
Application Management
AppMgmt      Stopped
Manual      Share Process
c:\winnt\system32\services.exe
Normal      LocalSystem      0
ASP.NET State Service      aspnet_state
Stopped      Manual      Own Process
c:\winnt\microsoft.net\framework\v2.0.50215
\aspnet_state.exe      Normal      .\ASPNET      0
Background Intelligent Transfer Service BITS
Stopped      Manual      Share Process
c:\winnt\system32\svchost.exe -k bitsgroup
Normal      LocalSystem      0
Computer Browser      Browser      Running      Auto
Share Process
c:\winnt\system32\services.exe
Normal      LocalSystem      0
Indexing Service      cisvc      Stopped      Manual
Share Process
c:\winnt\system32\cisvc.exe      Normal
LocalSystem      0
ClipBook      ClipSrv      Stopped      Manual      Own Process
c:\winnt\system32\clipsrv.exe      Normal
LocalSystem      0
.NET Runtime Optimization Service v2.0.50215_X86
clr_optimization_v2.0.50215_32
Stopped      Manual      Own Process
c:\winnt\microsoft.net\framework\v2.0.50215
\mscorlib.exe      Ignore      LocalSystem      0
Distributed File System      Dfs      Running
Auto      Own Process
c:\winnt\system32\dfssvc.exe      Normal
LocalSystem      0
DHCP Client      Dhcp      Stopped      Disabled
Share Process
c:\winnt\system32\services.exe
Normal      LocalSystem      0
Logical Disk Manager Administrative Service
dmadmin      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
IIS Admin Service      IISADMIN      Running      Auto
Share Process

```

```

c:\winnt\system32\inetrv\inetinfo.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
Disabled      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
Stopped      Manual      Own Process
c:\winnt\system32\msdtc.exe      Normal
LocalSystem      0
Windows Installer      MSI Server      Stopped      Manual
Share Process
c:\winnt\system32\msiexec.exe /v
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
Stopped      Manual      Share Process
c:\winnt\system32\lsass.exe      Normal
LocalSystem      0

```

```

Removable Storage NtmsSvc Running Auto
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 Running
Auto 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 Auto Connection Manager RasAuto
Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Remote Access Connection Manager RasMan
Running 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 Command Service RMSYS Running
Auto Own Process
c:\benchcraft\rsys.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 Running Auto
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 Running
Auto 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
Performance Logs and Alerts SysmonLog Stopped
Manual Own Process
c:\winnt\system32\smlogsvc.exe
Normal LocalSystem 0
Telephony Tapisrv Running Manual Share Process
c:\winnt\system32\svchost.exe -k tapisrv
Normal LocalSystem 0
Terminal Services TermService Running
Auto 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
Running Auto 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
World Wide Web Publishing Service W3SVC
Stopped Auto Share Process
c:\winnt\system32\inetsrv\inetinfo.exe
Normal LocalSystem 0
Windows Management Instrumentation WinMgmt
Running Auto Own Process
c:\winnt\system32\wbem\winmgmt.exe
Ignore LocalSystem 0
Windows Management Instrumentation Driver Extensions
Wmi Running Manual Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
Automatic Updates wuauclt Running Auto
Share Process

```

```

c:\winnt\system32\svchost.exe -k wugroup
Normal LocalSystem 0
Wireless Configuration WZCSVC Stopped
Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs
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\Communications All
Users:Accessories\Communications All Users
Accessories\Entertainment All
Users:Accessories\Entertainment All Users
Accessories\Microsoft Script Debugger All
Users:Accessories\Microsoft Script Debugger All
Users
Accessories\System Tools All
Users:Accessories\System Tools All Users
Administrative Tools All
Users:Administrative Tools All Users
HP System Tools All Users:HP System Tools All
Users
HP System Tools\HP Array Diagnostic Utility All
Users:HP System Tools\HP Array Diagnostic Utility All
Users
Microsoft SQL Server 2005 CTP All Users:Microsoft SQL
Server 2005 CTP All Users
Microsoft SQL Server 2005 CTP\Configuration Tools All
Users:Microsoft SQL Server 2005 CTP\Configuration
Tools All Users
Microsoft SQL Server 2005 CTP\Documentation and
Tutorials All Users:Microsoft SQL Server 2005
CTP\Documentation and Tutorials All Users
Microsoft SQL Server 2005 CTP\Documentation and
Tutorials\Tutorials All Users:Microsoft SQL Server
2005 CTP\Documentation and Tutorials\Tutorials All
Users
Startup All Users:Startup All Users
Tardis All Users:Tardis All Users
Accessories CL31\Administrator:Accessories
CL31\Administrator
Accessories\Accessibility
CL31\Administrator:Accessories\Accessibilit
y
CL31\Administrator
Accessories\Entertainment
CL31\Administrator:Accessories\Entertainmen
t
CL31\Administrator
Accessories\System Tools
CL31\Administrator:Accessories\System Tools
CL31\Administrator

```

Administrative Tools
 CL31\Administrator:Administrative Tools
 CL31\Administrator

Startup CL31\Administrator:Startup
 CL31\Administrator

[Startup Programs]

Program	Command	User Name	Location
Tardis 2000	c:\progra-1\tardis-1.4\tardis.exe	All Users	Common Startup
ACLntUsr	c:\program files\altiris\aclient\aclntusr.exe	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
Available	
DDSContainerCtl Class	Not Available
Bitmap Image	mspaint.exe

[Internet Explorer 5]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Version	5.0.3700.1000
Build	53700.1000
Product ID	51876-270-9567332-05753
Application Path	C:\Program Files\Internet Explorer
Language	English (United States)
Active Printer	Not Available
Cipher Strength	168-bit
Content Advisor	Disabled
IEAK Install	No

[File Versions]

File	Version	Size	Date	Path
advapi32.dll	5.0.2195.6710	378 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation

advpack.dll	5.0.3502.6601	87 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
browseic.dll	5.0.3700.6661	35 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
browseui.dll	5.0.3700.6661	789 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
ckcnv.exe	5.0.2189.1	9 KB	12/7/1999 8:00:00 AM	C:\WINNT\system32 Microsoft Corporation
comctl32.dll	5.81.3502.6601	538 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
crypt32.dll	5.131.2195.6661	468 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
ehnsig.dll	<File Missing>	Not Available	Not Available	Not Available
Available				
iemigrat.dll	<File Missing>	Not Available	Not Available	Not Available
Available				
iesetup.dll	5.0.3502.6601	57 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
iexplore.exe	5.0.2920.0	59 KB	12/7/1999 8:00:00 AM	C:\Program Files\Internet Explorer Microsoft Corporation
imagehlp.dll	5.0.2195.6613	126 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
imghelp.dll	<File Missing>	Not Available	Not Available	Not Available
Available				
inseng.dll	5.0.3502.6601	72 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
jobexec.dll	5.0.0.1	47 KB	12/7/1999 8:00:00 AM	C:\WINNT\system32 Microsoft Corporation
jscript.dll	5.1.0.8513	476 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
jsproxy.dll	5.0.2920.0	13 KB	12/7/1999 8:00:00 AM	C:\WINNT\system32 Microsoft Corporation
msaahtml.dll	<File Missing>	Not Available	Not Available	Not Available
Available				
mshtml.dll	5.0.3700.6699	2299 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
msoss.dll	<File Missing>	Not Available	Not Available	Not Available
Available				
msxml.dll	8.0.6730.0	502 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
occache.dll	5.0.3502.6601	86 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation

ole32.dll	5.0.2195.6692	973 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
oleaut32.dll	2.40.4522.0	612 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
olepro32.dll	5.0.4522.0	160 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
rsabase.dll	5.0.2195.6619	129 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
rsaenh.dll	5.0.2195.6611	132 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
rsapi32.dll	<File Missing>	Not Available	Not Available	Not Available
Available				
rsasig.dll	<File Missing>	Not Available	Not Available	Not Available
Available				
schannel.dll	5.1.2195.6705	144 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
shdoc401.dll	<File Missing>	Not Available	Not Available	Not Available
Available				
shdocvw.dll	5.0.3700.6668	1082 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
shell32.dll	5.0.3700.6705	2328 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
shlwapi.dll	5.0.3502.6601	283 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
url.dll	5.0.3502.6601	82 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
urlmon.dll	5.0.3700.6705	443 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
vbscript.dll	5.1.0.7426	428 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
webcheck.dll	5.0.3502.6601	252 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
win.com	5.0.2134.1	24 KB	12/7/1999 8:00:00 AM	C:\WINNT\system32 Microsoft Corporation
wininet.dll	5.0.3700.6713	456 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
winsock.dll	3.10.0.103	3 KB	12/7/1999 8:00:00 AM	C:\WINNT\system32 Microsoft Corporation
wintrust.dll	5.131.2195.6624	162 KB	6/19/2003 12:05:04 PM	C:\WINNT\system32 Microsoft Corporation
wsock.vxd	<File Missing>	Not Available	Not Available	Not Available
Available				


```
wsock32.dll      5.0.2195.6603      21 KB
                 6/19/2003 12:05:04 PM
                 C:\WINNT\system32 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      wininet.dll
AutoProxyDetectMode Disabled
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\Administrator\Local Settings\Temporary
Internet Files
Total Disk Space      34723 MB
Available Disk Space      31363 MB
Maximum Cache Size      542 MB
Available Cache Size      542 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 9/13/2002 to
8/20/2102 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
Local intranet      Medium-low
Trusted sites      Low
Internet Medium
Restricted sites      High
```

Microsoft SQL Server 2005 Installation Procedures

Microsoft SQL Server 2005 Installation Procedures
Type of installation: custom
During the custom installation, use the default settings for all except the following two areas:
Services accounts:
SQL Server - local system account
SQL Server Agent - local system account
Set the sort order/collation as Latin1_General_BIN

Microsoft COM Component Configuration Parameters

The component services tool in Windows 2000 was used to change the queue settings for the TPCC COM+ single queue component. The single queue component was set to enable object pooling, object construction, just in time activation, and component supports events and statistics. The min and max pool size for the single queue component on each client was 54. Delivery threads were set under the TPCC key in the registry. The construction string was Dummy String

Appendix D: 60-Day Space

TPC-C 60 Day Space Requirements

Warehouses	12,400				TpmC	140,264
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	12,400	1,328	104	72		1,504
District	124,000	13,784	128	696		14,608
Customer	372,000,000	270,545,456	16,880,192	14,371,282		301,796,930
History	372,000,000	21,722,632	81,168		4,273,321	21,803,800
New_order	111,600,000	1,988,424	5,032	99,673		2,093,129
Orders	372,000,000	12,146,944	27,704		5,273,969	12,174,648
Order_line	3,719,992,214	243,933,920	574,552		81,361,232	244,508,472
Item	100,000	9,416	120	477		10,013
Stock	1,240,000,000	396,800,000	836,816	19,881,841		417,518,657
Total		947,161,904	18,405,816	34,354,040	90,908,522	999,921,760
	MB					
Dynamic Space	271,292	Sum of Data for Order, Orderline and History				
Static Space	705,194	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)				
Daily Growth	49,100	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed				
60 Day Space MB	3,651,199					
60 Day Space GB	3,565.62	GB				
Log Size	520,000.00	MB				
KB Per New Order	6.56	KB				
8 hr log MB	431,258	MB				
8 hr log GB	421.15	GB				
Space Usage	GB Needed	Disks Measured	GB Priced	Disk Size	Formatted Size	
60 Day Space DB	3,566	504	17,035.20	36GB	33.80	
			0.00			
			0.00			
Total DB			17,035.20			
8-hr log + mirror	842	16	1,081.60	72GB	67.60	
OS, Swap	3	2	67.60			
Total Storage	4,410.92	GB	18,184.40	GB		

	MSSQL_cs_fg	MSSQL_misc_fg
		1,504
		14,608
	301,796,930	
		26,077,121
		2,093,129
		17,448,617
		325,869,704
		10,013
	417,518,657	
	719,315,587	371,514,695
files=	12	12
size=	8,192,000	4,352,000
Total=	98,304,000	52,224,000
8K blocks	786,432,000	417,792,000
	OK	OK

tpmC	140,264									
	Data Before KB	Index Before KB	Data After KB	Index After KB	Data Grow KB	Index Grow KB	Total Grow KB	KB/New- Order	8-Hr Growth KB	8-Hr Growth MB
History	21,722,632	81,168	23,831,776	155,200	2,109,144	74,032	2,183,176	0.0635	4,273,321.08	4,173.17
Order	12,146,944	27,704	14,816,696	52,344	2,669,752	24,640	2,694,392	0.0783	5,273,968.81	5,150.36
Order-Line	243,933,920	574,552	284,980,416	1,094,296	41,046,496	519,744	41,566,240	1.2085	81,361,232.31	79,454.33
										88,777.85
	sum(*) Before		sum(*) After		Num New-Order					
d_next_o_id	372,124,000		406,520,217		34,396,217					
	Before MB		After MB		Grow MB			KB/New- Order	8-Hr Growth MB	8-Hr Growth GB
Log	27,167.97		247,491.11		220,323.14			6.5592	431,257.72	421.15
								6,716.5978	bytes	
520,000	5.2246099		47.594444							
Database tpcc log used (%)										

Appendix E: *Third Party Quotes*

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
<http://www.microsoft.com/>

Microsoft

June 21, 2006

Hewlett-Packard Company
David Adams
20555 SH 249
Houston, TX 77070

Mr. Adams:

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars (\$).

Part Number	Description	Unit Price	Quantity	Price
810-03150	SQL Server 2005 Enterprise x64 Edition <i>Per Processor License</i> <i>Discount Schedule: Open Program - No Level</i> <i>Unit Price reflects a 4% discount from the retail unit price of \$24,999.</i>	\$23,911	2	\$47,822
P72-00274	Windows Server 2003 Enterprise x64 Edition <i>Server License Only - No CALs</i> <i>Discount Schedule: Open Program - No Level</i> <i>Unit Price reflects a 42% discount from the retail unit price of \$3,999.</i>	\$2,334	1	\$2,334
C11-00821	Windows 2000 Server <i>Server License Only - No CALs</i> <i>Discount Schedule: No Level</i> <i>Unit Price reflects a 8% discount from the retail unit price of \$799.</i>	\$738	8	\$5,904
254-00170	Visual C++ Standard Edition <i>No Discounts Applied</i>	\$109	1	\$109
N/A	Microsoft Problem Resolution Services <i>Professional Support</i> <i>(1 Incident)</i>	\$245	1	\$245

All products are currently orderable through Microsoft's normal distribution channels. A list of these distribution channels can be found at <http://www.microsoft.com/products/info/render.aspx?type=mnpp&content=22%2flicensing&View=22>.

Defect support is included in the purchase price. Additional support is available from Microsoft PSS on an incident by incident basis at \$245 per call.

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.

Reference ID: PCbrca0618058267.

Please include this Reference ID in any correspondence regarding this price quote.

- Home
- WE ARE ANTI SPAM
- Blacklisted Brands
- Software
- Storage
- Barcode
- Hardware
- Housewares And Tools
- Networking
- Print servers
- SCSI
- Miscellaneous Items
- Printing Supplies and Cables
- Cables
- Network Cables & Parts
Cat5 Cat5e Cat6
- Power

LanAdapters.com



10 foot Category 5E Non Booted Network Patch Cables (Cat 5e)(backwards c

Cat 5E LIFETIME WARRANTY

cblc5ENB10 Regular price: \$3.00 Sale price: \$2.02, 50/\$76.00, 100/\$141.00, 200/\$262.00, 400/\$484.00 color:

- Show Order
- Privacy Policy
- Info & Shipping Notes
& Ways to delay
Processing of order
- Search
- Index
-

Appendix F:

Price Verification

Description	Part Number	Order Date	Order Method	Price Verification
HP Smart Array P800/512MB SAS Controller	381513-B21	11/22/2006	Note 1	Note 2
HP StorageWorks MSA-60 Storage	418408-B21	11/22/2006	Note 1	Note 2
HP StorageWorks MSA-60 Storage (10% Spares)	418408-B21	11/22/2006	Note 1	Note 2

Note 1 = HP Direct : 800-203-6748.
Note 2 = These components are not immediately orderable. For price verification before order date: e-mail hp.pricing.desk@hp.com