



Hewlett-Packard Company

TPC Benchmark™ C
Full Disclosure Report
for
ProLiant DL380 G3
using
Microsoft SQL Server 2000 Enterprise Edition
and
Windows Server 2003 Enterprise Edition

First Edition
May 2003

First Edition – May 2003

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 2003 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., 2003

HP, NonStop, ProLiant DL380G3, and ProLiant are registered trademarks of Hewlett-Packard Company.

Microsoft, Windows Server 2003 and SQL Server 2000 are registered trademarks of Microsoft Corporation.

Pentium III 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.

Table of Contents

TABLE OF CONTENTS	III
PREFACE	V
TPC BENCHMARK C OVERVIEW	V
ABSTRACT	VI
OVERVIEW	VI
TPC BENCHMARK C METRICS	VI
STANDARD AND EXECUTIVE SUMMARY STATEMENTS	VI
AUDITOR	VI
GENERAL ITEMS	10
TEST SPONSOR	10
APPLICATION CODE AND DEFINITION STATEMENTS	10
PARAMETER SETTINGS	10
CONFIGURATION ITEMS	10
CLAUSE 1 RELATED ITEMS	13
TABLE DEFINITIONS	13
PHYSICAL ORGANIZATION OF DATABASE	13
<i>Benchmarked Configuration:</i>	13
PRICED CONFIGURATION VS. MEASURED CONFIGURATION:	13
INSERT AND DELETE OPERATIONS	14
PARTITIONING	14
REPLICATION, DUPLICATION OR ADDITIONS	14
CLAUSE 2 RELATED ITEMS	15
RANDOM NUMBER GENERATION	15
INPUT/OUTPUT SCREEN LAYOUT	15
PRICED TERMINAL FEATURE VERIFICATION	15
PRESENTATION MANAGER OR INTELLIGENT TERMINAL	15
TRANSACTION STATISTICS	15
QUEUEING MECHANISM	16
CLAUSE 3 RELATED ITEMS	17
TRANSACTION SYSTEM PROPERTIES (ACID)	17
ATOMICITY	17
<i>Completed Transactions</i>	17
<i>Aborted Transactions</i>	17
CONSISTENCY	17
ISOLATION	17
DURABILITY	18
<i>Durable Media Failure</i>	18
<i>Instantaneous Interruption and Loss of Memory</i>	18
CLAUSE 4 RELATED ITEMS	20
INITIAL CARDINALITY OF TABLES	20
DATABASE LAYOUT	20
TYPE OF DATABASE	20
DATABASE MAPPING	21
60 DAY SPACE	21
CLAUSE 5 RELATED ITEMS	22

THROUGHPUT.....	22
KEYING AND THINK TIMES	22
RESPONSE TIME FREQUENCY DISTRIBUTION CURVES AND OTHER GRAPHS.....	23
STEADY STATE DETERMINATION.....	28
WORK PERFORMED DURING STEADY STATE	28
MEASUREMENT PERIOD DURATION	28
REGULATION OF TRANSACTION MIX	29
TRANSACTION STATISTICS.....	29
CHECKPOINT COUNT AND LOCATION	30
CHECKPOINT DURATION	30
CLAUSE 6 RELATED ITEMS	31
RTE DESCRIPTIONS	31
EMULATED COMPONENTS.....	31
FUNCTIONAL DIAGRAMS	31
NETWORKS	31
OPERATOR INTERVENTION.....	31
CLAUSE 7 RELATED ITEMS	32
SYSTEM PRICING.....	32
AVAILABILITY, THROUGHPUT, AND PRICE PERFORMANCE.....	32
COUNTRY SPECIFIC PRICING.....	32
USAGE PRICING	32
CLAUSE 9 RELATED ITEMS	33
AUDITOR'S REPORT	33
AVAILABILITY OF THE FULL DISCLOSURE REPORT	33

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 5.1, released February 11, 2003.

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 DL380G3. The operating system used for the benchmark was Windows Server 2003 Standard Edition. The DBMS used was Microsoft SQL Server 2000 Standard Edition.

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:

43,230.66 tpmC
\$3.71 per tpmC

The availability date is May 27, 2003.

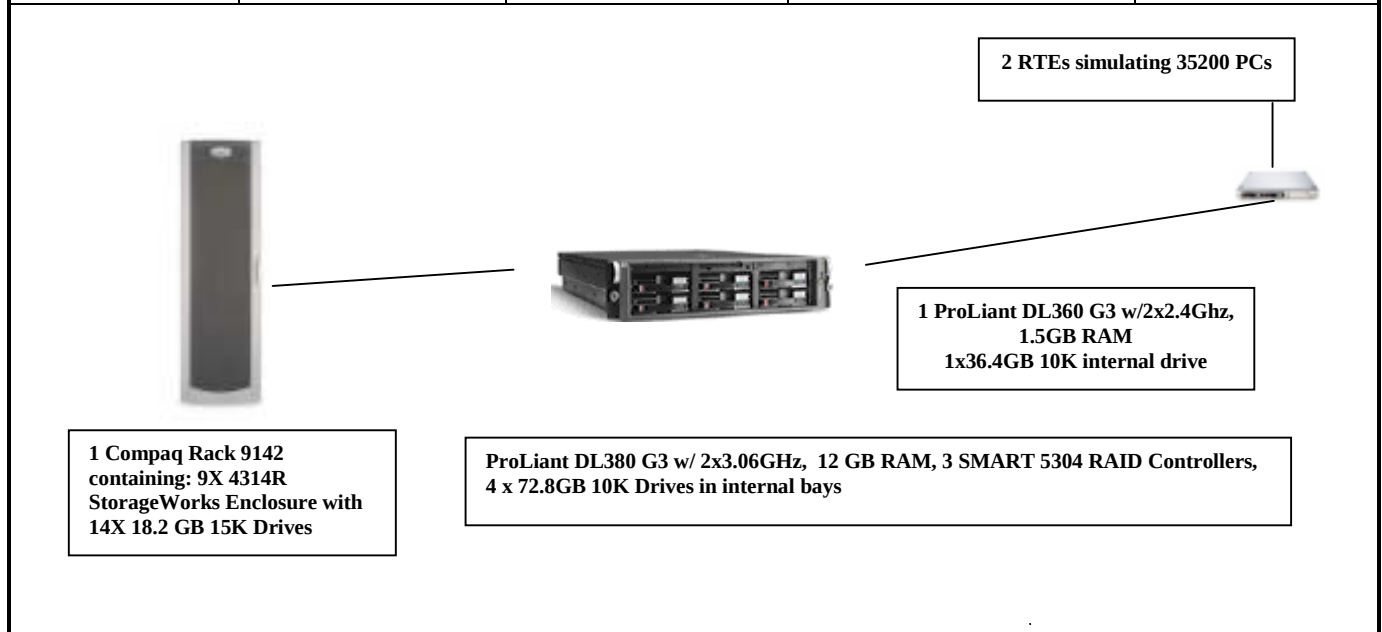
Standard and Executive Summary Statements

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		ProLiant DL380 G3 2P		TPC-C Rev. 5.1	
		c/s with 1 ProLiant DL360 G3		Report Date: May 29, 2003	
Total System Cost		TPC-C Throughput		Price/Performance	
\$160,353		43,230.66		\$3.71	
Processors		Database Manager		Operating System	
2 Intel Xeon 3.06GHz – Server 2 Intel Xeon 2.4GHz – Client		Microsoft SQL Server 2000 Enterprise Edition (SP3)		Microsoft Windows Server 2003, Enterprise Edition	
Other Software		Number of Users			
Microsoft Visual C++ Microsoft COM+		35200			



	Server		Each Client	
System Components	Quantity	Description	Quantity	Description
Processor	2	3.06 GHz Intel Xeon w/ 512K Cache	2	2.4GHz Intel Xeon w/ 512K cache
Memory	3	4 GB REG (2x2GB)	1	512MB (2x256MB)
Disk Controllers	1	Integrated SMART Array Controller	1	1024MB (2x512MB)
	3	SMART 5304 Array Controller	1	Integrated SMART Array Controller
Disk Drives	126	18.2 GB SCSI Drive	1	36.4 GB SCSI Drive
	4	72.8 GB SCSI Drive		
Total Storage		2584.4 GB		36.4 GB
9Tape Drives	1	12/24 GB DAT		

Hewlett-Packard		HP ProLiant DL380 G3 2P		TPC-C Rev. 5.1		
Company		Client/Server		Report Date:		29-May-03
Description	Part Number	Third Party	Unit Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware		Brand Pricing				
ProLiant DL380 G3 1x3.06GHz/533MHz FSB (512KB L2), 512MB (2x256MB), Two integrated Gigabit NIC, Integrated Smart Array Controller	310587-001	1	3,699	1	3,699	
3.06/533MHz Xeon 2nd processor DL380		1	1,159	1	1,159	
4G Reg PC2100 (2 x 2Gb)	300682-B21	1	11,404	3	34,212	
StorageWorks Enclosure Model 4314R - Rack-mountable	190209-001	1	2,955	9	26,595	
Rack Model 9142 (42U - Opal) - Flat Pallet	120663-B21	1	1,321	1	1,321	
Side Panel Kit - 9142 Rack	120670-B21	1	207	1	207	
Smart Array 5304/128 Controller	158939-B21	1	2,052	3	6,156	
S5500 15 carbon / silver monitor	261602-001	1	129	1	129	
3-Button Mouse-Carbon	231947-B21	1	5	1	5	
PS/2 Easy Access Internet Keyboard	265977-001	1	12	1	12	
TR5 10/20-Gigabyte Tape Drive - Carbon	294243-B22	1	299	1	299	
Pro UPS 500 (500VA/300 Watts; 110-127 VAC, 60Hz)	136386-001	1	146	1	146	
18.2GB 15Krpm U320 UNI HDD	286775-B22	1	399	126	50,274	
18.2GB 15Krpm U320 UNI HDD (10% spares)	286775-B22	1	399	13		5,187
72.8GB 10Krpm U320 UNI HDD (internal OS & log drives)	286714-B22	1	649	4	2,596	
FM-EL724-36 3YR 24X7 4HR ENTRY 300 SVR	162675-002	1	599	1		599
FM-4E724-36 3YR 24X7 4HR EMPTY DISK ENCL	171242-002	1	157	9		1,413
Subtotal					126,810	7,199
Server Software						
Database Server Support Package	PRO-PRORS-161	Microsoft	2	1,950	3	5,850
SQL Server 2000 Enterprise Edition (per processor)	810-00845	Microsoft	2	17,279	2	34,558
Visual C++ .Net Standard	254-00170	Microsoft	2	109	1	109
Windows Server 2003, Enterprise Edition	P72-00264	Microsoft	2	2,399	1	2,399
Subtotal					37,066	5,850
Client Hardware						
ProLiant DL360 G3 1x2.4GHz, 512MB (2x256MB), Two integrated Gigabit NIC, Integrated Smart Array Controller	292887-001	1	2,599	1	2,599	
2.4GHz Xeon Processor Option Kit - DL360G3	292891-B21	1	729	1	729	
1GB PC2100 DDR (2x512MB)	300679-B21	1	649	1	649	
36.4GB 10Krpm U320 UNI HDD (internal OS drive)	286713-B22	1	339	1	339	
FM-EL724-36 3YR 24X7 4HR ENTRY 300 SVR	162675-002	1	599	1		599
Subtotal					4,316	599
Client Software						
Windows 2000 Server 32-bit	C11-00821	Microsoft	2	738	1	738
Subtotal					738	0
User Connectivity						
7 ft. CAT5e Patch cable	CBLC57	LanAdapters	3	1	3	
Subtotal					3	0
Large Purchase and Net 30 discount (See Note 1)		16.0%	1		(\$20,980)	(\$1,248)
Total					\$147,953	\$12,400
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: \$160,353		
				tpmC Rating: 43230.66		
				\$ / tpmC: \$3.71		
Pricing: 1=HP 2=Microsoft 3=LanAdapters.com						
Note 1 = Discount based on HP Direct guidance with large purchase and Net 30 discount.						
Note: The benchmark results and test methodology were audited by Lorna Livingtree of Performance Metrics, Inc.						

Numerical Quantities Summary

MQTH, Computed Maximum Qualified Throughput

43,230.66 tpmC

Response Times (in seconds)	Average	90%	Maximum
New-Order	0.69	1.02	8.47
Payment	0.58	0.91	4.47
Order-Status	0.61	0.93	9.94
Delivery (interactive portion)	0.13	0.20	0.76
Delivery (deferred portion)	0.23	0.35	0.95
Stock-Level	1.56	2.03	4.38
Menu	0.13	0.21	0.87

Transaction Mix, in percent of total transaction

New-Order	44.88%
Payment	43.03%
Order-Status	4.04%
Delivery	4.03%
Stock-Level	4.03%

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	18.00/0.00	18.01/12.17	18.69/122.71
Payment	3.00/0.00	3.02/12.16	3.72/122.71
Order-Status	2.00/0.00	2.01/10.15	2.69/101.50
Delivery (interactive)	2.00/0.00	2.01/5.12	2.71/51.00
Stock-Level	2.00/0.00	2.01/5.10	2.48/51.00

Test Duration

Ramp-up time	20 minutes
Measurement interval	120 minutes
Transactions (all types) completed during measurement interval	12,026,116
Ramp down time	2 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 diagrams for both the tested and priced systems are included on the following pages.

Figure 1. Benchmarked Configuration

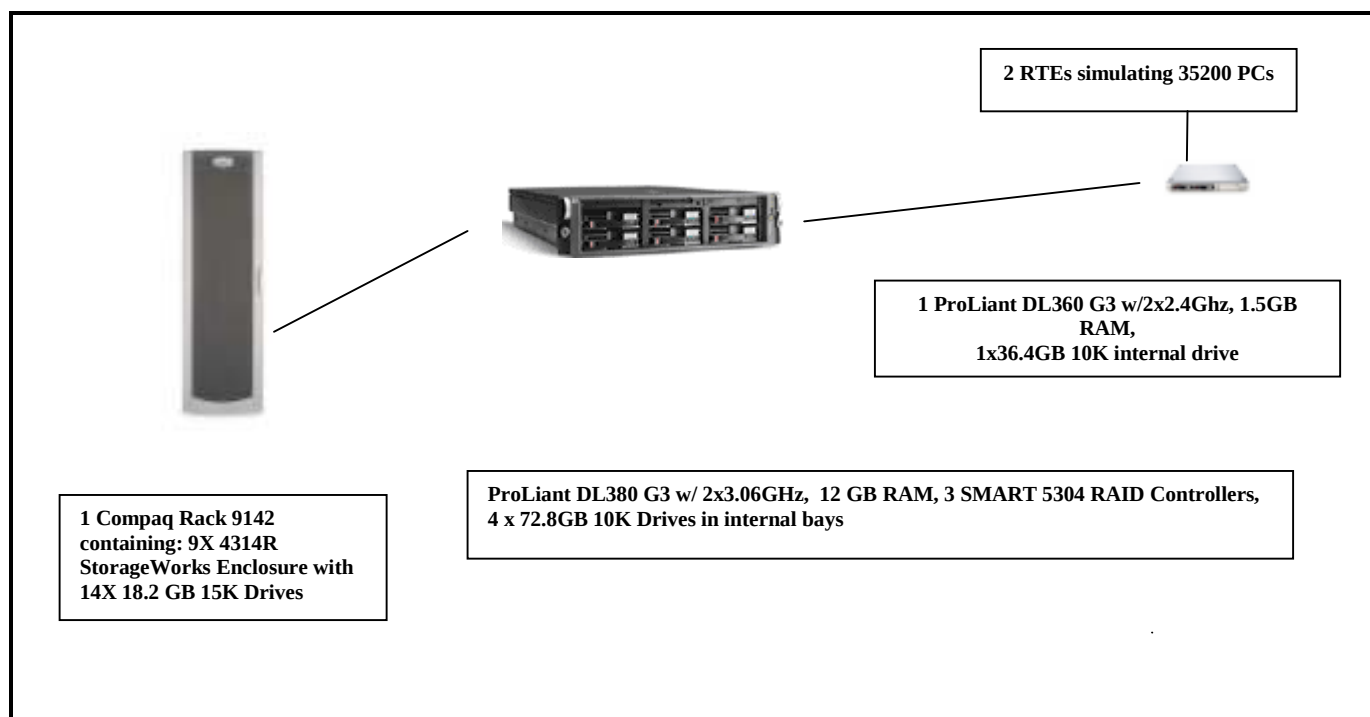
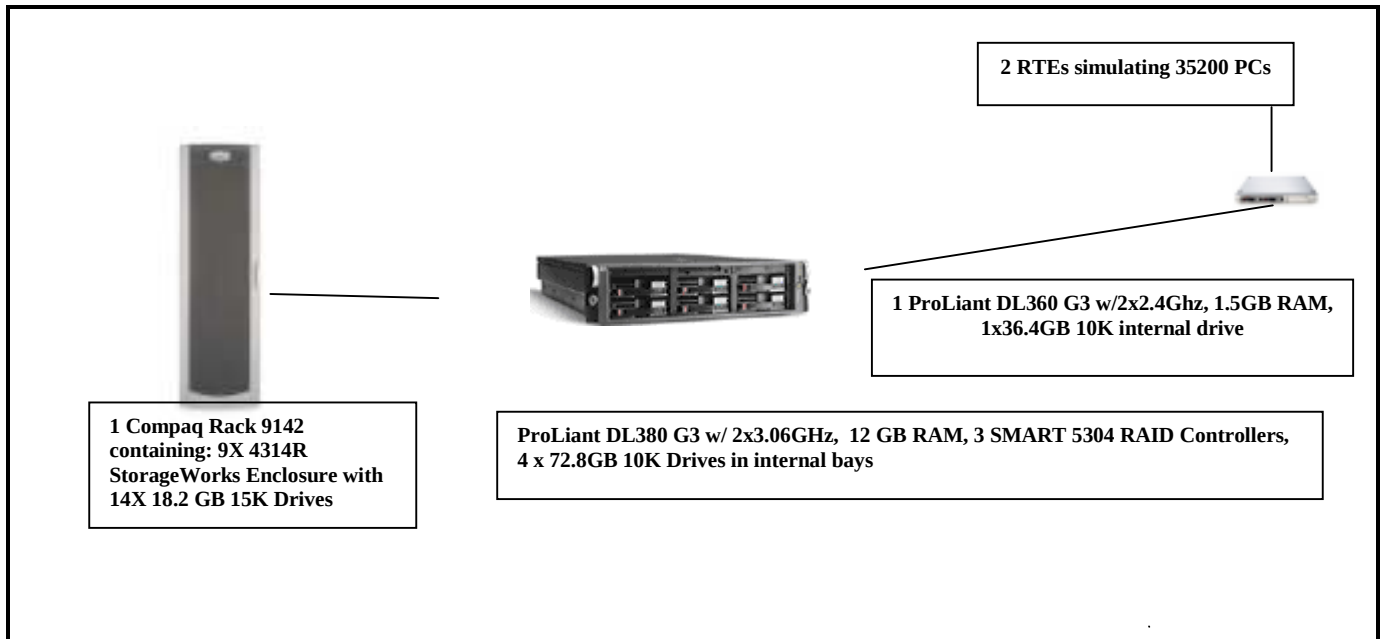


Figure 2. Priced Configuration



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: 126 18.2GB 15K drives for the database data, and 4 72.8GB 10K drives for the transaction log and operating system. Fourteen drives (18.2GB 15K) were connected to each of the controllers in slots 1-3 and the four drives (72.8GB 10K) were connected to the integrated controller.

Benchmarked Configuration:

Integrated SMART Controller, Port 1

<u>LOGICAL DRIVE C:</u>	<u>Total Capacity = 16.00 GB</u>	<u>RAID 0+1</u>
Microsoft Server 2003 Standard Edition, MSSQL_tpcc_root.mdf		
<u>LOGICAL DRIVE E:</u>	<u>Total Capacity = 119.66 GB</u>	<u>RAID 0+1</u>
MSSQL_tpcc_log		

SMART-5304 Controller, Slot 1, Array A

<u>LOGICAL DRIVE F:</u>	<u>Total Capacity = 68.55 GB</u>	<u>RAID 0</u>
MSSQL_cs0		
<u>LOGICAL DRIVE G:</u>	<u>Total Capacity = 32.26 GB</u>	<u>RAID 0</u>
MSSQL_misc0		
<u>LOGICAL DRIVE X:</u>	<u>Total Capacity = 305.20 GB</u>	<u>RAID 0+1</u>
Tpccbackup0		

SMART-5304 Controller, Slot 2, Array A

<u>LOGICAL DRIVE H:</u>	<u>Total Capacity = 68.55 GB</u>	<u>RAID 0</u>
MSSQL_cs1		
<u>LOGICAL DRIVE I:</u>	<u>Total Capacity = 32.26 GB</u>	<u>RAID 0</u>
MSSQL_misc1		
<u>LOGICAL DRIVE Y:</u>	<u>Total Capacity = 305.20 GB</u>	<u>RAID 0+1</u>
Tpccback1		

SMART-5304 Controller, Slot 3, Array A

<u>LOGICAL DRIVE J:</u>	<u>Total Capacity = 68.55 GB</u>	<u>RAID 0</u>
MSSQL_cs2		
<u>LOGICAL DRIVE K:</u>	<u>Total Capacity = 32.26 GB</u>	<u>RAID 0</u>
MSSQL_misc2		
<u>LOGICAL DRIVE Z:</u>	<u>Total Capacity = 305.20 GB</u>	<u>RAID 0+1</u>
Tpccbackup2		

Priced Configuration vs. Measured Configuration:

The measured and priced configuration differ in that the measured configuration used disk drives for database backup and the priced configuration used a DAT drive for backup.

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 were 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	84.99%
	Remote warehouse payments	15.01%

Statistic		Value
	Accessed by last name	59.98%
Order Status	Accessed by last name	60.04%
Transaction Mix	New Order	44.88%
	Payment	43.03%
	Order status	4.04%
	Delivery	4.03%
	Stock level	4.03%

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 that 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:

- A new database containing 352 warehouses (10% of the warehouses of the full database) was created and was backed up to extra disks.
- 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 3520 users.
- The test was allowed to run for a minimum of 10 minutes.
- One log disk was removed from the drive cabinet.
- Since the disk was mirrored, processing was not interrupted. This was verified by checking the users status on the RTE.
- One of the data disks was removed from the drive cabinet.
- When Microsoft SQL Server recorded errors about not being able to access the database, the RTE was shut down.
- A dump of the transaction log was taken and the Microsoft SQL Server was shutdown.
- A new log disk was inserted into the log drive cabinet. A new data disk was inserted into the data drive cabinet. After the RAID recovery process finished, the system was rebooted and Microsoft SQL Server was started.
- 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 step 13 and 14 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 3520 warehouses under a full load of 35200 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 35200 users.
- The test was allowed to run for a minimum of 10 minutes.
- A checkpoint was performed.
- The system crash and loss of memory were induced by switching the power off. The power cords were then physically removed from the SUT. No battery backup or Uninterruptible Power Supply (UPS) were used to preserve the contents of memory.
- The RTE was shutdown.
- 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 10 and 11 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

Table2	Cardinality as built
Warehouse	3,520
District	35,200
Customer	105,600,000
History	105,600,000
Orders	105,600,000
New Order	31,680,000
Order Line	1,056,005,900
Stock	352,000,000
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 3 SMART-5304 Array controllers with 4 SCSI channels and 1 integrated SMART Array controller with 2 SCSI channels. Each controller is capable of accessing up to 14 disk drives per channel, and supports RAID 0, RAID 0+1, and RAID 5 per each logical volume configured. The data tables were stored on 3 RAID arrays of (42) 18.42GB 15K drives each. Each of these controllers also housed a RAID 0+1 volume used for backup of the database. The integrated SMART Array controller had one array consisting of (4) 72.8GB 10K drives with a RAID 0+1 logical volume for the database log and operating system. The Array Accelerators on the data controllers were disabled for all logical drives on these controllers. The Array Accelerator on the integrated controller for the transaction log had the cache disabled. All RAID volumes used hardware RAID.

Section 1.2 of this report details the distribution of database tables across all disks. The code that creates the filegroups 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/1, 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 2000 Enterprise Edition 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.

The details of both the 8-hour transaction log space requirement and the 60-day space requirement is shown in Appendix D.

Clause 5 Related Items

Throughput

Measured tpmC must be reported

Measured tpmC 43,230.66 tpmC
Price per tpmC \$3.71 per tpmC

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	0.69	1.02	8.47
Payment	0.58	0.91	4.47
Order-Status	0.61	0.93	9.94
Interactive Delivery	0.13	0.20	0.76
Deferred Delivery	0.23	0.35	0.95
Stock-Level	1.56	2.03	4.38
Menu	0.13	0.21	0.87

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	18.00	18.01	18.69
Payment	3.00	3.02	3.72
Order-Status	2.00	2.01	2.69
Interactive Delivery	2.00	2.01	2.71
Stock-Level	2.00	2.01	2.48

Table 5.4: Think Times

Type	Minimum	Average	Maximum
New-Order	0.00	12.17	121.71
Payment	0.00	12.16	121.71
Order-Status	0.00	10.15	101.50
Interactive Delivery	0.00	5.12	51.00
Stock-Level	0.00	5.10	51.00

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

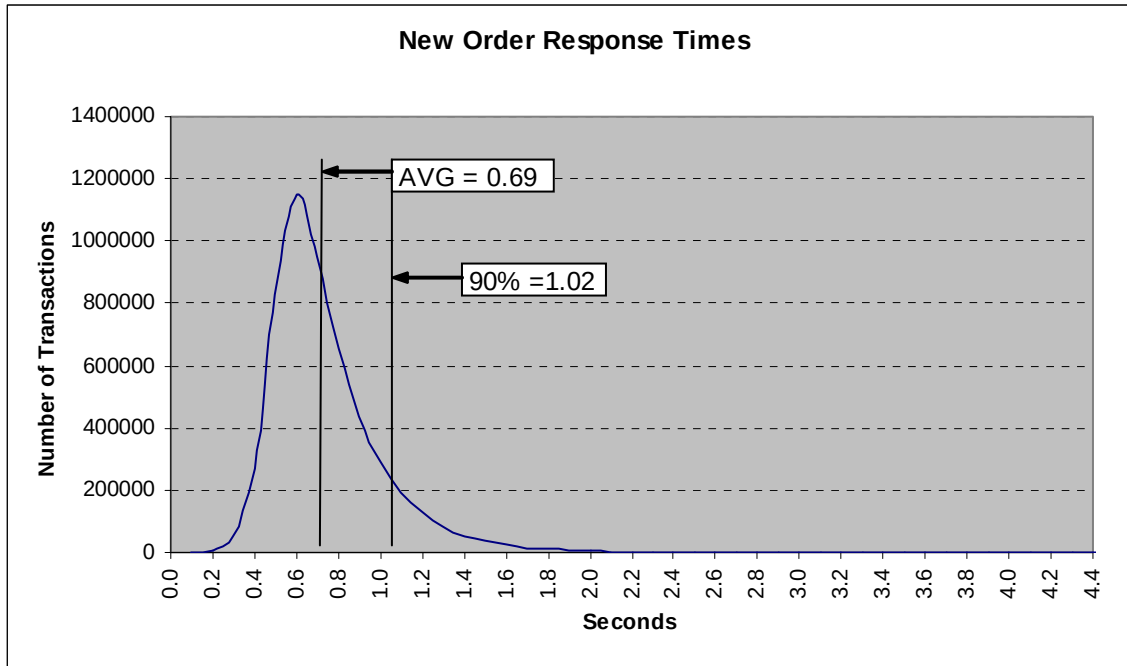


Figure 4. Payment Response Time Distribution

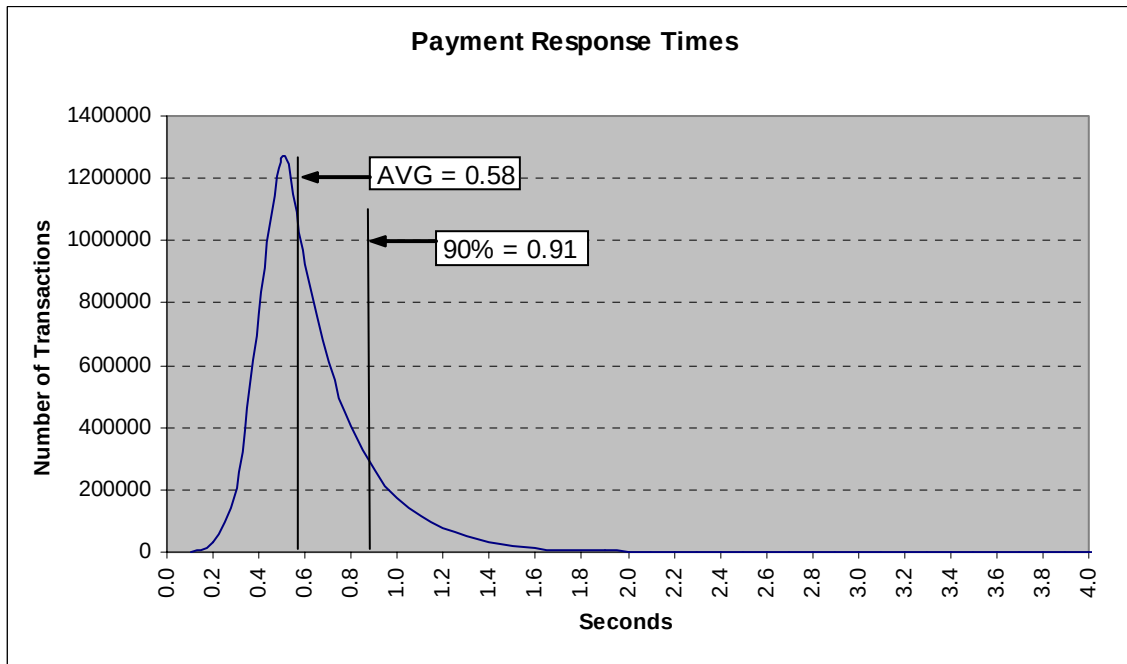


Figure 5. Order Status Response Time Distribution

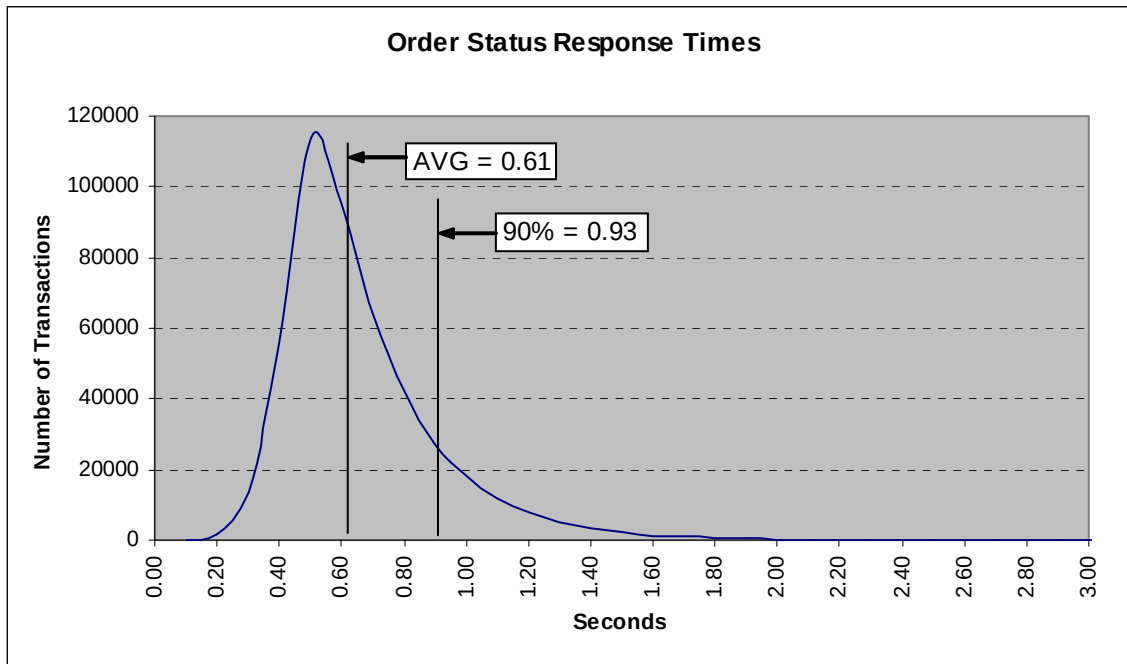


Figure 6. Delivery Response Time Distribution

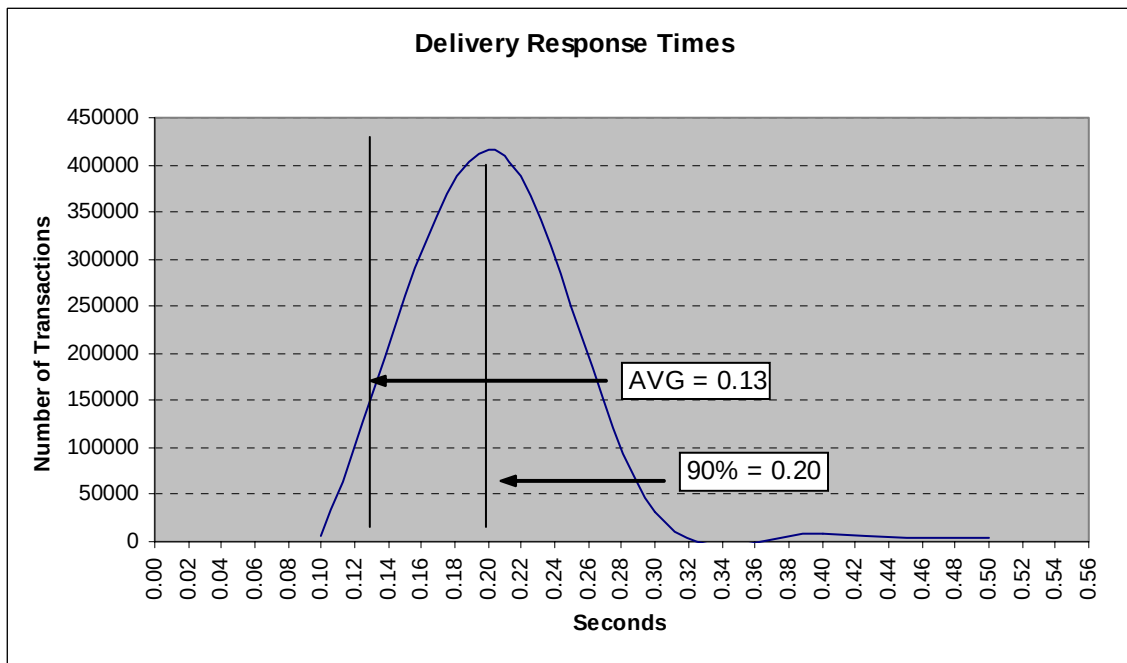


Figure 7. Stock Level Response Time Distribution

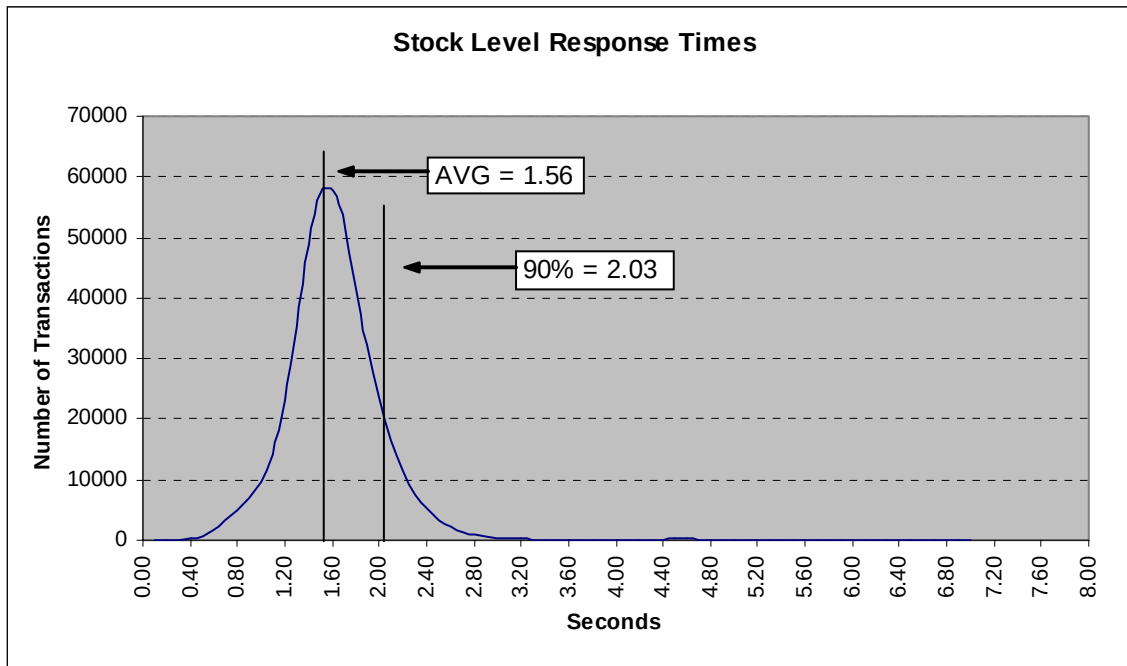


Figure 8. Response Time vs. Throughput

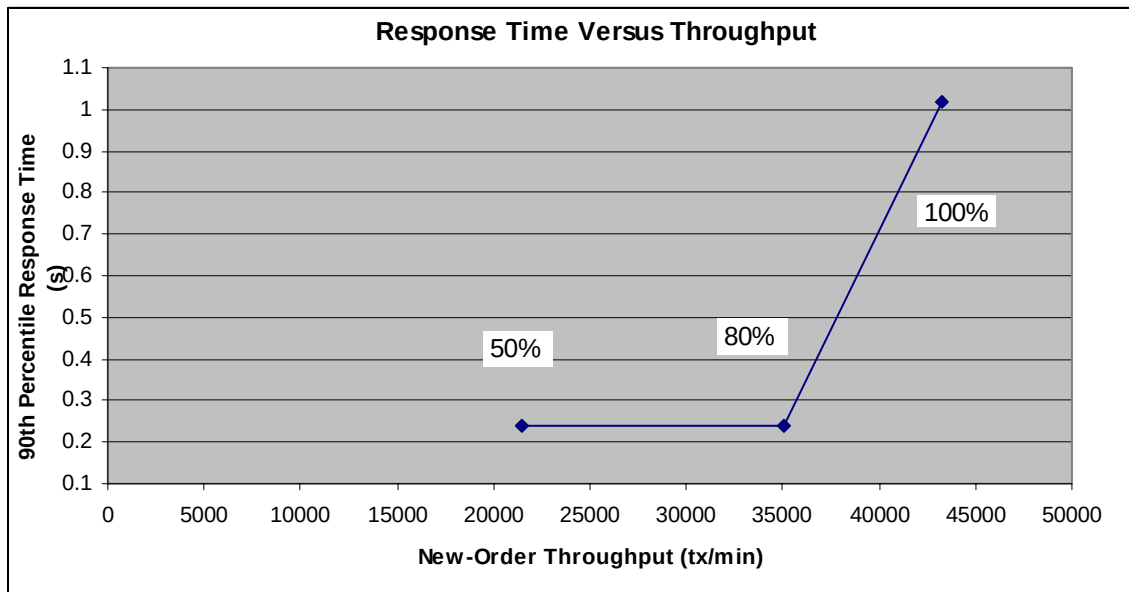
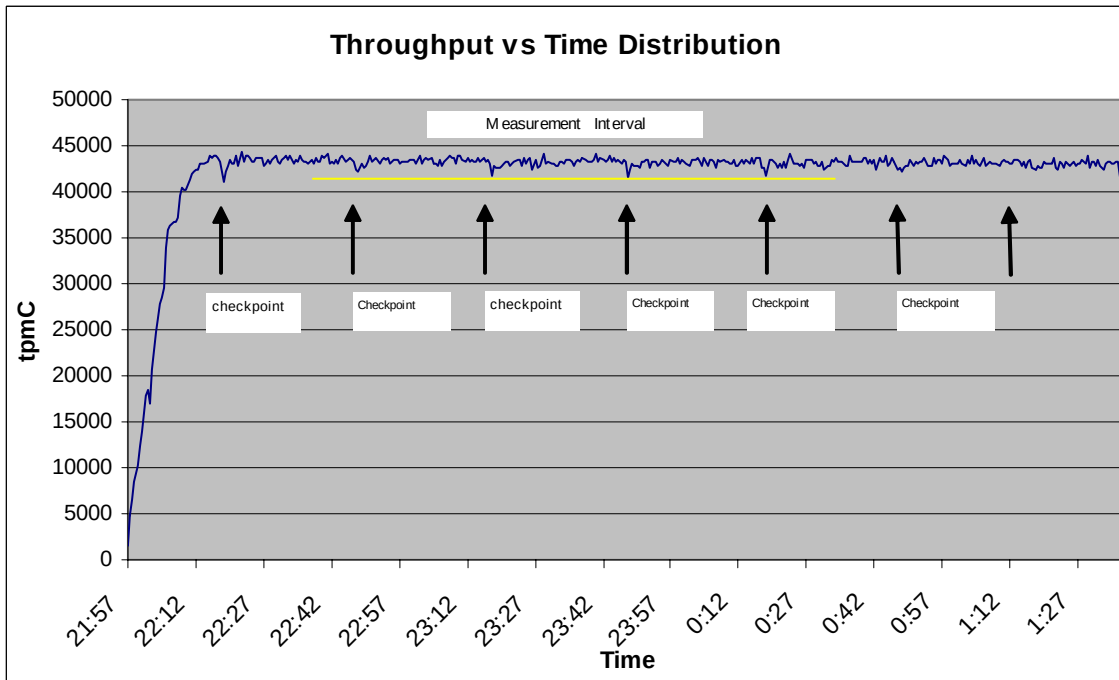


Figure 9. New Order Think Time Distribution



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 Ethernet LANs using ODBC and RPC calls.

To perform checkpoints at specific intervals, the SQL Server *recovery interval* was set to 60 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 so that the checkpoint interval was an integral multiple of the measurement interval, which 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 10.

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	84.99%
	Remote warehouse payments	15.01%
	Accessed by last name	59.98%
Delivery	Skipped transactions (interactive)	0
	Skipped transactions (deferred)	0
Order Status	Accessed by last name	60.04%
Transaction Mix	New Order	44.88%
	Payment	43.03%
	Order status	4.04%
	Delivery	4.03%
	Stock level	4.03%

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 21 minutes after the start of the ramp-up. Subsequent checkpoints occurred every 30 minutes. 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
11:02:25 pm	14 minutes, 59 seconds
11:32:14 pm	14 minutes, 52 seconds
12:02:20 pm	15 minutes, 1 seconds
12:32:13 pm	14 minutes, 57 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 are 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 1 HP ProLiant server. This driver machine 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 which 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, 2 driver (RTE) machines were connected through a 10/100/1000 switch to the client machine at 1000Mbps, thus providing the path from the RTE to the client. The server (SUT) was connected to the client through a CAT5e cable at 1000Mbps.

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	43,230.66 tpmC
• Price per tpmC	\$3.71 per tpmC
• Availability	May 27, 2003

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:

- 1 Microsoft Windows 2000 Server
- 1 Microsoft Windows Server 2003, Enterprise Edition
- 1 Microsoft SQL Server 2000 Enterprise Edition (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.
137 Yankton St., Suite 101
Folsom, CA 95630
(phone) (916) 985-1131
(fax) (916) 985-1185
e-mail: lorna@perfmetrics.com

Availability of the Full Disclosure Report

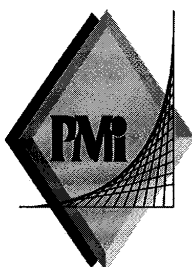
The Full Disclosure Report must be readily available to the public at a reasonable charge, similar to the charges for similar documents by the test sponsor. The report must be made available when results are made public. In order to use the phrase "TPC Benchmark™ C", the Full Disclosure Report must have 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:

Transaction Processing Performance Council
c/o Shanley Public Relations
777 North First Street, Suite 600
San Jose, CA 95112-6311

or

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



PERFORMANCE METRICS INC.
TPC Certified Auditors

May 28, 2003

Mr. Jim Barrett
Systems Software Engineer
Compaq Computer Corporation
20555 SH 249
Houston, TX 77070

I have verified by remote and in real time the TPC Benchmark™ C client/server for the following configuration on each node:

Platform: ProLiant DL380 G3
Database Manager: Microsoft SQL Server 2000 Enterprise Edition 32-bit
Operating System: Microsoft Windows 2003 Enterprise Edition
Transaction Monitor: Microsoft COM+

Servers: ProLiant DL380 G3 with:				
CPU's	Memory	Disks (total)	90% Response	TpmC
2 Pentium III Xeon @ 3.06 GHz	Main: 12 GB Cache: 512 KB	126 @ 18GB 4 @ 72GB	1.02	43,230.66
1 Client: ProLiant DL360 G3 with:				
2 Pentium III Xeon @ 2.4 GHz	Main: 1535 MB Cache: 512 KB	1 @ 36GB	Na	Na

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 and populated.
- The database was properly scaled with 3520 warehouses, all of which were active during the measured interval.
- The ACID properties were successfully demonstrated.

137 Yankton St. Suite 101, Folsom, CA 95630
(916) 985-1131 fax: (916) 985-1185 email: Lorna@PerfMetrics.com

Page 1

PERFORMANCE METRICS INC.
TPC Certified Auditors

- 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 day space calculation was verified.
- The controller cache was disabled on the log disk controllers.
- The steady state portion of the test was 120 minutes.
- One checkpoint was taken before the measured interval.
- Four checkpoints were taken during the measured interval.
- The system pricing was checked for major components and maintenance.
- Third party quotes were verified for compliance.

Auditor Notes: None.

Sincerely,



Lorna Livingtree
Auditor

Appendix A: Source Code

The client source code is listed below.

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;};
    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      NewOrder;
            PAYMENT_DATA         Payment;
            DELIVERY_DATA        Delivery;
            STOCK_LEVEL_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)
        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_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_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_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_ENTRY_CHAIN(CTPCC_Common)
    END_COM_MAP()

```

```

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;}
};

```

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 *)&Reg->szDbUser,
&size) != ERROR_SUCCESS )
            pReg->szDbUser[0] = 0;

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

        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];
} TPCCREGISTRYDATA, *PTPCCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );

```

WEBCLNT.DSP

```

# Microsoft Developer Studio Project File - Name="webclnt" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Application" 0x0101

```

```

CFG=webclnt - Win32 Release
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak" CFG="webclnt - Win32 Release"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "webclnt - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "webclnt - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

```

```

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

```

```

!IF "$(CFG)" == "webclnt - Win32 Release"

```

```

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir ".\Release"
# PROP BASE Intermediate_Dir ".\Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\Release"
# PROP Intermediate_Dir ".\Release"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386

```

```

!ELSEIF "$(CFG)" == "webclnt - Win32 Debug"

```

```

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir ".\Debug"
# PROP BASE Intermediate_Dir ".\Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1

```

```
# PROP Output_Dir ".\Debug"
# PROP Intermediate_Dir ".\Debug"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/c
# ADD CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386

!ENDIF

# Begin Target

# Name "webclnt - Win32 Release"
# Name "webclnt - Win32 Debug"
# End Target
# End Project
```

Webclnt.dsw

Microsoft Developer Studio Workspace File, Format Version 6.00
WARNING: DO NOT EDIT OR DELETE THIS WORKSPACE FILE!

```
#####

Project: "db_dblib_dll"=.\db_dblib_dll\db_dblib_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####

Project: "db_odbc_dll"=.\db_odbc_dll\db_odbc_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####
```

Project: "install"=.\install\install.dsp - Package Owner=<4>

```
Package=<5>
{{{
}}}

Package=<4>
{{{
  Begin Project Dependency
  Project_Dep_Name isapi_dll
  End Project Dependency
  Begin Project Dependency
  Project_Dep_Name tuxapp
  End Project Dependency
  Begin Project Dependency
  Project_Dep_Name db_dblib_dll
  End Project Dependency
  Begin Project Dependency
  Project_Dep_Name db_odbc_dll
  End Project Dependency
  Begin Project Dependency
  Project_Dep_Name tm_com_dll
  End Project Dependency
  Begin Project Dependency
  Project_Dep_Name tm_tuxedo_dll
  End Project Dependency
  Begin Project Dependency
  Project_Dep_Name tpcc_com_all
  End Project Dependency
  Begin Project Dependency
  Project_Dep_Name tpcc_com_ps
  End Project Dependency
}}}
```

#####

Project: "isapi_dll"=.\isapi_dll\isapi_dll.dsp - Package Owner=<4>

```
Package=<5>
{{{
}}}

Package=<4>
{{{
  Begin Project Dependency
  Project_Dep_Name db_dblib_dll
  End Project Dependency
  Begin Project Dependency
  Project_Dep_Name db_odbc_dll
  End Project Dependency
  Begin Project Dependency
  Project_Dep_Name tm_tuxedo_dll
  End Project Dependency
  Begin Project Dependency
  Project_Dep_Name tm_com_dll
  End Project Dependency
  Begin Project Dependency
  Project_Dep_Name tm_encina_dll
  End Project Dependency
}}}
```

#####


```

Project: "tm_com_dll"=.\tm_com_dll\tm_com_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_all
    End Project Dependency
}}}

#####

Project: "tm_encina_dll"=.\tm_encina_dll\tm_encina_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####

Project: "tm_tuxedo_dll"=.\tm_tuxedo_dll\tm_tuxedo_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####

Project: "tpcc_com_all"=.\tpcc_com_all\tpcc_com_all.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
}}}

#####

Project: "tpcc_com_ps"=.\tpcc_com_ps\tpcc_com_ps.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

```

```

Package=<4>
{{{
}}}

#####

Project: "tuxapp"=.\tuxapp\tuxapp.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
}}}

#####

Global:

Package=<5>
{{{
}}}

Package=<3>
{{{
}}}

#####

```

db_dblib_dll.dsp

```

# Microsoft Developer Studio Project File - Name="db_dblib_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=db_dblib_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "db_dblib_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "db_dblib_dll.mak" CFG="db_dblib_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "db_dblib_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link
Library")
!MESSAGE "db_dblib_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_dblib_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")

```

```

!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF "$(CFG)" == "db_dblib_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ntwdlib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /machine:I386 /out:".bin\tpcc_dblib.dll"

!ELSEIF "$(CFG)" == "db_dblib_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe

```

```

# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ntwdlib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_dblib.dll"
/pdbtype:sept

!ELSEIF "$(CFG)" == "db_dblib_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_dblib"
# PROP BASE Intermediate_Dir "db_dblib"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /ZI /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 ntwdlib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_dblib.dll"
/pdbtype:sept
# ADD LINK32 icap.lib ntwdlib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_dblib.dll"
/pdbtype:sept

!ENDIF

# Begin Target

# Name "db_dblib_dll - Win32 Release"
# Name "db_dblib_dll - Win32 Debug"
# Name "db_dblib_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp"
# Begin Source File

SOURCE=.src\tpcc_dblib.cpp
# End Source File
# End Group
# Begin Group "Header"

```

```
# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=..\src\tpcc_dblib.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project
```

db_odbc_dll.dsp

```
# Microsoft Developer Studio Project File - Name="db_odbc_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=db_odbc_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "db_odbc_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "db_odbc_dll.mak" CFG="db_odbc_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "db_odbc_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_odbc_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_odbc_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF "$(CFG)" == "db_odbc_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
```

```
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386 /out:".bin\tpcc_odbc.dll"

!ELSEIF "$(CFG)" == "db_odbc_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_odbc.dll"
/pdbtype:sept

!ELSEIF "$(CFG)" == "db_odbc_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_odbc_"
# PROP BASE Intermediate_Dir "db_odbc_"
```

```

# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /Mdd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /Zi /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_odbc.dll"
/pdbtype:sept
# ADD LINK32 icap.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_odbc.dll"
/pdbtype:sept

!ENDIF

# Begin Target

# Name "db_odbc_dll - Win32 Release"
# Name "db_odbc_dll - Win32 Debug"
# Name "db_odbc_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp"
# Begin Source File

SOURCE=.\src\tpcc_odbc.cpp
# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=.\common\src\error.h
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_odbc.h
# End Source File
# Begin Source File

SOURCE=.\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=.\common\src\txn_base.h
# End Source File

```

```

# End Group
# End Target
# End Project

```

dllldata.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
/dllldata 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 dllldata file */

```

error.h

```

/* FILE: ERROR.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 error exception classes.
*
* Change history:
* 4.20.000 - updated rev number to match kit
* 4.21.000 - fixed bug: ~CBaseErr needed to be declared virtual
*/

```

```

#pragma once

#ifndef _INC_STRING
#include <string.h>
#endif

const int m_szMsg_size = 512;
const int m_szApp_size = 64;
const int m_szLoc_size = 64;

//error message structure used in ErrorText routines
typedef struct _SERRORMSG
{
    int iError; //error id of
    char szMsg[256]; //message to sent to
} SERRORMSG;

typedef enum _ErrorLevel
{
    ERR_FATAL_LEVEL = 1,
    ERR_WARNING_LEVEL = 2,
    ERR_INFORMATION_LEVEL = 3
} ErrorLevel;

#define ERR_TYPE_LOGIC -1 //logic error in program; internal error
#define ERR_SUCCESS 0 //success (a non-error error)
#define ERR_BAD_ITEM_ID 1 //expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST 2 //expected delivery post failed
#define ERR_TYPE_WEBDLL 3 //tpcc web generated error
#define ERR_TYPE_SQL 4 //sql server generated error
#define ERR_TYPE_DBLIB 5 //dblib generated error
#define ERR_TYPE_ODBC 6 //odbc generated error
#define ERR_TYPE_SOCKET 7 //error on communication socket client rte only
#define ERR_TYPE_DEADLOCK 8 //dblib and odbc only deadlock condition
#define ERR_TYPE_COM 9 //error from COM call
#define ERR_TYPE_TUXEDO 10 //tuxedo error
#define ERR_TYPE_OS 11 //operating system error
#define ERR_TYPE_MEMORY 12 //memory allocation error
#define ERR_TYPE_TPCC_ODBC 13 //error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB 14 //error from tpcc dblib txn module
#define ERR_TYPE_DELISRV 15 //delivery server error
#define ERR_TYPE_TXNLOG 16 //txn log error

```

```

#define ERR_TYPE_BCCONN 17 //Benchcraft connection class
#define ERR_TYPE_TPCC_CONN 18 //Benchcraft connection class
#define ERR_TYPE_ENCINA 19 //Encina error
#define ERR_TYPE_COMPONENT 20 //error from COM component
#define ERR_TYPE_RTE 21 //Benchcraft rte
#define ERR_TYPE_AUTOMATION 22 //Benchcraft automation errors
#define ERR_TYPE_DRIVER 23 //Driver engine errors
#define ERR_TYPE_RTE_BASE 24 //Framework errors
#define ERR_BUF_OVERFLOW 25 //Buffer overflow during receive

// TPC-W error types
#define ERR_TYPE_TPCW_CONN 50 //Benchcraft connection class
#define ERR_TYPE_TPCW_HTML 51 //error from TpcwHtml dll
#define ERR_TYPE_TPCW_USER 52 //error from TPC-W user class
#define ERR_TYPE_TPCW_ENG_BASE 53
#define ERR_TYPE_TPCW_ENG_OS 54
#define ERR_TYPE_HTML_RESP 55
#define ERR_TYPE_TPCC_ODBC 56
#define ERR_TYPE_SCHANNEL 57

#define ERR_INS_MEMORY "Insufficient Memory to continue."
#define ERR_UNKNOWN "Unknown error."
#define ERR_MSG_BUF_SIZE 512
#define INV_ERROR_CODE -1
#define ERR_INS_BUF_OVERFLOW "Insufficient Buffer size to recieve HTML pages."

class CBaseErr
{
public:
    CBaseErr(LPCTSTR szLoc = NULL)
    {
        m_idMsg = INV_ERROR_CODE;

        if (szLoc)
        {
            m_szLoc = new char[m_szLoc_size];
            strcpy(m_szLoc, szLoc);
        }
        else
            m_szLoc = NULL;

        m_szApp = new char[m_szApp_size];
        GetModuleFileName(GetModuleHandle(NULL), m_szApp, m_szApp_size);
    }
}

```

```

CBaseErr(int idMsg, LPCTSTR szLoc = NULL)
{
    m_idMsg          = idMsg;

    if (szLoc)
    {
        m_szLoc = new char[m_szLoc_size];
        strcpy(m_szLoc, szLoc);
    }
    else
        m_szLoc = NULL;

    m_szApp          = new char[m_szApp_size];
    GetModuleFileName(GetModuleHandle(NULL), m_szApp, m_szApp_size);
}

virtual ~CBaseErr(void)
{
    if (m_szApp)
        delete [] m_szApp;
    if (m_szLoc)
        delete [] m_szLoc;
};

virtual void Draw(HWND hwnd, LPCTSTR szStr = NULL)
{
    int      j = 0;
    char     szTmp[512];

    if (szStr)
        j = wsprintf(szTmp, "%s\n", szStr);
    if (ErrorNum() != INV_ERROR_CODE)
        j += wsprintf(szTmp+j, "Error = %d\n", ErrorNum());
    if (m_szLoc)
        j += wsprintf(szTmp+j, "Location = %s\n",
        GetLocation());

    j += wsprintf(szTmp+j, "%s\n", ErrorText());

    ::MessageBox(hwnd, szTmp, m_szApp, MB_OK);
}

char *GetApp(void) { return m_szApp; }
char *GetLocation(void) { return m_szLoc; }
virtual int ErrorNum() { return m_idMsg; }

virtual int ErrorType() = 0; // a value which distinguishes the kind of
error that occurred
virtual char *ErrorText() = 0; // a string (i.e., human readable)
representation of the error

protected:
char     *m_szApp;
char     *m_szLoc; // code location where the error occurred
int      m_idMsg;

//short   m_errType;
};

```

```

class CSocketErr : public CBaseErr
{
public:
    enum Action
    {
        eNone = 0,
        eSend,
        eSocket,
        eBind,
        eConnect,
        eListen,
        eHost,
        eRecv,
        eGetHostByName,
        eWSACreateEvent,
        eWSASend,
        eWSASendImage,
        eWSAGetOverlappedResult,
        eWSARecv,
        eWSARecvImage,
        eWSAWaitForMultipleEvents,
        eWSAStartup,
        eWSAResetEvent,
        eNonRetryable,
    };

    CSocketErr(Action eAction, LPCTSTR szLocation = NULL);

    ~CSocketErr()
    {
        if (m_szErrorText != NULL)
            delete [] m_szErrorText;
    };

    Action   m_eAction;
    char     *m_szErrorText;

    int ErrorType() { return ERR_TYPE_SOCKET;};
    char *ErrorText(void);
};

class CSystemErr : public CBaseErr
{
public:
    enum Action
    {
        eNone = 0,
        eTransactNamedPipe,
        eWaitNamedPipe,
        eSetNamedPipeHandleState,
        eCreateFile,
        eCreateProcess,
        eCallNamedPipe,
        eCreateEvent,
        eCreateThread,
        eVirtualAlloc,
        eReadFile = 10,
        eWriteFile,
        eMapViewOfFile,
        eCreateFileMapping,
        eInitializeSecurityDescriptor,
        eSetSecurityDescriptorDacl,
    };
};

```

```

        eCreateNamedPipe,
        eConnectNamedPipe,
        eWaitForSingleObject,
        eRegOpenKeyEx,
        eRegQueryValueEx = 20,
        eBeginThread,
        eRegEnumValue,
        eRegSetValueEx,
        eRegCreateKeyEx,
        eWaitForMultipleObjects,
        eRegisterClassEx,
        eCreateWindow,
        eCreateSemaphore,
        eFSeek,
        eFRead,
        eFWrite,
        eTmpFile,
        eSetFilePointer,
        eNew,
    };

    CSystemErr(Action eAction, LPCTSTR szLocation);
    CSystemErr(int iError, Action eAction, LPCTSTR
szLocation);
    int      ErrorType() { return ERR_TYPE_OS;};
    char      *ErrorText(void);
    void      Draw(HWND hwnd, LPCTSTR szStr = NULL);

    Action    m_eAction;

private:
    char m_szMsg[ERR_MSG_BUF_SIZE];
};

class CMemoryErr : public CBaseErr
{
public:
    CMemoryErr();

    int ErrorType() {return ERR_TYPE_MEMORY;}
    char *ErrorText() {return ERR_INS_MEMORY;}
};

class CBufferOverflowErr : public CBaseErr
{
public:
    CBufferOverflowErr(int, LPCTSTR);

    int ErrorType() {return ERR_BUF_OVERFLOW;}

    char *ErrorText() {return ERR_INS_BUF_OVERFLOW;}
};

```

install.c

```

/*      FILE:          INSTALL.C
 *
 *      Microsoft TPC-C Kit Ver. 4.20.000
 *      Copyright Microsoft, 1999
 *
 *      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
 */

#include <windows.h>
#include <direct.h>
#include <io.h>
#include <stdlib.h>
#include <stdio.h>
#include <commctrl.h>
#include "..\..\common\src\ReadRegistry.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 iThreadTimeout;
static int iListenBackLog;
static int iAcceptExOutstanding;

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);
static void ReadRegistrySettings(void);
static void WriteRegistrySettings(char *szDllPath);
static BOOL RegisterDLL(char *szFileName);
static int CopyFiles(HWND hDlg, char *szDllPath);
static BOOL GetInstallPath(char *szDllPath);
static void GetVersionInfo(char *szDLLPath, char *szExePath);
static void CheckWebService(void);
static void StartWebService(void);
static void StopWebService(void);
static void UpdateDialog(HWND hDlg);

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);
        }
    }
}

```

```

        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        szExePath[256];

    switch(uMsg)
    {
        case WM_INITDIALOG:
            GlobalMemoryStatus(&memoryStatus);
            iMaxPhysicalMemory = (memoryStatus.dwTotalPhys/
1048576);

            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();

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

    GetVersionInfo(szDllPath, szExePath);

    wsprintf(szTmp, "Version %d.%2.2d.%3.3d",
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
    VI.dwOSVersionInfoSize = sizeof(VI);

```

option

szDllPath);

```

GetVersionEx( &VI );
if (VI.dwMajorVersion < 5)
{
    HWND hDlg = GetDlgItem( hwnd, IDC_TM_MTS );
    EnableWindow( hDlg, 0 ); // disable COM

    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);
        return TRUE;
    }
    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,
                return TRUE;
            case IDCANCEL:
                EndDialog(hwnd, FALSE);
                return TRUE;
            default:
                return FALSE;
        }
    }
    break;
default:
    break;

```

```

    }
    return FALSE;
}

static void ProcessOK(HWND hwnd, char *szDllPath)
{
    int d;
    HWND hDlg;
    int rc;

    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);

    // write binaries to inetpub\wwwroot

```

```

rc = CopyFiles(hDlg, szDllPath);
if ( !rc )
{
    ShowWindow(hwnd, SW_SHOWNA);
    DestroyWindow(hDlg);
    strcpy( szErrTxt, "Error(s) occurred when creating " );
    strcat( szErrTxt, szLastFileName );
    MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
    EndDialog(hwnd, 0);
    return;
}

// 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 occurred 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 occurred 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,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters", 0, KEY_READ, &hKey) ==
ERROR_SUCCESS )
        {
            size = sizeof(iPoolThreadLimit);
            if ( RegQueryValueEx(hKey, "PoolThreadLimit", 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\\Parameters", 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);
        }
    }

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\\Parameters", 0, NULL,
REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) ==
ERROR_SUCCESS )
    {
        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\\Parameters", 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)
{
    BOOL          bSvcRunning;

    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);
    }

    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 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);

                    //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);
                        StartWebService();
                    }

                    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\PathWWWRoot
                // 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 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;
            free(ptr);
        }
    }

    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 CheckWWWebService(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);
    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);
    }
}

```

install.h

install.rc

```
//
//
// Dialog
//

IDD_DIALOG1 DIALOGEX 0, 0, 219, 351
STYLE DS_MODALFRAME | DS_CENTER | WS_MINIMIZEBOX | WS_POPUP | WS_CAPTION |
    WS_SYSMENU
CAPTION "TPC-C Web Client Installation Utility"
FONT 8, "MS Sans Serif"
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_RIGHT |
                    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, 39, 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_SETFOREGROUND | DS_3DLLOOK | 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 Successfull", 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 DISCARDABLE 0, 0, 91, 40
STYLE DS_SYSMODAL | DS_MODALFRAME | DS_3DLLOOK | DS_CENTER | WS_CAPTION
CAPTION "Installing TPC-C Web Client"
FONT 12, "Arial Black"
BEGIN
    CONTROL         "Progress1", IDC_PROGRESS1, "msctls_progress32", WS_BORDER,
7, 20, 77, 13
    CTEXT           "Static", IDC_STATUS, 7, 7, 77, 12, SS_SUNKEN
END

IDD_DIALOG4 DIALOG DISCARDABLE 0, 0, 291, 202
STYLE 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 DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 22
        RIGHTMARGIN, 209
        VERTGUIDE, 35
        VERTGUIDE, 198
        TOPMARGIN, 4
        BOTTOMMARGIN, 345
    END
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 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

////////////////////////////////////
//
// Icon
//

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

```



```

////////////////////////////////////
//
// TPCCDLL
//

IDR_TPCCDLL          TPCCDLL DISCARDABLE      "..\\..\\isapi_dll\\bin\\tpcc.dll"

#ifdef _MAC
////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,4,20,0
PRODUCTVERSION 0,4,20,0
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x1L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "Comments", "TPC-C Web Client Installer\0"
            VALUE "CompanyName", "Microsoft\0"
            VALUE "FileDescription", "install\0"
            VALUE "FileVersion", "0, 4, 20, 0\0"
            VALUE "InternalName", "install\0"
            VALUE "LegalCopyright", "Copyright © 1999\0"
            VALUE "OriginalFilename", "install.exe\0"
            VALUE "ProductName", "Microsoft install\0"
            VALUE "ProductVersion", "0, 4, 20, 0\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

#endif // !_MAC

////////////////////////////////////
//
// LICENSE
//

IDR_LICENSE1          LICENSE DISCARDABLE      "license.txt"

////////////////////////////////////
//
// DBLIB_DLL
//

```

```

IDR_DBLIB_DLL          DBLIB_DLL DISCARDABLE
"..\\..\\db_dblib_dll\\bin\\tpcc_dblib.dll"

////////////////////////////////////
//
// ODBC_DLL
//

IDR_ODBC_DLL          ODBC_DLL DISCARDABLE
"..\\..\\db_odbc_dll\\bin\\tpcc_odbc.dll"

////////////////////////////////////
//
// TUXEDO_APP
//

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

////////////////////////////////////
//
// TUXEDO_DLL
//

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

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

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

////////////////////////////////////
//
// COM_PS_DLL
//

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

////////////////////////////////////
//
// COM_ALL_DLL
//

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

////////////////////////////////////
//
// COM_TYPLIB
//

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

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

```

```

#ifndef APSTUDIO_INVOKED
//
// Generated from the TEXTINCLUDE 3 resource.
//

//
#endif // not APSTUDIO_INVOKED

```

install_com.cpp

```

/*
FILE:          INSTALL_COM.CPP
*
*              Microsoft TPC-C Kit Ver. 4.20.000
*              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 (wcscmp(vTmp.bstrVal, L"TPC-C"))
        {
            lCount--;
            continue;
        }
        else
        {
            hr = pCatalogCollectionApp->Remove(lCount - 1);
            if (!SUCCEEDED(hr)) goto Error;
            break;
        }
    }

    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";
    bstrTemp2 = bstrDllPath + "tpcc_com_all.dll"; // app name

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;

    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
        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;

>get_Count(&lCountMethod);
        hr = pCatalogCollectionMethod-
        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();
        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
        return FALSE;
}

```

isapi_dll.dsp

```

# Microsoft Developer Studio Project File - Name="isapi_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=isapi_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "isapi_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "isapi_dll.mak" CFG="isapi_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "isapi_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""

```

```

# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF "$(CFG)" == "isapi_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "NDEBUG" /D "WIN32" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\common\txnlog\lib\release\rtetime.lib
..\common\txnlog\lib\release\spinlock.lib ..\common\txnlog\lib\release\error.lib
..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
/nodefaultlib:"LIBCMT" /out:".bin\tpcc.dll"
# SUBTRACT LINK32 /nodefaultlib

!ELSEIF "$(CFG)" == "isapi_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /GX /ZI /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo

```

```

# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ..\common\txnlog\lib\debug\rtetime.lib
..\common\txnlog\lib\debug\spinlock.lib ..\common\txnlog\lib\debug\error.lib
..\common\txnlog\lib\debug\txnlog.lib wsock32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/nodefaultlib:"LIBCMTD" /out:".bin\tpcc.dll" /pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /nodefaultlib

!ELSEIF "$(CFG)" == "isapi_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "isapi_dll"
# PROP BASE Intermediate_Dir "isapi_dll"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /GX /Zi /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /GX /Zi /O2 /D "NDEBUG" /D "ICECAP" /D "WIN32" /D
"_WINDOWS" /FR /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc.dll"
/pdbtype:sept
# SUBTRACT BASE LINK32 /profile /pdb:none
# ADD LINK32 icap.lib ..\common\txnlog\lib\release\rtetime.lib
..\common\txnlog\lib\release\spinlock.lib ..\common\txnlog\lib\release\error.lib
..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug
/machine:I386 /out:".bin\tpcc.dll" /pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /map

!ENDIF

# Begin Target

# Name "isapi_dll - Win32 Release"
# Name "isapi_dll - Win32 Debug"
# Name "isapi_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp, *.def, *.rc"
# Begin Source File

```

```

SOURCE=.\src\tpcc.cpp
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.def
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.rc
# End Source File
# End Group
# Begin Group "Header Files"

# PROP Default_Filter "*.h, *.hpp"
# Begin Source File

SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=..\common\src\ReadRegistry.h
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.h
# End Source File
# Begin Source File

SOURCE=..\db_dblib_dll\src\tpcc_dblib.h
# End Source File
# Begin Source File

SOURCE=..\db_odbc_dll\src\tpcc_odbc.h
# End Source File
# Begin Source File

SOURCE=..\tm_tuxedo_dll\src\tpcc_tux.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project

```

rtetime.h

```

/* FILE: rtetime.h : header file
 * Copyright 1997 Microsoft Corp., All rights reserved.
 *
 * Source code licensed to Tandem Computers for Internal
 * use only. Redistribution of source or object files or
 * any derivative works is prohibited. By agreement, this
 * notice may not be removed.
 *
 * Authors: Charles Levine, Philip Durr

```

```

*
* Microsoft Corp.
*/

//FILE: RTETIME.H

#define MAX_JULIAN_TIME 0x7FFFFFFFFFFFFFFF
#define JULIAN_TIME __int64
#define TC_TIME DWORD
extern "C"
{
    BOOL InitJulianTime(LPSYSTEMTIME lpInitTime);
    JULIAN_TIME GetJulianTime(void);
    DWORD MyTickCount(void);
    void GetJulianAndTC(JULIAN_TIME *pJulian, DWORD *pTC);
    JULIAN_TIME ConvertTo64BitTime(int iYear, int iMonth, int iDay, int iHour,
    int iMinute, int iSecond);
    JULIAN_TIME Get64BitTime(LPSYSTEMTIME lpInitTime);
    int JulianDay( int yr, int mm, int dd );
    void JulianToTime(JULIAN_TIME julianTS, int* yr, int* mm, int* dd,
    int *hh, int *mi, int *ss );
    void JulianToCalendar( int day, int* yr, int* mm, int* dd );
}

```

spinlock.h

```

/* FILE: SPINLOCK.H
 *
 * Copyright 1997 Microsoft Corp., All rights reserved.
 *
 * Source code licensed to Tandem Computers for Internal
 * use only. Redistribution of source or object files or
 * any derivative works is prohibited. By agreement, this
 * notice may not be removed.
 *
 * Authors: Mike Parkes, Charles Levine, Philip Durr
 * Microsoft Corp.
 */

#ifdef _INC_Spinlock

    const LONG LockClosed = 1;
    const LONG LockOpen = 0;

    /*****
     *
     * Spinlock and Semaphore locking.
     *
     * This class provides a very conservative locking scheme.
     * The assumption behind the code is that locks will be
     * held for a very short time. When a lock is taken a memory
     * location is exchanged. All other threads that want this
     * lock wait by spinning and sometimes sleeping on a semaphore
     * until it becomes free again. The only other choice is not
     * to wait at all and move on to do something else. This
     * module should normally be used in conjunction with cache
     * aligned memory in minimize cache line misses.
     *****/

    class Spinlock
    {

```

```

// Private data.
HANDLE Semaphore;
volatile LONG m_Spinlock;
volatile LONG Waiting;

#ifdef _DEBUG
// Counters for debugging builds.
volatile LONG TotalLocks;
volatile LONG TotalSleeps;
volatile LONG TotalSpins;
volatile LONG TotalWaits;
#endif

public:
// Public functions.

Spinlock( void );

inline BOOL ClaimLock( BOOL Wait = TRUE );
inline void ReleaseLock( void );
~Spinlock( void );
// Disabled operations.
Spinlock( const Spinlock & Copy );
void operator=( const Spinlock & Copy );

private:
// Private functions.
inline BOOL ClaimSpinlock( volatile LONG *sl );
void WaitForLock( void );
void WakeAllSleepers( void );
};

/*****
*
* A guaranteed atomic exchange.
*
* An attempt is made to claim the Spinlock. This action is
* guaranteed to be atomic.
*****/

inline BOOL Spinlock::ClaimSpinlock( volatile LONG *Spinlock )
{
#ifdef _DEBUG
InterlockedIncrement( (LPLONG) & TotalLocks );
#endif
return ( (*Spinlock) == LockOpen ) && ( InterlockedExchange(
(LPLONG)Spinlock, LockClosed ) == LockOpen );
}

/*****
*
* Claim the Spinlock.
*
* Claim the lock if available else wait or exit.
*****/

inline BOOL Spinlock::ClaimLock( BOOL Wait )
{
if ( ! ClaimSpinlock( (volatile LONG*) & m_Spinlock ) )
{
if ( Wait )

```

```

WaitForLock();
return Wait;
}
return TRUE;
}

/*****
*
* Release the Spinlock.
*
* Release the lock and if needed wakeup any sleepers.
*
*****/

inline void Spinlock::ReleaseLock( void )
{
m_Spinlock = LockOpen;
if ( Waiting > 0 )
WakeAllSleepers();
}

#define _INC_Spinlock

#endif

```

tm_com_dll.dsp

```

# Microsoft Developer Studio Project File - Name="tm_com_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=tm_com_dll - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tm_com_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tm_com_dll.mak" CFG="tm_com_dll - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tm_com_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "tm_com_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF "$(CFG)" == "tm_com_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0

```

```

# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386 /out:".bin\tpcc_com.dll"

!ELSEIF "$(CFG)" == "tm_com_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_com.dll"
/pdbtype:sept

!ENDIF

# Begin Target

```

```

# Name "tm_com_dll - Win32 Release"
# Name "tm_com_dll - Win32 Debug"
# Begin Source File

```

```

SOURCE=. \src\tpcc_com.cpp
# End Source File
# Begin Source File

```

```

SOURCE=. \src\tpcc_com.h
# End Source File
# End Target
# End Project

```

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 "410"

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

#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
DWORD ul_reason_for_call reason for
call
* LPVOID lpReserved
* 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 CWEBCLNT_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");

            szDllName );

            hLibInstanceTm = LoadLibrary(

            if (hLibInstanceTm == NULL)
                throw new CWEBCLNT_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 CWEBCLNT_ERR(

ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
        }
        else if (Reg.eTxnMon == ENCINA)
        {
            strcpy( szDllName, Reg.szPath );
            strcat( szDllName,

"tpcc_encina.dll");

            szDllName );

            hLibInstanceTm = LoadLibrary(

            if (hLibInstanceTm == NULL)
                throw new CWEBCLNT_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 CWEBCLNT_ERR(

ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
        }
        else if (Reg.eTxnMon == COM)
        {
            strcpy( szDllName, Reg.szPath );
            strcat( szDllName,

"tpcc_com.dll");

            szDllName );

            hLibInstanceTm = LoadLibrary(

            if (hLibInstanceTm == NULL)

```

```

            throw new CWEBCLNT_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 CWEBCLNT_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");

                LoadLibrary( szDllName );

                hLibInstanceDb =

                if (hLibInstanceDb ==

NULL)
                    throw new

CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                // get function pointer
                for class constructor
                    pCTPCC_DBLIB_new =
                    (TYPE_CTPCC_DBLIB*) GetProcAddress(hLibInstanceDb, "CTPCC_DBLIB_new");
                if (pCTPCC_DBLIB_new ==

NULL)
                    throw new

CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
            }
            else if (Reg.eDB_Protocol == ODBC)
            {
                strcpy( szDllName,

Reg.szPath );

                strcat( szDllName,

"tpcc_odbc.dll");

                LoadLibrary( szDllName );

                hLibInstanceDb =

                if (hLibInstanceDb ==

NULL)
                    throw new

CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                // get function pointer
                for class constructor
                    pCTPCC_ODBC_new =
                    (TYPE_CTPCC_ODBC*) GetProcAddress(hLibInstanceDb, "CTPCC_ODBC_new");
                if (pCTPCC_ODBC_new ==

NULL)
                    throw new

CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
            }
        }

        if (dwNumDeliveryThreads)

```

```

{
    // for deferred delivery txns:
    hDoneEvent = CreateEvent( NULL,
TRUE /* manual reset */, FALSE /* initially not signalled */, NULL );

    InitializeCriticalSection(&DelBuffCriticalSection);
    hWorkerSemaphore =
CreateSemaphore( NULL, 0, dwDelBuffSize, NULL );
    dwDelBuffFreeCount =
dwDelBuffSize;

    InitJulianTime(NULL);

    // create unique log file name
    SYSTEMTIME Time;
    GetLocalTime( &Time );
    wsprintf( szLogFile, "%sdelivery-
%2.2d%2.2d%2.2d-%2.2d%2.2d.log",
    Time.wYear % 100, Time.wMonth, Time.wDay, Time.wHour, Time.wMinute );
    CTxnLog(szLogFile, TXN_LOG_WRITE);

    //write event into txn log for
START
    >WriteCtrlRecToLog(TXN_EVENT_START, szMyComputerName, sizeof(szMyComputerName));

    // allocate structures for
    pDeliHandles = new
    pDelBuff = new
    // launch DeliveryWorkerThread to
    perform actual delivery txns
    for(i=0; i<dwNumDeliveryThreads;
    i++)
    {
        pDeliHandles[i] =
        if (pDeliHandles[i] ==
INVALID_HANDLE_VALUE)
            throw new
CWEBCINT_ERR( ERR_DELIVERY_THREAD_FAILED );
    }
    break;

case DLL_PROCESS_DETACH:
    if (dwNumDeliveryThreads)
    {
        if (txnDelilog != NULL)
        {
            //write event into txn
log for STOP
            >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(&DelBuffCriticalSection);
}

DeleteCriticalSection(&TermCriticalSection);

if (hLibInstanceTm != NULL)
    FreeLibrary( hLibInstanceTm );
hLibInstanceTm = NULL;

if (hLibInstanceDb != NULL)
    FreeLibrary( hLibInstanceDb );
hLibInstanceDb = NULL;

Sleep(500);
break;

default:
    /* nothing */;
}
}
catch (CBaseErr *e)
{
    WriteMessageToEventLog( e->ErrorText() );
    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);
    lstrcpy(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++)
            WaitForSingleObject( 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 CWEBCLNT_ERR( ERR_INVALID_TERMID
);
            }

            //must have a valid syncid here since termid is valid
            if (iSyncId != Term.pClientData[TermId].iSyncId)
                throw new CWEBCLNT_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)
            pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
        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. "
Database="%s",
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:
            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, "Error in Delivery Txn thread. %s",
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:
    delete pTxn;
    _endthread();
}

/* FUNCTION: PostDeliveryInfo
 * PURPOSE:      This function enters the delivery txn into the deferred delivery
 *               buffer.
 * RETURNS:      BOOL      FALSE      delivery information posted
 *               TRUE       error cannot
 *               successfully
 * post delivery info
 */

BOOL PostDeliveryInfo(short w_id, short o_carrier_id)
{
    BOOL bError;

    EnterCriticalSection(&DelBuffCriticalSection);
    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(&DelBuffCriticalSection);

        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 CWEBCLNT_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>"
Client</TITLE></HEAD><BODY>"

Web Client (ver 4.20)</BIG></B> <BR> <BR>"

New"><PRE>"

"__TIME__" <BR>"

("__TIMESTAMP__") <BR>"

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

NAME="\"STATUSID\" VALUE="\"0\">"

NAME="\"ERROR\" VALUE="\"0\">"

NAME="\"FORMID\" VALUE="\"1\">"

NAME="\"TERMID\" VALUE="\"0\">"

NAME="\"SYNCID\" VALUE="\"0\">"

NAME="\"VERSION\" VALUE="\"\" WEBCLIENT_VERSION \"\">"

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

        "Txn Monitor          =

        "Database protocol    =

        "Max Connections      =

        "# of Delivery Threads =

        "Max Pending Deliveries =

        , 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>"

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>"

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=4><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 CWEBCLNT_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 );
        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 iTot;

    EnterCriticalSection(&TermCriticalSection);

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

    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>"
, iTot );
}

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,
        "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,
        "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,
        "Order Item_Id field entered without a corresponding Supp_W."
    },
    {
        ERR_NEWORDER_MISSING_IID_KEY,
        "Order missing Item Id key \"IID*\"."
    },
    {
        ERR_NEWORDER_MISSING_QTY_KEY,
        "Order Missing Qty key \"Qty##*\"."
    },
    {
        ERR_NEWORDER_MISSING_SUPPW_KEY,
        "New Order missing Supp_W key \"SP##*\"."
    },
    {
        ERR_NEWORDER_NOITEMS_ENTERED,
        "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."
    }
},

```



```

        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 *pKey
 *               key value to look for
 *               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=value\0. 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 CWEBCLNT_ERR( NoKeyErr );
        return 0;
    }
}

```

```

    *pQueryString = ptr;
    return atoi(ptr0);

ErrorNoKey:
    if (NoKeyErr != NO_ERR)
        throw new CWEBCLNT_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 CWEBCLNT_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 CWEBCLNT_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=\"TERMINID\" 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=\"TERMINID\" 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=\"TERMINID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">"
        "Stock-Level<BR>"
        "Warehouse: %4.4d 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> <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></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 = wsprintf(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=\"SYNCDID\" 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: %4.4d  ",
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:
</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: %4.4d  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 += sprintf(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,
                100.0 * pNewOrderData->w_tax,
                100.0 * pNewOrderData->d_tax);

        for(i=0; i<pNewOrderData->o_ol_cnt; i++)
        {
            c += sprintf(szForm+c, " %4.4d %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_i_price,
        pNewOrderData->OL[i].ol_amount );
    }
    else
    {
        c += sprintf(szForm+c,
            "%Disc:<BR>"
            "Order Number: %8.8d  Number of Lines:
            " Supp_W Item_Id Item Name
            , 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 += sprintf(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..\">"
            "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 = sprintf(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=\"\"%d\">"
            "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"\"%d\">"
            "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"\"%d\">"
            "<PRE><font face=\"Courier\">"

Payment<BR>"

            "Date: "
            , PAYMENT_FORM, iTermId, Term.pClientData[iTermId].iSyncId);

        if ( !bInput )
        {
            c += sprintf(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 += sprintf(szForm+c,
                "<BR> <BR>Warehouse: %4.4d"
                "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>"
                "New Cust-Balance:<BR>"
                "<BR></font></PRE><HR>"
                "Amount Paid: $<INPUT NAME=\"HAM*\" SIZE=7>"
                "Credit Limit:<BR> <BR>Cust-Data: <BR> <BR> <BR> <BR>"
                "<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: %4.4d
                District: %2.2d<BR>"
                "%-20s %-20s"
                "%-20s %-20s"
                "%-20s %-2s %5.5s-%4.4s %-20s %-2s %5.5s-"
                "%4.4s<BR> <BR>"
                "Customer: %4.4d Cust-Warehouse: %4.4d 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_city, pPaymentData->w_state,
pPaymentData->w_zip, pPaymentData->w_zip+5
        , pPaymentData->d_city, pPaymentData->d_state,
pPaymentData->d_zip, pPaymentData->d_zip+5
        , pPaymentData->c_id, pPaymentData->c_w_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 += wsprintf(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 += wsprintf(szForm+c,
            "Cust-Data: %-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, "          <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>");
    }

}

/* 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
    static char szBR[] = " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR>";

    c = wsprintf(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=\"0\">"
    "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"0\">"
    "<INPUT TYPE=\"hidden\" NAME=\"SYCID\" VALUE=\"0\">"
    "<PRE><font face=\"Courier\">"
    "Order-Status<BR>"
    "Warehouse: %4.4d          ",
    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 += wsprintf(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\n",
            %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, "    %4.4d    %6.6d    %2.2d\n",
                %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=\"TERMID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">"

Delivery<BR>"
        "Warehouse: %4.4d<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>
</font></PRE><HR>"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                "</BODY></FORM></HTML>" );
        }
        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 lpszQueryString
 *               client 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] =
    { "SP00", "SP01", "SP02", "SP03", "SP04",
      "SP05", "SP06", "SP07", "SP08", "SP09",
      "SP10", "SP11", "SP12", "SP13", "SP14" };
    static char szIID[MAX_OL_NEW_ORDER_ITEMS][7] =
    { "IID00", "IID01", "IID02", "IID03", "IID04",
      "IID05", "IID06", "IID07", "IID08", "IID09",
      "IID10", "IID11", "IID12", "IID13", "IID14" };
    static char szQty[MAX_OL_NEW_ORDER_ITEMS][7] =
    { "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] )
        {
            if ( !IsNumeric(szTmp) )
                throw new CWBCLNT_ERR(
ERR_NEWORDER_SUPPW_INVALID );
            pNewOrderData->OL[items].ol_supply_w_id =
(short)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 CWEBCLNT_ERR(
ERR_NEWORDER_QTY_WITHOUT_SUPPW );
    }
    if ( items == 0 )
        throw new CWEBCLNT_ERR( ERR_NEWORDER_NOITEMS_ENTERED );
    pNewOrderData->o_o1_cnt = items;
}

/* FUNCTION: GetPaymentData
 * PURPOSE:      This function extracts and validates the payment form data from
an http command string.
 * ARGUMENTS:    LPSTR          lpszQueryString
                  client browser http command string
                  PAYMENT_DATA  *pPaymentData
                  pointer to payment data structure
 */
void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData)
{
    char        szTmp[26];
    char        *ptr = lpszQueryString;
    BOOL        bCustIdBlank;

    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 CWEBCLNT_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 CWEBCLNT_ERR( ERR_PAYMENT_MISSING_CID_CLT );

        _strupr( szTmp );
        if ( strlen(pPaymentData->c_last) > LAST_NAME_LEN )
            throw new CWEBCLNT_ERR( ERR_PAYMENT_LAST_NAME_TO_LONG
);
    }
}

```

```

        strcpy(pPaymentData->c_last, szTmp);
    }
    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 CWEBCLNT_ERR( ERR_PAYMENT_CID_AND_CLT );
    }

    GetKeyValue(&ptr, "HAM*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_HAM_KEY);
    if ( !IsDecimal(szTmp) )
        throw new CWEBCLNT_ERR( ERR_PAYMENT_HAM_INVALID );
    pPaymentData->h_amount = atof(szTmp);
    if ( pPaymentData->h_amount >= 10000.00 || pPaymentData->h_amount < 0 )
        throw new CWEBCLNT_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;

    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 CWEBCLNT_ERR(
ERR_ORDERSTATUS_MISSING_CID_CLT );

        _strupr( szTmp );
        if ( strlen(pOrderStatusData->c_last) > LAST_NAME_LEN )
            throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_CLT_RANGE );
        strcpy(pOrderStatusData->c_last, szTmp);
    }
    else
    {
        // parse customer id and verify that last name was NOT entered
        if ( !IsNumeric(szTmp) )
            throw new CWEBCLNT_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 CWEBCLNT_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.def

LIBRARY TPCC.DLL

EXPORTS

```

GetExtensionVersion @1
HttpExtensionProc @2
TerminateExtension @3

```

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 //beginning form no term id assigned, form id 1
#define MAIN_MENU_FORM //term id assigned main menu form id 2
#define NEW_ORDER_FORM //new order form id 3
#define PAYMENT_FORM //payment form id 4
#define DELIVERY_FORM //delivery form id 5
#define ORDER_STATUS_FORM //order status id 6

```

```

#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 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_TO_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;};
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(short 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"

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

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

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifndef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

#ifdef _MAC
////////////////////////////////////
//
// 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

#endif // !_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

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

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//

////////////////////////////////////
#endif // 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 = 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;
    }

    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:
    VARIANT m_vTxn;
    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();

} // 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;

////////////////////////////////////
// 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 );
        }
        else if (dwReason == DLL_PROCESS_DETACH)
            _Module.Term();

    }
    catch (CBaseErr *e)
    {
        WriteMessageToEventLog(e->ErrorText());
        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)
        wprintf( 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()
{
    if (m_pTxn)
        delete m_pTxn;
}

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
        {
            if (Reg.eDB_Protocol == ODBC)
                m_pTxn = pCTPCC_ODBC_new( Reg.szDbServer,
                Reg.szDbUser, Reg.szDbPassword, szMyComputerName, Reg.szDbName );
            else if (Reg.eDB_Protocol == DBLIB)
                m_pTxn = pCTPCC_DBLIB_new( Reg.szDbServer,
                Reg.szDbUser, Reg.szDbPassword, szMyComputerName, Reg.szDbName );
        }
        catch (CBaseErr *e)
        {
            WriteMessageToEventLog(e->ErrorText());
            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,

            >rgsabound->cElements, txn_in.parray-
            >rgsabound->cElements);
            pData = (COM_DATA*) txn_out->parray->pvData;
            memcpy(pData, &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,

            >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)) ||
        == 10054) )
            ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum()
            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)) ||
        == 10054) )
            ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum()
            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,
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)) ||
        == 10054) )
            ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum()
            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.def

```
; tpcc_com_all.def : Declares the module parameters.
```

```

LIBRARY      "tpcc_com_all.dll"

EXPORTS
   DllCanUnloadNow      @1 PRIVATE
   DllGetClassObject    @2 PRIVATE
   DllRegisterServer    @3 PRIVATE
   DllUnregisterServer  @4 PRIVATE

```

tpcc_com_all.dsp

```

# Microsoft Developer Studio Project File - Name="tpcc_com_all" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

```

```
# TARGETTYPE "Win32 (x86) Dynamic-Link Library" 0x0102
```

```

CFG=tpcc_com_all - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_all.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_all.mak" CFG="tpcc_com_all - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_all - Win32 Release" (based on "Win32 (x86) Dynamic-Link
Library")
!MESSAGE "tpcc_com_all - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

```

```

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

```

```
!IF "$(CFG)" == "tpcc_com_all - Win32 Release"
```

```

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0

```

```

# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\db_dblib_dll\bin\tpcc_dblib.lib ..\db_odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
/dll /machine:I386

```

```
!ELSEIF "$(CFG)" == "tpcc_com_all - Win32 Debug"
```

```

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ..\db_dblib_dll\bin\tpcc_dblib.lib ..\db_odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
/dll /debug /machine:I386 /pdbtype:sept

```

```
!ENDIF
```

```
# Begin Target
```

```

# Name "tpcc_com_all - Win32 Release"
# Name "tpcc_com_all - Win32 Debug"
# Begin Group "Source"

```

```

# PROP Default_Filter "*.cpp, *.c"
# Begin Source File

SOURCE=.\\src\\tpcc_com_all.cpp
# SUBTRACT CPP /YX
# End Source File
# Begin Source File

SOURCE=.\\src\\tpcc_com_all.def
# End Source File
# Begin Source File

SOURCE=.\\src\\tpcc_com_all.idl

!IF "$(CFG)" == "tpcc_com_all - Win32 Release"

# PROP Ignore_Default_Tool 1
# Begin Custom Build - Performing MIDL step
InputPath=.\\src\\tpcc_com_all.idl

BuildCmds= \
    midl /Oicf /h "tpcc_com_all.h" /iid "tpcc_com_all_i.c"
".\\src\\tpcc_com_all.idl" /out ".\\src"

".\\src\\tpcc_com_all.tlb" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)

".\\src\\tpcc_com_all.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)

".\\src\\tpcc_com_all_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)
# End Custom Build

!ELSEIF "$(CFG)" == "tpcc_com_all - Win32 Debug"

# PROP Ignore_Default_Tool 1
# Begin Custom Build - Performing MIDL step
InputPath=.\\src\\tpcc_com_all.idl

BuildCmds= \
    midl /Oicf /h "tpcc_com_all.h" /iid "tpcc_com_all_i.c"
".\\src\\tpcc_com_all.idl" /out ".\\src"

".\\src\\tpcc_com_all.tlb" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)

".\\src\\tpcc_com_all.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)

".\\src\\tpcc_com_all_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)
# End Custom Build

!ENDIF

# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

```

```

SOURCE=.\\src\\Methods.h
# End Source File
# Begin Source File

SOURCE=.\\src\\resource.h
# End Source File
# End Group
# Begin Source File

SOURCE=.\\src\\tpcc_com_all.rc
# End Source File
# End Target
# End Project

```

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 5.03.0280 */
/* at Thu Dec 13 23:13:14 2001
*/
/* Compiler settings for .\\src\\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=win32 (32b run), 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*/
#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__

/* 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__ */

#ifdef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

#ifdef __cplusplus
typedef class OrderStatus OrderStatus;
#else
typedef struct OrderStatus OrderStatus;
#endif /* __cplusplus */

#endif /* __OrderStatus_FWD_DEFINED__ */

#ifdef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

#ifdef __cplusplus
typedef class Payment Payment;
#else
typedef struct Payment Payment;
#endif /* __cplusplus */

#endif /* __Payment_FWD_DEFINED__ */

#ifdef __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_FAR * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void __RPC_FAR * );

/* 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 */

```

```

#ifndef __cplusplus
}
#endif

#endif

```

tpcc_com_all.idl

```

/*      FILE:          TPCC.IDL
 *
 *      Microsoft TPC-C Kit Ver. 4.20.000
 *      Copyright Microsoft, 1999
 *
 *      All Rights Reserved
 *
 *      not yet audited
 *
 *      PURPOSE:  IDL source for TPCC.dll.  This file is processed by the MIDL
 *      tool to      produce the type library (TPCC.tlb) and
 *      marshalling code.
 *
 *      Change history:
 *      4.20.000 - first version
 */

interface TPCC;
interface NewOrder;
interface OrderStatus;
interface Payment;
interface StockLevel;

import "oidl.idl";
import "ocidl.idl";
import "..\tpcc_com_ps\src\tpcc_com_ps.idl";

[
    uuid(122A3117-2520-11D3-BA71-00C04FBFE08B),
    version(1.0),
    helpstring("TPC-C 1.0 Type Library")
]
library TPCCLib
{
    importlib("stdole32.tlb");
    importlib("stdole2.tlb");

    [
        uuid(122A3128-2520-11D3-BA71-00C04FBFE08B),
        helpstring("All Txns Class")
    ]
    coclass TPCC
    {
        [default] interface ITPCC;
    };

    [
        uuid(975BAABF-84A7-11D2-BA47-00C04FBFE08B),
        helpstring("NewOrder Class")
    ]

```

```

coclass NewOrder
{
    [default] interface ITPCC;
};

[
    uuid(266836AD-A50D-11D2-BA4E-00C04FBFE08B),
    helpstring("OrderStatus Class")
]
coclass OrderStatus
{
    [default] interface ITPCC;
};

[
    uuid(CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B),
    helpstring("Payment Class")
]
coclass Payment
{
    [default] interface ITPCC;
};

[
    uuid(2668369E-A50D-11D2-BA4E-00C04FBFE08B),
    helpstring("StockLevel Class")
]
coclass StockLevel
{
    [default] interface ITPCC;
};

};

```

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 _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif

```

```

#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

```

```

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

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

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
1 TYPELIB "tpcc_com_all.tlb"

////////////////////////////////////
#endif // 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 5.03.0280 */
/* at Thu Dec 13 23:13:14 2001 */
/*
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), 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_AXP64)

#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 = {l,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,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x00,0xC0,0x4F,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,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* !defined(_M_IA64) && !defined(_M_AXP64) */

#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 5.03.0280 */
/* at Thu Dec 13 23:13:14 2001 */
/*
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), 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_HEADERING( )

#if defined(_M_IA64) || defined(_M_AXP64)

#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 = {l,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,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

```

```

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x00,0xC0,0x4F,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,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* defined(_M_IA64) || defined(_M_AXP64) */

```

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.def

```

LIBRARY      "tpcc_com_ps"

DESCRIPTION  'Proxy/Stub DLL'

EXPORTS
    DllGetClassObject      @1    PRIVATE
    DllCanUnloadNow        @2    PRIVATE
    GetProxyDllInfo        @3    PRIVATE
    DllRegisterServer       @4    PRIVATE
    DllUnregisterServer     @5    PRIVATE

```

tpcc_com_ps.dsp

```

# Microsoft Developer Studio Project File - Name="tpcc_com_ps" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

```

```

# TARGETTYPE "Win32 (x86) Application" 0x0101

```

```

CFG=tpcc_com_ps - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak" CFG="tpcc_com_ps - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_ps - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "tpcc_com_ps - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

```

```

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

```

```

!IF "$(CFG)" == "tpcc_com_ps - Win32 Release"

```

```

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WIN32_WINNT=0x0400" /D
"REGISTER_PROXY_DLL" /FD /c
# SUBTRACT CPP /YX
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386

```



```

# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /subsystem:windows /dll /pdb:none /machine:I386
/def:".src\tpcc_com_ps.def"
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=. \bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

".. \tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    copy . \src\tpcc_com_ps.h .. \tpcc_com_all\src\

# End Custom Build

!ELSEIF "$(CFG)" == "tpcc_com_ps - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ". \bin"
# PROP Intermediate_Dir ". \obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/ FD /c
# ADD CPP /nologo /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WIN32_WINNT=0x0400" /D
"REGISTER_PROXY_DLL" /FD /c
# ADD BASE MTL /nologo /D " _DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D " _DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d " _DEBUG"
# ADD RSC /l 0x409 /d " _DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /dll /debug /machine:IX86 /def:".src\tpcc_com_ps.def"
/pdbtype:sept
# SUBTRACT LINK32 /pdb:none
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=. \bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

".. \tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    copy . \src\tpcc_com_ps.h .. \tpcc_com_all\src\

# End Custom Build

!ENDIF

# Begin Target

# Name "tpcc_com_ps - Win32 Release"
# Name "tpcc_com_ps - Win32 Debug"
# Begin Group "Source"

# PROP Default_Filter ""
# Begin Source File

```

```

SOURCE=. \src\dlldata.c
# End Source File
# Begin Source File

SOURCE=. \src\tpcc_com_ps.def
# PROP Exclude_From_Build 1
# End Source File
# Begin Source File

SOURCE=. \src\tpcc_com_ps.idl

!IF "$(CFG)" == "tpcc_com_ps - Win32 Release"

# PROP Ignore_Default_Tool 1
# Begin Custom Build
InputPath=. \src\tpcc_com_ps.idl

BuildCmds= \
    midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c"
".. \src\tpcc_com_ps.idl" /out ". \src"

".. \src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)

".. \src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)

".. \src\dlldata.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)

".. \src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)
# End Custom Build

!ELSEIF "$(CFG)" == "tpcc_com_ps - Win32 Debug"

# PROP Ignore_Default_Tool 1
# Begin Custom Build
InputPath=. \src\tpcc_com_ps.idl

BuildCmds= \
    midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c"
".. \src\tpcc_com_ps.idl" /out ". \src"

".. \src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)

".. \src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)

".. \src\dlldata.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)

".. \src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    $(BuildCmds)
# End Custom Build

!ENDIF

# End Source File
# Begin Source File

```

```

SOURCE=.\src\tpcc_com_ps_i.c
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps_p.c
# End Source File
# End Group
# End Target
# End Project

```

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 5.03.0280 */
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), 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*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

#include "rpc.h"
#include "rpcndr.h"

#ifndef __RPCNDR_H_VERSION__
#error this stub requires an updated version of <rpcndr.h>
#endif // __RPCNDR_H_VERSION__

#ifndef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

#ifndef __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"

```

```

#ifndef __cplusplus
extern "C"{
#endif

void __RPC_FAR * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void __RPC_FAR * );

/* 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;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

/* interface ITPCC */
/* [unique][helpstring][uuid][oleautomation][object] */

EXTERN_C const IID IID_ITPCC;

#if defined(__cplusplus) && !defined(CINTERFACE)

    MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT __stdcall NewOrder(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall Payment(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall Delivery(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall StockLevel(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall OrderStatus(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall CallSetComplete( void) = 0;

    };
#else /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

        HRESULT ( STDMETHODCALLTYPE __RPC_FAR *QueryInterface )(

```

```

        ITPCC __RPC_FAR * This,
        /* [in] */ REFIID riid,
        /* [iid_is][out] */ void __RPC_FAR *__RPC_FAR *ppvObject);

ULONG ( STDMETHODCALLTYPE __RPC_FAR *AddRef )(
    ITPCC __RPC_FAR * This);

ULONG ( STDMETHODCALLTYPE __RPC_FAR *Release )(
    ITPCC __RPC_FAR * This);

HRESULT ( STDMETHODCALLTYPE __RPC_FAR *NewOrder )(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);

HRESULT ( STDMETHODCALLTYPE __RPC_FAR *Payment )(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);

HRESULT ( STDMETHODCALLTYPE __RPC_FAR *Delivery )(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);

HRESULT ( STDMETHODCALLTYPE __RPC_FAR *StockLevel )(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);

HRESULT ( STDMETHODCALLTYPE __RPC_FAR *OrderStatus )(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);

HRESULT ( STDMETHODCALLTYPE __RPC_FAR *CallSetComplete )(
    ITPCC __RPC_FAR * This);

END_INTERFACE
} ITPCCVtbl;

interface ITPCC
{
    CONST_VTBL struct ITPCCVtbl __RPC_FAR *lpVtbl;
};

#ifdef 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 __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);

void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_Payment_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);

void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_Delivery_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);

void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_StockLevel_Proxy(

```

```

ITPCC __RPC_FAR * This,
/* [in] */ VARIANT txn_in,
/* [out] */ VARIANT __RPC_FAR *txn_out);

void __RPC_STUB ITPCC_StockLevel_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *_pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_OrderStatus_Proxy(
ITPCC __RPC_FAR * This,
/* [in] */ VARIANT txn_in,
/* [out] */ VARIANT __RPC_FAR *txn_out);

void __RPC_STUB ITPCC_OrderStatus_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *_pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
ITPCC __RPC_FAR * 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 __RPC_FAR
*, unsigned long __RPC_FAR *, VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER VARIANT_UserMarshal( unsigned long __RPC_FAR
*, unsigned char __RPC_FAR *, VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER VARIANT_UserUnmarshal(unsigned long __RPC_FAR
*, unsigned char __RPC_FAR *, VARIANT __RPC_FAR * );
void __RPC_USER VARIANT_UserFree( unsigned long __RPC_FAR
*, VARIANT __RPC_FAR * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

tpcc_com_ps.idl

```

/* FILE: ITPCC.IDL
 *
 * Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 *
 * All Rights Reserved
 *
 * not yet audited
 *
 * PURPOSE: Defines the interface used by TPCC. This interface can be
 * implemented by C++ components.
 *
 * Change history:
 * 4.20.000 - first version
 */

// Forward declare all types defined
interface ITPCC;
import "oidl.idl";
import "ocidl.idl";

[
    object,
    oleautomation,
    uuid(FEEEE6AA2-84B1-11d2-BA47-00C04FBFE08B),
    helpstring("ITPCC Interface"),
    pointer_default(unique)
]
interface ITPCC : IUnknown
{
    HRESULT __stdcall NewOrder(
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );

    HRESULT __stdcall Payment(
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );

    HRESULT __stdcall Delivery(
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );

    HRESULT __stdcall StockLevel(
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );

    HRESULT __stdcall OrderStatus(
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );

    HRESULT __stdcall CallSetComplete

```

```

(
);

}; // interface ITPCC


```

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 5.03.0280 */
/* at Thu Dec 13 23:13:08 2001 */
/*
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), 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_AXP64)

#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 = {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,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* !defined(_M_IA64) && !defined(_M_AXP64) */

#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 5.03.0280 */
/* at Thu Dec 13 23:13:08 2001 */
/*
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), 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_AXP64)

#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 = {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,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* defined(_M_IA64) || defined(_M_AXP64) */

```

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 5.03.0280 */
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com.ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), 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(  )

#ifdef __cplusplus
extern "C" {
#endif

/* 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 997
#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;

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")

```

```

static const unsigned short ITPCC_FormatStringOffsetTable[] =
{
    0,
    34,
    68,
    102,
    136,
    170
};

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
{
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0
};

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
{
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
};

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy,
    (void *)-1 /* ITPCC::NewOrder */ ,
    (void *)-1 /* ITPCC::Payment */ ,
    (void *)-1 /* ITPCC::Delivery */ ,
    (void *)-1 /* ITPCC::StockLevel */ ,
    (void *)-1 /* ITPCC::OrderStatus */ ,
    (void *)-1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,

```

```

    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* Reserved3 */
    0, /* Reserved4 */
    0 /* Reserved5 */
};

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        VARIANT_UserSize,
        VARIANT_UserMarshal,
        VARIANT_UserUnmarshal,
        VARIANT_UserFree
    }
};

#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_marshal] or [user_marshal] 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 */

        0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object, Oi2 */

        /* 2 */ NdrFcLong( 0x0 ), /* 0 */
        /* 6 */ NdrFcShort( 0x3 ), /* 3 */
#ifdef _ALPHA_
#ifdef _PPC_

```

```

#if !defined(_MIPS_)
/* 8 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x7, /* 0i2 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, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 18 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 20 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 22 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 24 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 26 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

/* 28 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 30 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */

```

```

#endif
#else
NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 32 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure Payment */

/* 34 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, 0i2 */
/* 36 */ NdrFcLong( 0x0 ), /* 0 */
/* 40 */ NdrFcShort( 0x4 ), /* 4 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 42 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 44 */ NdrFcShort( 0x0 ), /* 0 */
/* 46 */ NdrFcShort( 0x8 ), /* 8 */
/* 48 */ 0x7, /* 0i2 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, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 52 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 54 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 56 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 58 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */

```



```

#endif
#else
                                NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */

#endif
#else
                                NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */

#endif
/* 60 */ NdrFcShort( 0x3da ),          /* Type Offset=986 */

/* Return value */

/* 62 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 64 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                                NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */

#endif
#else
                                NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */

#endif
/* 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 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 76 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                                NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */

#endif
#else
                                NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */

#endif
#else
                                NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */

#endif
/* 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, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 86 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
                                NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */

#endif

```

```

#else
                                NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */

#endif
#else
                                NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */

#endif
/* 88 */ NdrFcShort( 0x3c8 ),          /* Type Offset=968 */

/* Parameter txn_out */

/* 90 */ NdrFcShort( 0x4113 ),          /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 92 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                                NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */

#endif
#else
                                NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */

#endif
#else
                                NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */

#endif
/* 94 */ NdrFcShort( 0x3da ),          /* Type Offset=986 */

/* Return value */

/* 96 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 98 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                                NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */

#endif
#else
                                NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */

#endif
#else
                                NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */

#endif
/* 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 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                                NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */

#endif
#else
                                NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */

#endif

```

```

NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* 114 */ NdrFcShort( 0x8 ), /* 8 */
/* 116 */ 0x7, /* 0i2 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, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 122 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 128 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

/* 130 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 132 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 134 */ 0x8, /* FC_LONG */
0x0, /* 0 */

```

```

/* Procedure OrderStatus */

/* 136 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, 0i2 */
/* 138 */ NdrFcLong( 0x0 ), /* 0 */
/* 142 */ NdrFcShort( 0x7 ), /* 7 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 144 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 146 */ NdrFcShort( 0x0 ), /* 0 */
/* 148 */ NdrFcShort( 0x8 ), /* 8 */
/* 150 */ 0x7, /* 0i2 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, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 154 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 156 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 162 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

```

```

/* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
#ifdef _PPC_
#ifdef _MIPS_
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 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 */
#ifdef _ALPHA_
/* 178 */ NdrFcShort( 0x8 ), /* x86, MIPS, PPC Stack size/offset = 8 */
#else
NdrFcShort( 0x10 ), /* Alpha Stack size/offset = 16 */
#endif
/* 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, */
#ifdef _ALPHA_
/* 188 */ NdrFcShort( 0x4 ), /* x86, MIPS, PPC Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 190 */ 0x8, /* FC_LONG */
0x0, /* 0 */

0x0

};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
0,
{
/* 2 */ NdrFcShort( 0x0 ), /* 0 */
0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x3b0 ), /* Offset= 944 (948) */
/* 6 */
0x2b, /* FC_NON_ENCAPSULATED_UNION */
0x9, /* FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /* */
}
}

```

```

/* 10 */ NdrFcShort( 0xffff ), /* -8 */
/* 12 */ NdrFcShort( 0x2 ), /* Offset= 2 (14) */
/* 14 */ NdrFcShort( 0x10 ), /* 16 */
/* 16 */ NdrFcShort( 0x2b ), /* 43 */
/* 18 */ NdrFcLong( 0x3 ), /* 3 */
/* 22 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 24 */ NdrFcLong( 0x11 ), /* 17 */
/* 28 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 30 */ NdrFcLong( 0x2 ), /* 2 */
/* 34 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 36 */ NdrFcLong( 0x4 ), /* 4 */
/* 40 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 42 */ NdrFcLong( 0x5 ), /* 5 */
/* 46 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 48 */ NdrFcLong( 0xb ), /* 11 */
/* 52 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 54 */ NdrFcLong( 0xa ), /* 10 */
/* 58 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 60 */ NdrFcLong( 0x6 ), /* 6 */
/* 64 */ NdrFcShort( 0xd6 ), /* Offset= 214 (278) */
/* 66 */ NdrFcLong( 0x7 ), /* 7 */
/* 70 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 72 */ NdrFcLong( 0x8 ), /* 8 */
/* 76 */ NdrFcShort( 0xd0 ), /* Offset= 208 (284) */
/* 78 */ NdrFcLong( 0xd ), /* 13 */
/* 82 */ NdrFcShort( 0xe2 ), /* Offset= 226 (308) */
/* 84 */ NdrFcLong( 0x9 ), /* 9 */
/* 88 */ NdrFcShort( 0xee ), /* Offset= 238 (326) */
/* 90 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 94 */ NdrFcShort( 0xfa ), /* Offset= 250 (344) */
/* 96 */ NdrFcLong( 0x24 ), /* 36 */
/* 100 */ NdrFcShort( 0x308 ), /* Offset= 776 (876) */
/* 102 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 106 */ NdrFcShort( 0x302 ), /* Offset= 770 (876) */
/* 108 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 112 */ NdrFcShort( 0x300 ), /* Offset= 768 (880) */
/* 114 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 118 */ NdrFcShort( 0x2fe ), /* Offset= 766 (884) */
/* 120 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 124 */ NdrFcShort( 0x2fc ), /* Offset= 764 (888) */
/* 126 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 130 */ NdrFcShort( 0x2fa ), /* Offset= 762 (892) */
/* 132 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 136 */ NdrFcShort( 0x2f8 ), /* Offset= 760 (896) */
/* 138 */ NdrFcLong( 0x400b ), /* 16395 */
/* 142 */ NdrFcShort( 0x2e6 ), /* Offset= 742 (884) */
/* 144 */ NdrFcLong( 0x400a ), /* 16394 */
/* 148 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (888) */
/* 150 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 154 */ NdrFcShort( 0x2ea ), /* Offset= 746 (900) */
/* 156 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 160 */ NdrFcShort( 0x2e0 ), /* Offset= 736 (896) */
/* 162 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 166 */ NdrFcShort( 0x2e2 ), /* Offset= 738 (904) */
/* 168 */ NdrFcLong( 0x400d ), /* 16397 */
/* 172 */ NdrFcShort( 0x2e0 ), /* Offset= 736 (908) */
/* 174 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 178 */ NdrFcShort( 0x2de ), /* Offset= 734 (912) */
/* 180 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 184 */ NdrFcShort( 0x2dc ), /* Offset= 732 (916) */
/* 186 */ NdrFcLong( 0x400c ), /* 16396 */
/* 190 */ NdrFcShort( 0x2da ), /* Offset= 730 (920) */
/* 192 */ NdrFcLong( 0x10 ), /* 16 */

```

```

/* 196 */ NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 198 */ NdrFcLong( 0x12 ), /* 18 */
/* 202 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 204 */ NdrFcLong( 0x13 ), /* 19 */
/* 208 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 210 */ NdrFcLong( 0x16 ), /* 22 */
/* 214 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 216 */ NdrFcLong( 0x17 ), /* 23 */
/* 220 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 222 */ NdrFcLong( 0xe ), /* 14 */
/* 226 */ NdrFcShort( 0x2be ), /* Offset= 702 (928) */
/* 228 */ NdrFcLong( 0x400e ), /* 16398 */
/* 232 */ NdrFcShort( 0x2c4 ), /* Offset= 708 (940) */
/* 234 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 238 */ NdrFcShort( 0x2c2 ), /* Offset= 706 (944) */
/* 240 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 244 */ NdrFcShort( 0x280 ), /* Offset= 640 (884) */
/* 246 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 250 */ NdrFcShort( 0x27e ), /* Offset= 638 (888) */
/* 252 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 256 */ NdrFcShort( 0x278 ), /* Offset= 632 (888) */
/* 258 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 262 */ NdrFcShort( 0x272 ), /* Offset= 626 (888) */
/* 264 */ NdrFcLong( 0x0 ), /* 0 */
/* 268 */ NdrFcShort( 0x0 ), /* Offset= 0 (268) */
/* 270 */ NdrFcLong( 0x1 ), /* 1 */
/* 274 */ NdrFcShort( 0x0 ), /* Offset= 0 (274) */
/* 276 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (275) */
/* 278 */
0x15, /* FC_STRUCT */
0x7, /* 7 */
/* 280 */ NdrFcShort( 0x8 ), /* 8 */
/* 282 */ 0xb, /* FC_HYPER */
0x5b, /* FC_END */
/* 284 */
0x12, 0x0, /* FC_UP */
/* 286 */ NdrFcShort( 0xc ), /* Offset= 12 (298) */
/* 288 */
0x1b, /* FC_CARRAY */
0x1, /* 1 */
/* 290 */ NdrFcShort( 0x2 ), /* 2 */
/* 292 */ 0x9, /* Corr desc: FC_ULONG */
0x0, /* */
/* 294 */ NdrFcShort( 0xfffc ), /* -4 */
/* 296 */ 0x6, /* FC_SHORT */
0x5b, /* FC_END */
/* 298 */
0x17, /* FC_CSTRUCT */
0x3, /* 3 */
/* 300 */ NdrFcShort( 0x8 ), /* 8 */
/* 302 */ NdrFcShort( 0xffffffff2 ), /* Offset= -14 (288) */
/* 304 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 306 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 308 */
0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 310 */ NdrFcLong( 0x0 ), /* 0 */
/* 314 */ NdrFcShort( 0x0 ), /* 0 */
/* 316 */ NdrFcShort( 0x0 ), /* 0 */
/* 318 */ 0xc0, /* 192 */
0x0, /* 0 */

```

```

/* 320 */ 0x0, /* 0 */
0x0, /* 0 */
/* 322 */ 0x0, /* 0 */
0x0, /* 0 */
/* 324 */ 0x0, /* 0 */
0x46, /* 70 */
/* 326 */
0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 328 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 332 */ NdrFcShort( 0x0 ), /* 0 */
/* 334 */ NdrFcShort( 0x0 ), /* 0 */
/* 336 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 338 */ 0x0, /* 0 */
0x0, /* 0 */
/* 340 */ 0x0, /* 0 */
0x0, /* 0 */
/* 342 */ 0x0, /* 0 */
0x46, /* 70 */
/* 344 */
0x12, 0x10, /* FC_UP [pointer_deref] */
/* 346 */ NdrFcShort( 0x2 ), /* Offset= 2 (348) */
/* 348 */
0x12, 0x0, /* FC_UP */
/* 350 */ NdrFcShort( 0x1fc ), /* Offset= 508 (858) */
/* 352 */
0x2a, /* FC_ENCAPSULATED_UNION */
0x49, /* 73 */
/* 354 */ NdrFcShort( 0x18 ), /* 24 */
/* 356 */ NdrFcShort( 0xa ), /* 10 */
/* 358 */ NdrFcLong( 0x8 ), /* 8 */
/* 362 */ NdrFcShort( 0x58 ), /* Offset= 88 (450) */
/* 364 */ NdrFcLong( 0xd ), /* 13 */
/* 368 */ NdrFcShort( 0x78 ), /* Offset= 120 (488) */
/* 370 */ NdrFcLong( 0x9 ), /* 9 */
/* 374 */ NdrFcShort( 0x94 ), /* Offset= 148 (522) */
/* 376 */ NdrFcLong( 0xc ), /* 12 */
/* 380 */ NdrFcShort( 0xbc ), /* Offset= 188 (568) */
/* 382 */ NdrFcLong( 0x24 ), /* 36 */
/* 386 */ NdrFcShort( 0x114 ), /* Offset= 276 (662) */
/* 388 */ NdrFcLong( 0x800d ), /* 32781 */
/* 392 */ NdrFcShort( 0x130 ), /* Offset= 304 (696) */
/* 394 */ NdrFcLong( 0x10 ), /* 16 */
/* 398 */ NdrFcShort( 0x148 ), /* Offset= 328 (726) */
/* 400 */ NdrFcLong( 0x2 ), /* 2 */
/* 404 */ NdrFcShort( 0x160 ), /* Offset= 352 (756) */
/* 406 */ NdrFcLong( 0x3 ), /* 3 */
/* 410 */ NdrFcShort( 0x178 ), /* Offset= 376 (786) */
/* 412 */ NdrFcLong( 0x14 ), /* 20 */
/* 416 */ NdrFcShort( 0x190 ), /* Offset= 400 (816) */
/* 418 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (417) */
/* 420 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 422 */ NdrFcShort( 0x4 ), /* 4 */
/* 424 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 426 */ NdrFcShort( 0x0 ), /* 0 */
/* 428 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 430 */

```

```

0x48, /* FC_VARIABLE_REPEAT */
0x49, /* FC_FIXED_OFFSET */
/* 432 */ NdrFcShort( 0x4 ), /* 4 */
/* 434 */ NdrFcShort( 0x0 ), /* 0 */
/* 436 */ NdrFcShort( 0x1 ), /* 1 */
/* 438 */ NdrFcShort( 0x0 ), /* 0 */
/* 440 */ NdrFcShort( 0x0 ), /* 0 */
/* 442 */ 0x12, 0x0, /* FC_UP */
/* 444 */ NdrFcShort( 0xfffff6e ), /* Offset= -146 (298) */
/* 446 */

0x5b, /* FC_END */

0x8, /* FC_LONG */
/* 448 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 450 */

0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 452 */ NdrFcShort( 0x8 ), /* 8 */
/* 454 */

0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 456 */

0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 458 */ NdrFcShort( 0x4 ), /* 4 */
/* 460 */ NdrFcShort( 0x4 ), /* 4 */
/* 462 */ 0x11, 0x0, /* FC_RP */
/* 464 */ NdrFcShort( 0xffffffd4 ), /* Offset= -44 (420) */
/* 466 */

0x5b, /* FC_END */

0x8, /* FC_LONG */
/* 468 */ 0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 470 */

0x21, /* FC_BOGUS_ARRAY */
0x3, /* 3 */
/* 472 */ NdrFcShort( 0x0 ), /* 0 */
/* 474 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 476 */ NdrFcShort( 0x0 ), /* 0 */
/* 478 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 482 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 484 */ NdrFcShort( 0xfffff50 ), /* Offset= -176 (308) */
/* 486 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 488 */

0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 490 */ NdrFcShort( 0x8 ), /* 8 */
/* 492 */ NdrFcShort( 0x0 ), /* 0 */
/* 494 */ NdrFcShort( 0x6 ), /* Offset= 6 (500) */
/* 496 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 498 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 500 */

0x11, 0x0, /* FC_RP */
/* 502 */ NdrFcShort( 0xffffffe0 ), /* Offset= -32 (470) */
/* 504 */

0x21, /* FC_BOGUS_ARRAY */

```

```

0x3, /* 3 */
/* 506 */ NdrFcShort( 0x0 ), /* 0 */
/* 508 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 510 */ NdrFcShort( 0x0 ), /* 0 */
/* 512 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 516 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 518 */ NdrFcShort( 0xfffff40 ), /* Offset= -192 (326) */
/* 520 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 522 */

0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 524 */ NdrFcShort( 0x8 ), /* 8 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 532 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 534 */

0x11, 0x0, /* FC_RP */
/* 536 */ NdrFcShort( 0xffffffe0 ), /* Offset= -32 (504) */
/* 538 */

0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 540 */ NdrFcShort( 0x4 ), /* 4 */
/* 542 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */

0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 548 */

0x48, /* FC_VARIABLE_REPEAT */
0x49, /* FC_FIXED_OFFSET */
/* 550 */ NdrFcShort( 0x4 ), /* 4 */
/* 552 */ NdrFcShort( 0x0 ), /* 0 */
/* 554 */ NdrFcShort( 0x1 ), /* 1 */
/* 556 */ NdrFcShort( 0x0 ), /* 0 */
/* 558 */ NdrFcShort( 0x0 ), /* 0 */
/* 560 */ 0x12, 0x0, /* FC_UP */
/* 562 */ NdrFcShort( 0x182 ), /* Offset= 386 (948) */
/* 564 */

0x5b, /* FC_END */

0x8, /* FC_LONG */
/* 566 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 568 */

0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 570 */ NdrFcShort( 0x8 ), /* 8 */
/* 572 */ NdrFcShort( 0x0 ), /* 0 */
/* 574 */ NdrFcShort( 0x6 ), /* Offset= 6 (580) */
/* 576 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 578 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 580 */

0x11, 0x0, /* FC_RP */
/* 582 */ NdrFcShort( 0xffffffd4 ), /* Offset= -44 (538) */

```

```

/* 584 */
                                0x2f,          /* FC_IP */
                                0x5a,          /* FC_CONSTANT_IID */

/* 586 */ NdrFcLong( 0x2f ), /* 47 */
/* 590 */ NdrFcShort( 0x0 ), /* 0 */
/* 592 */ NdrFcShort( 0x0 ), /* 0 */
/* 594 */ 0xc0,             /* 192 */
                                0x0,          /* 0 */
/* 596 */ 0x0,             /* 0 */
                                0x0,          /* 0 */
/* 598 */ 0x0,             /* 0 */
                                0x0,          /* 0 */
/* 600 */ 0x0,             /* 0 */
                                0x46,          /* 70 */
/* 602 */
                                0x1b,          /* FC_CARRAY */
                                0x0,          /* 0 */

/* 604 */ NdrFcShort( 0x1 ), /* 1 */
/* 606 */ 0x19,             /* Corr desc: field pointer, FC_ULONG */
                                0x0,          /* */
/* 608 */ NdrFcShort( 0x4 ), /* 4 */
/* 610 */ 0x1,             /* FC_BYTE */
                                0x5b,          /* FC_END */
/* 612 */
                                0x1a,          /* FC_BOOLEAN_STRUCT */
                                0x3,          /* 3 */

/* 614 */ NdrFcShort( 0x10 ), /* 16 */
/* 616 */ NdrFcShort( 0x0 ), /* 0 */
/* 618 */ NdrFcShort( 0xa ), /* Offset= 10 (628) */
/* 620 */ 0x8,             /* FC_LONG */
                                0x8,          /* FC_LONG */
/* 622 */ 0x4c,            /* FC_EMBEDDED_COMPLEX */
                                0x0,          /* 0 */
/* 624 */ NdrFcShort( 0xffffffffd8 ), /* Offset= -40 (584) */
/* 626 */ 0x36,            /* FC_POINTER */
                                0x5b,          /* FC_END */
/* 628 */
                                0x12, 0x0,      /* FC_UP */
/* 630 */ NdrFcShort( 0xffffffffe4 ), /* Offset= -28 (602) */
/* 632 */
                                0x1b,          /* FC_CARRAY */
                                0x3,          /* 3 */
/* 634 */ NdrFcShort( 0x4 ), /* 4 */
/* 636 */ 0x19,             /* Corr desc: field pointer, FC_ULONG */
                                0x0,          /* */
/* 638 */ NdrFcShort( 0x0 ), /* 0 */
/* 640 */
                                0x4b,          /* FC_PP */
                                0x5c,          /* FC_PAD */
/* 642 */
                                0x48,          /* FC_VARIABLE_REPEAT */
                                0x49,          /* FC_FIXED_OFFSET */

/* 644 */ NdrFcShort( 0x4 ), /* 4 */
/* 646 */ NdrFcShort( 0x0 ), /* 0 */
/* 648 */ NdrFcShort( 0x1 ), /* 1 */
/* 650 */ NdrFcShort( 0x0 ), /* 0 */
/* 652 */ NdrFcShort( 0x0 ), /* 0 */
/* 654 */ 0x12, 0x0,        /* FC_UP */
/* 656 */ NdrFcShort( 0xffffffffd4 ), /* Offset= -44 (612) */
/* 658 */
                                0x5b,          /* FC_END */
                                0x8,          /* FC_LONG */

```

```

/* 660 */ 0x5c,            /* FC_PAD */
                                0x5b,          /* FC_END */
/* 662 */
                                0x1a,          /* FC_BOOLEAN_STRUCT */
                                0x3,          /* 3 */

/* 664 */ NdrFcShort( 0x8 ), /* 8 */
/* 666 */ NdrFcShort( 0x0 ), /* 0 */
/* 668 */ NdrFcShort( 0x6 ), /* Offset= 6 (674) */
/* 670 */ 0x8,             /* FC_LONG */
                                0x36,          /* FC_POINTER */
/* 672 */ 0x5c,            /* FC_PAD */
                                0x5b,          /* FC_END */
/* 674 */
                                0x11, 0x0,      /* FC_UP */
/* 676 */ NdrFcShort( 0xffffffffd4 ), /* Offset= -44 (632) */
/* 678 */
                                0x1d,          /* FC_SMFARRAY */
                                0x0,          /* 0 */
/* 680 */ NdrFcShort( 0x8 ), /* 8 */
/* 682 */ 0x1,             /* FC_BYTE */
                                0x5b,          /* FC_END */
/* 684 */
                                0x15,          /* FC_STRUCT */
                                0x3,          /* 3 */

/* 686 */ NdrFcShort( 0x10 ), /* 16 */
/* 688 */ 0x8,             /* FC_LONG */
                                0x6,          /* FC_SHORT */
/* 690 */ 0x6,            /* FC_SHORT */
                                0x4c,          /* FC_EMBEDDED_COMPLEX */
/* 692 */ 0x0,            /* 0 */
                                NdrFcShort( 0xfffffffff1 ), /* Offset= -15 (678) */
/* 696 */ 0x5b,            /* FC_END */

                                0x1a,          /* FC_BOOLEAN_STRUCT */
                                0x3,          /* 3 */
/* 698 */ NdrFcShort( 0x18 ), /* 24 */
/* 700 */ NdrFcShort( 0x0 ), /* 0 */
/* 702 */ NdrFcShort( 0xa ), /* Offset= 10 (712) */
/* 704 */ 0x8,             /* FC_LONG */
                                0x36,          /* FC_POINTER */
/* 706 */ 0x4c,            /* FC_EMBEDDED_COMPLEX */
                                0x0,          /* 0 */
/* 708 */ NdrFcShort( 0xffffffffe8 ), /* Offset= -24 (684) */
/* 710 */ 0x5c,            /* FC_PAD */
                                0x5b,          /* FC_END */
/* 712 */
                                0x11, 0x0,      /* FC_UP */
/* 714 */ NdrFcShort( 0xfffffffff0c ), /* Offset= -244 (470) */
/* 716 */
                                0x1b,          /* FC_CARRAY */
                                0x0,          /* 0 */
/* 718 */ NdrFcShort( 0x1 ), /* 1 */
/* 720 */ 0x19,             /* Corr desc: field pointer, FC_ULONG */
                                0x0,          /* */
/* 722 */ NdrFcShort( 0x0 ), /* 0 */
/* 724 */ 0x1,             /* FC_BYTE */
                                0x5b,          /* FC_END */
/* 726 */
                                0x16,          /* FC_PSTRUCT */
                                0x3,          /* 3 */
/* 728 */ NdrFcShort( 0x8 ), /* 8 */
/* 730 */
                                0x4b,          /* FC_PP */

```

```

/* 732 */          0x5c,          /* FC_PAD */
/* 734 */ NdrFcShort( 0x4 ), /* 4 */
/* 736 */ NdrFcShort( 0x4 ), /* 4 */
/* 738 */ 0x12, 0x0, /* FC_UP */
/* 740 */ NdrFcShort( 0xfffffffffe8 ), /* Offset= -24 (716) */
/* 742 */
          0x5b,          /* FC_END */
          0x8,          /* FC_LONG */
/* 744 */ 0x8, /* FC_LONG */
/* 746 */          0x5b,          /* FC_END */
          0x1b,          /* FC_CARRAY */
          0x1,          /* 1 */
/* 748 */ NdrFcShort( 0x2 ), /* 2 */
/* 750 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
          0x0,          /* */
/* 752 */ NdrFcShort( 0x0 ), /* 0 */
/* 754 */ 0x6, /* FC_SHORT */
/* 756 */          0x5b,          /* FC_END */
          0x16,          /* FC_PSTRUCT */
          0x3,          /* 3 */
/* 758 */ NdrFcShort( 0x8 ), /* 8 */
/* 760 */
          0x4b,          /* FC_PP */
          0x5c,          /* FC_PAD */
/* 762 */
          0x46,          /* FC_NO_REPEAT */
          0x5c,          /* FC_PAD */
/* 764 */ NdrFcShort( 0x4 ), /* 4 */
/* 766 */ NdrFcShort( 0x4 ), /* 4 */
/* 768 */ 0x12, 0x0, /* FC_UP */
/* 770 */ NdrFcShort( 0xfffffffffe8 ), /* Offset= -24 (746) */
/* 772 */
          0x5b,          /* FC_END */
          0x8,          /* FC_LONG */
/* 774 */ 0x8, /* FC_LONG */
          0x5b,          /* FC_END */
/* 776 */
          0x1b,          /* FC_CARRAY */
          0x3,          /* 3 */
/* 778 */ NdrFcShort( 0x4 ), /* 4 */
/* 780 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
          0x0,          /* */
/* 782 */ NdrFcShort( 0x0 ), /* 0 */
/* 784 */ 0x8, /* FC_LONG */
          0x5b,          /* FC_END */
/* 786 */
          0x16,          /* FC_PSTRUCT */
          0x3,          /* 3 */
/* 788 */ NdrFcShort( 0x8 ), /* 8 */
/* 790 */
          0x4b,          /* FC_PP */
          0x5c,          /* FC_PAD */
/* 792 */
          0x46,          /* FC_NO_REPEAT */
          0x5c,          /* FC_PAD */
/* 794 */ NdrFcShort( 0x4 ), /* 4 */

```

```

/* 796 */ NdrFcShort( 0x4 ), /* 4 */
/* 798 */ 0x12, 0x0, /* FC_UP */
/* 800 */ NdrFcShort( 0xfffffffffe8 ), /* Offset= -24 (776) */
/* 802 */
          0x5b,          /* FC_END */
          0x8,          /* FC_LONG */
/* 804 */ 0x8, /* FC_LONG */
          0x5b,          /* FC_END */
/* 806 */
          0x1b,          /* FC_CARRAY */
          0x7,          /* 7 */
/* 808 */ NdrFcShort( 0x8 ), /* 8 */
/* 810 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
          0x0,          /* */
/* 812 */ NdrFcShort( 0x0 ), /* 0 */
/* 814 */ 0xb, /* FC_HYPER */
          0x5b,          /* FC_END */
/* 816 */
          0x16,          /* FC_PSTRUCT */
          0x3,          /* 3 */
/* 818 */ NdrFcShort( 0x8 ), /* 8 */
/* 820 */
          0x4b,          /* FC_PP */
          0x5c,          /* FC_PAD */
/* 822 */
          0x46,          /* FC_NO_REPEAT */
          0x5c,          /* FC_PAD */
/* 824 */ NdrFcShort( 0x4 ), /* 4 */
/* 826 */ NdrFcShort( 0x4 ), /* 4 */
/* 828 */ 0x12, 0x0, /* FC_UP */
/* 830 */ NdrFcShort( 0xfffffffffe8 ), /* Offset= -24 (806) */
/* 832 */
          0x5b,          /* FC_END */
          0x8,          /* FC_LONG */
/* 834 */ 0x8, /* FC_LONG */
          0x5b,          /* FC_END */
/* 836 */
          0x15,          /* FC_STRUCT */
          0x3,          /* 3 */
/* 838 */ NdrFcShort( 0x8 ), /* 8 */
/* 840 */ 0x8, /* FC_LONG */
          0x8,          /* FC_LONG */
          0x5c,          /* FC_PAD */
          0x5b,          /* FC_END */
/* 844 */
          0x1b,          /* FC_CARRAY */
          0x3,          /* 3 */
/* 846 */ NdrFcShort( 0x8 ), /* 8 */
/* 848 */ 0x7, /* Corr desc: FC_USHORT */
          0x0,          /* */
/* 850 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 852 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
          0x0,          /* 0 */
/* 854 */ NdrFcShort( 0xfffffffffee ), /* Offset= -18 (836) */
/* 856 */ 0x5c, /* FC_PAD */
          0x5b,          /* FC_END */
/* 858 */
          0x1a,          /* FC_BOOLEAN_STRUCT */
          0x3,          /* 3 */
/* 860 */ NdrFcShort( 0x28 ), /* 40 */
/* 862 */ NdrFcShort( 0xfffffffffee ), /* Offset= -18 (844) */

```

```

/* 864 */ NdrFcShort( 0x0 ), /* Offset= 0 (864) */
/* 866 */ 0x6, /* FC_SHORT */
/* 868 */ 0x38, /* FC_ALIGNM4 */
/* 870 */ 0x8, /* FC_LONG */
/* 872 */ 0x0, /* FC_EMBEDDED_COMPLEX */
/* 876 */ NdrFcShort( 0xfffffd7 ), /* Offset= -521 (352) */
/* 878 */ 0x5b, /* FC_END */
/* 880 */ 0x12, 0x0, /* FC_UP */
/* 882 */ NdrFcShort( 0xfffffe6 ), /* Offset= -266 (612) */
/* 884 */ 0x12, 0x8, /* FC_UP [simple_pointer] */
/* 886 */ 0x5c, /* FC_BYTE */
/* 888 */ 0x5c, /* FC_PAD */
/* 890 */ 0x12, 0x8, /* FC_UP [simple_pointer] */
/* 892 */ 0x5c, /* FC_BYTE */
/* 894 */ 0x5c, /* FC_PAD */
/* 896 */ 0x12, 0x8, /* FC_UP [simple_pointer] */
/* 898 */ 0x5c, /* FC_FLOAT */
/* 900 */ 0x5c, /* FC_PAD */
/* 902 */ 0x12, 0x0, /* FC_UP */
/* 904 */ NdrFcShort( 0xfffffd90 ), /* Offset= -624 (278) */
/* 906 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 908 */ NdrFcShort( 0xfffffd92 ), /* Offset= -622 (284) */
/* 910 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 912 */ NdrFcShort( 0xfffffda6 ), /* Offset= -602 (308) */
/* 914 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 916 */ NdrFcShort( 0xfffffdb4 ), /* Offset= -588 (326) */
/* 918 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 920 */ NdrFcShort( 0xfffffdc2 ), /* Offset= -574 (344) */
/* 922 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 924 */ NdrFcShort( 0x2 ), /* Offset= 2 (924) */
/* 926 */ 0x12, 0x0, /* FC_UP */
/* 928 */ NdrFcShort( 0x16 ), /* Offset= 22 (948) */
/* 930 */ 0x15, /* FC_STRUCT */
/* 932 */ 0x7, /* 7 */
/* 934 */ 0x1, /* FC_BYTE */
/* 936 */ 0x8, /* FC_ALIGNM4 */

```

```

/* 938 */ 0xb, /* FC_HYPER */
/* 940 */ 0x5b, /* FC_END */
/* 942 */ 0x12, 0x0, /* FC_UP */
/* 944 */ NdrFcShort( 0xfffffff2 ), /* Offset= -14 (928) */
/* 946 */ 0x2, /* FC_UP [simple_pointer] */
/* 948 */ 0x5c, /* FC_CHAR */
/* 950 */ 0x1a, /* FC_PAD */
/* 952 */ 0x7, /* FC_BOGUS_STRUCT */
/* 954 */ NdrFcShort( 0x20 ), /* 32 */
/* 956 */ NdrFcShort( 0x0 ), /* 0 */
/* 958 */ 0x8, /* Offset= 0 (954) */
/* 960 */ 0x6, /* FC_LONG */
/* 962 */ 0x6, /* FC_SHORT */
/* 964 */ 0x6, /* FC_SHORT */
/* 966 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 968 */ 0x0, /* 0 */
/* 970 */ NdrFcShort( 0xfffffc42 ), /* Offset= -958 (6) */
/* 972 */ 0x5c, /* FC_PAD */
/* 974 */ 0x5b, /* FC_END */
/* 976 */ 0xb4, /* FC_USER_MARSHAL */
/* 978 */ 0x83, /* 131 */
/* 980 */ NdrFcShort( 0x0 ), /* 0 */
/* 982 */ NdrFcShort( 0x10 ), /* 16 */
/* 984 */ NdrFcShort( 0x0 ), /* 0 */
/* 986 */ 0xb4, /* FC_USER_MARSHAL */
/* 988 */ 0x83, /* 131 */
/* 990 */ NdrFcShort( 0x0 ), /* 0 */
/* 992 */ NdrFcShort( 0x10 ), /* 16 */
/* 994 */ NdrFcShort( 0xfffffff4 ), /* Offset= -12 (982) */
/* 996 */ 0x0
};

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &ITPCProxyVtbl,
    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_AXP64) */

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the proxy stub code */

/* File created by MIDL compiler version 5.03.0280 */
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), 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_AXP64)
#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__ 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 979
#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;

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
   GUID={0x00000000, 0x0000, 0x0000, {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}} */

/* Object interface: IUnknown, ver. 0.0,
   GUID={0x00000000, 0x0000, 0x0000, {0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x46}} */

/* Object interface: ITPCC, ver. 0.0,
   GUID={0xFEEE6AA2, 0x84B1, 0x11d2, {0xBA, 0x47, 0x00, 0xC0, 0x4F, 0xBF, 0xE0, 0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
{
    0,
    44,
    88,
    132,
    176,
    220
};

static const MIDL_SERVER_INFO ITPCC_ServerInfo =

```

```

{
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0,
    0
};

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
{
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
};

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy,
    (void *)-1 /* ITPCC::NewOrder */ ,
    (void *)-1 /* ITPCC::Payment */ ,
    (void *)-1 /* ITPCC::Delivery */ ,
    (void *)-1 /* ITPCC::StockLevel */ ,
    (void *)-1 /* ITPCC::OrderStatus */ ,
    (void *)-1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
};

```

```

0,
UserMarshalRoutines,
0, /* notify & notify_flag routine table */
0x1, /* MIDL flag */
0, /* Reserved3 */
0, /* Reserved4 */
0 /* Reserved5 */
};

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        VARIANT_UserSize,
        VARIANT_UserMarshal,
        VARIANT_UserUnmarshal,
        VARIANT_UserFree
    }
};

#ifdef __RPC_WIN64__
#error Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
{
    0,
    {
        /* Procedure NewOrder */

        0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object, Oi2 */

        /* 2 */ NdrFcLong( 0x0 ), /* 0 */
        /* 6 */ NdrFcShort( 0x3 ), /* 3 */
#ifdef _ALPHA_
        /* 8 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
        NdrFcShort( 0x30 ), /* xpp64 Stack size/offset = 48 */
#endif
        /* 10 */ NdrFcShort( 0x0 ), /* 0 */
        /* 12 */ NdrFcShort( 0x8 ), /* 8 */
        /* 14 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
        0x3, /* 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, */
#ifdef _ALPHA_
        /* 28 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */

```

```

#else
                                NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 30 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 32 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 34 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
                                NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 36 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 38 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
/* 40 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
                                NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 42 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure Payment */

/* 44 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, 0i2 */
/* 46 */ NdrFcLong( 0x0 ), /* 0 */
/* 50 */ NdrFcShort( 0x4 ), /* 4 */
#ifndef _ALPHA_
/* 52 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
                                NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 54 */ NdrFcShort( 0x0 ), /* 0 */
/* 56 */ NdrFcShort( 0x8 ), /* 8 */
/* 58 */ 0x47, /* 0i2 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, */
#ifndef _ALPHA_
/* 72 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
                                NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 74 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

```

```

/* 76 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 78 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
                                NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 80 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 82 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
/* 84 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
                                NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 86 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure Delivery */

/* 88 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, 0i2 */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
#ifndef _ALPHA_
/* 96 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
                                NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 98 */ NdrFcShort( 0x0 ), /* 0 */
/* 100 */ NdrFcShort( 0x8 ), /* 8 */
/* 102 */ 0x47, /* 0i2 Flags: srv must size, clt must size, has
return, has ext, */
0x3, /* 3 */
/* 104 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 106 */ NdrFcShort( 0x20 ), /* 32 */
/* 108 */ NdrFcShort( 0x20 ), /* 32 */
/* 110 */ NdrFcShort( 0x0 ), /* 0 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 114 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifndef _ALPHA_
/* 116 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
                                NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 118 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 120 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 122 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
                                NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif

```

```

/* 124 */ NdrFcShort( 0x3c8 ),          /* Type Offset=968 */

        /* Return value */

/* 126 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
/* 128 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
        NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 130 */ 0x8,                /* FC_LONG */
                                0x0,                /* 0 */

        /* Procedure StockLevel */

/* 132 */ 0x33,                /* FC_AUTO_HANDLE */
                                0x6c,                /* Old Flags: object, 0i2 */
/* 134 */ NdrFcLong( 0x0 ),    /* 0 */
/* 138 */ NdrFcShort( 0x6 ),   /* 6 */
#ifdef _ALPHA_
/* 140 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
        NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 142 */ NdrFcShort( 0x0 ),   /* 0 */
/* 144 */ NdrFcShort( 0x8 ),   /* 8 */
/* 146 */ 0x47,                /* 0i2 Flags: srv must size, clt must size, has
return, has ext, */
                                0x3,                /* 3 */
/* 148 */ 0xa,                /* 10 */
                                0x7,                /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 150 */ NdrFcShort( 0x20 ), /* 32 */
/* 152 */ NdrFcShort( 0x20 ), /* 32 */
/* 154 */ NdrFcShort( 0x0 ),   /* 0 */
/* 156 */ NdrFcShort( 0x0 ),   /* 0 */

        /* Parameter txn_in */

/* 158 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
/* 160 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
        NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 162 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

        /* Parameter txn_out */

/* 164 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef _ALPHA_
/* 166 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
        NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 168 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

        /* Return value */

/* 170 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
/* 172 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */

```

```

#else
        NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 174 */ 0x8,                /* FC_LONG */
                                0x0,                /* 0 */

        /* Procedure OrderStatus */

/* 176 */ 0x33,                /* FC_AUTO_HANDLE */
                                0x6c,                /* Old Flags: object, 0i2 */
/* 178 */ NdrFcLong( 0x0 ),    /* 0 */
/* 182 */ NdrFcShort( 0x7 ),   /* 7 */
#ifdef _ALPHA_
/* 184 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
        NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 186 */ NdrFcShort( 0x0 ),   /* 0 */
/* 188 */ NdrFcShort( 0x8 ),   /* 8 */
/* 190 */ 0x47,                /* 0i2 Flags: srv must size, clt must size, has
return, has ext, */
                                0x3,                /* 3 */
/* 192 */ 0xa,                /* 10 */
                                0x7,                /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 194 */ NdrFcShort( 0x20 ), /* 32 */
/* 196 */ NdrFcShort( 0x20 ), /* 32 */
/* 198 */ NdrFcShort( 0x0 ),   /* 0 */
/* 200 */ NdrFcShort( 0x0 ),   /* 0 */

        /* Parameter txn_in */

/* 202 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
/* 204 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
        NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 206 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

        /* Parameter txn_out */

/* 208 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef _ALPHA_
/* 210 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
        NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 212 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

        /* Return value */

/* 214 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
/* 216 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
        NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 218 */ 0x8,                /* FC_LONG */
                                0x0,                /* 0 */

        /* Procedure CallSetComplete */

```

```

/* 220 */ 0x33,          /* FC_AUTO_HANDLE */
0x6c,                  /* Old Flags: object, 0i2 */

/* 222 */ NdrFcLong( 0x0 ), /* 0 */
/* 226 */ NdrFcShort( 0x8 ), /* 8 */
/* 228 */ NdrFcShort( 0x10 ), /* 1a64, axp64 Stack size/offset = 16 */
/* 230 */ NdrFcShort( 0x0 ), /* 0 */
/* 232 */ NdrFcShort( 0x8 ), /* 8 */
/* 234 */ 0x44,          /* 0i2 Flags: has return, has ext, */
0x1,                  /* 1 */
/* 236 */ 0xa,          /* 10 */
0x1,                  /* Ext Flags: new corr desc, */

/* 238 */ NdrFcShort( 0x0 ), /* 0 */
/* 240 */ NdrFcShort( 0x0 ), /* 0 */
/* 242 */ NdrFcShort( 0x0 ), /* 0 */
/* 244 */ NdrFcShort( 0x0 ), /* 0 */

/* Return value */

/* 246 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 248 */ NdrFcShort( 0x8 ), /* 1a64, axp64 Stack size/offset = 8 */
/* 250 */ 0x8,          /* FC_LONG */
0x0,                  /* 0 */

0x0

}
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
    0,
    {
        NdrFcShort( 0x0 ), /* 0 */

/* 2 */
0x12, 0x0,          /* FC_UP */
/* 4 */ NdrFcShort( 0x39e ), /* Offset= 926 (930) */
/* 6 */
0x2b,              /* FC_NON_ENCAPSULATED_UNION */
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( 0x2b ), /* 43 */
/* 20 */ NdrFcLong( 0x3 ), /* 3 */
/* 24 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 26 */ NdrFcLong( 0x11 ), /* 17 */
/* 30 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 32 */ NdrFcLong( 0x2 ), /* 2 */
/* 36 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 38 */ NdrFcLong( 0x4 ), /* 4 */
/* 42 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 44 */ NdrFcLong( 0x5 ), /* 5 */
/* 48 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 50 */ NdrFcLong( 0xb ), /* 11 */
/* 54 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 56 */ NdrFcLong( 0xa ), /* 10 */
/* 60 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 62 */ NdrFcLong( 0x6 ), /* 6 */
/* 66 */ NdrFcShort( 0xd6 ), /* Offset= 214 (280) */
/* 68 */ NdrFcLong( 0x7 ), /* 7 */

```

```

/* 72 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 74 */ NdrFcLong( 0x8 ), /* 8 */
/* 78 */ NdrFcShort( 0xd0 ), /* Offset= 208 (286) */
/* 80 */ NdrFcLong( 0xd ), /* 13 */
/* 84 */ NdrFcShort( 0xe4 ), /* Offset= 228 (312) */
/* 86 */ NdrFcLong( 0x9 ), /* 9 */
/* 90 */ NdrFcShort( 0xf0 ), /* Offset= 240 (330) */
/* 92 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 96 */ NdrFcShort( 0xfc ), /* Offset= 252 (348) */
/* 98 */ NdrFcLong( 0x24 ), /* 36 */
/* 102 */ NdrFcShort( 0x2f4 ), /* Offset= 756 (858) */
/* 104 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 108 */ NdrFcShort( 0x2ee ), /* Offset= 750 (858) */
/* 110 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 114 */ NdrFcShort( 0x2ec ), /* Offset= 748 (862) */
/* 116 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 120 */ NdrFcShort( 0x2ea ), /* Offset= 746 (866) */
/* 122 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 126 */ NdrFcShort( 0x2e8 ), /* Offset= 744 (870) */
/* 128 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 132 */ NdrFcShort( 0x2e6 ), /* Offset= 742 (874) */
/* 134 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 138 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (878) */
/* 140 */ NdrFcLong( 0x400b ), /* 16395 */
/* 144 */ NdrFcShort( 0x2d2 ), /* Offset= 722 (866) */
/* 146 */ NdrFcLong( 0x400a ), /* 16394 */
/* 150 */ NdrFcShort( 0x2d0 ), /* Offset= 720 (870) */
/* 152 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 156 */ NdrFcShort( 0x2d6 ), /* Offset= 726 (882) */
/* 158 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 162 */ NdrFcShort( 0x2cc ), /* Offset= 716 (878) */
/* 164 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 168 */ NdrFcShort( 0x2ce ), /* Offset= 718 (886) */
/* 170 */ NdrFcLong( 0x400d ), /* 16397 */
/* 174 */ NdrFcShort( 0x2cc ), /* Offset= 716 (890) */
/* 176 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 180 */ NdrFcShort( 0x2ca ), /* Offset= 714 (894) */
/* 182 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 186 */ NdrFcShort( 0x2c8 ), /* Offset= 712 (898) */
/* 188 */ NdrFcLong( 0x400c ), /* 16396 */
/* 192 */ NdrFcShort( 0x2c6 ), /* Offset= 710 (902) */
/* 194 */ NdrFcLong( 0x10 ), /* 16 */
/* 198 */ NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 200 */ NdrFcLong( 0x12 ), /* 18 */
/* 204 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 206 */ NdrFcLong( 0x13 ), /* 19 */
/* 210 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 212 */ NdrFcLong( 0x16 ), /* 22 */
/* 216 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 218 */ NdrFcLong( 0x17 ), /* 23 */
/* 222 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 224 */ NdrFcLong( 0xe ), /* 14 */
/* 228 */ NdrFcShort( 0x2aa ), /* Offset= 682 (910) */
/* 230 */ NdrFcLong( 0x400e ), /* 16398 */
/* 234 */ NdrFcShort( 0x2b0 ), /* Offset= 688 (922) */
/* 236 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 240 */ NdrFcShort( 0x2ae ), /* Offset= 686 (926) */
/* 242 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 246 */ NdrFcShort( 0x26c ), /* Offset= 620 (866) */
/* 248 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 252 */ NdrFcShort( 0x26a ), /* Offset= 618 (870) */
/* 254 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 258 */ NdrFcShort( 0x264 ), /* Offset= 612 (870) */

```

```

/* 260 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 264 */ NdrFcShort( 0x25e ), /* Offset= 606 (870) */
/* 266 */ NdrFcLong( 0x0 ), /* 0 */
/* 270 */ NdrFcShort( 0x0 ), /* Offset= 0 (270) */
/* 272 */ NdrFcLong( 0x1 ), /* 1 */
/* 276 */ NdrFcShort( 0x0 ), /* Offset= 0 (276) */
/* 278 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (277) */
/* 280 */
/* FC_STRUCT */
0x15, /* 7 */
0x7, /* 8 */
/* 282 */ NdrFcShort( 0x8 ), /* FC_HYPER */
/* 284 */ 0xb, /* FC_END */
0x5b, /* 286 */
/* FC_UP */
0x12, 0x0, /* Offset= 14 (302) */
/* 288 */ NdrFcShort( 0xe ), /* FC_CARRAY */
/* 290 */
0x1b, /* 1 */
0x1, /* 2 */
/* 292 */ NdrFcShort( 0x2 ), /* Corr desc: FC_ULONG */
/* 294 */ 0x9, /* 3 */
/* 296 */ NdrFcShort( 0xffffc ), /* -4 */
/* 298 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 300 */ 0x6, /* FC_SHORT */
0x5b, /* FC_END */
/* 302 */
/* FC_CSTRUCT */
0x17, /* 3 */
0x3, /* 8 */
/* 304 */ NdrFcShort( 0x8 ), /* Offset= -16 (290) */
/* 306 */ NdrFcShort( 0xffffffff0 ), /* FC_LONG */
/* 308 */ 0x8, /* FC_LONG */
/* 310 */ 0x5c, /* FC_PAD */
/* 312 */
0x5b, /* FC_END */
/* FC_IP */
0x2f, /* FC_CONSTANT_IID */
0x5a, /* 0 */
/* 314 */ NdrFcLong( 0x0 ), /* 0 */
/* 318 */ NdrFcShort( 0x0 ), /* 0 */
/* 320 */ NdrFcShort( 0x0 ), /* 0 */
/* 322 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 324 */ 0x0, /* 0 */
0x0, /* 0 */
/* 326 */ 0x0, /* 0 */
0x0, /* 0 */
/* 328 */ 0x0, /* 70 */
0x46, /* 330 */
/* FC_IP */
0x2f, /* FC_CONSTANT_IID */
0x5a, /* 132096 */
/* 332 */ NdrFcLong( 0x20400 ), /* 0 */
/* 336 */ NdrFcShort( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 342 */ 0x0, /* 0 */
0x0, /* 0 */
/* 344 */ 0x0, /* 0 */
0x0, /* 0 */
/* 346 */ 0x0, /* 70 */
0x46,

```

```

/* 348 */
/* FC_UP [pointer_deref] */
0x12, 0x10, /* Offset= 2 (352) */
/* 350 */ NdrFcShort( 0x2 ), /* FC_UP */
/* 352 */
0x12, 0x0, /* Offset= 486 (840) */
/* 354 */ NdrFcShort( 0x1e6 ), /* FC_ENCAPSULATED_UNION */
/* 356 */
0x2a, /* 137 */
0x89, /* 32 */
/* 358 */ NdrFcShort( 0x20 ), /* 10 */
/* 360 */ NdrFcShort( 0xa ), /* 8 */
/* 362 */ NdrFcLong( 0x8 ), /* Offset= 80 (446) */
/* 366 */ NdrFcShort( 0x50 ), /* 13 */
/* 368 */ NdrFcLong( 0xd ), /* Offset= 112 (484) */
/* 372 */ NdrFcShort( 0x70 ), /* 9 */
/* 374 */ NdrFcLong( 0x9 ), /* Offset= 144 (522) */
/* 378 */ NdrFcShort( 0x90 ), /* 12 */
/* 380 */ NdrFcLong( 0xc ), /* Offset= 176 (560) */
/* 384 */ NdrFcShort( 0xb0 ), /* 36 */
/* 386 */ NdrFcLong( 0x24 ), /* Offset= 260 (650) */
/* 390 */ NdrFcShort( 0x104 ), /* 32781 */
/* 392 */ NdrFcLong( 0x800d ), /* Offset= 288 (684) */
/* 396 */ NdrFcShort( 0x120 ), /* 16 */
/* 398 */ NdrFcLong( 0x10 ), /* Offset= 314 (716) */
/* 402 */ NdrFcShort( 0x13a ), /* 2 */
/* 404 */ NdrFcLong( 0x2 ), /* Offset= 336 (744) */
/* 408 */ NdrFcShort( 0x150 ), /* 3 */
/* 410 */ NdrFcLong( 0x3 ), /* Offset= 358 (772) */
/* 414 */ NdrFcShort( 0x166 ), /* 20 */
/* 416 */ NdrFcLong( 0x14 ), /* Offset= 380 (800) */
/* 420 */ NdrFcShort( 0x17c ), /* Offset= -1 (421) */
/* 422 */ NdrFcShort( 0xffffffff ), /* FC_BOOLEAN_ARRAY */
/* 424 */
0x21, /* 3 */
0x3, /* 0 */
/* 426 */ NdrFcShort( 0x0 ), /* Corr desc: field pointer, FC_ULONG */
/* 428 */ 0x19, /* 0 */
0x0, /* 0 */
/* 430 */ NdrFcShort( 0x0 ), /* 0 */
/* 432 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 434 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 438 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 440 */
0x12, 0x0, /* FC_UP */
/* 442 */ NdrFcShort( 0xffffffff74 ), /* Offset= -140 (302) */
/* 444 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 446 */
0x1a, /* FC_BOOLEAN_STRUCT */
0x3, /* 3 */
/* 448 */ NdrFcShort( 0x10 ), /* 16 */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ NdrFcShort( 0x6 ), /* Offset= 6 (458) */
/* 454 */ 0x8, /* FC_LONG */
0x39, /* FC_ALIGNM8 */
/* 456 */ 0x36, /* FC_POINTER */
0x5b, /* FC_END */
/* 458 */
0x11, 0x0, /* FC_RP */
/* 460 */ NdrFcShort( 0xffffffffdc ), /* Offset= -36 (424) */
/* 462 */
0x21, /* FC_BOOLEAN_ARRAY */
0x3, /* 3 */

```

```

/* 464 */ NdrFcShort( 0x0 ), /* 0 */
/* 466 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 468 */ NdrFcShort( 0x0 ), /* 0 */
/* 470 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 472 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 476 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 478 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 480 */ NdrFcShort( 0xffffffff58 ), /* Offset= -168 (312) */
/* 482 */ 0x5c, /* FC_PAD */
/* 484 */ 0x5b, /* FC_END */
/* 486 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 488 */ 0x3, /* 3 */
/* 490 */ NdrFcShort( 0x10 ), /* 16 */
/* 492 */ NdrFcShort( 0x0 ), /* 0 */
/* 494 */ NdrFcShort( 0x6 ), /* Offset= 6 (496) */
/* 496 */ 0x8, /* FC_LONG */
/* 498 */ 0x39, /* FC_ALIGNM8 */
/* 500 */ 0x36, /* FC_POINTER */
/* 502 */ 0x5b, /* FC_END */
/* 504 */ 0x11, 0x0, /* FC_RP */
/* 506 */ NdrFcShort( 0xffffffffdc ), /* Offset= -36 (462) */
/* 508 */ 0x21, /* FC_BOGUS_ARRAY */
/* 510 */ 0x3, /* 3 */
/* 512 */ NdrFcShort( 0x0 ), /* 0 */
/* 514 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 516 */ 0x0, /* 0 */
/* 518 */ NdrFcShort( 0x0 ), /* Corr flags: early, */
/* 520 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 522 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 524 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 526 */ 0x0, /* 0 */
/* 528 */ NdrFcShort( 0xffffffff44 ), /* Offset= -188 (330) */
/* 530 */ 0x5c, /* FC_PAD */
/* 532 */ 0x5b, /* FC_END */
/* 534 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 536 */ 0x3, /* 3 */
/* 538 */ NdrFcShort( 0x10 ), /* 16 */
/* 540 */ NdrFcShort( 0x0 ), /* 0 */
/* 542 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 544 */ 0x8, /* FC_LONG */
/* 546 */ 0x39, /* FC_ALIGNM8 */
/* 548 */ 0x36, /* FC_POINTER */
/* 550 */ 0x5b, /* FC_END */
/* 552 */ 0x11, 0x0, /* FC_RP */
/* 554 */ NdrFcShort( 0xffffffffdc ), /* Offset= -36 (500) */
/* 556 */ 0x21, /* FC_BOGUS_ARRAY */
/* 558 */ 0x3, /* 3 */
/* 560 */ NdrFcShort( 0x0 ), /* 0 */
/* 562 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 564 */ 0x0, /* 0 */
/* 566 */ NdrFcShort( 0x0 ), /* Corr flags: early, */
/* 568 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 570 */ NdrFcShort( 0x0 ), /* Corr flags: */

```

```

/* 554 */ 0x12, 0x0, /* FC_UP */
/* 556 */ NdrFcShort( 0x176 ), /* Offset= 374 (930) */
/* 558 */ 0x5c, /* FC_PAD */
/* 560 */ 0x5b, /* FC_END */
/* 562 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 564 */ 0x3, /* 3 */
/* 566 */ NdrFcShort( 0x10 ), /* 16 */
/* 568 */ NdrFcShort( 0x0 ), /* 0 */
/* 570 */ NdrFcShort( 0x6 ), /* Offset= 6 (572) */
/* 572 */ 0x8, /* FC_LONG */
/* 574 */ 0x39, /* FC_ALIGNM8 */
/* 576 */ 0x36, /* FC_POINTER */
/* 578 */ 0x5b, /* FC_END */
/* 580 */ 0x11, 0x0, /* FC_RP */
/* 582 */ NdrFcShort( 0xffffffffdc ), /* Offset= -36 (538) */
/* 584 */ 0x2f, /* FC_IP */
/* 586 */ 0x5a, /* FC_CONSTANT_IID */
/* 588 */ NdrFcLong( 0x2f ), /* 47 */
/* 590 */ NdrFcShort( 0x0 ), /* 0 */
/* 592 */ NdrFcShort( 0x0 ), /* 0 */
/* 594 */ 0xc0, /* 192 */
/* 596 */ 0x0, /* 0 */
/* 598 */ 0x0, /* 0 */
/* 600 */ 0x0, /* 0 */
/* 602 */ 0x0, /* 0 */
/* 604 */ 0x46, /* 70 */
/* 606 */ 0x1b, /* FC_CARRAY */
/* 608 */ 0x0, /* 0 */
/* 610 */ NdrFcShort( 0x1 ), /* 1 */
/* 612 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 614 */ 0x0, /* 0 */
/* 616 */ NdrFcShort( 0x4 ), /* 4 */
/* 618 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 620 */ 0x1, /* FC_BYTE */
/* 622 */ 0x5b, /* FC_END */
/* 624 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 626 */ 0x3, /* 3 */
/* 628 */ NdrFcShort( 0x18 ), /* 24 */
/* 630 */ NdrFcShort( 0x0 ), /* 0 */
/* 632 */ NdrFcShort( 0xc ), /* Offset= 12 (624) */
/* 634 */ 0x8, /* FC_LONG */
/* 636 */ 0x8, /* FC_LONG */
/* 638 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 640 */ 0x0, /* 0 */
/* 642 */ NdrFcShort( 0xffffffffd6 ), /* Offset= -42 (576) */
/* 644 */ 0x39, /* FC_ALIGNM8 */
/* 646 */ 0x36, /* FC_POINTER */
/* 648 */ 0x5c, /* FC_PAD */
/* 650 */ 0x5b, /* FC_END */
/* 652 */ 0x12, 0x0, /* FC_UP */
/* 654 */ NdrFcShort( 0xffffffffe0 ), /* Offset= -32 (594) */
/* 656 */ 0x21, /* FC_BOGUS_ARRAY */
/* 658 */ 0x3, /* 3 */

```

```

/* 630 */ NdrFcShort( 0x0 ), /* 0 */
/* 632 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 634 */ NdrFcShort( 0x0 ), /* 0 */
/* 636 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 638 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 642 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 644 */
/* 646 */ NdrFcShort( 0xffffffff8 ), /* Offset= -40 (606) */
/* 648 */ 0x5c, /* FC_PAD */
/* 650 */ 0x5b, /* FC_END */
/* 652 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 654 */ NdrFcShort( 0x10 ), /* 16 */
/* 656 */ NdrFcShort( 0x0 ), /* 0 */
/* 658 */ NdrFcShort( 0x6 ), /* Offset= 6 (662) */
/* 660 */ 0x8, /* FC_LONG */
/* 662 */ 0x39, /* FC_ALIGNM8 */
/* 664 */ 0x36, /* FC_POINTER */
/* 666 */ 0x5b, /* FC_END */
/* 668 */ 0x11, 0x0, /* FC_RP */
/* 670 */ NdrFcShort( 0xffffffffc ), /* Offset= -36 (628) */
/* 672 */ 0x1d, /* FC_SMFARRAY */
/* 674 */ 0x0, /* 0 */
/* 676 */ 0x8, /* FC_BYTE */
/* 678 */ 0x1, /* FC_BYTE */
/* 680 */ 0x5b, /* FC_END */
/* 682 */ 0x15, /* FC_STRUCT */
/* 684 */ 0x3, /* 3 */
/* 686 */ NdrFcShort( 0x10 ), /* 16 */
/* 688 */ NdrFcShort( 0x0 ), /* 0 */
/* 690 */ NdrFcShort( 0xa ), /* Offset= 10 (700) */
/* 692 */ 0x8, /* FC_LONG */
/* 694 */ 0x39, /* FC_ALIGNM8 */
/* 696 */ 0x36, /* FC_POINTER */
/* 698 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 700 */ 0x0, /* 0 */
/* 702 */ NdrFcShort( 0xfffffffff1 ), /* Offset= -15 (666) */
/* 704 */ 0x5b, /* FC_END */
/* 706 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 708 */ 0x3, /* 3 */
/* 710 */ NdrFcShort( 0x20 ), /* 32 */
/* 712 */ NdrFcShort( 0x0 ), /* 0 */
/* 714 */ NdrFcShort( 0xa ), /* Offset= 10 (700) */
/* 716 */ 0x8, /* FC_LONG */
/* 718 */ 0x39, /* FC_ALIGNM8 */
/* 720 */ 0x36, /* FC_POINTER */
/* 722 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 724 */ 0x0, /* 0 */
/* 726 */ NdrFcShort( 0xfffffffff7 ), /* Offset= -25 (672) */
/* 728 */ 0x5b, /* FC_END */
/* 730 */ 0x11, 0x0, /* FC_RP */
/* 732 */ NdrFcShort( 0xffffffff10 ), /* Offset= -240 (462) */
/* 734 */ 0x1b, /* FC_CARRY */
/* 736 */ 0x0, /* 0 */
/* 738 */ NdrFcShort( 0x1 ), /* 1 */
/* 740 */ 0x19, /* Corr desc: field pointer, FC_ULONG */

```

```

0x0, /* */
/* 710 */ NdrFcShort( 0x0 ), /* 0 */
/* 712 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 714 */ 0x1, /* FC_BYTE */
/* 716 */ 0x5b, /* FC_END */
/* 718 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 720 */ 0x3, /* 3 */
/* 722 */ NdrFcShort( 0x10 ), /* 16 */
/* 724 */ NdrFcShort( 0x0 ), /* 0 */
/* 726 */ NdrFcShort( 0x6 ), /* Offset= 6 (728) */
/* 728 */ 0x8, /* FC_LONG */
/* 730 */ 0x39, /* FC_ALIGNM8 */
/* 732 */ 0x36, /* FC_POINTER */
/* 734 */ 0x5b, /* FC_END */
/* 736 */ 0x12, 0x0, /* FC_UP */
/* 738 */ NdrFcShort( 0xfffffffff6 ), /* Offset= -26 (704) */
/* 740 */ 0x1b, /* FC_CARRY */
/* 742 */ 0x1, /* 1 */
/* 744 */ 0x2, /* 2 */
/* 746 */ NdrFcShort( 0x2 ), /* Corr desc: field pointer, FC_ULONG */
/* 748 */ 0x19, /* 0 */
/* 750 */ NdrFcShort( 0x0 ), /* 0 */
/* 752 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 754 */ 0x6, /* FC_SHORT */
/* 756 */ 0x5b, /* FC_END */
/* 758 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 760 */ 0x3, /* 3 */
/* 762 */ NdrFcShort( 0x10 ), /* 16 */
/* 764 */ NdrFcShort( 0x0 ), /* 0 */
/* 766 */ NdrFcShort( 0x6 ), /* Offset= 6 (756) */
/* 768 */ 0x8, /* FC_LONG */
/* 770 */ 0x39, /* FC_ALIGNM8 */
/* 772 */ 0x36, /* FC_POINTER */
/* 774 */ 0x5b, /* FC_END */
/* 776 */ 0x12, 0x0, /* FC_UP */
/* 778 */ NdrFcShort( 0xfffffffff6 ), /* Offset= -26 (732) */
/* 780 */ 0x1b, /* FC_CARRY */
/* 782 */ 0x3, /* 3 */
/* 784 */ 0x4, /* 4 */
/* 786 */ NdrFcShort( 0x4 ), /* Corr desc: field pointer, FC_ULONG */
/* 788 */ 0x19, /* 0 */
/* 790 */ NdrFcShort( 0x0 ), /* 0 */
/* 792 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 794 */ 0x8, /* FC_LONG */
/* 796 */ 0x5b, /* FC_END */
/* 798 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 800 */ 0x3, /* 3 */
/* 802 */ NdrFcShort( 0x10 ), /* 16 */
/* 804 */ NdrFcShort( 0x0 ), /* 0 */
/* 806 */ NdrFcShort( 0x6 ), /* Offset= 6 (784) */
/* 808 */ 0x8, /* FC_LONG */
/* 810 */ 0x39, /* FC_ALIGNM8 */
/* 812 */ 0x36, /* FC_POINTER */
/* 814 */ 0x5b, /* FC_END */
/* 816 */ 0x12, 0x0, /* FC_UP */

```



```

/* 786 */ NdrFcShort( 0xffffffffe6 ), /* Offset= -26 (760) */
/* 788 */
/* 790 */ NdrFcShort( 0x8 ), /* 8 */
/* 792 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 794 */ NdrFcShort( 0x0 ), /* 0 */
/* 796 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 798 */ 0xb, /* FC_HYPER */
/* 800 */ 0x5b, /* FC_END */
/* 802 */ NdrFcShort( 0x1a ), /* FC_BOGUS_STRUCT */
/* 804 */ 0x3, /* 3 */
/* 806 */ NdrFcShort( 0x10 ), /* 16 */
/* 808 */ 0x0, /* 0 */
/* 810 */ 0x8, /* Offset= 6 (812) */
/* 812 */ 0x39, /* FC_ALIGNM8 */
/* 814 */ 0x36, /* FC_POINTER */
/* 816 */ 0x5b, /* FC_END */
/* 818 */ NdrFcShort( 0x12, 0x0, /* FC_UP */
/* 820 */ 0x8, /* Offset= -26 (788) */
/* 822 */ 0x15, /* FC_STRUCT */
/* 824 */ 0x3, /* 3 */
/* 826 */ NdrFcShort( 0x8 ), /* 8 */
/* 828 */ 0x8, /* FC_LONG */
/* 830 */ 0x5c, /* FC_PAD */
/* 832 */ 0x5b, /* FC_END */
/* 834 */ NdrFcShort( 0x1b, /* FC_CARRAY */
/* 836 */ 0x3, /* 3 */
/* 838 */ 0x7, /* Corr desc: FC_USHORT */
/* 840 */ 0x0, /* 0 */
/* 842 */ NdrFcShort( 0xffc8 ), /* -56 */
/* 844 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 846 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 848 */ 0x0, /* 0 */
/* 850 */ NdrFcShort( 0xffffffffec ), /* Offset= -20 (816) */
/* 852 */ 0x5c, /* FC_PAD */
/* 854 */ 0x5b, /* FC_END */
/* 856 */ NdrFcShort( 0x1a, /* FC_BOGUS_STRUCT */
/* 858 */ 0x3, /* 3 */
/* 860 */ NdrFcShort( 0x38 ), /* 56 */
/* 862 */ NdrFcShort( 0xffffffffec ), /* Offset= -20 (824) */
/* 864 */ NdrFcShort( 0x0 ), /* Offset= 0 (846) */
/* 866 */ 0x6, /* FC_SHORT */
/* 868 */ 0x6, /* FC_ALIGNM4 */
/* 870 */ 0x8, /* FC_LONG */
/* 872 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 874 */ 0x4, /* 4 */
/* 876 */ NdrFcShort( 0xffffffffe0d ), /* Offset= -499 (356) */
/* 878 */ 0x5b, /* FC_END */
/* 880 */ NdrFcShort( 0x12, 0x0, /* FC_UP */
/* 882 */ 0x02, /* Offset= -254 (606) */

```

```

/* 862 */
/* 864 */ 0x1, /* 0x12, 0x8, /* FC_UP [simple_pointer] */
/* 866 */ 0x5c, /* FC_BYTE */
/* 868 */ 0x6, /* 0x12, 0x8, /* FC_UP [simple_pointer] */
/* 870 */ 0x5c, /* FC_SHORT */
/* 872 */ 0x8, /* 0x12, 0x8, /* FC_UP [simple_pointer] */
/* 874 */ 0x5c, /* FC_LONG */
/* 876 */ 0xa, /* 0x12, 0x8, /* FC_UP [simple_pointer] */
/* 878 */ 0x5c, /* FC_FLOAT */
/* 880 */ 0xc, /* 0x12, 0x8, /* FC_UP [simple_pointer] */
/* 882 */ 0x5c, /* FC_DOUBLE */
/* 884 */ NdrFcShort( 0x12, 0x0, /* FC_UP */
/* 886 */ 0xffffda4, /* Offset= -604 (280) */
/* 888 */ NdrFcShort( 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 890 */ 0xffffda6, /* Offset= -602 (286) */
/* 892 */ NdrFcShort( 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 894 */ 0xffffdbc, /* Offset= -580 (312) */
/* 896 */ NdrFcShort( 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 898 */ 0xffffdca, /* Offset= -566 (330) */
/* 900 */ NdrFcShort( 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 902 */ 0xffffdd8, /* Offset= -552 (348) */
/* 904 */ NdrFcShort( 0x2, /* FC_UP [pointer_deref] */
/* 906 */ 0x12, 0x10, /* Offset= 2 (906) */
/* 908 */ NdrFcShort( 0x16, /* FC_UP */
/* 910 */ 0x12, 0x0, /* Offset= 22 (930) */
/* 912 */ NdrFcShort( 0x15, /* FC_STRUCT */
/* 914 */ 0x7, /* 7 */
/* 916 */ 0x6, /* 0x1, /* FC_BYTE */
/* 918 */ 0x8, /* 0x1, /* FC_BYTE */
/* 920 */ 0xb, /* 0x38, /* FC_ALIGNM4 */
/* 922 */ 0x5b, /* 0x39, /* FC_ALIGNM8 */
/* 924 */ NdrFcShort( 0x12, 0x0, /* FC_UP */
/* 926 */ 0xfffffffff2, /* Offset= -14 (910) */
/* 928 */ 0x2, /* 0x12, 0x8, /* FC_UP [simple_pointer] */
/* 930 */ 0x5c, /* FC_CHAR */
/* 932 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 934 */ 0x7, /* 7 */

```

```

/* 932 */ NdrFcShort( 0x20 ), /* 32 */
/* 934 */ NdrFcShort( 0x0 ), /* 0 */
/* 936 */ NdrFcShort( 0x0 ), /* Offset= 0 (936) */
/* 938 */ 0x8, /* FC_LONG */
/* 940 */ 0x6, /* FC_SHORT */
/* 942 */ 0x6, /* FC_SHORT */
/* 944 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 946 */ NdrFcShort( 0xfffffc54 ), /* Offset= -940 (6) */
/* 948 */ 0x5c, /* FC_PAD */
/* 950 */ 0xb4, /* FC_USER_MARSHAL */
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x18 ), /* 24 */
/* 956 */ NdrFcShort( 0x0 ), /* 0 */
/* 958 */ NdrFcShort( 0xfffffc44 ), /* Offset= -956 (2) */
/* 960 */
/* 962 */ NdrFcShort( 0x6 ), /* Offset= 6 (968) */
/* 964 */
/* 966 */ NdrFcShort( 0xffffffdc ), /* Offset= -36 (930) */
/* 968 */ 0xb4, /* FC_USER_MARSHAL */
/* 970 */ NdrFcShort( 0x0 ), /* 0 */
/* 972 */ NdrFcShort( 0x18 ), /* 24 */
/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xfffffff4 ), /* Offset= -12 (964) */

0x0
    }
};

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_AXP64)*/

```

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. 4.20.000
 * Copyright Microsoft, 1999
 *
 * 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:
*       4.20.000 - 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 <sqlldb.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[] = "4.10.000";

const iMaxRetries = 10; // how many retries on
deadlock
static long iConnectionCount = 0; // number of current dblib connections

const int iErr0leDbProvider = 7312;
const char sErrTimeoutExpired[] = "Timeout expired";

BOOL APIENTRY 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 msgno
*               message number
*               int
*               msgstate message state
*               int
*               severity message severity
*               char
*               printable message description *msgtext
*
* RETURNS:      int INT_CONTINUE
*               continue if error is SQLETIME else INT_CANCEL action
*               INT_CANCEL cancel operation
*
* COMMENTS:     This function also sets the dead lock dbproc variable if
necessary.
*/

// typedef INT (SQLAPI *DBMSGHANDLE_PROC)(PDBPROCESS, 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 pDst 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 SERRMSG 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 (dbprocmshandle(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 (dbsqlxexec(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      // but the error handlers were not called.
        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 = 0;
    const BYTE   *pData;

    ResetError();

    while (TRUE)
    {
        try
        {
            dbrpcinit(m_dbproc, "tpcc_stocklevel", 0);

            dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_txn.StockLevel.w_id);    // @w_id smallint
            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) != SUCCEED)
                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 ||
            (e->m_msgno == iErrOleDbProvider &&
             strstr(e->m_msgtext, sErrTimeoutExpired) !=
             NULL)) &&
            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
    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, SQLINT2, -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;
                    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, SQLINT2, -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) != SUCCEEDED)
            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_generic, 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 (pData=dbdata(m_dbproc, 7))
                {
                    datetime = (*(DBDATETIME *) pData);
                    dbdatecrack(m_dbproc, &daterec, &datetime);
                    m_txn.NewOrder.o_entry_d.year =

                        m_txn.NewOrder.o_entry_d.month =

                        m_txn.NewOrder.o_entry_d.day =

                        m_txn.NewOrder.o_entry_d.hour =

                        m_txn.NewOrder.o_entry_d.minute =

                        m_txn.NewOrder.o_entry_d.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 ||
            (e->m_msgno == iErrOleDbProvider &&
            strstr(e->m_msgtext, sErrTimeoutExpired) !=
            NULL)) &&
            (++iTryCount <= iMaxRetries))
        {
            // hit deadlock; backoff for increasingly
            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;
    DBDATETIME          daterec;

    int                  iTryCount = 0;
    const BYTE          *pData;

    ResetError();

    while (TRUE)
    {
        try
        {
            dbrpcinit(m_dbproc, "tpcc_payment", 0);

            dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_txn.Payment.w_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -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) != SUCCEED)
    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,
(LPCBYTE)pData, dbdatlen(m_dbproc,24), SQLFLT8, (BYTE *)&m_txn.Payment.c_credit_lim,
8);
if (pData=dbdata(m_dbproc, 25))
    dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,25), SQLFLT8, (BYTE *)&m_txn.Payment.c_discount,
8);
if (pData=dbdata(m_dbproc, 26))
    dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)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;

```



```

        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 =
                m_txn.OrderStatus.o_entry_d.month =
                m_txn.OrderStatus.o_entry_d.day =
                m_txn.OrderStatus.o_entry_d.hour =
                m_txn.OrderStatus.o_entry_d.minute =
                m_txn.OrderStatus.o_entry_d.second =
                daterec.year;
            daterec.month;
            daterec.day;
            daterec.hour;
            daterec.minute;
            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 ||
(e->m_msgno == iErr0leDbProvider &&
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::Delivery()
    {
        int i;
        int iTryCount = 0;
        const BYTE *pData;

        ResetError();

        while (TRUE)
        {
            try
            {
                dbrpcinit(m_dbproc, "tpcc_delivery", 0);

                dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -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);
            }
            catch (...)
            {
                // 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, SQLINT2, -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);
            }
            catch (...)
            {
                // hit deadlock; backoff for increasingly
                // longer period
                delete e;
                Sleep(10 * iTryCount);
            }
            else
                throw;
        }
    }

```

```

        m_txn.Delivery.exec_status_code = eOK;
        return;
    }
    catch (CSQLERR *e)
    {
        if ((e->m_msgno == 1205 ||
            (e->m_msgno == 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::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: TPCC_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;};
    int ErrorNum() {return m_msgno;};
    char *ErrorText() {return m_msgtext;};
};

class CDBLIBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eLogin, // error from
        eDbOpen, // error from dbopen
        eDbUse, // error from
        eDbSqlExec, // error from
        eDbSet, // error from
        eDbNextRow, // error from
        eWrongRowCount, // more or less rows
        eWrongNumCols, // more or less columns
        eDbResults, // error from
        eDbRpcExec, // error from
    };
};

```

```

        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;};
    int ErrorNum() {return m_dberror;};
    char *ErrorText() {return m_dberrstr;};

};

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;};
        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
        {
            NEW_ORDER_DATA    NewOrder;
            PAYMENT_DATA        Payment;
            DELIVERY_DATA        Delivery;
            STOCK_LEVEL_DATA    StockLevel;
            ORDER_STATUS_DATA    OrderStatus;
            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_handler
        // 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
msgtxt );
};

```

```
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_odbc.cpp

```
/* FILE: TPC_C.ODBC.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: Implements ODBC calls for TPC-C txns.
 * Contact: Charles Levine (clevine@microsoft.com)
 *
 * Change history:
 * 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>
#include <odbcss.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_odbc.h"

// version string; must match return value from tpcc_version stored proc
const char sVersion[] = "4.10.000";

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(
LPCSTR szServer, // name of SQL server
LPCSTR szUser, // user name for login
LPCSTR szPassword, // password for login
LPCSTR szHost, // not used
LPCSTR szDatabase ) // name of database to use
{
```

```

        return new CTPCC_ODBC( szServer, szUser, szPassword, szHost, szDatabase );
    }

CTPCC_ODBC::CTPCC_ODBC (
    LPCSTR szServer,                // name of SQL server
    LPCSTR szUser,                  // user name
    for login
    LPCSTR szPassword,              // password for login
    LPCSTR szHost,                  // not used
    LPCSTR szDatabase               // name of database to
    use
    )
{
    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;

    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;

        sprintf( szConnectStr, "DRIVER=SQL
Server;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
                szServer, szUser, szPassword, szDatabase );

        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;

            // capture the (first) database error
            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_SSHORT,
SQL_SMALLINT, 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);
    }

void CTPCC_ODBC::StockLevel()
{
    RETCODE rc;
    int iTryCount = 0;

    m_hstmt = m_hstmtStockLevel;

    while (TRUE)
    {
        try
        {
            rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)"L"call
tpcc_stocklevel(?,?,?)", 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) || (++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::InitNewOrderParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmtNewOrder) !=
SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc, &m_descNewOrderCols1)
        != SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc, &m_descNewOrderCols2)
        != 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_SSHORT,
SQL_SMALLINT, 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_SSHORT, SQL_SMALLINT, 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);
    }

void CTPCC_ODBC::NewOrder()
{
    int RETCODE rc;
    int i;
    iTryCount = 0;

    // 0 1 2
    // 012345678901234567890123456789
    wchar_t szSqlTemplate[] = L"{call
tpcc_neworder(?,?,?,?,?,
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
    i = 29 + 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
            break;
        }
    }

    while (TRUE)
    {
        try
        {
            m_BindOffset = 0;
            rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)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
            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_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 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, 0, 0, &m_txn.Payment.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, 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);
}

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, (SQLWCHAR*)L"call
tpcc_payment(?,?,?,?,,?)", 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 (CODBCERR *e)
    {
        if ((!e->m_bDeadLock) || (++iTryCount > iMaxRetries))
            throw;

        // hit deadlock; backoff for increasingly longer
        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(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtOrderStatus;

    if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols1, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 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(CODBCERR::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(CODBCERR::eSetStmtAttr);

    i = 0;

```

```

        if ( SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&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(CODBCERR::eBindCol);

        if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols2, SQL_IS_POINTER ) != SQL_SUCCESS )
            ThrowError(CODBCERR::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.o_id, 0, NULL) != SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindCol);
    }

void CTPCC_ODBC::OrderStatus()
{
    int                                     rc;          iTryCount = 0;
    RETCODE

    m_hstmt = m_hstmtOrderStatus;

    if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols1, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::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(CODBCERR::eSetStmtAttr);

```

```

        rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)"call
tpcc_orderstatus(?,?,?,?)", 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) || (++iTryCount > iMaxRetries))
            throw;

        // hit deadlock; backoff for increasingly longer
        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_SSHORT,
SQL_SMALLINT, 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, (WORD)(i+1), SQL_C_SLONG,
&m_txn.Delivery.o_id[i], 0, NULL) != SQL_SUCCESS )
            ThrowError(CODBCERR::eBindCol);
    }
}

void CTPCC_ODBC::Delivery()
{
    RETCODE rc;
    int iTryCount = 0;

    m_hstmt = m_hstmtDelivery;

    while (TRUE)
    {
        try
        {
            rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)"L{call
tpcc_delivery(?,?)", 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) || (++iTryCount > iMaxRetries))
                throw;

            // hit deadlock; backoff for increasingly longer
            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

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
        eSetStmtAttr // error from
SQLSetStmtAttr
    };

    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;};
int ErrorNum() {return m_NativeError;};
char *ErrorText() {return m_odbcerrstr;};

};

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;};
    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_hstmtPayment;
SQLHSTMT     m_hstmtDelivery;
SQLHSTMT     m_hstmtOrderStatus;
SQLHSTMT     m_hstmtStockLevel;

SQLHDESC     m_descNewOrderCols1;
SQLHDESC     m_descNewOrderCols2;
SQLHDESC     m_descOrderStatusCols1;
SQLHDESC     m_descOrderStatusCols2;

// new-order specific fields
SQLINTEGER    m_BindOffset;
SQLINTEGER    m_RowsFetched;
int           m_no_commit_flag;

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;
}

public:
    CTPCC_ODBC(LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword,
LPCSTR szHost, LPCSTR szDatabase);
~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

```

```
( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR
szDatabase );
```

```
typedef CTPOCC_ODBC* (TYPE_CTPCC_ODBC)(LPCSTR, LPCSTR, LPCSTR, LPCSTR, LPCSTR);
```

trans.h

```
/* FILE: TRANS.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 structure templates.
 *
 * Change history:
 * 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 /* SQLSMALLINT */
year;
unsigned short /* SQLUSMALLINT */ month;
unsigned short /* SQLUSMALLINT */ day;
unsigned short /* SQLUSMALLINT */ hour;
unsigned short /* SQLUSMALLINT */ minute;
unsigned short /* SQLUSMALLINT */ 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
short 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
short w_id;
short d_id;
long c_id;
short 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
short w_id;
short d_id;
long c_id;
```

```

short                c_d_id;
short                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;
    short               ol_supply_w_id;
    short               ol_quantity;
    double              ol_amount;
    TIMESTAMP_STRUCT    ol_delivery_d;
} OL_ORDER_STATUS_DATA;

typedef struct
{
    // input params
    short               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
    short               w_id;
    short               o_carrier_id;

    // output params
    EXEC_STATUS          exec_status_code;
    SYSTEMTIME           queue_time;
    long                 o_id[10];          // id's of
} DELIVERED_ORDERS, *PDELIVERED_ORDERS;

//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   w_id;              //delivery warehouse
    short               o_carrier_id;       //carrier id
} DELIVERY_TRANSACTION;

typedef struct
{
    // input params
    short               w_id;
    short               d_id;
    short               threshold;

    // output params
    EXEC_STATUS          exec_status_code;
    long                 low_stock;
} STOCK_LEVEL_DATA, *PSTOCK_LEVEL_DATA;

```

txnlog.h

```

/*      FILE:                TXNLOG.H
 *
 *      Microsoft TPC-C Kit Ver. 4.10.000
 *      not yet audited
 *
 *      PURPOSE:  Header file for txn log class
 *      Copyright Microsoft, 1999
 *
 *      All Rights Reserved
 */

#pragma once

typedef struct _TXN_NEWORDER
{
    BYTE    OL_Count;          //range 0 to 31
    BYTE    OL_Remote_Count;   //range 0 to 31
    WORD    c_id;
    int     o_id;
} TXN_NEWORDER;

typedef struct _TXN_PAYMENT
{
    BYTE    CustByName;
} TXN_PAYMENT;

```



```

        BYTE        IsRemote;
    } TXN_PAYMENT;

typedef struct _TXN_ORDERSTATUS
{
    BYTE        CustByName;
} TXN_ORDERSTATUS;

typedef union _TXN_DETAILS
{
    TXN_NEWORDER    NewOrder;
    TXN_PAYMENT     Payment;
    TXN_ORDERSTATUS OrderStatus;
} TXN_DETAILS;

// Common header for all records in txn log. The TxnType field is
// a switch which identifies the particular variant.
#define TXN_REC_TYPE_CONTROL    1
#define TXN_REC_TYPE_TPCC      2    // replaces
TRANSACTION_TYPE_TPCC
#define TXN_REC_TYPE_TPCC_DELIV_DEF    3

typedef struct _TXN_RECORD_HEADER
{
    JULIAN_TIME    TxnStartT0;    // start of
    BYTE        TxnType;    // one of TXN_REC_TYPE_*
    BYTE        TxnSubType;    // depends on
    TxnType
} TXN_RECORD_HEADER, *PTXN_RECORD_HEADER;

typedef struct _TXN_RECORD_CONTROL
{
    // common header; must exactly match TXN_RECORD_HEADER
    JULIAN_TIME    TxnStartT0;    // start of
    BYTE        TxnType;    // =
    TXN_REC_TYPE_CONTROL
    BYTE        TxnSubType;    // depends on
    TxnType
    // end of common header
    DWORD        Len;    // number of
    bytes after this field
} TXN_RECORD_CONTROL, *PTXN_RECORD_CONTROL;

// TPC-C Txn Record Layout:
//
//TxnStartT0' is a Julian timestamp corresponding to the moment the
//txn is sent to the SUT, i.e., beginning of response time. Deltas
//are in milliseconds. Note that if RTDelay > 0, then the txn was
//delayed by this amount. The delay occurs at the beginning of the
//response time. So if RTDelay > 0, then the txn was actually sent
//at TxnStartT0 + RTDelay.
//
//Graphically:
//
// time -->
//
// |--- Menu ---|-- Keying --|-- Response --|--- Think --|

```

```

//      <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4 -> <- DeltaT3 ->
//
//      ^
//      ^ TxnStartT0
//
//RTDelay is the amount of response time delay included in DeltaT4.
//RTDelay is recorded per txn because this value can be changed on
//the fly, and so may vary from txn to txn.
//
//TxnStatus is the txn completion code. It is used to indicate errors.
//For example, in the New Order txn, 1% of txns abort. TxnStatus will
//reflect this.

typedef struct _TXN_RECORD_TPCC
{
    // common header; must exactly match TXN_RECORD_HEADER
    JULIAN_TIME    TxnStartT0;    // start of
    Txn
    BYTE        TxnType;    // = TXN_REC_TYPE_TPCC
    BYTE        TxnSubType;    // depends on
    TxnType
    // end of common header
    int        DeltaT1;    // menu time (ms)
    int        DeltaT2;    // keying time (ms)
    int        DeltaT3;    // think time (ms)
    int        DeltaT4;    // response time (ms)
    int        RTDelay;    // response time delay (ms)
    int        TxnError;    // error code providing
    more detail for TxnStatus
    int        w_id;    // warehouse
    ID
    BYTE        d_id;    // assigned district ID
    for this thread
    BYTE        d_id_ThisTxn;    // district ID chosen for this
    particular
    BYTE        TxnStatus;    // completion status for
    txn to indicate errors
    BYTE        reserved;    // for word alignment
    TXN_DETAILS    TxnDetails;    //
} TXN_RECORD_TPCC, *PTXN_RECORD_TPCC;

//
// TPC-C Deferred Delivery Txn Record Layout:
//
//Incorporating delivery transaction information into the above
//structure would increase the size of TXN_DETAILS from 8 to 42 bytes.
//Hence, we store delivery transaction details in a separate structure.
//
typedef struct _TXN_RECORD_TPCC_DELIV_DEF
{
    // common header; must exactly match TXN_RECORD_HEADER
    JULIAN_TIME    TxnStartT0;    // start of
    Txn
    BYTE        TxnType;    // =
    TXN_REC_TYPE_TPCC_DELIV_DEF
    BYTE        TxnSubType;    // = 0
    // end of common header
    int        DeltaT4;    // response time (ms)
    int        DeltaTxnExec;    // execution time (ms)
    int        w_id;    // warehouse
    ID
    BYTE        TxnStatus;    // completion status for
    txn to indicate errors

```

```

        BYTE    reserved;           // for word alignment
        short   o_carrier_id;       // carrier id
        long    o_id[10];           // returned delivery transaction
ids      } TXN_RECORD_TPCC_DELIV_DEF, *PTXN_RECORD_TPCC_DELIV_DEF;

#define TXN_LOG_VERSION 2
#define TXN_DATA_START 4096 // offset in log file
where log records start
#define TXN_LOG_EYE_CATCHER "BC" // signature bytes at the start of
log file

////////////////////////////////////
//
// The transaction log has a header as the first 4K block.
//
typedef struct _TXN_LOG_HEADER
{
    char            EyeCatcher[2]; // signature
    int             LogVersion;
    // set to TXN_LOG_VERSION
    int             BeginTxnTS;     //
    timestamp of first (lowest) txn start
    int             EndTxnTS;       // timestamp
    of last (highest) txn completion time
    int             iRecCount;
    // number of records in log file
    BOOL            bLogSorted;
    int             iFileSize;
    // file size in bytes

    // the record map provides a fast way to get close to a
    particular timestamp in a sorted log file.
    //
    struct
    {
        //
        // timestamp of record
        int             TS;
        //
        // byte position in file
        int             iPos;
        //
        //
        } RecMap[RecMapSize];
    } TXN_LOG_HEADER, *PTXN_LOG_HEADER;

    /* Header of the sorted pointers blocks in Temp file (in merging). */
    typedef struct BLOCK_HEADER {
        long            BlockPos;
        __int64         CurPos;
        DWORD           BytesRead;
        int             nRecords;
        BYTE            *offset; /* offset of pointers to records in the log
file */
    } BLOCK_HEADER, *PBLOCK_HEADER;

#define READ_BUFFER_SIZE 64*1024
#define WRITE_BUFFER_SIZE 8*1024

```

```

#define NUM_READ_BUFFERS 1
#define NUM_WRITE_BUFFERS 2
#define MAX_NUM_BUFFERS 2

// flags passed in to the constructor
#define TXN_LOG_WRITE 0x01
#define TXN_LOG_READ 0x02
#define TXN_LOG_SORTED 0x04
#define TXN_LOG_CRASHOPEN 0x08 // if set, invalid headers will be
tolerated; used for recovery

#define TXN_LOG_OS_ERROR 1
#define TXN_LOG_NOT_SORTED 2

#define SKIP_CTRL_RECS 1

class CTxnLog
{
private:
    DWORD            iBufferSize;
    //buffer allocated size
    DWORD            iBytesFreeInBuffer; //total bytes
available for use in buffer
    int             iNumBuffers;
    //buffers in use
    int             iActiveBuffer;
    //indicates which buffer is active: 0 or 1
    int             iIoBuffer;
    //buffer for any pending IO operation
    int             iFilePointer;
    //
    //position in file.
    LARGE_INTEGER    lFilePointer; //position in
file.
    int             iNextRec;
    //when reading, ordinal value of next record

    // A "save point" is remembered each time GetNextRecord is
    called with a start time specified.
    // The next time it is called, if start time is after the save
    point, we start scanning from the
    // save point. This is particularly useful in FindBestInterval,
    where the log is scanned repeatedly.
    int             SavePtTime;
    //
    int             iSavePtFilePointer;
    LARGE_INTEGER    lSavePtFilePointer;
    int             iSavePtNextRec;

    int             lastTS;
    //when writing sorted output, used to verify records are sorted
    BOOL            bWrite;
    //writing log file
    BOOL            bCrashOpen;
    // tolerate bad headers and consistency checks

    BOOL            bLogSorted;
    // is log file sorted? applies to both
    int             input and output
    int             BeginTxnTS;
    // timestamp of first (lowest) txn start
    int             EndTxnTS;
    timestamp of last (highest) txn completion time
    int             iRecCount;
    //
    // number of records in log file

```

```

        BYTE                *pCurrent;
//ptr to current buffer
        BYTE                *pBuffer[MAX_NUM_BUFFERS];

        PTXN_RECORD_HEADER  *TxnArray;                //transaction
record pointer array for sort

        DWORD              dwError;
        HANDLE             hTxnFile;
//handle to log file
        HANDLE             hMapFile;
//map file used when sorting the log
        HANDLE             hIoComplete;
//event to signify that there are no pending IOs
        HANDLE             hLogFileIo;
//event to signal the IO thread to write the inactive buffer

        Spinlock Spin;
//spin lock to protect the txn log file buffers

        FILE               *tmpFile;
//temp file for merging sorted pieces
        PBLOCK_HEADER       tmpHeaders;
//sorted pointers block header
        BYTE               **recPointers;
//record pointer buffers for each sorted block
        PTXN_RECORD_HEADER  *recBuffers;            //record buffers for
each sorted block
        int                *PointersRead;
//# of pointers processed in each block
        BOOL               *BlockAvailable;        //whether to
check a particular block for jmin

        int                nBlocks;
        int                jmin;
//index (block-wise) of the lowest timestamp record
        int                iAvgRecordLen;
//average record length

        int                iSortedReturnedCount;
//keeps track of the # of sorted records returned through
GetSortedRecord()

        int Write(BYTE *ptr, DWORD Size);
        static void LogFileIO(CTxnLog *);

        void LoadBuffers(int j);                //used in
sort/merge to load record buffers

    public:

        CTxnLog::CTxnLog(LPCTSTR szFileName, DWORD dwOpts);
        ~CTxnLog(void);

        int WriteToLog(PTXN_RECORD_TPCC pTxnRcrd);
        int WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF pTxnRcrd);
        int WriteToLog(PTXN_RECORD_CONTROL pCtrlRec);
        int WriteToLog(PTXN_RECORD_HEADER pCtrlRec);

        int WriteCtrlRecToLog(BYTE SubType, LPTSTR lpStr, DWORD dwLen);

        void CloseTransactionLogFile(void);

```

```

        PTXN_RECORD_HEADER GetNextRecord(BOOL bSkipCtrlRecs = FALSE);
        PTXN_RECORD_HEADER GetNextRecord(JULIAN_TIME SeekTimeT0, BOOL
bSkipCtrlRecs = FALSE);

        int Sort(void);
        PTXN_RECORD_HEADER GetSortedRecord();

        inline BOOL IsSorted(void) { return bLogSorted; };
        inline JULIAN_TIME BeginTS(void) { return BeginTxnTS; };
        inline JULIAN_TIME EndTS(void) { return EndTxnTS; };
        inline int RecordCount(void) { return iRecCount; };

};

class CTXNLOG_ERR : public CBaseErr
{
    public:
        enum CTXNLOG_ERRS
        {
            ERR_BAD_FILE_FORMAT,                // "File
format is invalid."
            ERR_UNKNOWN_LOG_VERSION,            // "Log file version is
unknown."
            ERR_BROKEN_LOG_FILE,                // "Log file
is broken."
            ERR_LOG_NOT_SORTED,                // "Log file
is not sorted"
            ERR_INVALID_TIME_SEQ,                // "Internal
Error: Record Time Sequence invalid."
        };

        CTXNLOG_ERR(int iErr) : CBaseErr(iErr) {};

        int ErrorType() {return ERR_TYPE_TXNLOG;};

        char *ErrorText()
        {
            static char *szMsgs[] = {
                "File format is invalid.",
                "Log file version is unknown.",
                "Log file is broken.",
                "Log file is not sorted",
                "Internal Error: Record Time Sequence
invalid.",
                ""
            };

            for(int i = 0; szMsgs[i][0]; i++)
            {
                if ( m_idMsg == i )
                    break;
            }

            return(szMsgs[i][0] ? szMsgs[i] : ERR_UNKNOWN);
        };
};

```

Appendix B:

Database Design

The TPC-C database was created with the following Transact-SQL scripts:

RunSQLCfg.sql

```
-- File:      RUNSQLCFG.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   This script file is used to set 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
```

VerifyTpccLoad.sql

```
-- File:      VERIFYTPCCLOAD.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Performs series of TPCC database checks to verify
--           that database load completed correctly

print      " "
select      convert(char(30), getdate(),9)
```

```
print      " "

use tpcc
go

--          *****
--          Check rows per table from SYSINDEXES
--          *****

print      'WAREHOUSE TABLE'

select      rows
from        sysindexes
where       id          = object_id("warehouse")
go

print      'DISTRICT TABLE = (10 * No of warehouses)'

select      rows
from        sysindexes
where       id          =object_id("district")
go

print      'ITEM TABLE = 100,000'

select      rows
from        sysindexes
where       id          =object_id("item")
go

print      'CUSTOMER TABLE = (30,000 * No of warehouses)'

select      rows
from        sysindexes
where       id          =object_id("customer")
go

print      'ORDERS TABLE = (30,000 * No of warehouses)'

select      rows
from        sysindexes
where       id          =object_id("orders")
go

print      'HISTORY TABLE = (30,000 * No of warehouses)'

select      rows
from        sysindexes
where       id          =object_id("history")
go

print      'STOCK TABLE = (100,000 * No of warehouses)'

select      rows
from        sysindexes
where       id          =object_id("stock")
go

print      'ORDER_LINE TABLE = (300,000 * No of warehouses + some change)'

select      rows
```

```

from sysindexes
where id =object_id("order_line")
go

print 'NEW_ORDER TABLE = (9000 * No of warehouses)'

select rows
from sysindexes
where id =object_id("new_order")
go

-- *****
--
-- Check indices
--
-- *****

print '*****Index Check*****'

use tpcc
go

sp_helpindex customer
go

sp_helpindex stock
go

sp_helpindex district
go

sp_helpindex item
go

sp_helpindex new_order
go

sp_helpindex orders
go

sp_helpindex order_line
go

sp_helpindex warehouse
go

```

backup.sql

```

-- File: BACKUP.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.41
-- Copyright Microsoft, 2001
-- Purpose: Creates backup of tpcc database

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

dump database tpcc to tpccback1, tpccback2, tpccback3 with init, stats = 1

select @enddate = getdate()

```

```

select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)
go

```

backupdev.sql

```

-- File: BACKUPDEVB.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.41
-- Copyright Microsoft, 2001
-- Purpose: Creates tpcc database Backup Devices

use master
go

-- create backup devices

exec sp_addumpdevice 'disk','tpccback1','X:\tpccback1.dmp'
go
exec sp_addumpdevice 'disk','tpccback2','Y:\tpccback2.dmp'
go
exec sp_addumpdevice 'disk','tpccback3','Z:\tpccback3.dmp'
go

```

config.sql

```

-- File: CONFIG.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.41
-- Copyright Microsoft, 2001
-- 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
sp_configure
go

```

createdb.sql

```

-- File: CREATEDB.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.41
-- Copyright Microsoft, 2001
-- Purpose: Creates tpcc database and backup files

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)
)

insert into tpcc_timer values (0,0)
go

-- Store starting time

update tpcc_timer
set start_date = (select convert(char(30), getdate(),9))
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_misc_fg
(
    NAME = MSSQL_misc1,
    FILENAME = "G:",
    SIZE = 34017MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc2,
    FILENAME = "I:",
    SIZE = 34017MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc3,
    FILENAME = "K:",
    SIZE = 34017MB,
    FILEGROWTH = 0),
FILEGROUP MSSQL_cs_fg
(
    NAME = MSSQL_cs1,
    FILENAME = "F:",
    SIZE = 70150MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs2,
    FILENAME = "H:",
    SIZE = 70150MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cs3,
    FILENAME = "J:",
    SIZE = 70150MB,
    FILEGROWTH = 0)
LOG ON
(
    NAME = MSSQL_tpcc_log,
    FILENAME = "E:",
    SIZE = 122496MB,
    FILEGROWTH = 0)
COLLATE Latin1_General_BIN
go

-- Store ending time

```

```

update tpcc_timer
set end_date = (select convert(char(30), getdate(),9))
go

select "Elapsed time (in seconds): ", datediff(second,(select start_date from
tpcc_timer),(select end_date from tpcc_timer))

-- 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. 4.41
--            Copyright Microsoft, 2001
-- Purpose:   Sets database options for data load

```

```

use master
go

exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
exec sp_dboption tpcc,'torn page detection',false
go

use tpcc
go

checkpoint
go

```

dbopt2.sql

```

-- File:      DBOPT2.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
-- Purpose:   Resets database options after data load

```

```

exec sp_dboption tpcc,'select into/bulkcopy',false
exec sp_dboption tpcc,'trunc. log on chkpt.',false
exec sp_dboption tpcc,'torn page detection',false
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

EXEC sp_tableoption 'district',      'pintable',true
EXEC sp_tableoption 'warehouse',      'pintable',true

```

```

EXEC sp_tableoption 'new_order',      'pintable',true
EXEC sp_tableoption 'item',          'pintable',true
GO

```

delivery.sql

```

-- File:      DELIVERY.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
-- Purpose:    Creates delivery transaction stored procedure
--
--            Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = 'tpcc_delivery' )
    drop procedure tpcc_delivery
go

create proc tpcc_delivery      @w_id                smallint,
                              @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

select @d_id = 0

begin tran 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 (serializable uplock)
        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 tran d

-- return delivery data to client

select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

```

go

getargs.c

```

// File: GETARGS.C
// Microsoft TPC-C Kit Ver. 4.41
// Copyright Microsoft, 1996, 1997, 1998, 1999,
// 2000, 2001
// 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 = LOG_PATH;
    pargs->pack_size = DEF_LD_PACKSIZE;
    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("TPCCLDR:\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("-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. 4.41
--            Copyright Microsoft, 2001
-- Purpose:    Creates clustered index on customer table

```

```

use tpcc
go

```

```

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

```

```

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,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

```

```
go
```

idxcusnc.sql

```

-- File:      IDXCUSNC.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
-- Purpose:    Creates non-clustered index on customer table

```

```

use tpcc
go

```

```

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

```

```

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,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

```

```
go
```

idxdiscl.sql

```

-- File:      IDXDISCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
-- Purpose:    Creates clustered index on district table

```

```

use tpcc
go

```

```

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

```

```

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,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxitmcl.sql

```

-- File:      IDXITMCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on item table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxnodcl.sql

```

-- File:      IDXNODCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on new_order table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxodlcl.sql

```

-- File:      IDXODLCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on order_line table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxordcl.sql

```

-- File:      IDXORDCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on orders table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordnc.sql

```
-- File:      IDXORDNC.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:    Creates non-clustered index on orders table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'orders_nc1' )
    drop index orders.orders_nc1

create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxstkcl.sql

```
-- File:      IDXSTKCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:    Creates clustered index on stock table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)
```

go

idxwarcl.sql

```
-- File:      IDXWARCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:    Creates clustered index on warehouse table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

neword.sql

```
-- File:      NEWORD.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:    Creates new order transaction stored procedure
--           Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = 'tpcc_neworder' )
    drop procedure tpcc_neworder

go

create proc tpcc_neworder

                                @w_id      smallint,
                                @d_id      tinyint,
                                @c_id      int,
                                @o_ol_cnt  tinyint,
                                @o_all_local tinyint,
                                @i_id1     int = 0, @s_w_id1

smallint = 0, @ol_qty1  smallint = 0,
                                @i_id2     int = 0, @s_w_id2

smallint = 0, @ol_qty2  smallint = 0,
                                @i_id3     int = 0, @s_w_id3

smallint = 0, @ol_qty3  smallint = 0,
```

```

smallint = 0, @ol_qty4 smallint = 0,
smallint = 0, @ol_qty5 smallint = 0,
smallint = 0, @ol_qty6 smallint = 0,
smallint = 0, @ol_qty7 smallint = 0,
smallint = 0, @ol_qty8 smallint = 0,
smallint = 0, @ol_qty9 smallint = 0,
smallint = 0, @ol_qty10 smallint = 0,
smallint = 0, @ol_qty11 smallint = 0,
smallint = 0, @ol_qty12 smallint = 0,
smallint = 0, @ol_qty13 smallint = 0,
smallint = 0, @ol_qty14 smallint = 0,
smallint = 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),
@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 smallint,
@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

```

```

@i_id4 int = 0, @s_w_id4
@i_id5 int = 0, @s_w_id5
@i_id6 int = 0, @s_w_id6
@i_id7 int = 0, @s_w_id7
@i_id8 int = 0, @s_w_id8
@i_id9 int = 0, @s_w_id9
@i_id10 int = 0, @s_w_id10
@i_id11 int = 0, @s_w_id11
@i_id12 int = 0, @s_w_id12
@i_id13 int = 0, @s_w_id13
@i_id14 int = 0, @s_w_id14
@i_id15 int = 0, @s_w_id15

```

```

where d_w_id = @w_id and
d_id = @d_id

-- process orderlines

while (@li_no < @o_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 @s_w_id10
when 11 then @s_w_id11
when 12 then @s_w_id12
when 13 then @s_w_id13
when 14 then @s_w_id14
when 15 then @s_w_id15
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 @ol_qty10
when 11 then @ol_qty11
when 12 then @ol_qty12
when 13 then @ol_qty13
when 14 then @ol_qty14

```

```

                                when 15 then @ol_qty15
                                end

-- get item data (no one updates item)

        select      @i_price = i_price,
                   @i_name  = i_name,
                   @i_data  = i_data
        from        item (tablock 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,
                                                @li_s_w_id,
                                                'dec 31, 1899',
                                                @li_qty,
                                                @i_price *
@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

                                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 (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,
                                    @o_entry_d,
                                    0,
                                    @o_ol_cnt,
                                    @o_all_local)

-- 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 (repeatableread)
        where       w_id        = @w_id

        if (@commit_flag = 1)
                commit transaction n
        else
                rollback transaction n

-- all that work for nuthn!!!

-- 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

```

null-txns.sql

```

-- File:      NULL-TXNS.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
--
-- Purpose:   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.
--
--           The purpose of these stored procs is to size and test the web client
without      the need of a fully scaled database.
--
--
drop proc tpcc_delivery
drop proc tpcc_neworder
drop proc tpcc_orderstatus
drop proc tpcc_payment
drop proc tpcc_stocklevel
drop proc tpcc_version
drop table order_line_null
go

create proc tpcc_delivery    @w_id          smallint,
                             @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

```

```

declare @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 proc tpcc_neworder

        @w_id          smallint,
        @d_id          tinyint,
        @c_id          int,
        @o_ol_cnt      tinyint,
        @o_all_local   tinyint,
        @i_id1 int = 0, @s_w_id1 smallint
= 0, @ol_qty1 smallint = 0,
        @i_id2 int = 0, @s_w_id2 smallint
= 0, @ol_qty2 smallint = 0,
        @i_id3 int = 0, @s_w_id3 smallint
= 0, @ol_qty3 smallint = 0,
        @i_id4 int = 0, @s_w_id4 smallint
= 0, @ol_qty4 smallint = 0,
        @i_id5 int = 0, @s_w_id5 smallint
= 0, @ol_qty5 smallint = 0,
        @i_id6 int = 0, @s_w_id6 smallint
= 0, @ol_qty6 smallint = 0,
        @i_id7 int = 0, @s_w_id7 smallint
= 0, @ol_qty7 smallint = 0,
        @i_id8 int = 0, @s_w_id8 smallint
= 0, @ol_qty8 smallint = 0,
        @i_id9 int = 0, @s_w_id9 smallint
= 0, @ol_qty9 smallint = 0,
        @i_id10 int = 0, @s_w_id10
smallint = 0, @ol_qty10 smallint = 0,
        @i_id11 int = 0, @s_w_id11
smallint = 0, @ol_qty11 smallint = 0,
        @i_id12 int = 0, @s_w_id12
smallint = 0, @ol_qty12 smallint = 0,
        @i_id13 int = 0, @s_w_id13
smallint = 0, @ol_qty13 smallint = 0,
        @i_id14 int = 0, @s_w_id14
smallint = 0, @ol_qty14 smallint = 0,
        @i_id15 int = 0, @s_w_id15
smallint = 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

```

```

declare @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 proc tpcc_orderstatus @w_id smallint,
tinyint,
int,
char(16) = ''
@d_id
@c_id
@c_last

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

declare @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 proc tpcc_payment @w_id          smallint,
                                @c_w_id
                                @h_amount
                                @d_id
                                @c_d_id
                                @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 numeric(12,2),
        @c_balance    numeric(12,2),
        @c_discount   numeric(4,4),
        @data         char(500),
        @c_data       char(500),
        @datetime     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   smallint,
        @c_id_local   int

declare @delaytime varchar(30)

-- uniform random delay of 0 - 0.3 second; avg = 0.15
select @delaytime = '00:00:0' + cast(cast((rand())*0.30) 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    = 'QpGd0Hjv8mR9vNI8V',
    @c_street_2    = 'dzKoC0bBqbC3yu',
    @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 proc tpcc_stocklevel    @w_id          smallint,
                               @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()*3.60) as decimal(4,3)) as
char(5))
        waitfor delay @delaytime

        select 49

GO

create proc tpcc_version
as
declare @version    char(8)

begin
    select @version = '4.10.000'
    select @version as 'Version'
end

GO

CREATE TABLE order_line_null (
    [ol_i_id] [int] NOT NULL ,
    [ol_supply_w_id] [smallint] 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. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Creates order status transaction stored procedure
--
--           Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = 'tpcc_orderstatus' )
    drop procedure    tpcc_orderstatus
go

create proc tpcc_orderstatus @w_id          smallint,
                             @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,
        @cnt            smallint

begin tran o

if (@c_id = 0)
    begin

        -- get customer id and info using last name

        select    @cnt      = (count(*)+1)/2
        from      customer (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 (repeatableread)
        where     c_last     = @c_last and
                  c_w_id     = @w_id and
                  c_d_id     = @d_id
    end

```

```

        order      by c_w_id, c_d_id, c_last, c_first
    end
    set      rowcount 0
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 (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 (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 (repeatableread)
        where      ol_o_id = @o_id and
                    ol_d_id = @d_id and
                    ol_w_id = @w_id

custnotfound:

commit tran 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. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Creates payment transaction stored procedure
--
--           Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = 'tpcc_payment' )
    drop procedure tpcc_payment
go

create proc tpcc_payment      @w_id      smallint,
                             @c_w_id    smallint,
                             @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_since         datetime,
         @c_credit        char(2),
         @c_credit_lim    numeric(12,2),
         @c_balance       numeric(12,2),
         @c_discount      numeric(4,4),
         @data            char(500),
         @c_data          char(500),

```

```

@datetime      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    smallint,
@c_id_local    int

select @screen_data = ''

begin tran 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 (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 (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      = 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,
        @data      = c_data,

```

```

        @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) +
                        substring(@data, 1, 458)

-- update customer info

update  customer
set     c_data      = @c_data
where   c_id        = @c_id and
        c_w_id      = @c_w_id and
        c_d_id      = @c_d_id

select  @screen_data = substring (@c_data,1,200)

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 tran 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

```

random.c

```

//      File:                RANDOM.C
//
//      Microsoft TPC-C Kit Ver. 4.41
//      Copyright Microsoft, 1996, 1997, 1998, 1999,
2000, 2001
//      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;
}

```

```

#ifdef 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. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Removes tpcc database and backup files
```

```
use master
go
```

```
-- remove any existing database and backup files
```

```
exec sp_dbremove tpcc, dropdev
go
```

```
exec sp_dropdevice 'tpccback1'
exec sp_dropdevice 'tpccback2'
exec sp_dropdevice 'tpccback3'
go
```

restore.sql

```
-- File:      RESTORE.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Loads database backup from backup files
```

```
declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

load database tpcc from tpccback1, tpccback2, tpccback3 with stats = 1

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

sqlshutdown.sql

```
-- File:      SQLSHUTDOWN.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   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. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Creates stock level transaction stored procedure
--
--           Interface Level: 4.10.000
```

```
use tpcc
go
```

```
if exists (select name from sysobjects where name = 'tpcc_stocklevel' )
drop procedure tpcc_stocklevel
go
```

```
create proc tpcc_stocklevel    @w_id          smallint,
                                @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
```

```
go
```

strings.c

```
// File:      STRINGS.C
//           Microsoft TPC-C Kit Ver. 4.41
//           Copyright Microsoft, 1996, 1997, 1998, 1999,
//           2000, 2001
// 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++)
    {
        cc = chArray[RandomNumber(0, chArrayMax)];
        str[i] = cc;
    }

    //if ( len < z )
    //    memset(str+len, ' ', z - len);
    str[len] = 0;
}

```



```

    return len;
}

//=====
//
// 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 + y) <= 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 strlen(str);
}

//=====
//
// 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. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Creates TPC-C tables

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                smallint,
    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),
    w_tax              numeric(4,4),
    w_ytd              numeric(12,2)
) on MSSQL_misc_fg
go

create table district
(
    d_id                tinyint,
    d_w_id              smallint,
    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),
    d_tax              numeric(4,4),
    d_ytd              numeric(12,2),
    d_next_o_id         int
) on MSSQL_misc_fg
go

create table customer
(
    c_id                int,
    c_d_id              tinyint,
    c_w_id              smallint,
    c_first              char(16),
    c_middle             char(2),
    c_last              char(16),
    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        numeric(12,2),
        c_discount          numeric(4,4),
        c_balance           numeric(12,2),
        c_ytd_payment       numeric(12,2),
        c_payment_cnt       smallint,
        c_delivery_cnt      smallint,
        c_data              char(500)
    ) on MSSQL_cs_fg
go

create table history
(
    h_c_id                int,
    h_c_d_id              tinyint,
    h_c_w_id              smallint,
    h_d_id                tinyint,
    h_w_id                smallint,
    h_date                datetime,
    h_amount              numeric(6,2),
    h_data                char(24)
) on MSSQL_misc_fg
go

create table new_order
(
    no_o_id                int,
    no_d_id                tinyint,
    no_w_id                smallint
) on MSSQL_misc_fg
go

create table orders
(
    o_id                  int,
    o_d_id                tinyint,
    o_w_id                smallint,
    o_c_id                int,
    o_entry_d             datetime,
    o_carrier_id          tinyint,
    o_ol_cnt              tinyint,
    o_all_local           tinyint
) on MSSQL_misc_fg
go

create table order_line
(
    ol_o_id                int,
    ol_d_id                tinyint,
    ol_w_id                smallint,
    ol_number              tinyint,
    ol_i_id                int,
    ol_supply_w_id         smallint,
    ol_delivery_d          datetime,
    ol_quantity            smallint,
    ol_amount              numeric(6,2),
    ol_dist_info           char(24)

```

```

    ) on MSSQL_misc_fg
go

create table item
(
    i_id                  int,
    i_im_id               int,
    i_name                char(24),
    i_price               numeric(5,2),
    i_data                char(50)
) on MSSQL_misc_fg
go

create table stock
(
    s_i_id                int,
    s_w_id                smallint,
    s_quantity            smallint,
    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),
    s_ytd                 int,
    s_order_cnt            smallint,
    s_remote_cnt          smallint,
    s_data                char(50)
) on MSSQL_cs_fg
go

```

time.c

```

//      File:                TIME.C
//
//      Microsoft TPC-C Kit Ver. 4.41
//      Copyright Microsoft, 1996, 1997, 1998, 1999,
//      2000, 2001
//      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: TPC.H
// Microsoft TPC-C Kit Ver. 4.41
// Copyright Microsoft, 1996, 1997, 1998, 1999,
// 2000, 2001
// Purpose: Header file for TPC-C database loader

// Build number of TPC Benchmark Kit
#define TPCKIT_VER "4.41"

// 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>

// ODBC headers
#include <sql.h>
#include <sqlext.h>
#include <odbcss.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 DEFLOADPACKSIZE 32768
#define LOADER_RES_FILE "C:\\MSTPCC.440\\SETUP\\logs\\load.out"
#define LOG_PATH "C:\\MSTPCC.440\\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;
    BOOL // set if loading all tables
    BOOL table_item;
    BOOL // 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;
} TPCCCLR_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 MakeOriginalAlphaString();
int MakeNumberString();
int MakeZipNumberString();
void InitString();
void InitAddress();
void PaddString();



---


tpccldr.c


---


// File: TPCCLDR.C
// Microsoft TPC-C Kit Ver. 4.41
// Copyright Microsoft, 1996, 1997, 1998, 1999,
2000, 2001
// 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

```

```

// Functions declarations

void HandleErrorDBC (SQLHDBC hdbc1);

void CheckDataBase();

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 OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures

typedef struct
{
    long ol;
    long ol_i_id;
    short 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;
    short 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;
    short 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;
// fix to avoid ODBC float to numeric conversion problem.
// double      c_balance;
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;

ORDERS_STRUCT orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];

```

```

long          orders_rows_loaded;
long          new_order_rows_loaded;
long          order_line_rows_loaded;
long          history_rows_loaded;
long          customer_rows_loaded;
long          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;

TPCCLDR_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*****");
    printf("\n*");
    printf("\n* Microsoft SQL Server");
    printf("\n*");
    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);

    // verify database and tables exist before attempting to load
    //CheckDataBase();

    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);

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]);
}

```

```

                                &dwThreadID[0]);
                                {
                                    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()
{
    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        bcphint[128];
    char        err_log_path[256];

```

```

// Seed with unique number
seed(1);

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");

//rc = bcp_init(i_hdbc1, name, NULL, "logs\\item.err", DB_IN);
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(bcphint, "tablock, order (i_id), ROWS_PER_BATCH =
100000");
    rc = bcp_control(i_hdbc1, BCPHINTS, (void*) bcphint);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);
}

rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
if (rc != SUCCEED)
    HandleErrorDBC(i_hdbc1);

rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 2);
if (rc != SUCCEED)
    HandleErrorDBC(i_hdbc1);

rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN, NULL, 0, 0, 3);
if (rc != SUCCEED)
    HandleErrorDBC(i_hdbc1);

rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 4);
if (rc != SUCCEED)
    HandleErrorDBC(i_hdbc1);

rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL, 0, 0, 5);
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);

    MakeAlphaString(14, 24, I_NAME_LEN, i_name);

```



```

        i_price = ((float) RandomNumber(100L, 10000L))/100.0;

        MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data, 10);

        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()
{
    short 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   bcphint[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("idxwarcl");

        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");

        //rc = bcp_init(w_hdbc1, name, NULL, "logs\\whouse.err", DB_IN);
        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 != SUCCEED)
            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 != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        }

        rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 1);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN, NULL, 0, 0, 2);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0, ADDRESS_LEN, NULL, 0, 0,
3);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0, ADDRESS_LEN, NULL, 0, 0,
4);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0, ADDRESS_LEN, NULL, 0, 0, 5);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0, STATE_LEN, NULL, 0, 0, 6);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN, NULL, 0, 0, 7);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 8);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 9);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

```

```

        time_start = (TimeNow() / MILLI);
        warehouse_rows_loaded = 0;
        for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses;
w_id++)
        {
            MakeAlphaString(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 != SUCCEED)
                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("idxwarcl");

        stock_rows_loaded = 0;
        district_rows_loaded = 0;

        District();
        Stock();
    }

//=====
//
// Function   : District
//
//=====

void District()
{
    short d_id;
    short 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;
    char   name[20];

```

```

    long d_next_o_id;
    long   time_start;
    int     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");

    //rc = bcp_init(w_hdbc1, name, NULL, "logs\\district.err", DB_IN);
    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 != SUCCEED)
        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 != SUCCEED)
            HandleErrorDBC(w_hdbc1);
    }

    rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0, D_NAME_LEN, NULL, 0, 0, 3);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1, 0, ADDRESS_LEN, NULL, 0, 0,
4);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2, 0, ADDRESS_LEN, NULL, 0, 0,
5);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0, ADDRESS_LEN, NULL, 0, 0, 6);
    if (rc != SUCCEED)

```

```

        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0, STATE_LEN, NULL, 0, 0, 7);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN, NULL, 0, 0, 8);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

SQLFLT8, 9);
rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0, SQL_VARLEN_DATA, NULL, 0,
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

SQLFLT8, 10);
rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

SQLINT4, 11);
rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0, SQL_VARLEN_DATA, NULL, 0,
if (rc != SUCCEED)
    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++)
    {
        MakeAlphaString(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 != SUCCEED)
            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()
{
    long s_i_id;
    short 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("idxstkcl");

    sprintf(name, "%s..%s", aptr->database, "stock");

    //rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err", DB_IN);
    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 != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, order (s_i_id, s_w_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 100000));
        rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
    }
}

```

```

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN, NULL, 0, 0, 4);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN, NULL, 0, 0, 5);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN, NULL, 0, 0, 6);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN, NULL, 0, 0, 7);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN, NULL, 0, 0, 8);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN, NULL, 0, 0, 9);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN, NULL, 0, 0, 10);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN, NULL, 0, 0, 11);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN, NULL, 0, 0, 12);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN, NULL, 0, 0, 13);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 14);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 15);

```

```

        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 16);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN, NULL, 0, 0, 17);
        if (rc != SUCCEED)
            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 = (short)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 != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

                stock_rows_loaded++;
                CheckForCommit(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;
    short                 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];
    // SQLRETURN            rc_1;
    // SQLSMALLINT           recnum, MsgLen;
    // SQLCHAR               SqlState[6],
    Msg[SQL_MAX_MESSAGE_LENGTH];
    // SQLINTEGER            NativeError;

    // 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("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");
    }

    // Initialize bulk copy
    sprintf(name, "%s..%s", aptr->database, "customer");

    //rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err", DB_IN);
    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 != SUCCEED)
        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 = (short)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 loading threads here...

            // 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\" >
logs\\nurand_load.log",
        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()
{
    int 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;

        // fix to avoid ODBC float to numeric conversion problem.
        // customer_buf[i].c_balance = 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, int 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);

        MakeAlphaString(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;

        // Generate CUSTOMER and HISTORY data

        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;

        // fix to avoid ODBC float to numeric conversion problem.

        // customer_buf[i].c_balance = -10.0;
        strcpy(customer_buf[i].c_balance,"-10.0");

        MakeAlphaString(300, 500, C_DATA_LEN, customer_buf[i].c_data);

        // Generate HISTORY data
        MakeAlphaString(12, 24, H_DATA_LEN, customer_buf[i].h_data);
    }

}

//=====
//
// Function : LoadCustomerTable
//
//=====

void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
    int i;

    long c_id;
    short c_d_id;
    short 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;

    // fix to avoid ODBC float to numeric conversion problem.
    // double      c_balance;
    char          c_balance[6];

double     c_ytd_payment;
short     c_payment_cnt;
short     c_delivery_cnt;
char      c_data[C_DATA_LEN+1];
char      c_since[C_SINCE_LEN+1];
RETCODE   rc;

rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0, 0, 4);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN, NULL, 0, 0, 5);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0, 6);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 7);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL, 0, 0, 8);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
if (rc != SUCCEED)

```

```

    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0, 12);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN, NULL, 0,
SQLCHARACTER, 13);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0, 14);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 15);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 16);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

    // fix to avoid ODBC float to numeric conversion problem.

    // rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 17);
    // if (rc != SUCCEED)
    //     HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0, SQLCHARACTER, 17);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 18);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 19);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 20);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, 500, NULL, 0, 0, 21);
if (rc != SUCCEED)
    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;

        // fix to avoid ODBC float to numeric conversion problem.

        // c_balance = customer_buf[i].c_balance;
        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 != SUCCEED)
            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)
{
    int i;
    long c_id;
    short c_d_id;
    short c_w_id;
    double h_amount;
    char h_data[H_DATA_LEN+1];
    char h_date[H_DATE_LEN+1];
    RETCODE rc;

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);

```

```

        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
3);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
4);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
5);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0, H_DATE_LEN, NULL, 0,
SQLCHARACTER, 6);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
7);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0, 8);
        if (rc != SUCCEED)
            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 != SUCCEED)
                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;
short    w_id;
        d_id;
        dwThreadID[MAX_ORDER_THREADS];
HANDLE    hThread[MAX_ORDER_THREADS];
char    name[20];
        rc;
        bcphint[128];
        err_log_path_ord[256];
        err_log_path_nord[256];
        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("idxordcl");
    BuildIndex("idxnodcl");
    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 != SUCCEED)
    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 != SUCCEED)
        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 != SUCCEED)
    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 != SUCCEED)
        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 != SUCCEED)
    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 != SUCCEED)
        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 = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses;
w_id++)
{
    for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
    {
        OrdersBufLoad(d_id, w_id);

        // start parallel loading threads here...

        // start Orders table thread

        printf("...Loading Order Table for: d_id = %d, w_id =
%d\n", d_id, w_id);

        hThread[0] = CreateThread(NULL,

            0,

            (LPTHREAD_START_ROUTINE) 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);
        hThread[1] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadNewOrderTable,
&new_order_time_start,
0,
&dwThreadID[1]);
        if (hThread[1] == NULL)
        {
            printf("Error, failed in creating creating
thread = 1.\n");
            exit(-1);
        }
        // start Order-Line table thread
        printf("...Loading Order-Line Table for: d_id = %d,
w_id = %d\n", d_id, w_id);
        hThread[2] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) 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());
        }
    }
    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(int d_id, int w_id)
{
    int    cust[ORDERS_PER_DISTRICT+1];
    long   o_id;
    short  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;
                // Added to insure ol_delivery_d set
properly during load

                FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_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_delivery_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;
    short  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
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
3);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
4);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0, O_ENTRY_D_LEN, NULL, 0,
SQLCHARACTER, 5);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 6);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
7);
}

```

```

        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc1);

        rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 8);
        if (rc != SUCCEED)
            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 != SUCCEED)
                HandleErrorDBC(o_hdbc1);

            orders_rows_loaded++;
            CheckForCommit(o_hdbc1, o_hstmt1, orders_rows_loaded, "orders",
&orders_time_start->time_start);
        }

        // rcint = bcp_batch(o_hdbc1);
        // if (rcint < 0)
        //     HandleErrorDBC(o_hdbc1);

        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("idxordcl");

            // build non-clustered index
            if (aptr->build_index == 1)
                BuildIndex("idxordnc");
        }
    }

//=====
//
// Function   : LoadNewOrderTable
//
//=====

void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)

```

```

{
    int      i;
    long     o_id;
    short    o_d_id;
    short    o_w_id;

    RETCODE  rc;
    DBINT    rcint;

    // Bind NEW-ORDER data

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
3);
    if (rc != SUCCEED)
        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 != SUCCEED)
            HandleErrorDBC(o_hdbc2);

        new_order_rows_loaded++;
        CheckForCommit(o_hdbc2, o_hstmt2, new_order_rows_loaded,
"new_order", &new_order_time_start->time_start);
    }

    // rcint = bcp_batch(o_hdbc2);
    // if (rcint < 0)
    //     HandleErrorDBC(o_hdbc2);

    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("idxnodcl");
    }
}

//=====

```

```

//
// Function   : LoadOrderLineTable
//
//=====
void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
    int         i,j;
    long         o_id;
    short        o_d_id;
    short        o_w_id;
    long         ol;
    long         ol_i_id;
    short        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
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
3);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 4);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
5);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 6);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0, OL_DELIVERY_D_LEN,
NULL, 0, SQLCHARACTER, 7);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 8);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 9);
    if (rc != SUCCEED)

```

```

        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL, 0, 0, 10);
    if (rc != SUCCEED)
        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].ol_dist_info);

            rc = bcp_sendrow(o_hdbc3);
            if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

            order_line_rows_loaded++;
            CheckForCommit(o_hdbc3, o_hstmt3,
order_line_rows_loaded, "order_line", &order_line_time_start->time_start);
        }

        // rcint = bcp_batch(o_hdbc3);
        // if (rcint < 0)
        //     HandleErrorDBC(o_hdbc3);

        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("idxodlcl");

        }

    }

//=====
//
// 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,
                   int rows_loaded,
                   char *table_name,
                   long *time_start)
{
    long time_end, time_diff;
    // DBINT rcint;

    if ( !(rows_loaded % aptr->batch) )
    {
        // rcint = bcp_batch(hdbc);
        // if (rcint < 0)
        //     HandleErrorDBC(hdbc);

        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   : 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 != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = SQLDriverConnect ( i_hdbc1,
                           NULL,
                           (SQLCHAR*)&szDriverString[0] ,
                           SQL_NTS,

```

```

(SQLCHAR*)&szDriverStringOut[0],
sizeof(szDriverStringOut),
                                &cbDriverStringOut,
                                SQL_DRIVER_NOPROMPT );

if (rc != SUCCEED)
    HandleErrorDBC(i_hdbc1);

// 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 != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = SQLDriverConnect ( w_hdbc1,
                                NULL,

                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,

                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,
                                SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

// 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 != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = SQLDriverConnect ( c_hdbc1,
                                NULL,

                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,

                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,

```

```

SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

// 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 != SUCCEED)
    HandleErrorDBC(c_hdbc2);

rc = SQLDriverConnect ( c_hdbc2,
                                NULL,

                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,

                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,
                                SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

// 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 != SUCCEED)
    HandleErrorDBC(o_hdbc1);

rc = SQLDriverConnect ( o_hdbc1,
                                NULL,

                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,

                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,
                                SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc1);

```



```

// 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 != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    rc = SQLDriverConnect ( o_hdbc2,

                                NULL,

                                (SQLCHAR*)&szDriverString[0] ,

                                SQL_NTS,

                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,

                                SQL_DRIVER_NOPROMPT );
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

// 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 != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = SQLDriverConnect ( o_hdbc3,

                                NULL,

                                (SQLCHAR*)&szDriverString[0] ,

                                SQL_NTS,

                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,

                                SQL_DRIVER_NOPROMPT );
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
}

//=====
//
// 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%s.log",
                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->index_script_path,
                                index_script,
                                aptr->log_path,
                                index_script);

    system(cmd);

    printf("Finished index creation:  %s\n",index_script);
}

void HandleErrorDBC (SQLHDBC hdbc1)
{
    SQLCHAR      SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLINTEGER    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_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" , datebuf, timebuf, szLastError);

        strcpy(err_log_path,aptr->log_path);
        strcat(err_log_path,"tpccldr.err");
        fp1 = fopen(err_log_path,"w");
        //fp1 = fopen("logs\\tpccldr.err","w");
        if (fp1 == NULL)
            printf("ERROR:  Unable to open errorlog file.\n");
        else
        {
            fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);

            fclose(fp1);
        }
    }
}

```

```

        i++;
    }
}

void HandleErrorSTMT (HSTMT hstmt1)
{
    SQLCHAR          SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLINTEGER 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 )
    {
        sprintf( szLastError , "%s" , Msg );

        _strtime(timebuf);
        _strdate(datebuf);

        printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

        strcpy(err_log_path,aptr->log_path);
        strcat(err_log_path,"tpccldr.err");
        fp1 = fopen(err_log_path,"w");
        //fp1 = fopen("logs\\tpccldr.err","w");
        if (fp1 == NULL)
            printf("ERROR: Unable to open errorlog file.\n");
        else
        {
            fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);
            fclose(fp1);
        }

        i++;
    }
}

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;
}

//=====
//
// Function : CheckDataBase
//
//=====

void CheckDataBase()
{
    RETCODE rc;

    char szDriverString[300];
    char szDriverStringOut[1024];
    char TablesBitMap[9] = {"0000000000"};
    int i, ExitFlag;

    SQLSMALLINT cbDriverStringOut;
    SQLCHAR TabName[10];
    SQLINTEGER TabNameInd, TabCount, TabCountInd;

    ExitFlag = 0;

    SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );

    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0 );

    SQLAllocHandle(SQL_HANDLE_DBC, henv , &v_hdbc);

    SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );

    // Open connection to SQL Server

    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->database );

    rc = SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE, (SQLPOINTER)aptr-
>pack_size, SQL_IS_INTEGER );
    if (rc != SQL_SUCCESS)
        HandleErrorDBC(v_hdbc);

    rc = SQLDriverConnect ( v_hdbc,
                                NULL,

                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,

                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),

                                &cbDriverStringOut,
                                SQL_DRIVER_NOPROMPT );
}

```

```

// if the rc is SQL_ERROR, the the TPCC database probably does not exist
if (rc == SQL_ERROR)
{
    printf("The database TPCC does not appear to exist!\n");
    printf("\nCheck LOGS\\ directory for database creation
errors.\n");

    // cleanup database connections and handles
    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
    SQLDisconnect(v_hdbc);
    SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

    // since there is not a database, exit back to SETUP.CMD
    exit(1);
}

if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt) != SQL_SUCCESS )
    HandleErrorDBC(v_hdbc);

if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0, &TabCountInd) !=
SQL_SUCCESS )
    HandleErrorSTMT(v_hstmt);

// count the number of user tables from sysobjects
rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects where xtype =
\'U\'", SQL_NTS);
if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
    HandleErrorSTMT(v_hstmt);

if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
    HandleErrorSTMT(v_hstmt);

// if the number of tables is less than 9, select all the user tables in
TPCC
if (TabCount != 9)
{
    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);

    SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt);

    if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR, &TabName,
sizeof(TabName), &TabNameInd) != SQL_SUCCESS )
        HandleErrorSTMT(v_hstmt);

    // select the list of user tables into a result set
    rc = SQLExecDirect(v_hstmt, "select * from sysobjects where
xtype = \'U\'", SQL_NTS);
    if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
        HandleErrorSTMT(v_hstmt);

    // go through the result set and set the bitmap for each found
    table
    // set the bitmap to '1' if the table name is found
    while ((rc = SQLFetch(v_hstmt)) != SQL_NO_DATA)
    {
        switch( TabName[0] )
        {
            case 'w':
                TablesBitMap[0] = '1';
                break;
            case 'd':
                TablesBitMap[1] = '1';

```

```

                break;
            case 'c':
                TablesBitMap[2] = '1';
                break;
            case 'h':
                TablesBitMap[3] = '1';
                break;
            case 'n':
                TablesBitMap[4] = '1';
                break;
            case 'o':
                if (TabName[5] == 's')
                    TablesBitMap[5] = '1';
                if (TabName[5] == '-')
                    TablesBitMap[6] = '1';
                break;
            case 'i':
                TablesBitMap[7] = '1';
                break;
            case 's':
                TablesBitMap[8] = '1';
                break;
        }
    }

    // a '0' ExitFlag means do NOT exit the loader early, a '1'
    means exit the loader early
    ExitFlag = 0;

    // iterate through the bitmap to display which table(s) is
    actually missing
    for (i = 0; i <= 8; i++)
    {
        switch(i)
        {
            case 0:
                if (TablesBitMap[i] == '0')
                {
                    printf("The Warehouse table is
missing or damaged.\n");
                    ExitFlag = 1;
                }
                break;
            case 1:
                if (TablesBitMap[i] == '0')
                {
                    printf("The District table is
missing or damaged.\n");
                    ExitFlag = 1;
                }
                break;
            case 2:
                if (TablesBitMap[i] == '0')
                {
                    printf("The Customer table is
missing or damaged.\n");
                    ExitFlag = 1;
                }
                break;
            case 3:
                if (TablesBitMap[i] == '0')
                {

```

```

missing or damaged.\n");
        printf("The History table is
        ExitFlag = 1;
    }
    break;
case 4:
    if (TablesBitMap[i] == '0')
    {
        printf("The New_Order table is
        ExitFlag = 1;
    }
    break;
case 5:
    if (TablesBitMap[i] == '0')
    {
        printf("The Orders table is
        ExitFlag = 1;
    }
    break;
case 6:
    if (TablesBitMap[i] == '0')
    {
        printf("The Order_Line table is
        ExitFlag = 1;
    }
    break;
case 7:
    if (TablesBitMap[i] == '0')
    {
        printf("The Item table is missing
        ExitFlag = 1;
    }
    break;
case 8:
    if (TablesBitMap[i] == '0')
    {
        printf("The Stock table is missing
        ExitFlag = 1;
    }
    break;
    }
}

// if one or more tables are missing, display message and exit
the loader
if (ExitFlag = 1)
{
    printf("\nExiting TPC-C Loader!\n");
    printf("\nCheck LOGS\\ directory for database\n");
    printf("or table creation errors.\n");

    // cleanup database connections and handles
    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
    SQLDisconnect(v_hdbc);
    SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

    exit(1);
}

```

```

    }

    // cleanup database connections and handles
    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
    SQLDisconnect(v_hdbc);
    SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

    return;
}

```

version.sql

```

-- File:      VERSION.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:   Returns version level of TPC-C stored procs
-- Note:      Always update the return value of this proc for
--           any interface changes or 'must have' bug fixes.
--
-- The value returned by this SP defines the 'interface level',
-- which must match between the stored procs and the client code.
-- The interface level may be down rev from the current kit. This
-- indicates that the interface hasn't changed since that version.

use tpcc
go

if exists ( select name from sysobjects where name = 'tpcc_version' )
    drop procedure tpcc_version
go

create proc tpcc_version
as
declare   @version char(8)

begin
    select @version = '4.10.000'
    select @version as 'Version'

end

go

```

Appendix C: Tunable Parameters

Microsoft SQL Server 2000 Startup Parameters

```
C:\Program Files\Microsoft SQL
Server\MSSQL\BINN\sqlservr.exe
-eC:\Program Files\Microsoft SQL
Server\MSSQL\LOG\ERRORLOG -x -c -t3502 -g100
```

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
- g100 Specify the amount of virtual address space in MB, SQL Server will leave available for memory allocations, excluding the buffer pool and threads stack, such as dynamically- loaded DLLs, extended procedure calls, etc. Incorrect use of this option can lead to conditions under which SQL Server may not start or may encounter runtime errors.

Boot.ini Parameters

```
[boot loader]
timeout=10
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 [PAE]" /fastdetect /pae
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows
Server 2003, Enterprise" /fastdetect
```

Microsoft SQL Server 2000 Configuration Parameters

```
1> 2> 3> 4> 5> 6> 7> 8> -- File: VERSION.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.41
-- Copyright Microsoft, 2001
-- Purpose: Extracts current version of SQL Server
```

```
use master
1> 2> 3>
SELECT CONVERT(char(20),
SERVERPROPERTY('ProductVersion'))
```

8.00.761

```
(1 row affected)
1> 2> 3>
SELECT CONVERT(char(20),
SERVERPROPERTY('ProductLevel'))
```

```
SP3
(1 row affected)
1> 2> 3>
SELECT CONVERT(char(30), getdate(),9)
```

May 28 2003 1:57:50:097AM

```
(1 row affected)
1> 2> 3> 4> 5>
```

```
1> 2> 3> 4> 5> 6> 7> 8> 9> 10>
-- File: CONFIG.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.41
-- Copyright Microsoft, 2001
-- Purpose: Collects SQL Server configuration
parameters
```

```
PRINT " "
SELECT convert(char(30), getdate(),9)
PRINT " "
```

May 28 2003 1:57:50:500AM

```
(1 row affected)
```

1> 2> 3> DBCC execution completed. If DBCC printed error messages, contact your system administrator. Configuration option 'show advanced options' changed from 1 to 1. Run the RECONFIGURE statement to install.

```
sp_configure "show advanced",1
1> 2> reconfigure with override
1> 2> sp_configure
```

name	config_value	run_value	minimum
------	--------------	-----------	---------

affinity mask			-2147483648
2147483647	15	15	
allow updates			0
1	0	0	
awe enabled			0
1	1	1	
c2 audit mode			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	
default full-text language			0
2147483647	1033	1033	
default language			0
9999	0	0	
fill factor (%)			0
100	0	0	
index create memory (KB)			704
2147483647	0	0	
lightweight pooling			0
1	1	1	
locks			5000
2147483647	0	0	
max degree of parallelism			0
32	1	1	
max server memory (MB)			4
2147483647	2147483647	2147483647	
max text repl size (B)			0
2147483647	65536	65536	
max worker threads			32
32767	340	340	
media retention			0
365	0	0	
min memory per query (KB)			512
2147483647	1024	1024	
min server memory (MB)			0
2147483647	0	0	
nested triggers			0
1	1	1	
network packet size (B)			512
65536	4096	4096	
open objects			0
2147483647	0	0	
priority boost			0
1	1	1	
query governor cost limit			0
2147483647	0	0	

```

query wait (s)                -1          -1          -1
2147483647                    -1          -1          0
recovery interval (min)       60          60          0
32767                          60          60          0
remote access                  1          1          0
1                              1          1          0
remote login timeout (s)      2147483647  20          0
2147483647                    20          20          0
remote proc trans              1          0          0
1                              0          0          0
remote query timeout (s)      2147483647  600         0
2147483647                    600         600         0
scan for startup procs        1          0          0
1                              0          0          0
set working set size          1          0          0
1                              0          0          0
show advanced options         1          1          0
1                              1          1          0
two digit year cutoff         9999        2049       2049       1753
user connections              32767        0          0          0
user options                   32767        0          0          0
32767                          0          0          0

```

1>

Benchcraft Profile

Profile: ap2_3520
File Path: C:\benchcraft\ap2_3520.pro
Version: 3

Number of Engines: 8

Name: cl74a
Description:
Directory: c:\blog\cl74a.log
Machine: N15
Parameter Set: 4.0
Index: 500000000
Seed: 18546
Configured Users: 4400
Pipe Name: DRIVER286005718
Connect Rate: 0
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: cl74b
Description:
Directory: c:\blog\cl74b.log
Machine: N15
Parameter Set: 4.0
Index: 1000000000
Seed: 18546
Configured Users: 4400

Pipe Name: DRIVER2149515765
Connect Rate: 0
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: cl74c
Description:
Directory: c:\blog\cl74c.log
Machine: N15
Parameter Set: 4.0
Index: 200000000
Seed: 18546
Configured Users: 4400
Pipe Name: DRIVER34355890
Connect Rate: 0
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: cl74d
Description:
Directory: c:\blog\cl74d.log
Machine: N15
Parameter Set: 4.0
Index: 300000000
Seed: 18546
Configured Users: 4400
Pipe Name: DRIVER44400187
Connect Rate: 0
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: cl74e
Description:
Directory: c:\blog\cl74e.log
Machine: N16
Parameter Set: 4.0
Index: 400000000
Seed: 18546
Configured Users: 4400
Pipe Name: DRIVER565600609
Connect Rate: 0
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: cl74f
Description:
Directory: c:\blog\cl74f.log
Machine: N16
Parameter Set: 4.0
Index: 500000000
Seed: 18546

Configured Users: 4400
Pipe Name: DRIVER665661734
Connect Rate: 0
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: cl74g
Description:
Directory: c:\blog\cl74g.log
Machine: N16
Parameter Set: 4.0
Index: 600000000
Seed: 18546
Configured Users: 4400
Pipe Name: DRIVER7172283812
Connect Rate: 0
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: cl74h
Description:
Directory: c:\blog\cl74h.log
Machine: N16
Parameter Set: 4.0
Index: 700000000
Seed: 18546
Configured Users: 4400
Pipe Name: DRIVER8172323828
Connect Rate: 0
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Number of User groups: 8

Driver Engine: cl74a
IIS Server: cr74
SQL Server: apark2
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1 - 440
w_id Min Warehouse: 1
w_id Max Warehouse: 3520
Scale: Normal
User Count: 4400
District id: 1
Scale Down: No

Driver Engine: cl74b
IIS Server: cr74
SQL Server: apark2
Database: tpcc
User: sa
Protocol: HTML

w_id Range: 441 - 880
w_id Min Warehouse: 1
w_id Max Warehouse: 3520
Scale: Normal
User Count: 4400
District id: 1
Scale Down: No

Driver Engine: cl74c
IIS Server: cr74
SQL Server: apark2
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 881 - 1320
w_id Min Warehouse: 1
w_id Max Warehouse: 3520
Scale: Normal
User Count: 4400
District id: 1
Scale Down: No

Driver Engine: cl74d
IIS Server: cr74
SQL Server: apark2
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1321 - 1760
w_id Min Warehouse: 1
w_id Max Warehouse: 3520
Scale: Normal
User Count: 4400
District id: 1
Scale Down: No

Driver Engine: cl74e
IIS Server: cr74
SQL Server: apark2
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1761 - 2200
w_id Min Warehouse: 1
w_id Max Warehouse: 3520
Scale: Normal
User Count: 4400
District id: 1
Scale Down: No

Driver Engine: cl74f
IIS Server: cr74
SQL Server: apark2
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2201 - 2640
w_id Min Warehouse: 1
w_id Max Warehouse: 3520
Scale: Normal
User Count: 4400
District id: 1
Scale Down: No

Driver Engine: cl74g
IIS Server: cr74
SQL Server: apark2
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2641 - 3080
w_id Min Warehouse: 1
w_id Max Warehouse: 3520
Scale: Normal
User Count: 4400
District id: 1
Scale Down: No

Driver Engine: cl74h
IIS Server: cr74
SQL Server: apark2
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3081 - 3520
w_id Min Warehouse: 1
w_id Max Warehouse: 3520
Scale: Normal
User Count: 4400
District id: 1
Scale Down: No

Number of Parameter Sets: 42

-Default
Default Parameter Set

Key	RT	RT	Menu	Txn	Think
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
12.05	18.01		New Order	44.75	
			0.10	5.00	0.10
12.05	3.01		Payment	43.10	
			0.10	5.00	0.10
5.05	2.01		Delivery	4.05	
			0.10	5.00	0.10
5.05	2.01		Stock Level	4.05	
			0.10	20.00	0.10

Order Status 4.05
10.05 2.01 0.10 5.00 0.10

No Think

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
0.00	0.00		New Order	10.00	
			0.00	5.00	0.00
0.00	0.00		Payment	10.00	
			0.00	5.00	0.00
0.00	0.00		Delivery	1.00	
			0.00	5.00	0.00
0.00	0.00		Stock Level	1.00	
			0.00	20.00	0.00
0.00	0.00		Order Status	1.00	
			0.00	5.00	0.00

95%

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
13.00	18.01		New Order	44.75	
			0.10	5.00	0.10
13.00	3.01		Payment	43.10	
			0.10	5.00	0.10
6.00	2.01		Delivery	4.05	
			0.10	5.00	0.10
6.00	2.01		Stock Level	4.05	
			0.10	20.00	0.10
11.00	2.01		Order Status	4.05	
			0.10	5.00	0.10

90%

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
16.00	18.01		New Order	44.75	
			0.10	5.00	0.10
16.00	3.01		Payment	43.10	
			0.10	5.00	0.10
9.00	2.01		Delivery	4.05	
			0.10	5.00	0.10
9.00	2.01		Stock Level	4.05	
			0.10	20.00	0.10
14.00	2.01		Order Status	4.05	
			0.10	5.00	0.10

1.6
1.6 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
19.28	18.01		New Order	44.75	
			0.10	5.00	0.10

19.28	3.01	Payment	0.10	5.00	43.10	0.10
8.08	2.01	Delivery	0.10	5.00	4.05	0.10
8.08	2.01	Stock Level	0.10	20.00	4.05	0.10
16.08	2.01	Order Status	0.10	5.00	4.05	0.10

2.0
2.0 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
24.10	18.01	New Order	0.10	5.00	44.88 0.10
24.10	3.01	Payment	0.10	5.00	43.03 0.10
10.10	2.01	Delivery	0.10	5.00	4.03 0.10
10.10	2.01	Stock Level	0.10	20.00	4.03 0.10
20.10	2.01	Order Status	0.10	5.00	4.03 0.10

2.6
2.6 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
31.33	18.01	New Order	0.10	5.00	44.75 0.10
31.33	3.01	Payment	0.10	5.00	43.10 0.10
13.13	2.01	Delivery	0.10	5.00	4.05 0.10
13.13	2.01	Stock Level	0.10	20.00	4.05 0.10
26.13	2.01	Order Status	0.10	5.00	4.05 0.10

3.0
3.0 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
36.15	18.01	New Order	0.10	5.00	44.75 0.10
36.15	3.01	Payment	0.10	5.00	43.10 0.10
15.15	2.01	Delivery	0.10	5.00	4.05 0.10
15.15	2.01	Stock Level	0.10	20.00	4.05 0.10
30.15	2.01	Order Status	0.10	5.00	4.05 0.10

4.0
4.0 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
48.20	18.01	New Order	0.10	5.00	44.75 0.10
48.20	3.01	Payment	0.10	5.00	43.10 0.10
20.20	2.01	Delivery	0.10	5.00	4.05 0.10
20.20	2.01	Stock Level	0.10	20.00	4.05 0.10
40.20	2.01	Order Status	0.10	5.00	4.05 0.10

3.8
3.8 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
45.80	18.01	New Order	0.10	5.00	44.75 0.10
45.80	3.01	Payment	0.10	5.00	43.10 0.10
19.20	2.01	Delivery	0.10	5.00	4.05 0.10
19.20	2.01	Stock Level	0.10	20.00	4.05 0.10
38.20	2.01	Order Status	0.10	5.00	4.05 0.10

3.6
3.6 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
43.38	18.01	New Order	0.10	5.00	44.75 0.10
43.38	3.01	Payment	0.10	5.00	43.10 0.10
18.18	2.01	Delivery	0.10	5.00	4.05 0.10
18.18	2.01	Stock Level	0.10	20.00	4.05 0.10
36.18	2.01	Order Status	0.10	5.00	4.05 0.10

3.4
3.4 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
40.97	18.01	New Order	0.10	5.00	44.75 0.10
40.97	3.01	Payment	0.10	5.00	43.10 0.10
17.17	2.01	Delivery	0.10	5.00	4.05 0.10
17.17	2.01	Stock Level	0.10	20.00	4.05 0.10

34.17	2.01	Order Status	0.10	5.00	4.05 0.10
		3.2			
		3.2 tt			

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
38.56	18.01	New Order	0.10	5.00	44.75 0.10
38.56	3.01	Payment	0.10	5.00	43.10 0.10
16.16	2.01	Delivery	0.10	5.00	4.05 0.10
16.16	2.01	Stock Level	0.10	20.00	4.05 0.10
32.16	2.01	Order Status	0.10	5.00	4.05 0.10

2.8
2.8 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
33.74	18.01	New Order	0.10	5.00	44.75 0.10
33.74	3.01	Payment	0.10	5.00	43.10 0.10
14.14	2.01	Delivery	0.10	5.00	4.05 0.10
14.14	2.01	Stock Level	0.10	20.00	4.05 0.10
28.14	2.01	Order Status	0.10	5.00	4.05 0.10

2.4
2.4 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
28.92	18.01	New Order	0.10	5.00	44.88 0.10
28.92	3.01	Payment	0.10	5.00	43.03 0.10
12.12	2.01	Delivery	0.10	5.00	4.03 0.10
12.12	2.01	Stock Level	0.10	20.00	4.03 0.10
24.12	2.01	Order Status	0.10	5.00	4.03 0.10

2.2
2.2 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
26.51	18.01	New Order	0.10	5.00	44.86 0.10

26.51	3.01	Payment	0.10	5.00	43.05	0.10
11.11	2.01	Delivery	0.10	5.00	4.03	0.10
11.11	2.01	Stock Level	0.10	20.00	4.03	0.10
22.11	2.01	Order Status	0.10	5.00	4.03	0.10

1.1
1.1 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
13.25	18.01	New Order	0.10	5.00	44.86
13.25	3.01	Payment	0.10	5.00	43.05
5.55	2.01	Delivery	0.10	5.00	4.03
5.55	2.01	Stock Level	0.10	20.00	4.03
11.05	2.01	Order Status	0.10	5.00	4.03

1.2
1.2 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
14.46	18.01	New Order	0.10	5.00	44.86
14.46	3.01	Payment	0.10	5.00	43.05
6.06	2.01	Delivery	0.10	5.00	4.03
6.06	2.01	Stock Level	0.10	20.00	4.03
12.06	2.01	Order Status	0.10	5.00	4.03

1.05
1.05tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.65	18.01	New Order	0.10	5.00	44.86
12.65	3.01	Payment	0.10	5.00	43.05
5.30	2.01	Delivery	0.10	5.00	4.03
5.30	2.01	Stock Level	0.10	20.00	4.03
10.55	2.01	Order Status	0.10	5.00	4.03

1.01
1.01tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.17	18.01	New Order	0.10	5.00	44.86
12.17	3.01	Payment	0.10	5.00	43.05
5.10	2.01	Delivery	0.10	5.00	4.03
5.10	2.01	Stock Level	0.10	20.00	4.03
10.15	2.01	Order Status	0.10	5.00	4.03

1.02
1.02tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.29	18.01	New Order	0.10	5.00	44.86
12.29	3.01	Payment	0.10	5.00	43.05
5.15	2.01	Delivery	0.10	5.00	4.03
5.15	2.01	Stock Level	0.10	20.00	4.03
10.25	2.01	Order Status	0.10	5.00	4.03

1.08
1.08 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
13.01	18.01	New Order	0.10	5.00	44.86
13.01	3.01	Payment	0.10	5.00	43.05
5.45	2.01	Delivery	0.10	5.00	4.03
5.45	2.01	Stock Level	0.10	20.00	4.03
10.85	2.01	Order Status	0.10	5.00	4.03

1.06
1.06tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.77	18.01	New Order	0.10	5.00	44.86
12.77	3.01	Payment	0.10	5.00	43.05
5.35	2.01	Delivery	0.10	5.00	4.03
5.35	2.01	Stock Level	0.10	20.00	4.03

10.65	2.01	Order Status	0.10	5.00	4.03	0.10
		1.07				
		1.07tt				

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.89	18.01	New Order	0.10	5.00	44.86
12.89	3.01	Payment	0.10	5.00	43.05
5.40	2.01	Delivery	0.10	5.00	4.03
5.40	2.01	Stock Level	0.10	20.00	4.03
10.75	2.01	Order Status	0.10	5.00	4.03

1.03
1.03tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.41	18.01	New Order	0.10	5.00	44.86
12.41	3.01	Payment	0.10	5.00	43.05
5.20	2.01	Delivery	0.10	5.00	4.03
5.20	2.01	Stock Level	0.10	20.00	4.03
10.35	2.01	Order Status	0.10	5.00	4.03

1.04
1.04tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.53	18.01	New Order	0.10	5.00	44.86
12.53	3.01	Payment	0.10	5.00	43.05
5.25	2.01	Delivery	0.10	5.00	4.03
5.25	2.01	Stock Level	0.10	20.00	4.03
10.45	2.01	Order Status	0.10	5.00	4.03

1.005
1.005

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.11	18.01	New Order	0.10	5.00	44.86

12.11	3.01	Payment	43.05	0.10
		0.10	5.00	4.03
5.08	2.01	Delivery	5.00	0.10
		0.10		
5.08	2.01	Stock Level	4.03	0.10
		0.10	20.00	4.03
10.10	2.01	Order Status	5.00	0.10
		0.10		

1.8tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
21.69	18.01	New Order	44.86	0.10	
		0.10	5.00	4.03	
21.69	3.01	Payment	43.05	0.10	
		0.10	5.00	4.03	
9.09	2.01	Delivery	5.00	0.10	
		0.10			
9.09	2.01	Stock Level	4.03	0.10	
		0.10	20.00	4.03	
18.09	2.01	Order Status	5.00	0.10	
		0.10			

1.9

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
22.90	18.01	New Order	44.86	0.10	
		0.10	5.00	4.03	
22.90	3.01	Payment	43.05	0.10	
		0.10	5.00	4.03	
9.60	2.01	Delivery	5.00	0.10	
		0.10			
9.60	2.01	Stock Level	4.03	0.10	
		0.10	20.00	4.03	
19.10	2.01	Order Status	5.00	0.10	
		0.10			

1.4

1.4 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
16.87	18.01	New Order	44.86	0.10	
		0.10	5.00	4.03	
16.87	3.01	Payment	43.05	0.10	
		0.10	5.00	4.03	
7.07	2.01	Delivery	5.00	0.10	
		0.10			
7.07	2.01	Stock Level	4.03	0.10	
		0.10	20.00	4.03	
14.07	2.01	Order Status	5.00	0.10	
		0.10			

1.5

1.5 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
18.07	18.01	New Order	44.86	0.10	
		0.10	5.00	4.03	
18.07	3.01	Payment	43.05	0.10	
		0.10	5.00	4.03	
7.58	2.01	Delivery	5.00	0.10	
		0.10			
7.58	2.01	Stock Level	4.03	0.10	
		0.10	20.00	4.03	
15.07	2.01	Order Status	5.00	0.10	
		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	44.86	0.10	
		0.10	5.00	4.03	
15.66	3.01	Payment	43.05	0.10	
		0.10	5.00	4.03	
6.57	2.01	Delivery	5.00	0.10	
		0.10			
6.57	2.01	Stock Level	4.03	0.10	
		0.10	20.00	4.03	
13.07	2.01	Order Status	5.00	0.10	
		0.10			

1.49

1.49 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
17.95	18.01	New Order	44.86	0.10	
		0.10	5.00	4.03	
17.95	3.01	Payment	43.05	0.10	
		0.10	5.00	4.03	
7.52	2.01	Delivery	5.00	0.10	
		0.10			
7.52	2.01	Stock Level	4.03	0.10	
		0.10	20.00	4.03	
14.97	2.01	Order Status	5.00	0.10	
		0.10			

1.48

1.48 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
17.83	18.01	New Order	44.86	0.10	
		0.10	5.00	4.03	
17.83	3.01	Payment	43.05	0.10	
		0.10	5.00	4.03	
7.47	2.01	Delivery	5.00	0.10	
		0.10			
7.47	2.01	Stock Level	4.03	0.10	
		0.10	20.00	4.03	

14.87	2.01	Order Status	4.03	0.10
		0.10	5.00	1.47
				1.47 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
17.71	18.01	New Order	44.86	0.10	
		0.10	5.00	4.03	
17.71	3.01	Payment	43.05	0.10	
		0.10	5.00	4.03	
7.42	2.01	Delivery	5.00	0.10	
		0.10			
7.42	2.01	Stock Level	4.03	0.10	
		0.10	20.00	4.03	
14.77	2.01	Order Status	5.00	0.10	
		0.10			

1.46

1.46 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
17.59	18.01	New Order	44.86	0.10	
		0.10	5.00	4.03	
17.59	3.01	Payment	43.05	0.10	
		0.10	5.00	4.03	
7.37	2.01	Delivery	5.00	0.10	
		0.10			
7.37	2.01	Stock Level	4.03	0.10	
		0.10	20.00	4.03	
14.67	2.01	Order Status	5.00	0.10	
		0.10			

1.45

1.45 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
17.47	18.01	New Order	44.86	0.10	
		0.10	5.00	4.03	
17.47	3.01	Payment	43.05	0.10	
		0.10	5.00	4.03	
7.32	2.01	Delivery	5.00	0.10	
		0.10			
7.32	2.01	Stock Level	4.03	0.10	
		0.10	20.00	4.03	
14.57	2.01	Order Status	5.00	0.10	
		0.10			

1.44

1.44 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
17.35	18.01	New Order	44.86	0.10	
		0.10	5.00		

			Payment	43.05	
17.35	3.01	0.10	5.00	0.10	
		Delivery	4.03		
7.27	2.01	0.10	5.00	0.10	
		Stock Level	4.03		
7.27	2.01	0.10	20.00	0.10	
		Order Status	4.03		
14.47	2.01	0.10	5.00	0.10	

1.43
1.43 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.86	
17.23	18.01	0.10	5.00	0.10	
		Payment	43.05		
17.23	3.01	0.10	5.00	0.10	
		Delivery	4.03		
7.22	2.01	0.10	5.00	0.10	
		Stock Level	4.03		
7.22	2.01	0.10	20.00	0.10	
		Order Status	4.03		
14.37	2.01	0.10	5.00	0.10	

1.42
1.42 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.86	
17.11	18.01	0.10	5.00	0.10	
		Payment	43.05		
17.11	3.01	0.10	5.00	0.10	
		Delivery	4.03		
7.17	2.01	0.10	5.00	0.10	
		Stock Level	4.03		
7.17	2.01	0.10	20.00	0.10	
		Order Status	4.03		
14.27	2.01	0.10	5.00	0.10	

1.41
1.41 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
			New Order	44.86	
16.99	18.01	0.10	5.00	0.10	
		Payment	43.05		
16.99	3.01	0.10	5.00	0.10	
		Delivery	4.03		
7.12	2.01	0.10	5.00	0.10	
		Stock Level	4.03		
7.12	2.01	0.10	20.00	0.10	
		Order Status	4.03		
14.17	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,00,44,00,41,00,50,00,53,0
0,56,00,43,00,00,00,00,00
"PoolThreadLimit"=dword:000003fe
"ThreadTimeout"=dword:00015180
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000842
"Last Help"=dword:00000843
"First Counter"=dword:00000802
"First Help"=dword:00000803
"Library Validation
Code"=hex:30,bb,ee,43,77,5b,c2,01,10,25,00,00,00,00,0
0,00
"WbemAdapFileTime"=hex:00,73,79,5b,bc,d4,c0,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
```

World Wide Web Service Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,0
0,4e,00,54,00,5c,00,53,00,\
```

```
79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00
,6e,00,65,00,74,00,73,\
```

```
00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,00,69,00,6e
,00,66,00,6f,00,2e,00,\
65,00,78,00,65,00,00,00
```

```
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):49,00,49,00,53,00,41,00,44,0
0,4d,00,49,00,4e,00,00,00,\
00,00
```

```
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity and
administration through the Internet Information
Services snap-in."
"FailureActions"=hex:ff,ff,ff,ff,00,00,00,00,00,00,00
,00,03,00,00,00,50,a8,0d,\
```

```
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP]
"NOTE"="This is for backward compatibility only."
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP\Parameters]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\WINNT\System32\inetrv"
"CertMapList"="C:\WINNT\System32\inetrv\iisrmap
.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\WINNT\System32\LogFiles"
"AcceptExOutstanding"=dword:00000028
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots]
"/"="c:\inetpub\wwwroot,,207"
"/Scripts"="c:\inetpub\scripts,,1"
"/IISHelp"="c:\winnt\help\iishelp,,1"
"/IISAdmin"="C:\WINNT\System32\inetrv\iisadmin,,
1"
"/IISamples"="c:\inetpub\iisamples,,1"
"/MSADC"="c:\program files\common
files\system\msadc,,1"
```

```
"/vti_bin"="C:\\Program Files\\Common
Files\\Microsoft Shared\\Web Server
Extensions\\40\\isapi,1"
"/Printers"="C:\\WINNT\\web\\printers,,201"
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\W3SVC\\Performance]
"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:000008e6
"Last Help"=dword:000008e7
"First Counter"=dword:00000844
"First Help"=dword:00000845
"Library Validation
Code"=hex:de,61,7e,46,77,5b,c2,01,10,3d,00,00,00,00,0
0,00
"WbemAdapFileTime"=hex:00,73,79,5b,bc,d4,c0,01
"WbemAdapFileSize"=dword:00001d10
"WbemAdapStatus"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\W3SVC\\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14
,00,00,00,30,00,00,00,02,\\
```

```
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\\
```

```
00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00
,01,01,00,00,00,00,00,\\
```

```
05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01
,02,00,00,00,00,00,05,\\
```

```
20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01
,02,00,01,01,00,00,00,\\
```

```
00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02
,00,01,02,00,00,00,00,\\
```

```
00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00
,00,00,00,05,12,00,00,\\
00,01,01,00,00,00,00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\W3SVC\\Enum]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

TPCC Application Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\TPCC]
"Path"="C:\\Inetpub\\wwwroot\\"
"NumberOfDeliveryThreads"=dword:00000040
"MaxConnections"=dword:00009c40
"MaxPendingDeliveries"=dword:00000bb8
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="apark2"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"
```

Server Bus Performance Driver Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\hpcqissb]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:00000102
"ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,0
0,33,00,32,00,5c,00,44,00,\\
```

```
52,00,49,00,56,00,45,00,52,00,53,00,5c,00,68,00,70,00
,71,00,63,00,69,00,73,\\
00,73,00,62,00,2e,00,73,00,79,00,73,00,00,00
"DisplayName"="Smart Array Controllers Non-Miniport
Bus Driver"
"Group"="port"
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\hpcqissb\\Parameters]
"CompletionMode"=dword:00000002
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\hpcqissb\\Security]
"Security"=hex:01,00,14,80,90,00,00,00,9c,00,00,00,14
,00,00,00,30,00,00,00,02,\\
```

```
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\\
```

```
00,00,02,00,60,00,04,00,00,00,00,00,14,00,fd,01,02,00
,01,01,00,00,00,00,00,\\
```

```
05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00
,00,00,05,20,00,00,00,\\
```

```
20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00
,00,05,0b,00,00,00,00,\\
```

```
00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00
,00,23,02,00,00,01,01,\\
```

```
00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05
,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\hpcqissb\\Enum]
"0"="Root\\LEGACY_HPQCISB\\0000"
"Count"=dword:00000004
"NextInstance"=dword:00000004
"1"="PCI\\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\\3
&172e68dd&0&08"
"2"="PCI\\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\\3
&172e68dd&0&10"
"3"="PCI\\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\\3
&29e81982&0&08"
```

Server Disk Device Performance Driver Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\hpcqissd]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:00000102
"ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,0
0,33,00,32,00,5c,00,44,00,\\
```

```
52,00,49,00,56,00,45,00,52,00,53,00,5c,00,68,00,70,00
,71,00,63,00,69,00,73,\\
00,73,00,64,00,2e,00,73,00,79,00,73,00,00,00
"DisplayName"="Smart Array Controllers Non-Miniport
Disk Driver"
"Group"="Primary Disk"
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\hpcqissd\\Security]
"Security"=hex:01,00,14,80,90,00,00,00,9c,00,00,00,14
,00,00,00,30,00,00,00,02,\\
```

```
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\\
```

00,00,02,00,60,00,04,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,\

05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\

20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,0b,00,00,00,00,\

00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,01,01,\

00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpcissd\Enum]
"0"="Root\\LEGACY_HPQCISSD\\0000"
"Count"=dword:0000000a
"NextInstance"=dword:0000000a
"1"="HPQCIS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&1f72f2bd&0&0000000400000000"
"2"="HPQCIS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&1f72f2bd&0&0100004000000000"
"3"="HPQCIS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&1f72f2bd&0&0200004000000000"
"4"="HPQCIS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2e12b67&0&0000004000000000"
"5"="HPQCIS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2e12b67&0&0100004000000000"
"6"="HPQCIS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2e12b67&0&0200004000000000"
"7"="HPQCIS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&38eb4840&0&0000004000000000"
"8"="HPQCIS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&38eb4840&0&0100004000000000"
"9"="HPQCIS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&38eb4840&0&0200004000000000"

System Summary

System Information report written at: 05/25/03 13:30:58
System Name: APARK2
[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003, Enterprise Edition
Version	5.2.3790 Build 3790
OS Manufacturer	Microsoft Corporation
System Name	APARK2
System Manufacturer	HP
System Model	ProLiant DL380 G3
System Type	X86-based PC
Processor	x86 Family 15 Model 2 Stepping 7
GenuineIntel	~3054 Mhz

Processor x86 Family 15 Model 2 Stepping 7
GenuineIntel ~3054 Mhz
Processor x86 Family 15 Model 2 Stepping 7
GenuineIntel ~3054 Mhz
Processor x86 Family 15 Model 2 Stepping 7
GenuineIntel ~3054 Mhz
BIOS Version/Date HP P29, 3/25/2003
SMBIOS Version 2.3
Windows Directory C:\WINDOWS
System Directory C:\WINDOWS\system32
Boot Device \Device\HarddiskVolume1
Locale United States
Hardware Abstraction Layer Version = "5.2.3790.0 (srv03_rtm.030324-2048)"
User Name APARK2\Administrator
Time Zone Central Daylight Time
Total Physical Memory 12,288.00 MB
Available Physical Memory 111.67 MB
Total Virtual Memory 25.28 GB
Available Virtual Memory 2.15 GB
Page File Space 13.53 GB
Page File C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource	Device	
Memory Address 0xF7900000-0xF7AFFFFF	PCI bus	
Memory Address 0xF7900000-0xF7AFFFFF	Smart Array	
5300 Controller		
I/O Port 0x00000000-0x00000CFF	PCI bus	
I/O Port 0x00000000-0x00000CFF	PCI bus	
I/O Port 0x00000000-0x00000CFF	Direct memory access controller	
I/O Port 0x000003C0-0x000003DF	PCI bus	
I/O Port 0x000003C0-0x000003DF	RAGE XL PCI	
Family (Microsoft Corporation)		
I/O Port 0x00003000-0x000030FF	PCI bus	
I/O Port 0x00003000-0x000030FF	Smart Array	
51		
I/O Port 0x00005000-0x000054FF	PCI bus	
I/O Port 0x00005000-0x000054FF	Smart Array	
5300 Controller		
Memory Address 0xA0000-0xBFFFF	PCI bus	
Memory Address 0xA0000-0xBFFFF	RAGE XL PCI	
Family (Microsoft Corporation)		
Memory Address 0xF5F00000-0xF6FFFFFF	PCI bus	
Memory Address 0xF5F00000-0xF6FFFFFF	HP iLO	
Management Interface Driver		
I/O Port 0x000003B0-0x000003BB	PCI bus	
I/O Port 0x000003B0-0x000003BB	RAGE XL PCI	
Family (Microsoft Corporation)		

I/O Port 0x00004000-0x000040FF	PCI bus
I/O Port 0x00004000-0x000040FF	Smart Array
5300 Controller	

[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	
0x00000000-0x00000CFF	PCI bus	OK	
0x00000000-0x00000CFF	PCI bus	OK	
0x00000000-0x00000CFF	Direct memory access controller	OK	
0x000003B0-0x000003BB	PCI bus	OK	
0x000003B0-0x000003BB	RAGE XL PCI Family	OK	
(Microsoft Corporation)			
0x000003C0-0x000003DF	PCI bus	OK	
0x000003C0-0x000003DF	RAGE XL PCI Family	OK	
(Microsoft Corporation)			
0x00002400-0x000024FF	RAGE XL PCI Family	OK	
(Microsoft Corporation)			
0x00001800-0x000018FF	HP ProLiant iLO	OK	
Advanced System Management Controller	OK		
0x00002800-0x000028FF	HP iLO Management		
Interface Driver	OK		
0x00000A79-0x00000A79	ISAPNP Read Data Port	OK	
0x00000279-0x00000279	ISAPNP Read Data Port	OK	
0x00000274-0x00000277	ISAPNP Read Data Port	OK	
0x00000F50-0x00000F58	Motherboard resources	OK	
0x00000408-0x0000040F	Motherboard resources	OK	
0x00000092-0x00000092	Motherboard resources	OK	
0x00000900-0x00000903	Motherboard resources	OK	
0x00000910-0x00000911	Motherboard resources	OK	
0x00000920-0x00000923	Motherboard resources	OK	
0x00000930-0x00000937	Motherboard resources	OK	
0x00000940-0x00000947	Motherboard resources	OK	
0x00000950-0x00000957	Motherboard resources	OK	
0x00000C06-0x00000C08	Motherboard resources	OK	

0x00000C14-0x00000C14 OK Motherboard resources
 0x00000C49-0x00000C4A OK Motherboard resources
 0x00000C50-0x00000C52 OK Motherboard resources
 0x00000C6C-0x00000C6F OK Motherboard resources
 0x00000010-0x0000001F OK Motherboard resources
 0x00000230-0x00000233 OK Motherboard resources
 0x00000260-0x00000267 OK Motherboard resources
 0x000004D0-0x000004D1 OK Motherboard resources
 0x00000700-0x0000070F OK Motherboard resources
 0x00000800-0x0000081F OK Motherboard resources
 0x00000C80-0x00000C83 OK Motherboard resources
 0x00000CD4-0x00000CD7 OK Motherboard resources
 0x00000CF9-0x00000CF9 OK Motherboard resources
 0x00000020-0x00000021 OK Programmable interrupt controller
 0x000000A0-0x000000A1 OK Programmable interrupt controller
 0x00000C00-0x00000C01 OK Programmable interrupt controller
 0x00000040-0x00000043 OK System timer
 0x00000080-0x0000008F OK Direct memory access controller
 0x000000C0-0x000000DF OK Direct memory access controller
 0x0000040B-0x0000040B OK Direct memory access controller
 0x000004D6-0x000004D6 OK Direct memory access controller
 0x00000061-0x00000061 OK System speaker
 0x00000060-0x00000060 Standard 101/102-Key or Microsoft Natural PS/2 Keyboard OK
 0x00000064-0x00000064 Standard 101/102-Key or Microsoft Natural PS/2 Keyboard OK
 0x0000002E-0x0000002F OK Extended IO Bus
 0x00000220-0x00000223 OK Extended IO Bus
 0x00000240-0x0000025F OK Extended IO Bus
 0x00000070-0x00000073 OK Extended IO Bus
 0x000003F8-0x000003FF OK Communications Port (COM1)
 0x000003F2-0x000003F5 OK Standard floppy disk controller
 0x000003F7-0x000003F7 OK Standard floppy disk controller

0x00002000-0x0000200F CSB5 IDE Controller OK
 0x000001F0-0x000001F7 Primary IDE Channel OK
 0x000003F6-0x000003F6 Primary IDE Channel OK
 0x00000170-0x00000177 OK Secondary IDE Channel
 0x00000376-0x00000376 OK Secondary IDE Channel
 0x00003000-0x000030FF OK PCI bus
 0x00003000-0x000030FF OK Smart Array 5i
 0x00004000-0x000040FF OK PCI bus
 0x00004000-0x000040FF OK Smart Array 5300 Controller
 0x00005000-0x000054FF OK PCI bus
 0x00005000-0x000054FF OK Smart Array 5300 Controller
 0x00005400-0x000054FF OK Smart Array 5300 Controller

[IRQs]

Resource	Device	Status
IRQ 9	Microsoft ACPI-Compliant System	OK
IRQ 16	HP ProLiant iLO Advanced System Management Controller	OK
IRQ 17	HP iLO Management Interface Driver	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
IRQ 14	Primary IDE Channel	OK
IRQ 30	Smart Array 5i	OK
IRQ 29	Compaq NC7781 Gigabit Server Adapter #3	OK
IRQ 20	Smart Array 5300 Controller	OK
IRQ 24	Smart Array 5300 Controller	OK
IRQ 26	Smart Array 5300 Controller	OK
IRQ 18	hp ProLiant PCI Hot Plug Controller (CPQPHP installed)	OK

[Memory]

Resource	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	RAGE XL PCI Family (Microsoft Corporation)	OK
0xF5F00000-0xF6FFFFFF	PCI bus	OK
0xF5F00000-0xF6FFFFFF	HP iLO Management Interface Driver	OK
0xF6000000-0xF6FFFFFF	RAGE XL PCI Family (Microsoft Corporation)	OK
0xF5FF0000-0xF5FF0FFF	RAGE XL PCI Family (Microsoft Corporation)	OK
0xF5FE0000-0xF5FE01FF	HP ProLiant iLO Advanced System Management Controller	OK

0xF5FD0000-0xF5FD07FF HP iLO Management Interface Driver OK
 0xF5FC0000-0xF5FC1FFF HP iLO Management Interface Driver OK
 0xF7600000-0xF77FFFFF PCI bus OK
 0xF77C0000-0xF77FFFFF Smart Array 5i OK
 0xF76F0000-0xF76F3FFF Smart Array 5i OK
 0xF7800000-0xF78FFFFF PCI bus OK
 0xF78F0000-0xF78FFFFF Compaq NC7781 Gigabit Server Adapter #3 OK
 0xF7900000-0xF7AFFFFFFF PCI bus OK
 0xF7900000-0xF7AFFFFFFF Smart Array 5300 Controller OK
 0xF7AC0000-0xF7AFFFFFFF Smart Array 5300 Controller OK
 0xF7B00000-0xF7FFFFFFF PCI bus OK
 0xF7FC0000-0xF7FFFFFFF Smart Array 5300 Controller OK
 0xF7E00000-0xF7EFFFFFFF Smart Array 5300 Controller OK
 0xF7DC0000-0xF7DFFFFFFF Smart Array 5300 Controller OK
 0xF7C00000-0xF7CFFFFFFF Smart Array 5300 Controller OK
 0xF7BF0000-0xF7BF0FFF hp ProLiant PCI Hot Plug Controller (CPQPHP installed) OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size
c:\windows\system32\tssoft32.acm	INC.	DSP GROUP, INC.	OK	C:\WINDOWS\system32\TSSOFT32.ACM	1.01	9.50 KB (9,728 bytes)
c:\windows\system32\msg723.acm	Corporation	Microsoft Corporation	OK	C:\WINDOWS\system32\MSG723.ACM	4.4.0000	116.00 KB (118,784 bytes)
c:\windows\system32\msg711.acm	Corporation	Microsoft Corporation	OK	C:\WINDOWS\system32\MSG711.ACM	5.2.3790.0	(srv03_rtm.030324-2048) 10.00 KB (10,240 bytes)
c:\windows\system32\msadp32.acm	Corporation	Microsoft Corporation	OK	C:\WINDOWS\system32\MSADP32.ACM	5.2.3790.0	(srv03_rtm.030324-2048)

```

14.50 KB (14,848 bytes)      3/25/2003
12:00 AM
c:\windows\system32\imaadp32.acm      Microsoft
Corporation      OK
C:\WINDOWS\system32\IMAADP32.ACM
5.2.3790.0 (srv03_rtm.030324-2048)
15.50 KB (15,872 bytes)      3/25/2003
12:00 AM
c:\windows\system32\sl_anet.acm      Sipro Lab
Telecom Inc.      Sipro Lab Telecom Audio Codec OK
C:\WINDOWS\system32\SL_ANET.ACM
3.02      84.00 KB (86,016 bytes)
3/25/2003 12:00 AM
c:\windows\system32\msgsm32.acm      Microsoft
Corporation      OK
C:\WINDOWS\system32\MSGSM32.ACM
5.2.3790.0 (srv03_rtm.030324-2048)
20.50 KB (20,992 bytes)      3/25/2003
12:00 AM
c:\windows\system32\l3codeca.acm      Fraunhofer
Institut Integrierte Schaltungen IIS      Fraunhofer
IIS MPEG Layer-3 Codec      OK
C:\WINDOWS\system32\L3CODECA.ACM      1,
9, 0, 0305      284.00 KB (290,816 bytes)
3/25/2003 12:00 AM
c:\windows\system32\msaud32.acm      Microsoft
Corporation      Windows Media Audio Codec OK
C:\WINDOWS\system32\MSAUD32.ACM
8.00.00.4487      288.00 KB (294,912
bytes)      3/25/2003 12:00 AM

[Video Codecs]

CODEC      Manufacturer      Description
Status      File      Version      Size
Creation Date
c:\windows\system32\msh261.drv      Microsoft
Corporation      OK
C:\WINDOWS\system32\MSH261.DRV
4.4.4000      180.00 KB (184,320 bytes)
5/3/2003 3:32 PM
c:\windows\system32\iyuv_32.dll      Microsoft
Corporation      OK
C:\WINDOWS\system32\IYUV_32.DLL
5.2.3790.0 (srv03_rtm.030324-2048)
45.00 KB (46,080 bytes)      3/24/2003
7:49 PM
c:\windows\system32\tsbyuv.dll      Microsoft
Corporation      OK
C:\WINDOWS\system32\TSBYUV.DLL
5.2.3790.0 (srv03_rtm.030324-2048)
8.00 KB (8,192 bytes)      3/24/2003
7:50 PM
c:\windows\system32\msyuv.dll      Microsoft Corporation
OK
C:\WINDOWS\system32\MSYUV.DLL 5.2.3790.0
(srv03_rtm.030324-2048)      16.50 KB (16,896 bytes)
3/24/2003 7:49 PM
c:\windows\system32\msrle32.dll      Microsoft
Corporation      OK
C:\WINDOWS\system32\MSRLE32.DLL
5.2.3790.0 (srv03_rtm.030324-2048)

```

```

10.50 KB (10,752 bytes)      3/25/2003
12:00 AM
c:\windows\system32\msvidc32.dll      Microsoft
Corporation      OK
C:\WINDOWS\system32\MSVIDC32.DLL
5.2.3790.0 (srv03_rtm.030324-2048)
26.50 KB (27,136 bytes)      3/25/2003
12:00 AM
c:\windows\system32\msh263.drv      Microsoft
Corporation      OK
C:\WINDOWS\system32\MSH263.DRV
4.4.4000      284.00 KB (290,816 bytes)
3/24/2003 7:46 PM

[CD-ROM]

Item      Value
Drive      D:
Description      CD-ROM Drive
Media Loaded      No
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&FB0C83D&0&0.0.0
Driver      c:\windows\system32\drivers\cdrom.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 49.50 KB (50,688
bytes), 3/25/2003 12:00 AM)

[Sound Device]

Item      Value

[Display]

Item      Value
Name      RAGE XL PCI Family (Microsoft Corporation)

PNP Device ID      PCI\VEN_1002&DEV_4752&SUBSYS_001E0E11&REV_2
7\3&267A616A&0&18
Adapter Type      ATI RAGE XL PCI (B41), ATI
Technologies Inc. compatible
Adapter Description      RAGE XL PCI Family (Microsoft
Corporation)
Adapter RAM      8.00 MB (8,388,608 bytes)
Installed Drivers      ati2drad.dll
Driver Version      5.10.3663.6013
INF File      atixpad.inf (ati2mpad section)
Color Planes      1
Color Table Entries      65536
Resolution      800 x 600 x 60 hertz
Bits/Pixel      16
Memory Address      0xF6000000-0xF6FFFFFF
I/O Port      0x00002400-0x000024FF
Memory Address      0xF5FF0000-0xF5FFF0FF
I/O Port      0x000003B0-0x000003BB
I/O Port      0x000003C0-0x000003DF
Memory Address      0xA0000-0xBF5FF

```

```

Driver      c:\windows\system32\drivers\ati2mpad.sys
(5.10.3663.6013, 335.38 KB (343,424 bytes), 5/3/2003
9:23 AM)

[Infrared]

Item      Value

[Input]

[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&35118DFF&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
(5.2.3790.0 (srv03_rtm.030324-2048), 68.50 KB (70,144
bytes), 3/25/2003 12:00 AM)

[Pointing Device]

Item      Value
Hardware Type      PS/2 Compatible Mouse
Number of Buttons      5
Status      OK
PNP Device ID      ACPI\PNP0F13\4&35118DFF&0
Power Management Supported      No
Double Click Threshold      6
Handedness      Right Handed Operation
IRQ Channel      IRQ 12
Driver      c:\windows\system32\drivers\i8042prt.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 68.50 KB (70,144
bytes), 3/25/2003 12:00 AM)

[Modem]

Item      Value

[Network]

[Adapter]

Item      Value
Name      [00000001] Compaq NC7781 Gigabit Server
Adapter
Adapter Type      Not Available
Product Type      Compaq NC7781 Gigabit Server
Adapter
Installed Yes
PNP Device ID      Not Available
Last Reset      5/25/2003 1:43 AM
Index      1

```

Service Name q57w2k
 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] BCM5703 Gigabit Ethernet
 Adapter Type Not Available
 Product Type BCM5703 Gigabit Ethernet
 Installed Yes
 PNP Device ID Not Available
 Last Reset 5/25/2003 1:43 AM
 Index 2
 Service Name b57w2k
 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 [00000003] RAS Async Adapter
 Adapter Type Not Available
 Product Type RAS Async Adapter
 Installed Yes
 PNP Device ID Not Available
 Last Reset 5/25/2003 1:43 AM
 Index 3
 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 [00000004] WAN Miniport (L2TP)
 Adapter Type Not Available
 Product Type WAN Miniport (L2TP)
 Installed Yes
 PNP Device ID ROOT\MS_L2TPMINIPORT\0000
 Last Reset 5/25/2003 1:43 AM
 Index 4
 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
 (5.2.3790.0 (srv03_rtm.030324-2048), 77.00 KB (78,848 bytes), 3/25/2003 12:00 AM)

Name [00000005] 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 5/25/2003 1:43 AM
 Index 5
 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
 (5.2.3790.0 (srv03_rtm.030324-2048), 70.50 KB (72,192 bytes), 3/25/2003 12:00 AM)

Name [00000006] 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 5/25/2003 1:43 AM
 Index 6
 Service Name Rasppoe
 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\rasppoe.sys
 (5.2.3790.0 (srv03_rtm.030324-2048), 38.00 KB (38,912 bytes), 3/25/2003 12:00 AM)

Name [00000007] Direct Parallel
 Adapter Type Not Available
 Product Type Direct Parallel
 Installed Yes
 PNP Device ID ROOT\MS_PTMINIPORT\0000
 Last Reset 5/25/2003 1:43 AM
 Index 7
 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
 (5.2.3790.0 (srv03_rtm.030324-2048), 18.50 KB (18,944 bytes), 3/25/2003 12:00 AM)

Name [00000008] WAN Miniport (IP)
 Adapter Type Not Available

Product Type WAN Miniport (IP)
 Installed Yes
 PNP Device ID ROOT\MS_NDISWANIP\0000
 Last Reset 5/25/2003 1:43 AM
 Index 8
 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
 (5.2.3790.0 (srv03_rtm.030324-2048), 96.50 KB (98,816 bytes), 3/25/2003 12:00 AM)

Name [00000009] Compaq NC7781 Gigabit Server
 Adapter Ethernet 802.3
 Product Type Compaq NC7781 Gigabit Server
 Adapter
 Installed Yes
 PNP Device ID PCI\VEN_14E4&DEV_16A7&SUBSYS_00CB0E11&REV_02\3&1070020&0&08
 Last Reset 5/25/2003 1:43 AM
 Index 9
 Service Name q57w2k
 IP Address 130.168.202.9
 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 00:02:A5:EF:4F:BD
 Memory Address 0xF78F0000-0xF78FFFFF
 IRQ Channel IRQ 29
 Driver c:\windows\system32\drivers\q57xp32.sys
 (2.90.0.0 built by: WinDDK, 137.00 KB (140,288 bytes), 5/21/2003 5:39 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

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{345BEA79-4678-48A4-AAFA-8F66F6AF67E0}] SEQUENCE 4
 Connectionless Service No
 Guarantees Delivery Yes

Guarantees Sequencing Yes
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 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 No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{345BEA79-4678-48A4-AAFA-8F66F6AF67E0}] DATAGRAM 4
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 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 No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{FD0B4549-FE22-4DCC-A92E-E070C2A630B7}] SEQUENCE 0
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 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 No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{FD0B4549-FE22-4DCC-A92E-E070C2A630B7}] DATAGRAM 0
 Connectionless Service Yes
 Guarantees Delivery No

Guarantees Sequencing No
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 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 No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{98B7C9A4-D1AC-46D4-961D-5A4F2F08B1F2}] SEQUENCE 1
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 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 No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{98B7C9A4-D1AC-46D4-961D-5A4F2F08B1F2}] DATAGRAM 1
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 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 No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{063248C8-544A-4DB2-9D59-3812205D6848}] SEQUENCE 2
 Connectionless Service No
 Guarantees Delivery Yes

Guarantees Sequencing Yes
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 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 No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{063248C8-544A-4DB2-9D59-3812205D6848}] DATAGRAM 2
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 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 No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{30E6C8A5-7DC1-4DA0-B45A-B645F44A5D3C}] SEQUENCE 3
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 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 No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{30E6C8A5-7DC1-4DA0-B45A-B645F44A5D3C}] DATAGRAM 3
 Connectionless Service Yes
 Guarantees Delivery No

Guarantees Sequencing No
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 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 No

[WinSock]

Item Value
 File c:\windows\system32\winsock.dll
 Size 2.80 KB (2,864 bytes)
 Version 3.10

File c:\windows\system32\wsock32.dll
 Size 22.00 KB (22,528 bytes)
 Version 5.2.3790.0 (srv03_rtm.030324-2048)

[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
 (5.2.3790.0 (srv03_rtm.030324-2048), 76.00 KB (77,824 bytes), 3/25/2003 12:00 AM)

[Parallel]

Item Value

[Storage]

[Drives]

Item Value
 Drive A:
 Description 3 1/2 Inch Floppy Drive

Drive C:
 Description Local Fixed Disk
 Compressed No
 File System NTFS
 Size 16.00 GB (17,179,590,656 bytes)
 Free Space 12.39 GB (13,307,142,144 bytes)

Volume Name
 Volume Serial Number F445428C

Drive D:
 Description CD-ROM Disc

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 F:
 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 G:

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 H:
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 I:
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 J:
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 K:
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 X:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 305.20 GB (327,703,347,200 bytes)
Free Space 218.49 GB (234,605,576,192 bytes)

Volume Name Backup1
Volume Serial Number 18CA247F

Drive Y:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 305.20 GB (327,703,347,200 bytes)
Free Space 218.49 GB (234,605,576,192 bytes)

Volume Name Backup2
Volume Serial Number 40EDA029

Drive Z:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 305.20 GB (327,703,347,200 bytes)
Free Space 218.49 GB (234,605,641,728 bytes)

Volume Name Backup3
Volume Serial Number A80716A9

[Disks]

Item	Value
Description	Disk drive
Manufacturer	(Standard disk drives)
Model	COMPAQ 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	2
SCSI Target ID	4
Sectors/Track	32
Size	16.00 GB (17,179,607,040 bytes)
Total Cylinders	4,112
Total Sectors	33,553,920
Total Tracks	1,048,560
Tracks/Cylinder	255
Partition Disk #0,	Partition #0
Partition Size	16.00 GB (17,179,590,656 bytes)
Partition Starting Offset	16,384 bytes

Description	Disk drive
Manufacturer	(Standard disk drives)
Model	COMPAQ 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	2
SCSI Target ID	5
Sectors/Track	32
Size	119.66 GB (128,487,751,680 bytes)
Total Cylinders	30,754
Total Sectors	250,952,640
Total Tracks	7,842,270
Tracks/Cylinder	255
Partition Disk #1,	Partition #0
Partition Size	119.66 GB (128,483,557,376 bytes)
Partition Starting Offset	16,384 bytes

Description	Disk drive
Manufacturer	(Standard disk drives)
Model	COMPAQ 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	4
SCSI Target ID	4
Sectors/Track	63
Size	68.55 GB (73,608,030,720 bytes)
Total Cylinders	8,949
Total Sectors	143,765,685
Total Tracks	2,281,995
Tracks/Cylinder	255
Partition Disk #5,	Partition #0
Partition Size	68.55 GB (73,607,998,464 bytes)

Partition Starting Offset 32,256 bytes

Description	Disk drive
Manufacturer	(Standard disk drives)
Model	COMPAQ 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	4
SCSI Target ID	5
Sectors/Track	63
Size	33.26 GB (35,714,165,760 bytes)
Total Cylinders	4,342
Total Sectors	69,754,230
Total Tracks	1,107,210
Tracks/Cylinder	255
Partition Disk #6,	Partition #0
Partition Size	33.26 GB (35,714,133,504 bytes)

Partition Starting Offset 32,256 bytes

Description	Disk drive
Manufacturer	(Standard disk drives)
Model	COMPAQ 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	4
SCSI Target ID	6
Sectors/Track	63
Size	305.21 GB (327,711,605,760 bytes)
Total Cylinders	39,842
Total Sectors	640,061,730
Total Tracks	10,159,710
Tracks/Cylinder	255
Partition Disk #7,	Partition #0
Partition Size	305.20 GB (327,703,348,224 bytes)

Partition Starting Offset 32,256 bytes

Description	Disk drive
Manufacturer	(Standard disk drives)

Model COMPAQ 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 5
 SCSI Target ID 4
 Sectors/Track 63
 Size 68.55 GB (73,608,030,720 bytes)
 Total Cylinders 8,949
 Total Sectors 143,765,685
 Total Tracks 2,281,995
 Tracks/Cylinder 255
 Partition Disk #8, Partition #0
 Partition Size 68.55 GB (73,607,998,464 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ 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 5
 SCSI Target ID 5
 Sectors/Track 63
 Size 33.26 GB (35,714,165,760 bytes)
 Total Cylinders 4,342
 Total Sectors 69,754,230
 Total Tracks 1,107,210
 Tracks/Cylinder 255
 Partition Disk #9, Partition #0
 Partition Size 33.26 GB (35,714,133,504 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ 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 5
 SCSI Target ID 6
 Sectors/Track 63
 Size 305.21 GB (327,711,605,760 bytes)
 Total Cylinders 39,842
 Total Sectors 640,061,730
 Total Tracks 10,159,710
 Tracks/Cylinder 255
 Partition Disk #10, Partition #0
 Partition Size 305.20 GB (327,703,348,224 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ 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 3
 SCSI Target ID 4
 Sectors/Track 63
 Size 68.55 GB (73,608,030,720 bytes)
 Total Cylinders 8,949
 Total Sectors 143,765,685
 Total Tracks 2,281,995
 Tracks/Cylinder 255
 Partition Disk #2, Partition #0
 Partition Size 68.55 GB (73,607,998,464 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ 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 3
 SCSI Target ID 5
 Sectors/Track 63
 Size 33.26 GB (35,714,165,760 bytes)
 Total Cylinders 4,342
 Total Sectors 69,754,230
 Total Tracks 1,107,210
 Tracks/Cylinder 255
 Partition Disk #3, Partition #0
 Partition Size 33.26 GB (35,714,133,504 bytes)

Partition Starting Offset 32,256 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ 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 3
 SCSI Target ID 6
 Sectors/Track 63
 Size 305.21 GB (327,711,605,760 bytes)
 Total Cylinders 39,842
 Total Sectors 640,061,730
 Total Tracks 10,159,710
 Tracks/Cylinder 255
 Partition Disk #4, Partition #0

Partition Size 305.20 GB (327,703,348,224 bytes)

Partition Starting Offset 32,256 bytes

[SCSI]

Item Value
 Name Smart Array 5i
 Manufacturer Hewlett-Packard Company
 Status OK
 PNP Device ID
 PCI\VEN_0E11&DEV_B178&SUBSYS_40800E11&REV_0
 1\3&13C0B0C5&0&18
 Memory Address 0xF77C0000-0xF77FFFFF
 I/O Port 0x00003000-0x000030FF
 Memory Address 0xF76F0000-0xF76F3FFF
 IRQ Channel IRQ 30
 Driver c:\windows\system32\drivers\cpqccissm.sys
 (5.44.0.32 Build 1, 15.20 KB (15,568 bytes),
 5/21/2003 5:36 PM)

Name Smart Array 5300 Controller
 Manufacturer Hewlett-Packard Company
 Status OK
 PNP Device ID
 PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
 2\3&29E81982&0&08
 Memory Address 0xF7AC0000-0xF7AFFFFF
 Memory Address 0xF7900000-0xF7AFFFFF
 I/O Port 0x00004000-0x000040FF
 IRQ Channel IRQ 20
 Driver c:\windows\system32\drivers\cpqccissm.sys
 (5.44.0.32 Build 1, 15.20 KB (15,568 bytes),
 5/21/2003 5:36 PM)

Name Smart Array 5300 Controller
 Manufacturer Hewlett-Packard Company
 Status OK
 PNP Device ID
 PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
 2\3&172E68DD&0&08
 Memory Address 0xF7FC0000-0xF7FFFFFF
 Memory Address 0xF7E00000-0xF7EFFFFFFF
 I/O Port 0x00005000-0x000054FF
 IRQ Channel IRQ 24
 Driver c:\windows\system32\drivers\cpqccissm.sys
 (5.44.0.32 Build 1, 15.20 KB (15,568 bytes),
 5/21/2003 5:36 PM)

Name Smart Array 5300 Controller
 Manufacturer Hewlett-Packard Company
 Status OK
 PNP Device ID
 PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
 2\3&172E68DD&0&10
 Memory Address 0xF7DC0000-0xF7DFFFFFFF
 Memory Address 0xF7C00000-0xF7CFFFFFFF
 I/O Port 0x00005400-0x000054FF
 IRQ Channel IRQ 26
 Driver c:\windows\system32\drivers\cpqccissm.sys
 (5.44.0.32 Build 1, 15.20 KB (15,568 bytes),
 5/21/2003 5:36 PM)

[IDE]

Item	Value
Name	CSB5 IDE Controller
Manufacturer	ServerWorks
Status	OK
PNP Device ID	PCI\VEN_1166&DEV_0212&SUBSYS_02121166&REV_93\3&267A616A&0&79
I/O Port	0x00002000-0x0000200F
Driver	c:\windows\system32\drivers\pciide.sys (5.2.3790.0 (srv03_rtm.030324-2048), 5.50 KB (5,632 bytes), 3/25/2003 12:00 AM)
Name	Primary IDE Channel
Manufacturer	(Standard IDE ATA/ATAPI controllers)
Status	OK
PNP Device ID	PCIIDE\IDECHANNEL\4&1024D5C6&0&0
I/O Port	0x000001F0-0x000001F7
I/O Port	0x000003F6-0x000003F6
IRQ Channel	IRQ 14
Driver	c:\windows\system32\drivers\ataapi.sys (5.2.3790.0 (srv03_rtm.030324-2048), 89.00 KB (91,136 bytes), 3/25/2003 12:00 AM)
Name	Secondary IDE Channel
Manufacturer	(Standard IDE ATA/ATAPI controllers)
Status	OK
PNP Device ID	PCIIDE\IDECHANNEL\4&1024D5C6&0&1
I/O Port	0x00000170-0x00000177
I/O Port	0x00000376-0x00000376
Driver	c:\windows\system32\drivers\ataapi.sys (5.2.3790.0 (srv03_rtm.030324-2048), 89.00 KB (91,136 bytes), 3/25/2003 12:00 AM)

[Printing]

Name	Driver	Port	Name	Server	Name
------	--------	------	------	--------	------

[Problem Devices]

Device	PNP Device ID	Error Code
--------	---------------	------------

[USB]

Device	PNP Device ID
--------	---------------

[Software Environment]

[System Drivers]

Name	Description	File	Type
Started	Start Mode		State
Status	Error Control		Accept Pause
Accept	Stop		

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 Yes
acpiec	ACPIEC		
	c:\windows\system32\drivers\acpiec.sys		
	Kernel Driver	No	Disabled
	Stopped	OK	Normal No No
adpu160m	adpu160m	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Normal	No	No
adpu320	adpu320	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Normal	No	No
afcnt	afcnt	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Normal	No	No
afd	AFD Networking Support Environment		
	c:\windows\system32\drivers\afd.sys		
	Kernel Driver	Yes	Auto
	Running	OK	Normal No Yes
aha154x	Aha154x	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Normal	No	No
aic78u2	aic78u2	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Normal	No	No
aic78xx	aic78xx	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Normal	No	No
aliide	AliIde	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Normal	No	No
alkernel	Altiris Kernel Driver		
	c:\windows\system32\drivers\alkernel.sys		
	Kernel Driver	No	Manual
	Stopped	OK	Normal No No
asyncmac	RAS Asynchronous Media Driver		
	c:\windows\system32\drivers\asyncmac.sys		
	Kernel Driver	No	Manual
	Stopped	OK	Normal No No
atapi	Standard IDE/ESDI Hard Disk Controller		
	c:\windows\system32\drivers\ataapi.sys		
	Kernel Driver	Yes	Boot
	Running	OK	Normal No Yes
atdisk	Atdisk	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Ignore	No	No
ati2mpad	ati2mpad		
	c:\windows\system32\drivers\ati2mpad.sys		
	Kernel Driver	Yes	Manual
	Running	OK	Ignore No Yes

atmarpc	ATM ARP Client Protocol		
	c:\windows\system32\drivers\atmarpc.sys		
	Kernel Driver	No	Manual
	Stopped	OK	Normal No No
audstub	Audio Stub Driver		
	c:\windows\system32\drivers\audstub.sys		
	Kernel Driver	Yes	Manual
	Running	OK	Normal No Yes
b57w2k	BCM5703 Gigabit Ethernet		
	c:\windows\system32\drivers\b57xp32.sys		
	Kernel Driver	No	Manual
	Stopped	OK	Normal No No
beep	Beep		
	c:\windows\system32\drivers\beep.sys		
	Kernel Driver	Yes	System
	Running	OK	Normal No Yes
cbidf2k	cbidf2k		
	c:\windows\system32\drivers\cbidf2k.sys		
	Kernel Driver	No	Disabled
	Stopped	OK	Normal No No
cd20xrnt	cd20xrnt	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Normal	No	No
cdfs	Cdfs		
	c:\windows\system32\drivers\cdfs.sys		
	File System Driver	Yes	Disabled
	Running	OK	Normal No Yes
cdrom	CD-ROM Driver		
	c:\windows\system32\drivers\cdrom.sys		
	Kernel Driver	Yes	System
	Running	OK	Normal No Yes
changer	Changer	Not Available	Kernel Driver
	No	System	Stopped OK
	Ignore	No	No
clusdisk	Cluster Disk Driver		
	c:\windows\system32\drivers\clusdisk.sys		
	Kernel Driver	No	Disabled
	Stopped	OK	Normal No No
cmdide	CmdIde	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Normal	No	No
cpqarray	Cpqarray	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Normal	No	No
cpqarray2	cpqarray2	Not Available	Kernel Driver
	No	Disabled	Stopped OK
	Normal	No	No
cpqasm2	HP ProLiant iLO Advanced System Management Controller		
	c:\windows\system32\drivers\cpqasm2.sys		
	Kernel Driver	Yes	Manual
	Running	OK	Normal No Yes
cpqcidrv	HP Integrated Lights-Out		
	c:\windows\system32\drivers\cpqcidrv.sys		

	Kernel Driver	Yes	Manual	Yes		dpti2o	dpti2o	Not Available	Kernel Driver			Running	OK	Normal	No	Yes
	Running	OK	Normal	No	Yes		No	Disabled	Stopped	OK						
cpqcisse	CPQCISSSE					fastfat	Fastfat	No	No			iirsp	iirsp	Not Available	Kernel Driver	
	c:\windows\system32\drivers\cpqcisse.sys						c:\windows\system32\drivers\fastfat.sys						No	Disabled	Stopped	OK
	Kernel Driver	Yes	Manual				File System Driver	No	Disabled				Normal	No	No	
	Running	OK	Normal	No	Yes		Stopped	OK	Normal	No	No	imapi	CD-Burning Filter Driver			
													c:\windows\system32\drivers\imapi.sys			
cpqcissm	cpqcissm					fdc	Floppy Disk Controller Driver						Kernel Driver	No	System	No
	c:\windows\system32\drivers\cpqcissm.sys						c:\windows\system32\drivers\fdc.sys						Stopped	OK	Normal	No
	Kernel Driver	Yes	Boot				Kernel Driver	Yes	Manual			intelide	IntelIde	Not Available	Kernel Driver	
	Running	OK	Normal	No	Yes		Running	OK	Normal	No	Yes		No	Disabled	Stopped	OK
													Normal	No	No	
cpqfcalm	cpqfcalm	Not Available		Kernel Driver		fips	Fips					ipfilterdriver	IP Traffic Filter Driver			
	No	Disabled	Stopped	OK			c:\windows\system32\drivers\fips.sys						c:\windows\system32\drivers\ipfltdrv.sys			
	Normal	No	No				Kernel Driver	Yes	System				Kernel Driver	No	Manual	No
cpqphp	hp ProLiant PCI Hot Plug Controller (CPQPHP installed)						Running	OK	Normal	No	Yes		Stopped	OK	Normal	No
	c:\windows\system32\drivers\cpqphp.sys					flpydisk	Floppy Disk Driver					ipinip	IP in IP Tunnel Driver			
	Kernel Driver	Yes	Boot				c:\windows\system32\drivers\flpydisk.sys						c:\windows\system32\drivers\ipinip.sys			
	Running	OK	Normal	No	Yes		Kernel Driver	Yes	Manual				Kernel Driver	No	Manual	No
							Running	OK	Normal	No	Yes		Stopped	OK	Normal	No
cpqteam	HP Network Teaming and Configuration					ftdisk	Volume Manager Driver					ipnat	IP Network Address Translator			
	c:\windows\system32\drivers\cpqteam.sys						c:\windows\system32\drivers\ftdisk.sys						c:\windows\system32\drivers\ipnat.sys			
	Kernel Driver	No	Manual				Kernel Driver	Yes	Boot				Kernel Driver	No	Manual	No
	Stopped	OK	Normal	No	No		Running	OK	Normal	No	Yes		Stopped	OK	Normal	No
crcdisk	CRC Disk Filter Driver					gpc	Generic Packet Classifier					ipsec	IPSEC driver			
	c:\windows\system32\drivers\crcdisk.sys						c:\windows\system32\drivers\msgpc.sys						c:\windows\system32\drivers\ipsec.sys			
	Kernel Driver	Yes	Boot				Kernel Driver	Yes	Manual				Kernel Driver	Yes	System	Yes
	Running	OK	Normal	No	Yes		Running	OK	Normal	No	Yes		Running	OK	Normal	No
dac960nt	dac960nt	Not Available		Kernel Driver		hpn	hpn	Not Available	Kernel Driver			ipsraidn	ipsraidn	Not Available	Kernel Driver	
	No	Disabled	Stopped	OK			No	Disabled	Stopped	OK			No	Disabled	Stopped	OK
	Normal	No	No				Normal	No	No				Normal	No	No	
dellcerc	dellcerc	Not Available		Kernel Driver		hpqcissb	Smart Array Controllers Non-Miniport Bus Driver					isapnp	PnP ISA/EISA Bus Driver			
	No	Disabled	Stopped	OK			c:\windows\system32\drivers\hpqcissb.sys						c:\windows\system32\drivers\isapnp.sys			
	Normal	No	No				Kernel Driver	Yes	Boot				Kernel Driver	Yes	Boot	No
dfsdriver	DfsDriver						Running	OK	Normal	No	Yes		Running	OK	Critical	Yes
	c:\windows\system32\drivers\dfs.sys					hpqcissd	Smart Array Controllers Non-Miniport Disk Driver					kbdclass	Keyboard Class Driver			
	File System Driver	Yes	Boot				c:\windows\system32\drivers\hpqcissd.sys						c:\windows\system32\drivers\kbdclass.sys			
	Running	OK	Normal	No	Yes		Kernel Driver	Yes	Boot				Kernel Driver	Yes	System	Yes
							Running	OK	Normal	No	Yes		Running	OK	Normal	No
disk	Disk Driver					hpt3xx	hpt3xx	Not Available	Kernel Driver			ksecdd	KSecDD			
	c:\windows\system32\drivers\disk.sys						No	Disabled	Stopped	OK			c:\windows\system32\drivers\ksecdd.sys			
	Kernel Driver	Yes	Boot				Normal	No	No				Kernel Driver	Yes	Boot	Yes
	Running	OK	Normal	No	Yes		http	HTTP					Running	OK	Normal	No
dmboot	dmboot						c:\windows\system32\drivers\http.sys					lp6nds35	lp6nds35	Not Available	Kernel Driver	
	c:\windows\system32\drivers\dmboot.sys						Kernel Driver	No	Manual				No	Disabled	Stopped	OK
	Kernel Driver	No	Disabled		No		Stopped	OK	Normal	No	No		Normal	No	No	
	Stopped	OK	Normal	No	No								nmnmd	nmnmd		
dmio	Logical Disk Manager Driver					i2omgmt	i2omgmt	Not Available	Kernel Driver				c:\windows\system32\drivers\mnmd.sys			
	c:\windows\system32\drivers\dmio.sys						No	System	Stopped	OK			Kernel Driver	Yes	System	Yes
	Kernel Driver	Yes	Boot				Normal	No	No				Running	OK	Ignore	No
	Running	OK	Normal	No	Yes		i2omp	i2omp	Not Available	Kernel Driver						
							No	Disabled	Stopped	OK		modem	Modem			
dmload	dmload						Normal	No	No				c:\windows\system32\drivers\modem.sys			
	c:\windows\system32\drivers\dmload.sys					i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver						Kernel Driver	No	Manual	
	Kernel Driver	Yes	Boot				c:\windows\system32\drivers\i8042prt.sys									
	Running	OK	Normal	No	Yes		Kernel Driver	Yes	System							

	Stopped	OK	Ignore	No	No
mouclass	Mouse Class Driver				
	c:\windows\system32\drivers\mouclass.sys				
	Kernel Driver	Yes	System		
	Running	OK	Normal	No	Yes
mountmgr	Mount Point Manager				
	c:\windows\system32\drivers\mountmgr.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
mraid35x	mraid35x	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
mrxdav	WebDav Client Redirector				
	c:\windows\system32\drivers\mrxdav.sys				
	File System Driver	No	Manual		
	Stopped	OK	Normal	No	No
mrxsmb	MRXSMB				
	c:\windows\system32\drivers\mrxsmb.sys				
	File System Driver	Yes	System		
	Running	OK	Normal	No	Yes
msfs	Msfs				
	c:\windows\system32\drivers\msfs.sys				
	File System Driver	Yes	System		
	Running	OK	Normal	No	Yes
mup	Mup				
	c:\windows\system32\drivers\mup.sys				
	File System Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
ndis	NDIS System Driver				
	c:\windows\system32\drivers\ndis.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
ndistapi	Remote Access NDIS TAPI Driver				
	c:\windows\system32\drivers\ndistapi.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
ndisuio	NDIS Usermode I/O Protocol				
	c:\windows\system32\drivers\ndisuio.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
ndiswan	Remote Access NDIS WAN Driver				
	c:\windows\system32\drivers\ndiswan.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
ndproxy	NDIS Proxy				
	c:\windows\system32\drivers\ndproxy.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
netbios	NetBIOS Interface				
	c:\windows\system32\drivers\netbios.sys				
	File System Driver	Yes	System		

	Running	OK	Normal	No	Yes
netbt	NetBios over Tcpip				
	c:\windows\system32\drivers\netbt.sys				
	Kernel Driver	Yes	System		
	Running	OK	Normal	No	Yes
nfrd960	nfrd960	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
npfs	Npfs				
	c:\windows\system32\drivers\npfs.sys				
	File System Driver	Yes	System		
	Running	OK	Normal	No	Yes
ntfs	Ntfs				
	c:\windows\system32\drivers\ntfs.sys				
	File System Driver	Yes	Disabled		
	Running	OK	Normal	No	Yes
null	Null				
	c:\windows\system32\drivers\null.sys				
	Kernel Driver	Yes	System		
	Running	OK	Normal	No	Yes
parport	Parport				
	c:\windows\system32\drivers\parport.sys				
	Kernel Driver	No	Manual		
	Stopped	OK	Ignore	No	No
partmgr	Partition Manager				
	c:\windows\system32\drivers\partmgr.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
pci	PCI Bus Driver				
	c:\windows\system32\drivers\pci.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Critical	No	Yes
pciide	PCIIde				
	c:\windows\system32\drivers\pciide.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
pcmcia	Pcmcia				
	c:\windows\system32\drivers\pcmcia.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Normal	No	No
pdcomp	PDCOMP	Not Available		Kernel Driver	
	No	Manual	Stopped	OK	
	Ignore	No	No		
pdframe	PDFFRAME	Not Available		Kernel Driver	
	No	Manual	Stopped	OK	
	Ignore	No	No		
pdreli	PDRELI	Not Available		Kernel Driver	
	No	Manual	Stopped	OK	
	Ignore	No	No		
pdrframe	PDRFRAME	Not Available		Kernel Driver	
	No	Manual	Stopped	OK	
	Ignore	No	No		

perc2	perc2	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
perc2hib	perc2hib	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
pptpminiport	WAN Miniport (PPTP)				
	c:\windows\system32\drivers\rasppptp.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
processor	Processor Driver				
	c:\windows\system32\drivers\processr.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
ptilink	Direct Parallel Link Driver				
	c:\windows\system32\drivers\ptilink.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
q57w2k	Compaq NC7781 Gigabit Server Adapter				
	c:\windows\system32\drivers\q57xp32.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
ql1080	ql1080	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql10wnt	ql10wnt	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql12160	ql12160	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql1240	ql1240	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql1280	ql1280	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql2100	ql2100	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql2200	ql2200	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql2300	ql2300	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
rasacd	Remote Access Auto Connection Driver				
	c:\windows\system32\drivers\rasacd.sys				
	Kernel Driver	Yes	System		
	Running	OK	Normal	No	Yes
rasl2tp	WAN Miniport (L2TP)				
	c:\windows\system32\drivers\rasl2tp.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
raspppoe	Remote Access PPPOE Driver				
	c:\windows\system32\drivers\raspppoe.sys				
	Kernel Driver	Yes	Manual		

	Running	OK	Normal	No	Yes		swenum	Software Bus Driver c:\windows\system32\drivers\swenum.sys Kernel Driver Running	OK	Normal	No	Yes			Kernel Driver	No	Manual	No	No		
raspti	Direct Parallel c:\windows\system32\drivers\raspti.sys Kernel Driver Running													usbobhci Driver	Microsoft USB Open Host Controller Miniport c:\windows\system32\drivers\usbobhci.sys Kernel Driver Stopped	OK	Normal	No	Manual	No	
rdbss	Rdbss c:\windows\system32\drivers\rdbss.sys File System Driver Running						symc810	symc810 No Normal	Not Available Disabled No	Stopped	Kernel Driver	OK								No	
							symc8xx	symc8xx No Normal	Not Available Disabled No	Stopped	Kernel Driver	OK		vgasave	VGA Display Controller. c:\windows\system32\drivers\vga.sys Kernel Driver Running		Yes	System	No	Yes	
rdpcdd	RDPCDD c:\windows\system32\drivers\rdpcdd.sys Kernel Driver Running						symmpi	symmpi No Normal	Not Available Disabled No	Stopped	Kernel Driver	OK								Yes	
							sym_hi	sym_hi No Normal	Not Available Disabled No	Stopped	Kernel Driver	OK		viaide	ViaIde No Normal	Not Available Disabled No	Stopped	Kernel Driver	OK		
rdpdr	Terminal Server Device Redirector Driver c:\windows\system32\drivers\rdpdr.sys Kernel Driver Running						sym_u3	sym_u3 No Normal	Not Available Disabled No	Stopped	Kernel Driver	OK		volsnap	Storage volumes c:\windows\system32\drivers\volsnap.sys Kernel Driver Running		Yes	Boot	No	Yes	
rdpwd	RDPWD c:\windows\system32\drivers\rdpwd.sys Kernel Driver Running						sysmgmt Driver	HP ProLiant System Management Interface c:\windows\system32\drivers\sysmgmt.sys Kernel Driver Stopped			No	Manual	No		wanarp	Remote Access IP ARP Driver c:\windows\system32\drivers\wanarp.sys Kernel Driver Running		Yes	Manual	No	Yes
							tcpip	TCP/IP Protocol Driver c:\windows\system32\drivers\tcpip.sys Kernel Driver Running			Yes	System	No	Yes		wdica	WDICA No Ignore	Not Available Manual No	Stopped	Kernel Driver	OK
redbook	Digital CD Audio Playback Filter Driver c:\windows\system32\drivers\redbook.sys Kernel Driver Running						tdpipe	TDPIPE c:\windows\system32\drivers\tdpipe.sys Kernel Driver Stopped			No	Manual	No		wlbs	Network Load Balancing c:\windows\system32\drivers\wlbs.sys Kernel Driver Stopped		No	Manual	No	No
secdrv	Secdrv c:\windows\system32\drivers\secdrv.sys Kernel Driver Stopped						tdtcp	TDTCP c:\windows\system32\drivers\tdtcp.sys Kernel Driver Running			Yes	Manual	No	Yes	[Signed Drivers]						
serenum	Serenum Filter Driver c:\windows\system32\drivers\serenum.sys Kernel Driver Running						termdd	Terminal Device Driver c:\windows\system32\drivers\termdd.sys Kernel Driver Running			Yes	System	No	Yes	Device Name	Signed Driver Version Manufacturer Device ID	Device Class Driver Date INF Name	Driver Name			
serial	Serial port driver c:\windows\system32\drivers\serial.sys Kernel Driver Running													Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	
							toside	TosIde No Normal	Not Available Disabled No	Stopped	Kernel Driver	OK		Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	
sfloppy	Sfloppy c:\windows\system32\drivers\sfloppy.sys Kernel Driver Stopped						udfs	Udfs c:\windows\system32\drivers\udfs.sys File System Driver Stopped			No	Disabled	No	ACPI Multiprocessor PC 5.2.3790.0 computers)	hal.inf	Yes	COMPUTER	(Standard			
														ROOT\ACPI_HAL\0000	ACPI-Compliant System	Yes	10/1/2002				
simbad	Simbad No Normal	Not Available Disabled No	Stopped	Kernel Driver	OK		ultra	ultra No Normal	Not Available Disabled No	Stopped	Kernel Driver	OK		Microsoft acpi.inf	Not Available	Yes	PROCESSOR 5.2.3790.0	10/1/2002			
sparrow	Sparrow No Normal	Not Available Disabled No	Stopped	Kernel Driver	OK		update	Microcode Update Driver c:\windows\system32\drivers\update.sys Kernel Driver Running			Yes	Manual	No	Yes	ACPI_HAL\PNP0C08\0	Not Available	Yes	PROCESSOR 5.2.3790.0	10/1/2002 (Standard processor types)		
srv	Srv c:\windows\system32\drivers\srv.sys File System Driver Running													cpu.inf	Not Available	Yes	PROCESSOR 5.2.3790.0	10/1/2002 (Standard processor types)			
							usbhub	USB2 Enabled Hub c:\windows\system32\drivers\usbhub.sys							ACPI\GENUINEINTEL_- _X86_FAMILY_15_MODEL_2_0	Processor	Yes	PROCESSOR 5.2.3790.0	10/1/2002 (Standard processor types)		
														cpu.inf	Not Available	Yes	PROCESSOR 5.2.3790.0	10/1/2002 (Standard processor types)			
														ACPI\GENUINEINTEL_- _X86_FAMILY_15_MODEL_2_1		Yes	PROCESSOR 5.2.3790.0	10/1/2002 (Standard processor types)			

Processor Yes PROCESSOR 5.2.3790.0
10/1/2002 (Standard processor types)
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_X86_FAMILY_15_MODEL_2\6
Processor Yes PROCESSOR 5.2.3790.0
10/1/2002 (Standard processor types)
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_X86_FAMILY_15_MODEL_2\7
PCI bus Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A03\0
ServerWorks (RCC) CMIC_LE Processor to PCI Bridge(*)
Yes SYSTEM 5.2.3790.0
10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0014&SUBSYS_00000000&REV_3
1\3&267A616A&0&00
ServerWorks (RCC) CMIC_LE Processor to PCI Bridge(*)
Yes SYSTEM 5.2.3790.0
10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0014&SUBSYS_00000000&REV_0
0\3&267A616A&0&01
ServerWorks (RCC) CMIC_LE Processor to PCI Bridge(*)
Yes SYSTEM 5.2.3790.0
10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0014&SUBSYS_00000000&REV_0
0\3&267A616A&0&02
RAGE XL PCI Family (Microsoft Corporation) Yes
DISPLAY 5.10.2600.6014 8/8/2001 ATI
Technologies Inc. atixpad.inf Not Available
PCI\VEN_1002&DEV_4752&SUBSYS_001E0E11&REV_2
7\3&267A616A&0&18
Plug and Play Monitor Yes MONITOR
5.1.2001.0 6/6/2001 (Standard
monitor types) monitor.inf Not Available
DISPLAY\AV00402\4&89B5141&0&0000001&00&03
HP ProLiant iLO Advanced System Management Controller
Yes SYSTEM 5.30.3718.0
3/20/2003 Compaq oem0.inf Not Available
PCI\VEN_0E11&DEV_B203&SUBSYS_B2060E11&REV_0
1\3&267A616A&0&20
HP iLO Management Interface Driver Yes
MULTIFUNCTION 1.4.0.0 9/4/2002
Hewlett-Packard Company oem4.inf Not
Available
PCI\VEN_0E11&DEV_B204&SUBSYS_B2060E11&REV_0
1\3&267A616A&0&22
PCI standard ISA bridge Yes SYSTEM
5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_1166&DEV_0201&SUBSYS_00000000&REV_9
3\3&267A616A&0&78
ISAPNP Read Data Port Yes SYSTEM
5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ISAPNP\READDATAPORT\0

Motherboard resources Yes SYSTEM
5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0C02\0
Programmable interrupt controller Yes
SYSTEM 5.2.3790.0 10/1/2002
(Standard system devices) machine.inf
Not Available
ACPI\PNP0000\4&35118DFF&0
System timer Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0100\4&35118DFF&0
Direct memory access controller Yes
SYSTEM 5.2.3790.0 10/1/2002
(Standard system devices) machine.inf
Not Available
ACPI\PNP0200\4&35118DFF&0
System speaker Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0800\4&35118DFF&0
Standard 101/102-Key or Microsoft Natural PS/2
Keyboard Yes KEYBOARD 5.2.3790.0
10/1/2002 (Standard keyboards)
keyboard.inf Not Available
ACPI\PNP0303\4&35118DFF&0
PS/2 Compatible Mouse Yes MOUSE
5.2.3790.0 10/1/2002 Microsoft
msmouse.inf Not Available
ACPI\PNP0F13\4&35118DFF&0
Extended IO Bus Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A06\4&35118DFF&0
Communications Port Yes PORTS 5.2.3790.0
10/1/2002 (Standard port types)
msports.inf Not Available
ACPI\PNP0501\0
Standard floppy disk controller Yes FDC
5.2.3790.0 10/1/2002 (Standard
floppy disk controllers) fdc.inf Not Available
ACPI\PNP0700\5&13237358&0
Floppy disk drive Yes FLOPPYDISK
5.2.3790.0 10/1/2002 (Standard
floppy disk drives) flpydisk.inf Not Available
FDC\GENERIC_FLOPPY_DRIVE\6&1C650E5D&0&0
CSB5 IDE Controller Yes HDC 5.2.3790.0
10/1/2002 ServerWorks mshdc.inf Not
Available
PCI\VEN_1166&DEV_0212&SUBSYS_02121166&REV_9
3\3&267A616A&0&79
Primary IDE Channel Yes HDC 5.2.3790.0
10/1/2002 (Standard IDE ATA/ATAPI
controllers) mshdc.inf Not Available
PCIIDE\IDECHANNEL\4&1024D5C6&0&0
CD-ROM Drive Yes CDROM 5.2.3790.0
10/1/2002 (Standard CD-ROM drives)
cdrom.inf Not Available
IDE\CDROMCOMPAQ_CD-ROM-SN-
124 N104 \5&FB0C83D&0&0.0.0
Secondary IDE Channel Yes HDC
5.2.3790.0 10/1/2002 (Standard IDE

ATA/ATAPI controllers) mshdc.inf Not Available
PCIIDE\IDECHANNEL\4&1024D5C6&0&1
Serverworks Champion CSB5 - SouthBridge 5 LPC Yes
SYSTEM 5.2.3790.0 10/1/2002
ServerWorks (RCC) machine.inf Not
Available
PCI\VEN_1166&DEV_0225&SUBSYS_00000000&REV_0
0\3&267A616A&0&7B
ServerWorks Grand Champion CIOB_X2 - I/O Bridge 133
Mhz Yes SYSTEM 5.2.3790.0
10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0101&SUBSYS_00000000&REV_0
5\3&267A616A&0&80
ServerWorks Grand Champion CIOB_X2 - I/O Bridge 133
Mhz Yes SYSTEM 5.2.3790.0
10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0101&SUBSYS_00000000&REV_0
5\3&267A616A&0&82
ServerWorks Grand Champion CIOB_X2 - I/O Bridge 133
Mhz Yes SYSTEM 5.2.3790.0
10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0101&SUBSYS_00000000&REV_0
5\3&267A616A&0&88
ServerWorks Grand Champion CIOB_X2 - I/O Bridge 133
Mhz Yes SYSTEM 5.2.3790.0
10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0101&SUBSYS_00000000&REV_0
5\3&267A616A&0&8A
PCI bus Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A03\1
Smart Array 5i Yes SCSIADAPTER
5.44.0.32 1/31/2003 Hewlett-Packard Company
oem1.inf Not Available
PCI\VEN_0E11&DEV_B178&SUBSYS_40800E11&REV_0
1\3&13C0B0C5&0&18
Smart Array 5x and 6x Notification Driver Yes
SYSTEM 5.44.0.32 2/12/2003 Hewlett-
Packard Company oem2.inf Not Available
SCSI\OTHER&VEN_COMPAQ&PROD_SCSI_COMMUNICATE
&REV_CISS\4&73DC70A&0&000
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME&RE
V_2.38\4&73DC70A&0&040
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME&RE
V_2.38\4&73DC70A&0&050
PCI bus Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A03\2
Compaq NC7781 Gigabit Server Adapter Yes NET
2.90.0.0 12/12/2002 Compaq
oem5.inf Not Available

```

PCI\VEN_14E4&DEV_16A7&SUBSYS_00CB0E11&REV_0
2\3&1070020&0&08
PCI bus Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A03\3
Smart Array 5300 Controller Yes SCSIADAPTER
5.44.0.32 1/31/2003 Hewlett-Packard Company
oem1.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
2\3&29E81982&0&08
Smart Array 5x and 6x Notification Driver Yes
SYSTEM 5.44.0.32 2/12/2003 Hewlett-
Packard Company oem2.inf Not Available
SCSI\OTHER&VEN_COMPAQ&PROD_SCSI_COMMUNICATE
&REV_CISS\4&38EB4840&0&000
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME&RE
V_3.32\4&38EB4840&0&040
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME&RE
V_3.32\4&38EB4840&0&050
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME&RE
V_3.32\4&38EB4840&0&060
PCI bus Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A03\4
Smart Array 5300 Controller Yes SCSIADAPTER
5.44.0.32 1/31/2003 Hewlett-Packard Company
oem1.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
2\3&172E68DD&0&08
Smart Array 5x and 6x Notification Driver Yes
SYSTEM 5.44.0.32 2/12/2003 Hewlett-
Packard Company oem2.inf Not Available
SCSI\OTHER&VEN_COMPAQ&PROD_SCSI_COMMUNICATE
&REV_CISS\4&1F72F2BD&0&000
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME&RE
V_3.32\4&1F72F2BD&0&040
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME&RE
V_3.32\4&1F72F2BD&0&050
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME&RE
V_3.32\4&1F72F2BD&0&060
Smart Array 5300 Controller Yes SCSIADAPTER
5.44.0.32 1/31/2003 Hewlett-Packard Company
oem1.inf Not Available

```

```

PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
2\3&172E68DD&0&10
Smart Array 5x and 6x Notification Driver Yes
SYSTEM 5.44.0.32 2/12/2003 Hewlett-
Packard Company oem2.inf Not Available
SCSI\OTHER&VEN_COMPAQ&PROD_SCSI_COMMUNICATE
&REV_CISS\4&2E12B67&0&000
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME&RE
V_3.32\4&2E12B67&0&040
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME&RE
V_3.32\4&2E12B67&0&050
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME&RE
V_3.32\4&2E12B67&0&060
hp ProLiant PCI Hot Plug Controller (CPQPHP
installed) Yes SYSTEM 6.1.2.5
1/6/2003 HP oem3.inf Not Available
PCI\VEN_0E11&DEV_A0F7&SUBSYS_A2FE0E11&REV_1
4\3&172E68DD&0&F0
ACPI Thermal Zone Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\THERMALZONE\THM0
ACPI Fixed Feature Button Yes SYSTEM
5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\FIXEDBUTTON\2&DABA3FF&0
Logical Disk Manager Yes SYSTEM
5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ROOT\DMIO\0000
Volume Manager Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ROOT\FTDISK\0000
Generic volume Yes VOLUME 5.2.3790.0
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURECE68CE
680FFSET4000LENGTH3FFFBC000
Generic volume Yes VOLUME 5.2.3790.0
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE6B50EC
4A0FFSET4000LENGTH1DEA378000
Generic volume Yes VOLUME 5.2.3790.0
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREACAC1A
0B0FFSET7E00LENGTH112360EC00
Generic volume Yes VOLUME 5.2.3790.0
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREACAC1A
040FFSET7E00LENGTH850BA6E00

```

```

Generic volume Yes VOLUME 5.2.3790.0
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREACAC1A
060FFSET7E00LENGTH4C4CA4400
Generic volume Yes VOLUME 5.2.3790.0
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREACAC1A
070FFSET7E00LENGTH112360EC00
Generic volume Yes VOLUME 5.2.3790.0
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREACAC1A
000FFSET7E00LENGTH850BA6E00
Generic volume Yes VOLUME 5.2.3790.0
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREACAC1A
010FFSET7E00LENGTH4C4CA4400
Generic volume Yes VOLUME 5.2.3790.0
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREACAC1A
030FFSET7E00LENGTH112360EC00
Generic volume Yes VOLUME 5.2.3790.0
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREACAC1A
1D0FFSET7E00LENGTH850BA6E00
Generic volume Yes VOLUME 5.2.3790.0
10/1/2002 Microsoft volume.inf Not
Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREACAC1A
1E0FFSET7E00LENGTH4C4CA4400
AFD Networking Support Environment Not Available
LEGACYDRIVER Not Available Not
Available Not Available Not Available Not
Available ROOT\LEGACY_AFD\0000
Altiris Kernel Driver Not Available
LEGACYDRIVER Not Available Not
Available Not Available Not Available Not
Available ROOT\LEGACY_ALKERNEL\0000
Beep Not Available LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available ROOT\LEGACY_BEEP\0000

CRC Disk Filter Driver Not Available
LEGACYDRIVER Not Available Not
Available Not Available Not Available Not
Available ROOT\LEGACY_CRCDISK\0000
dmbot Not Available LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available ROOT\LEGACY_DMBOT\0000

dmlod Not Available LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available ROOT\LEGACY_DMLOAD\0000

Fips Not Available LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available ROOT\LEGACY_FIPS\0000

```

Generic Packet Classifier		Not Available	
LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available
Available	ROOT\LEGACY_GPC\0000		
Smart Array Controllers	Non-Miniport Bus Driver		Not Available
LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available
Available	ROOT\LEGACY_HPQCISSB\0000		
Smart Array Controllers	Non-Miniport Disk Driver		Not Available
LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available
Available	ROOT\LEGACY_HPQCISSD\0000		
IPSEC driver	Not Available	LEGACYDRIVER	
Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Not Available
ROOT\LEGACY_IPSEC\0000			
ksecdd	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Available	Not Available	ROOT\LEGACY_KSECD\0000	
mnmdd	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Available	Not Available	ROOT\LEGACY_MMDD\0000	
mountmgr	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Available	Not Available		
ROOT\LEGACY_MOUNTMGR\0000			
NDIS System Driver	Not Available	LEGACYDRIVER	
Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Not Available
ROOT\LEGACY_NDIS\0000			
Remote Access	NDIS TAPI Driver		Not Available
LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available
Available	ROOT\LEGACY_NDIS\TAPI\0000		
NDIS Usermode	I/O Protocol	Not Available	Not Available
LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available
Available	ROOT\LEGACY_NDIS\UIO\0000		
NDProxy	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Available	Not Available		
ROOT\LEGACY_NDPROXY\0000			
NetBios over Tcpi	Not Available	LEGACYDRIVER	
Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Not Available
ROOT\LEGACY_NETBT\0000			
Null	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Available	Not Available	ROOT\LEGACY_NULL\0000	
Partition Manager	Not Available	LEGACYDRIVER	
Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Not Available
ROOT\LEGACY_PARTMGR\0000			
Remote Access	Auto Connection Driver		Not Available
LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available
Available	ROOT\LEGACY_RASACD\0000		
RDPCCD	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available

Direct Parallel	Yes	NET	5.2.3790.0			
Available	10/1/2002	Microsoft netraser.inf		Not		
Terminal Server Device Redirector			Yes			
	SYSTEM	5.2.3790.0		10/1/2002		
	(Standard system devices)		machine.inf			
Terminal Server Keyboard Driver		ROOT\RDPDR\0000	Yes			
	SYSTEM	5.2.3790.0		10/1/2002		
	(Standard system devices)		machine.inf			
Terminal Server Mouse Driver		ROOT\RDP_KBD\0000	SYSTEM			
	5.2.3790.0		10/1/2002	(Standard		
system devices)	machine.inf			Not Available		
	ROOT\RDP_MOUSE\0000					
Plug and Play Software Device Enumerator			Yes			
	SYSTEM	5.2.3790.0		10/1/2002		
	(Standard system devices)		machine.inf			
	Not Available	ROOT\SYSTEM\0000				
Microcode Update Device		Yes	SYSTEM			
	5.2.3790.0		10/1/2002	(Standard		
system devices)	machine.inf			Not Available		
	ROOT\SYSTEM\0001					
[Environment Variables]						
Variable	Value	User Name				
ClusterLog		C:\WINDOWS\Cluster\cluster.log				
	<SYSTEM>					
ComSpec	%SystemRoot%\system32\cmd.exe	<SYSTEM>				
NUMBER_OF_PROCESSORS	4	<SYSTEM>				
OS	Windows_NT	<SYSTEM>				
Path	%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\Binn;C:\Program Files\Microsoft SQL Server\MSSQL\Binn;. <SYSTEM>					
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH <SYSTEM>					
PROCESSOR_ARCHITECTURE	x86	<SYSTEM>				
PROCESSOR_IDENTIFIER	x86 Family 15 Model 2					
Stepping 7, GenuineIntel		<SYSTEM>				
PROCESSOR_LEVEL	15	<SYSTEM>				
PROCESSOR_REVISION	0207	<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\LOCAL SERVICE						
TMP	%USERPROFILE%\Local Settings\Temp		NT			
AUTHORITY\LOCAL SERVICE						
TEMP	%USERPROFILE%\Local Settings\Temp		NT			
AUTHORITY\NETWORK SERVICE						
TMP	%USERPROFILE%\Local Settings\Temp		NT			
AUTHORITY\NETWORK SERVICE						
TEMP	%USERPROFILE%\Local Settings\Temp					
	APARK2\Administrator					

TMP %USERPROFILE%\Local Settings\Temp
APARK2\Administrator

[Print Jobs]

Document	Size	Owner	Notify	Status
Until Time	Time Submitted	Time	Start Time	Elapsed Time
Pages Printed	Job ID	Priority	Driver	Print
Parameters	Host	Print Queue	Data Type	Name

[Network Connections]

Local Name	Remote Name	Type
Status	User Name	
W:	\\inforb\mount	Disk Persistent

[Running Tasks]

Name	Path	Process ID	Priority	Min
Working Set	Version	Max Working Set	Start Time	
system	idle	process	Not Available	0
Available	Not Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
system	Not Available	4	8	0
1413120	Not Available	Not Available	Not Available	Not
smss.exe	Not Available	436	11	Not
204800	1413120	5/25/2003 1:44 AM	5.2.3790.0	Not
Available	Not Available	Not Available	Not Available	Not
csrss.exe	Not Available	560	13	Not
Available	Not Available	5/25/2003 1:44 AM	5.2.3790.0	Not
Available	Not Available	Not Available	Not Available	Not
winlogon.exe	c:\windows\system32\winlogon.exe	584	13	204800
5/25/2003 1:44 AM	5.2.3790.0	1413120		
(srv03_rtm.030324-2048)	536.50 KB (549,376 bytes)	3/25/2003 12:00 AM		
services.exe	c:\windows\system32\services.exe	628	9	204800
5/25/2003 1:44 AM	5.2.3790.0	1413120		
(srv03_rtm.030324-2048)	102.00 KB (104,448 bytes)	3/25/2003 12:00 AM		
lsass.exe	c:\windows\system32\lsass.exe	204800	1413120	5/25/2003 1:44 AM
5.2.3790.0 (srv03_rtm.030324-2048)	13.00 KB (13,312 bytes)	3/25/2003 12:00 AM		
12:00 AM				
svchost.exe	c:\windows\system32\svchost.exe	804	8	204800
5/25/2003 1:44 AM	5.2.3790.0	1413120		
(srv03_rtm.030324-2048)	13.00 KB (13,312 bytes)	3/25/2003 12:00 AM		
svchost.exe	c:\windows\system32\svchost.exe	852	8	204800
5/25/2003 1:44 AM	5.2.3790.0	1413120		
(srv03_rtm.030324-2048)	13.00 KB (13,312 bytes)	3/25/2003 12:00 AM		

svchost.exe	Not Available	996	8	
5/25/2003 1:44 AM	Not Available	Not Available	Not	
Available	Not Available			
svchost.exe	Not Available	1044	8	
5/25/2003 1:44 AM	Not Available	Not Available	Not	
Available	Not Available			
svchost.exe	c:\windows\system32\svchost.exe	1056	8	204800
5/25/2003 1:44 AM	5.2.3790.0	1413120		
(srv03_rtm.030324-2048)	13.00 KB (13,312 bytes)	3/25/2003 12:00 AM		
spoolsv.exe	c:\windows\system32\spoolsv.exe	1224	8	204800
5/25/2003 1:44 AM	5.2.3790.0	1413120		
(srv03_rtm.030324-2048)	55.00 KB (56,320 bytes)	3/25/2003 12:00 AM		
msdtc.exe	Not Available	1260	8	Not
Available	Not Available	5/25/2003 1:44 AM	Not	
Available	Not Available	Not Available	Not	
svchost.exe	c:\windows\system32\svchost.exe	1416	8	204800
5/25/2003 1:44 AM	5.2.3790.0	1413120		
(srv03_rtm.030324-2048)	13.00 KB (13,312 bytes)	3/25/2003 12:00 AM		
svchost.exe	Not Available	1484	8	
5/25/2003 1:44 AM	Not Available	Not Available	Not	
Available	Not Available			
dfssvc.exe	c:\windows\system32\dfssvc.exe	1816	8	204800
5/25/2003 1:44 AM	5.2.3790.0	1413120		
(srv03_rtm.030324-2048)	130.50 KB (133,632 bytes)	3/25/2003 12:00 AM		
explorer.exe	c:\windows\explorer.exe	1936	8	204800
5/25/2003 1:44 AM	6.00.3790.0	1413120		
(srv03_rtm.030324-2048)	1,008.50 KB (1,032,704 bytes)	3/25/2003 12:00 AM		
acntusr.exe	c:\program files\altiris\acntusr.exe	2004	8	
1413120	5/25/2003 1:44 AM	5, PM		
6, 0, 50	176.00 KB (180,224 bytes)	5/3/2003 3:48 PM		
cpqteam.exe	c:\windows\system32\cpqteam.exe	2012	8	204800
5/25/2003 1:44 AM	7.20.720.89	1413120		
56.00 KB (57,344 bytes)	11/21/2002			
8:59 AM				
wmiprvse.exe	Not Available	524	8	
5/25/2003 1:45 AM	Not Available	Not Available	Not	
Available	Not Available			
cmd.exe	c:\windows\system32\cmd.exe	1592	8	
204800	1413120	5/25/2003 1:57 AM		
5.2.3790.0 (srv03_rtm.030324-2048)	374.00 KB (382,976 bytes)	3/25/2003		
12:00 AM				
sqlservr.exe	c:\program files\microsoft sql server\mssql\binn\sqlservr.exe	1144	13	
204800	1413120	5/25/2003 1:57 AM		

2000.080.0761.00	7.11 MB (7,450,664 bytes)	5/22/2003 7:49 AM		
helpctr.exe	c:\windows\pchealth\helpctr\binaries\helpctr.exe	1040	8	204800
5/25/2003 1:30 PM	5.2.3790.0	1413120		
(srv03_rtm.030324-2048)	764.00 KB (782,336 bytes)	5/3/2003 3:32 PM		
helpsvc.exe	c:\windows\pchealth\helpctr\binaries\helpsv	516	8	204800
5/25/2003 1:30 PM	5.2.3790.0	1413120		
(srv03_rtm.030324-2048)	720.00 KB (737,280 bytes)	5/3/2003 3:32 PM		
wmiprvse.exe	Not Available	1424	8	
5/25/2003 1:30 PM	Not Available	Not Available	Not	
Available	Not Available			

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer
winlogon	5.2.3790.0 (srv03_rtm.030324-2048)	536.50 KB (549,376 bytes)	3/25/2003	Microsoft Corporation
12:00 AM	c:\windows\system32\winlogon.exe			
ntdll	5.2.3790.0 (srv03_rtm.030324-2048)	722.50 KB (739,840 bytes)	3/25/2003	Microsoft Corporation
12:00 AM	c:\windows\system32\ntdll.dll			
kernel32	5.2.3790.0 (srv03_rtm.030324-2048)	965.00 KB (988,160 bytes)	3/25/2003	Microsoft Corporation
12:00 AM	c:\windows\system32\kernel32.dll			
msvcrt	7.0.3790.0 (srv03_rtm.030324-2048)	319.50 KB (327,168 bytes)	3/25/2003	Microsoft Corporation
12:00 AM	c:\windows\system32\msvcrt.dll			
advapi32	5.2.3790.0 (srv03_rtm.030324-2048)	559.50 KB (572,928 bytes)	3/25/2003	Microsoft Corporation
12:00 AM	c:\windows\system32\advapi32.dll			
rpcrt4	5.2.3790.0 (srv03_rtm.030324-2048)	643.50 KB (658,944 bytes)	3/25/2003	Microsoft Corporation
12:00 AM	c:\windows\system32\rpcrt4.dll			
user32	5.2.3790.0 (srv03_rtm.030324-2048)	562.00 KB (575,488 bytes)	3/25/2003	Microsoft Corporation
12:00 AM	c:\windows\system32\user32.dll			
gdi32	5.2.3790.0 (srv03_rtm.030324-2048)	263.00 KB (269,312 bytes)	3/25/2003	Microsoft Corporation
12:00 AM	c:\windows\system32\gdi32.dll			
userenv	5.2.3790.0 (srv03_rtm.030324-2048)	732.50 KB (750,080 bytes)	3/25/2003	Microsoft Corporation
12:00 AM	c:\windows\system32\userenv.dll			
nddeapi	5.2.3790.0 (srv03_rtm.030324-2048)	16.00 KB (16,384 bytes)	3/25/2003	Microsoft Corporation
12:00 AM	c:\windows\system32\nddeapi.dll			

```

crypt32 5.131.3790.0 (srv03_rtm.030324-2048)
598.00 KB (612,352 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\crypt32.dll
msasn1 5.2.3790.0 (srv03_rtm.030324-2048)
58.00 KB (59,392 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\msasn1.dll
secur32 5.2.3790.0 (srv03_rtm.030324-2048)
63.00 KB (64,512 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\secur32.dll
winsta 5.2.3790.0 (srv03_rtm.030324-2048)
51.00 KB (52,224 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\winsta.dll
netapi32 5.2.3790.0 (srv03_rtm.030324-2048)
317.00 KB (324,608 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\netapi32.dll
profmap 5.2.3790.0 (srv03_rtm.030324-2048)
22.00 KB (22,528 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\profmap.dll
regapi 5.2.3790.0 (srv03_rtm.030324-2048)
48.50 KB (49,664 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\regapi.dll
ws2_32 5.2.3790.0 (srv03_rtm.030324-2048)
87.50 KB (89,600 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ws2_32.dll
ws2help 5.2.3790.0 (srv03_rtm.030324-2048)
19.50 KB (19,968 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ws2help.dll
psapi 5.2.3790.0 (srv03_rtm.030324-2048)
21.50 KB (22,016 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\psapi.dll
version 5.2.3790.0 (srv03_rtm.030324-2048)
17.00 KB (17,408 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\version.dll
setupapi 5.2.3790.0 (srv03_rtm.030324-2048)
1,014.50 KB (1,038,848 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\setupapi.dll
msgina 5.2.3790.0 (srv03_rtm.030324-2048)
1.14 MB (1,191,936 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\msgina.dll
shsvcs 6.00.3790.0 (srv03_rtm.030324-2048)
121.50 KB (124,416 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\shsvcs.dll
shlwapi 6.00.3790.0 (srv03_rtm.030324-2048)
281.00 KB (287,744 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\shlwapi.dll
sfc 5.2.3790.0 (srv03_rtm.030324-2048)
4.50 KB (4,608 bytes) 3/25/2003

```

```

12:00 AM Microsoft Corporation
c:\windows\system32\sfc.dll
sfc_os 5.2.3790.0 (srv03_rtm.030324-2048)
133.00 KB (136,192 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\sfc_os.dll
wintrust 5.131.3790.0 (srv03_rtm.030324-2048)
161.50 KB (165,376 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wintrust.dll
ole32 5.2.3790.0 (srv03_rtm.030324-2048)
1.13 MB (1,187,328 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ole32.dll
imagehlp 5.2.3790.0 (srv03_rtm.030324-2048)
142.50 KB (145,920 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\imagehlp.dll
comctl32 6.0 (srv03_rtm.030324-2048)
907.00 KB (928,768 bytes) 5/3/2003 9:18 AM Microsoft Corporation
c:\windows\winsxs\x86_microsoft.windows.com
mon-controls_6595b64144ccf1df.6.0.100.0_x-
ww_8417450b\comctl32.dll
winscard 5.2.3790.0 (srv03_rtm.030324-2048)
98.60 KB (100,864 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\winscard.dll
wtsapi32 5.2.3790.0 (srv03_rtm.030324-2048)
17.50 KB (17,920 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wtsapi32.dll
sxs 5.2.3790.0 (srv03_rtm.030324-2048)
733.00 KB (750,592 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\sxs.dll
winmm 5.2.3790.0 (srv03_rtm.030324-2048)
166.00 KB (169,984 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\winmm.dll
shell32 6.00.3790.0 (srv03_rtm.030324-2048)
7.79 MB (8,166,400 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\shell32.dll
wldap32 5.2.3790.0 (srv03_rtm.030324-2048)
158.00 KB (161,792 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wldap32.dll
rsaenh 5.2.3790.0 (srv03_rtm.030324-2048)
176.83 KB (181,072 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\rsaenh.dll
csddl 5.2.3790.0 (srv03_rtm.030324-2048)
99.00 KB (101,376 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\csddl.dll
wlnotify 5.2.3790.0 (srv03_rtm.030324-2048)
87.50 KB (89,600 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wlnotify.dll
winspool 5.2.3790.0 (srv03_rtm.030324-2048)
140.00 KB (143,360 bytes) 3/25/2003

```

```

12:00 AM Microsoft Corporation
c:\windows\system32\winspool.drv
mpr 5.2.3790.0 (srv03_rtm.030324-2048)
56.00 KB (57,344 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\mpr.dll
comctl32 5.82 (srv03_rtm.030324-2048)
561.00 KB (574,464 bytes) 5/3/2003 9:18 AM Microsoft Corporation
c:\windows\winsxs\x86_microsoft.windows.com
mon-controls_6595b64144ccf1df.5.82.0.0_x-
ww_8a69ba05\comctl32.dll
uxtheme 6.00.3790.0 (srv03_rtm.030324-2048)
196.00 KB (200,704 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\uxtheme.dll
samlib 5.2.3790.0 (srv03_rtm.030324-2048)
49.00 KB (50,176 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\samlib.dll
cscui 5.2.3790.0 (srv03_rtm.030324-2048)
305.00 KB (312,320 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\cscui.dll
drprov 5.2.3790.0 (srv03_rtm.030324-2048)
12.50 KB (12,800 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\drprov.dll
ntlanman 5.2.3790.0 (srv03_rtm.030324-2048)
41.00 KB (41,984 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ntlanman.dll
netui0 5.2.3790.0 (srv03_rtm.030324-2048)
75.50 KB (77,312 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\netui0.dll
netui1 5.2.3790.0 (srv03_rtm.030324-2048)
184.00 KB (188,416 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\netui1.dll
davclnt 5.2.3790.0 (srv03_rtm.030324-2048)
23.50 KB (24,064 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\davclnt.dll
mprui 5.2.3790.0 (srv03_rtm.030324-2048)
49.00 KB (50,176 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\mprui.dll
netui2 5.2.3790.0 (srv03_rtm.030324-2048)
309.50 KB (316,928 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\netui2.dll
comdlg32 6.00.3790.0 (srv03_rtm.030324-2048)
261.00 KB (267,264 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\comdlg32.dll
netmsg 5.2.3790.0 (srv03_rtm.030324-2048)
178.00 KB (182,272 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\netmsg.dll
oleaut32 5.2.3790.0
486.00 KB (497,664 bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows\system32\oleaut32.dll

```

```

clbcatq 2001.12.4720.0 (srv03_rtm.030324-2048)
481.00 KB (492,544 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\clbcatq.dll
comres 2001.12.4720.0 (srv03_rtm.030324-2048)
778.00 KB (796,672 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\comres.dll
ntmarta 5.2.3790.0 (srv03_rtm.030324-2048)
114.00 KB (116,736 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ntmarta.dll
wbemprox 5.2.3790.0 (srv03_rtm.030324-2048)
17.50 KB (17,920 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wbem\wbemprox.dll
wbemcomn 5.2.3790.0 (srv03_rtm.030324-2048)
211.50 KB (216,576 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wbem\wbemcomn.dll
wbemsvc 5.2.3790.0 (srv03_rtm.030324-2048)
42.50 KB (43,520 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wbem\wbemsvc.dll
fastprox 5.2.3790.0 (srv03_rtm.030324-2048)
443.00 KB (453,632 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wbem\fastprox.dll
msvcpc60 6.05.2144.0 388.00 KB (397,312
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows\system32\msvcpc60.dll
ntdsapi 5.2.3790.0 (srv03_rtm.030324-2048)
76.00 KB (77,824 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ntdsapi.dll
dnsapi 5.2.3790.0 (srv03_rtm.030324-2048)
147.50 KB (151,040 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\dnsapi.dll
services 5.2.3790.0 (srv03_rtm.030324-2048)
102.00 KB (104,448 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\services.exe
scserv 5.2.3790.0 (srv03_rtm.030324-2048)
316.50 KB (324,096 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\scserv.dll
authz 5.2.3790.0 (srv03_rtm.030324-2048)
67.00 KB (68,608 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\authz.dll
umpnpmgr 5.2.3790.0 (srv03_rtm.030324-2048)
121.50 KB (124,416 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\umpnpmgr.dll
ncobjapi 5.2.3790.0 (srv03_rtm.030324-2048)
34.50 KB (35,328 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ncobjapi.dll
eventlog 5.2.3790.0 (srv03_rtm.030324-2048)
60.50 KB (61,952 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\eventlog.dll

```

```

cryptnet 5.131.3790.0 (srv03_rtm.030324-2048)
59.50 KB (60,928 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\cryptnet.dll
sensapi 5.2.3790.0 (srv03_rtm.030324-2048)
6.00 KB (6,144 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\sensapi.dll
cabinet 5.2.3790.0 (srv03_rtm.030324-2048)
61.00 KB (62,464 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\cabinet.dll
imm32 5.2.3790.0 (srv03_rtm.030324-2048)
105.50 KB (108,032 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\imm32.dll
apphelp 5.2.3790.0 (srv03_rtm.030324-2048)
122.00 KB (124,928 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\apphelp.dll
lsass 5.2.3790.0 (srv03_rtm.030324-2048)
13.00 KB (13,312 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\lsass.exe
lsasrv 5.2.3790.0 (srv03_rtm.030324-2048)
780.50 KB (799,232 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\lsasrv.dll
samsrv 5.2.3790.0 (srv03_rtm.030324-2048)
452.00 KB (462,848 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\samsrv.dll
cryptdll 5.2.3790.0 (srv03_rtm.030324-2048)
34.00 KB (34,816 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\cryptdll.dll
msprivs 5.2.3790.0 (srv03_rtm.030324-2048)
46.50 KB (47,616 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\msprivs.dll
kerberos 5.2.3790.0 (srv03_rtm.030324-2048)
332.50 KB (340,480 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\kerberos.dll
msv1_0 5.2.3790.0 (srv03_rtm.030324-2048)
127.00 KB (130,048 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\msv1_0.dll
netlogon 5.2.3790.0 (srv03_rtm.030324-2048)
409.00 KB (418,816 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\netlogon.dll
w32time 5.2.3790.0 (srv03_rtm.030324-2048)
216.00 KB (221,184 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\w32time.dll
iphlpapi 5.2.3790.0 (srv03_rtm.030324-2048)
82.50 KB (84,480 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\iphlpapi.dll
schannel 5.2.3790.0 (srv03_rtm.030324-2048)
149.50 KB (153,088 bytes) 3/25/2003

```

```

12:00 AM Microsoft Corporation
c:\windows\system32\schannel.dll
wdigest 5.2.3790.0 (srv03_rtm.030324-2048)
61.00 KB (62,464 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wdigest.dll
rassfm 5.2.3790.0 (srv03_rtm.030324-2048)
20.50 KB (20,992 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\rassfm.dll
kdcsvc 5.2.3790.0 (srv03_rtm.030324-2048)
221.00 KB (226,304 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\kdcsvc.dll
ntdsa 5.2.3790.0 (srv03_rtm.030324-2048)
1.45 MB (1,520,640 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ntdsa.dll
ntdsatq 5.2.3790.0 (srv03_rtm.030324-2048)
32.00 KB (32,768 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ntdsatq.dll
mswsock 5.2.3790.0 (srv03_rtm.030324-2048)
254.00 KB (260,096 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\mswsock.dll
esent 5.2.3790.0 (srv03_rtm.030324-2048)
1.01 MB (1,056,256 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\esent.dll
scecli 5.2.3790.0 (srv03_rtm.030324-2048)
179.50 KB (183,808 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\scecli.dll
wshtcpip 5.2.3790.0 (srv03_rtm.030324-2048)
18.00 KB (18,432 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wshtcpip.dll
ipsecsvc 5.2.3790.0 (srv03_rtm.030324-2048)
162.50 KB (166,400 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ipsecsvc.dll
oakley 5.2.3790.0 (srv03_rtm.030324-2048)
325.50 KB (333,312 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\oakley.dll
winipsec 5.2.3790.0 (srv03_rtm.030324-2048)
34.50 KB (35,328 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\winipsec.dll
pstorsvc 5.2.3790.0 (srv03_rtm.030324-2048)
24.00 KB (24,576 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\pstorsvc.dll
psbase 5.2.3790.0 (srv03_rtm.030324-2048)
81.00 KB (82,944 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\psbase.dll
dssenh 5.2.3790.0 (srv03_rtm.030324-2048)
131.33 KB (134,480 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\dssenh.dll

```

```

wlbctrl 5.2.3790.0 (srv03_rtm.030324-2048)
78.00 KB (79,872 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wlbctrl.dll
svchost 5.2.3790.0 (srv03_rtm.030324-2048)
13.00 KB (13,312 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\svchost.exe
rpcss 5.2.3790.0 (srv03_rtm.030324-2048)
276.50 KB (283,136 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\rpcss.dll
termsrv 5.2.3790.0 (srv03_rtm.030324-2048)
216.50 KB (221,696 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\termsrv.dll
icaapi 5.2.3790.0 (srv03_rtm.030324-2048)
10.50 KB (10,752 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\icaapi.dll
mstlsapi 5.2.3790.0 (srv03_rtm.030324-2048)
104.50 KB (107,008 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\mstlsapi.dll
activeds 5.2.3790.0 (srv03_rtm.030324-2048)
189.00 KB (193,536 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\activeds.dll
adslsdp 5.2.3790.0 (srv03_rtm.030324-2048)
142.50 KB (145,920 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\adslsdp.dll
credui 5.2.3790.0 (srv03_rtm.030324-2048)
159.00 KB (162,816 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\credui.dll
atl 3.05.2283 83.00 KB (84,992 bytes)
3/25/2003 12:00 AM Microsoft Corporation
c:\windows\system32\atl.dll
rdpwsx 5.2.3790.0 (srv03_rtm.030324-2048)
80.13 KB (82,056 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\rdpwsx.dll
wzcsvc 5.2.3790.0 (srv03_rtm.030324-2048)
272.50 KB (279,040 bytes) 3/25/2003
6:15 AM Microsoft Corporation
c:\windows\system32\wzcsvc.dll
rtutils 5.2.3790.0 (srv03_rtm.030324-2048)
32.00 KB (32,768 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\rtutils.dll
wmi 5.2.3790.0 (srv03_rtm.030324-2048)
6.50 KB (6,656 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wmi.dll
dhcpcsvc 5.2.3790.0 (srv03_rtm.030324-2048)
101.50 KB (103,936 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\dhcpcsvc.dll
rastls 5.2.3790.0 (srv03_rtm.030324-2048)
155.00 KB (158,720 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\rastls.dll

```

```

cryptui 5.131.3790.0 (srv03_rtm.030324-2048)
473.50 KB (484,864 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\cryptui.dll
mprapi 5.2.3790.0 (srv03_rtm.030324-2048)
81.00 KB (82,944 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\mprapi.dll
rasapi32 5.2.3790.0 (srv03_rtm.030324-2048)
227.50 KB (232,960 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\rasapi32.dll
rasman 5.2.3790.0 (srv03_rtm.030324-2048)
56.50 KB (57,856 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\rasman.dll
tapi32 5.2.3790.0 (srv03_rtm.030324-2048)
175.00 KB (179,200 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\tapi32.dll
raschap 5.2.3790.0 (srv03_rtm.030324-2048)
106.00 KB (108,544 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\raschap.dll
schedsvc 5.2.3790.0 (srv03_rtm.030324-2048)
176.00 KB (180,224 bytes) 5/3/2003 3:32
PM Microsoft Corporation
c:\windows\system32\schedsvc.dll
msidle 6.00.3790.0 (srv03_rtm.030324-2048)
5.50 KB (5,632 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\msidle.dll
wkssvc 5.2.3790.0 (srv03_rtm.030324-2048)
125.00 KB (128,000 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wkssvc.dll
wiarpc 5.2.3790.0 (srv03_rtm.030324-2048)
30.00 KB (30,720 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wiarpc.dll
cryptsvc 5.2.3790.0 (srv03_rtm.030324-2048)
51.00 KB (52,224 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\cryptsvc.dll
certcli 5.2.3790.0 (srv03_rtm.030324-2048)
228.00 KB (233,472 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\certcli.dll
vssapi 5.2.3790.0 (srv03_rtm.030324-2048)
528.00 KB (540,672 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\vssapi.dll
dmserver 5.2.3790.0 (srv03_rtm.030324-2048)
24.00 KB (24,576 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\dmserver.dll
es 2001.12.4720.0 (srv03_rtm.030324-2048)
221.50 KB (226,816 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\es.dll
pchsvc 5.2.3790.0 (srv03_rtm.030324-2048)
31.50 KB (32,256 bytes) 5/3/2003 3:32
PM Microsoft Corporation

```

```

c:\windows\pchealth\helpctr\binaries\pchsvc
.dll
srsvsc 5.2.3790.0 (srv03_rtm.030324-2048)
89.00 KB (91,136 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\srsvsc.dll
seclogon 5.2.3790.0 (srv03_rtm.030324-2048)
16.50 KB (16,896 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\seclogon.dll
sens 5.2.3790.0 (srv03_rtm.030324-2048)
35.50 KB (36,352 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\sens.dll
trkwks 5.2.3790.0 (srv03_rtm.030324-2048)
85.00 KB (87,040 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\trkwks.dll
wmisvc 5.2.3790.0 (srv03_rtm.030324-2048)
131.00 KB (134,144 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wbem\wmisvc.dll
wuauerv 5.4.3790.0 (srv03_rtm.030324-2048)
10.50 KB (10,752 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wuauerv.dll
wuaueng 5.4.3790.0 (srv03_rtm.030324-2048)
188.50 KB (193,024 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wuaueng.dll
advpack 6.00.3790.0 (srv03_rtm.030324-2048)
93.50 KB (95,744 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\advpack.dll
wininet 6.00.3790.0 (srv03_rtm.030324-2048)
609.00 KB (623,616 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wininet.dll
winrnr 5.2.3790.0 (srv03_rtm.030324-2048)
15.00 KB (15,360 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\winrnr.dll
comsvcs 2001.12.4720.0 (srv03_rtm.030324-2048)
1.14 MB (1,199,616 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\comsvcs.dll
rasadhlp 5.2.3790.0 (srv03_rtm.030324-2048)
6.50 KB (6,656 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\rasadhlp.dll
browser 5.2.3790.0 (srv03_rtm.030324-2048)
70.50 KB (72,192 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\browser.dll
netrap 5.2.3790.0 (srv03_rtm.030324-2048)
11.50 KB (11,776 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\netrap.dll
netman 5.2.3790.0 (srv03_rtm.030324-2048)
209.00 KB (214,016 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\netman.dll

```

```

wzcsapi 5.2.3790.0 (srv03_rtm.030324-2048)
24.50 KB (25,088 bytes) 3/25/2003
6:15 AM Microsoft Corporation
c:\windows\system32\wzcsapi.dll
netshell 5.2.3790.0 (srv03_rtm.030324-2048)
1.67 MB (1,747,456 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\netshell.dll
clusapi 5.2.3790.0 (srv03_rtm.030324-2048)
56.00 KB (57,344 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\clusapi.dll
hnetcfg 5.2.3790.0 (srv03_rtm.030324-2048)
243.50 KB (249,344 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\hnetcfg.dll
wbemcore 5.2.3790.0 (srv03_rtm.030324-2048)
457.00 KB (467,968 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wbem\wbemcore.dll
esscli 5.2.3790.0 (srv03_rtm.030324-2048)
235.50 KB (241,152 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wbem\esscli.dll
wmiutils 5.2.3790.0 (srv03_rtm.030324-2048)
90.50 KB (92,672 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wbem\wmiutils.dll
repdrvfs 5.2.3790.0 (srv03_rtm.030324-2048)
165.00 KB (168,960 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wbem\repdrvfs.dll
wmiprvsd 5.2.3790.0 (srv03_rtm.030324-2048)
405.50 KB (415,232 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wbem\wmiprvsd.dll
wbemess 5.2.3790.0 (srv03_rtm.030324-2048)
256.50 KB (262,656 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wbem\wbemess.dll
rasdlg 5.2.3790.0 (srv03_rtm.030324-2048)
642.00 KB (657,408 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\rasdlg.dll
winhttp 5.2.3790.0 (srv03_rtm.030324-2048)
327.50 KB (335,360 bytes) 5/3/2003 9:18
AM Microsoft Corporation
c:\windows\winsxs\x86_microsoft.windows.win
http_6595b64144ccf1df_5.1.0.0_x-
ww_e0651936\winhttp.dll
ncprov 5.2.3790.0 (srv03_rtm.030324-2048)
43.00 KB (44,032 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\wbem\ncprov.dll
spoolsv 5.2.3790.0 (srv03_rtm.030324-2048)
55.00 KB (56,320 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\spoolsv.exe
spoolss 5.2.3790.0 (srv03_rtm.030324-2048)
79.00 KB (80,896 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\spoolss.dll

```

```

localspl 5.2.3790.0 (srv03_rtm.030324-2048)
304.50 KB (311,808 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\localspl.dll
cnbjmon 5.2.3680.0 (Lab03_dev(skatar).020509-1043)
45.50 KB (46,592 bytes) 3/24/2003
7:48 PM Microsoft Corporation
c:\windows\system32\cnbjmon.dll
pjlmmon 5.2.3790.0 (srv03_rtm.030324-2048)
15.00 KB (15,360 bytes) 3/24/2003
7:49 PM Microsoft Corporation
c:\windows\system32\pjlmmon.dll
tcpmon 5.2.3790.0 (srv03_rtm.030324-2048)
44.00 KB (45,056 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\tcpmon.dll
mgmtapi 5.2.3790.0 (srv03_rtm.030324-2048)
14.00 KB (14,336 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\mgmtapi.dll
snmpapi 5.2.3790.0 (srv03_rtm.030324-2048)
17.50 KB (17,920 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\snmpapi.dll
wsnmp32 5.2.3790.0 (srv03_rtm.030324-2048)
39.50 KB (40,448 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wsnmp32.dll
usbmon 5.2.3790.0 (srv03_rtm.030324-2048)
17.00 KB (17,408 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\usbmon.dll
wshqos 5.2.3790.0 (srv03_rtm.030324-2048)
23.00 KB (23,552 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wshqos.dll
win32spl 5.2.3790.0 (srv03_rtm.030324-2048)
94.50 KB (96,768 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\win32spl.dll
inetpp 5.2.3790.0 (srv03_rtm.030324-2048)
71.50 KB (73,216 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\inetpp.dll
icmp 5.2.3790.0 (srv03_rtm.030324-2048)
4.50 KB (4,608 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\icmp.dll
ersvc 5.2.3790.0 (srv03_rtm.030324-2048)
22.00 KB (22,528 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ersvc.dll
dfssvc 5.2.3790.0 (srv03_rtm.030324-2048)
130.50 KB (133,632 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\dfssvc.exe
resutils 5.2.3790.0 (srv03_rtm.030324-2048)
59.00 KB (60,416 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\resutils.dll
mfc42u 6.05.3014.0 960.00 KB (983,040
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows\system32\mfc42u.dll

```

```

wsock32 5.2.3790.0 (srv03_rtm.030324-2048)
22.00 KB (22,528 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\wsock32.dll
explorer 6.00.3790.0 (srv03_rtm.030324-2048)
1,008.50 KB (1,032,704 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\explorer.exe
browseui 6.00.3790.0 (srv03_rtm.030324-2048)
1.01 MB (1,057,280 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\browseui.dll
shdocvw 6.00.3790.0 (srv03_rtm.030324-2048)
1.33 MB (1,393,664 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\shdocvw.dll
themeui 6.00.3790.0 (srv03_rtm.030324-2048)
360.50 KB (369,152 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\themeui.dll
msimg32 5.2.3790.0 (srv03_rtm.030324-2048)
4.50 KB (4,608 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\msimg32.dll
actxprxy 6.00.3790.0 (srv03_rtm.030324-2048)
95.00 KB (97,280 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\actxprxy.dll
linkinfo 5.2.3790.0 (srv03_rtm.030324-2048)
16.50 KB (16,896 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\linkinfo.dll
ntshrui 6.00.3790.0 (srv03_rtm.030324-2048)
136.00 KB (139,264 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\ntshrui.dll
urlmon 6.00.3790.0 (srv03_rtm.030324-2048)
501.50 KB (513,536 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\urlmon.dll
webcheck 6.00.3790.0 (srv03_rtm.030324-2048)
261.50 KB (267,776 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\webcheck.dll
stobject 5.2.3790.0 (srv03_rtm.030324-2048)
117.50 KB (120,320 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\stobject.dll
batmeter 6.00.3790.0 (srv03_rtm.030324-2048)
28.50 KB (29,184 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\batmeter.dll
powrprof 6.00.3790.0 (srv03_rtm.030324-2048)
14.50 KB (14,848 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\powrprof.dll
printui 5.2.3790.0 (srv03_rtm.030324-2048)
536.50 KB (549,376 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\printui.dll
cfgmgr32 5.2.3790.0 (srv03_rtm.030324-2048)
17.50 KB (17,920 bytes) 3/25/2003

```



```

12:00 AM Microsoft Corporation
c:\windows\system32\cfgmgr32.dll
browsec 6.00.3790.0 (srv03_rtm.030324-2048)
62.00 KB (63,488 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\browsec.dll
duser 5.2.3790.0 (srv03_rtm.030324-2048)
270.00 KB (276,480 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\duser.dll
shdoclc 6.00.3790.0 (srv03_rtm.030324-2048)
588.50 KB (602,624 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\shdoclc.dll
zipfldr 6.00.3790.0 (srv03_rtm.030324-2048)
316.00 KB (323,584 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\zipfldr.dll
mydocs 6.00.3790.0 (srv03_rtm.030324-2048)
88.00 KB (90,112 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\mydocs.dll
shimgvw 6.00.3790.0 (srv03_rtm.030324-2048)
418.00 KB (428,032 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\shimgvw.dll
gdiplus 5.2.3790.0 (srv03_rtm.030324-2048)
1.67 MB (1,748,992 bytes) 5/3/2003 9:18
AM Microsoft Corporation
c:\windows\winsxs\x86_microsoft.windows.gdi
plus_6595b64144ccf1df.1.0.100.0_x-
ww_0d1f9f94\gdiplus.dll
diskcopy 6.00.3790.0 (srv03_rtm.030324-2048)
1.43 MB (1,503,744 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\diskcopy.dll
acIntusr 5, 6, 0, 50 176.00 KB (180,224
bytes) 5/3/2003 3:48 PM c:\program
files\altiris\acIntusr.exe
cpqteam 7.20.720.89 56.00 KB (57,344 bytes)
11/21/2002 8:59 AM Hewlett-Packard Company
c:\windows\system32\cpqteam.exe
cmd 5.2.3790.0 (srv03_rtm.030324-2048)
374.00 KB (382,976 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\cmd.exe
sqlservr 2000.080.0761.00 7.11 MB (7,450,664
bytes) 5/22/2003 7:49 AM Microsoft Corporation
c:\program files\microsoft sql
server\mssql\bin\sqlservr.exe
opends60 2000.080.0194.00 24.06 KB (24,639 bytes)
5/22/2003 7:49 AM Microsoft Corporation
c:\program files\microsoft sql
server\mssql\bin\opends60.dll
ums 2000.080.0760.00 52.55 KB (53,808 bytes)
5/22/2003 7:49 AM Microsoft Corporation
c:\program files\microsoft sql
server\mssql\bin\ums.dll
sqlsort 2000.080.0760.00 576.56 KB (590,396
bytes) 5/22/2003 7:49 AM Microsoft Corporation
c:\program files\microsoft sql
server\mssql\bin\sqlsort.dll

```

```

msvcirt 7.0.3790.0 (srv03_rtm.030324-2048)
50.00 KB (51,200 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\msvcirt.dll
sqllevn70 2000.080.0760.00 28.00 KB (28,672 bytes)
5/22/2003 7:49 AM Microsoft Corporation
c:\program files\microsoft sql
server\mssql\bin\resources\1033\sqllevn70.rll
xolehlp 2001.12.4720.0 (srv03_rtm.030324-2048)
8.50 KB (8,704 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\xolehlp.dll
msdtcprx 2001.12.4720.0 (srv03_rtm.030324-2048)
427.50 KB (437,760 bytes) 5/3/2003 3:28
PM Microsoft Corporation
c:\windows\system32\msdtcprx.dll
mtxclu 2001.12.4720.0 (srv03_rtm.030324-2048)
74.50 KB (76,288 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\mtxclu.dll
ssnetlib 2000.080.0760.00 80.56 KB (82,492 bytes)
5/22/2003 7:49 AM Microsoft Corporation
c:\program files\microsoft sql
server\mssql\bin\ssnetlib.dll
ssnmpn70 2000.080.0534.00 24.56 KB (25,148 bytes)
5/22/2003 7:49 AM Microsoft Corporation
c:\program files\microsoft sql
server\mssql\bin\ssnmpn70.dll
security 5.2.3790.0 (srv03_rtm.030324-2048)
5.50 KB (5,632 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\security.dll
ssmslpcn 2000.080.0760.00 28.56 KB (29,244 bytes)
5/22/2003 7:49 AM Microsoft Corporation
c:\program files\microsoft sql
server\mssql\bin\ssmslpcn.dll
sqloledb 2000.085.1022.00 (srv03_rtm.030324-2048)
536.00 KB (548,864 bytes) 5/3/2003 3:32
PM Microsoft Corporation c:\program
files\common files\system\ole db\sqloledb.dll
msdart 2.80.1022.0 (srv03_rtm.030324-2048)
164.00 KB (167,936 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\msdart.dll
msdatl3 2.80.1022.0 (srv03_rtm.030324-2048)
96.00 KB (98,304 bytes) 5/3/2003 3:32
PM Microsoft Corporation c:\program
files\common files\system\ole db\msdatl3.dll
oledb32 2.80.1022.0 (srv03_rtm.030324-2048)
500.00 KB (512,000 bytes) 5/3/2003 3:32
PM Microsoft Corporation c:\program
files\common files\system\ole db\oledb32.dll
oledb32r 2.80.1022.0 (srv03_rtm.030324-2048)
68.00 KB (69,632 bytes) 5/3/2003 3:32
PM Microsoft Corporation c:\program
files\common files\system\ole db\oledb32r.dll
helpctr 5.2.3790.0 (srv03_rtm.030324-2048)
764.00 KB (782,336 bytes) 5/3/2003 3:32
PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\helpctr
.exe
hcappres 5.2.3790.0 (srv03_rtm.030324-2048)
6.50 KB (6,656 bytes) 5/3/2003 3:32

```

```

PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\hcappre
es.dll
itss 5.2.3790.0 (srv03_rtm.030324-2048)
119.50 KB (122,368 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\itss.dll
msxml3 8.40.9419.0 1.28 MB (1,337,344
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows\system32\msxml3.dll
pchshell 5.2.3790.0 (srv03_rtm.030324-2048)
100.50 KB (102,912 bytes) 5/3/2003 3:32
PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\pchshe
ll.dll
mlang 6.00.3790.0 (srv03_rtm.030324-2048)
570.00 KB (583,680 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\mlang.dll
mshtml 6.00.3790.0 (srv03_rtm.030324-2048)
2.78 MB (2,916,352 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\mshtml.dll
msimtf 5.2.3790.0 (srv03_rtm.030324-2048)
149.00 KB (152,576 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\msimtf.dll
msctf 5.2.3790.0 (srv03_rtm.030324-2048)
287.00 KB (293,888 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\msctf.dll
jscript 5.6.0.8515 436.00 KB (446,464
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows\system32\jscript.dll
msls31 3.10.349.0 147.00 KB (150,528
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows\system32\msls31.dll
mshtml 6.00.3790.0 (srv03_rtm.030324-2048)
443.50 KB (454,144 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\mshtml.dll
vbscript 5.6.0.8515 404.00 KB (413,696
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows\system32\vbscript.dll
mfc42 6.05.3014.0 960.00 KB (983,040
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows\system32\mfc42.dll
msinfo 5.2.3790.0 (srv03_rtm.030324-2048)
358.50 KB (367,104 bytes) 5/3/2003 3:32
PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\msinfo
.dll
riched32 5.2.3790.0 (srv03_rtm.030324-2048)
3.50 KB (3,584 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows\system32\riched32.dll
riched20 5.31.23.1218 406.00 KB (415,744
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows\system32\riched20.dll
helpsvc 5.2.3790.0 (srv03_rtm.030324-2048)
720.00 KB (737,280 bytes) 5/3/2003 3:32
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	
Altiris Client Service	ACLient	Stopped	
files\altiris\aclient\aclient.exe -service	Disabled Own Process	c:\program	
Normal LocalSystem	0		
Alerter	Alerter	Running	Auto
c:\windows\system32\svchost.exe -k	Normal LocalSystem	0	Share Process
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
Windows Audio	AudioSrv	Stopped	Disabled
Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem
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
Indexing Service	CISvc	Stopped	Disabled
Share Process	c:\windows\system32\cisvc.exe	Normal	LocalSystem
ClipBook	ClipSrv	Stopped	Disabled
c:\windows\system32\clipsrv.exe	Normal	LocalSystem	0
COM+ System Application	COMSysApp	Stopped	
Manual Own Process	c:\windows\system32\dlh.exe	/processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}	Normal
Compaq Remote Monitor Service CpqRcmc	Disabled Own Process	c:\windows\system32\cpqrcmc.exe	Normal
Cryptographic Services	CryptSvc	Running	Auto
Auto Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem
Distributed File System	Dfs	Running	Auto
Auto Own Process	c:\windows\system32\dfssvc.exe	Normal	LocalSystem
DHCP Client	Dhcp	Running	Auto
Share Process	c:\windows\system32\svchost.exe -k		

networkservice	Normal	NT	
AUTHORITY\NetworkService	0		
Logical Disk Manager Administrative Service	dmadmin	Stopped	Manual
c:\windows\system32\dmadmin.exe /com	Normal	LocalSystem	0
Logical Disk Manager	dmserver	Running	Auto
Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem
DNS Client	Dnscache	Running	Auto
Share Process	c:\windows\system32\svchost.exe -k	Normal	LocalSystem
networkservice	Normal	NT	
AUTHORITY\NetworkService	0		
Error Reporting Service	ERSvc	Running	Auto
Share Process	c:\windows\system32\svchost.exe -k winerr	Ignore	LocalSystem
Event Log	Eventlog	Running	Auto
Share Process	c:\windows\system32\services.exe	Normal	LocalSystem
COM+ Event System	EventSystem	Running	Manual
Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem
Help and Support	helpsvc	Running	Auto
Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem
Human Interface Device Access HidServ	Disabled	Share Process	Stopped
c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
HTTP SSL	HTTPFilter	Stopped	Manual
Share Process	c:\windows\system32\lsass.exe	Normal	LocalSystem
IMAPI CD-Burning COM Service	ImapiService	Stopped	Disabled
IMAPI CD-Burning COM Service	ImapiService	Stopped	Disabled
Intersite Messaging IsmServ	Stopped	Disabled	Own
Process	c:\windows\system32\ismserv.exe	Normal	LocalSystem
Kerberos Key Distribution Center	kdc	Stopped	Disabled
Share Process	c:\windows\system32\lsass.exe	Normal	LocalSystem
Server	lanmanserver	Running	Auto
Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem
Workstation	lanmanworkstation	Running	Auto
Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem
License Logging	LicenseService	Stopped	Disabled
Own Process	c:\windows\system32\llssrv.exe	Normal	NT
AUTHORITY\NetworkService	0		

TCP/IP NetBIOS Helper	LmHosts	Running	Auto
Share Process	c:\windows\system32\svchost.exe -k	localservice	Normal
AUTHORITY\LocalService	0	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	LocalSystem	0
AUTHORITY\NetworkService	0	Windows Installer	MSIServer
Stopped	Manual	Share Process	c:\windows\system32\msiexec.exe /v
Normal	LocalSystem	0	MSSQLSERVER
MSSQLSERVER	MSSQLSERVER	Stopped	Manual
Own Process	c:\progra-1\microso-1\mssql\bin\sqlservr.exe	Normal	LocalSystem
0	MSSQLServerADHelper	MSSQLServerADHelper	Stopped
Manual	Own Process	c:\program	files\microsoft sql server\80\tools\bin\sqladhl.exe
LocalSystem	0	Network DDE	NetDDE
Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs
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	Running
Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal
LocalSystem	0	File Replication	NtFrs
Stopped	Manual	Own	Process
c:\windows\system32\ntfrs.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	Plug and Play	PlugPlay
Running	Auto	Share Process	c:\windows\system32\services.exe
Normal	LocalSystem	0	

IPSEC Services	PolicyAgent	Running	
Auto	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	RDSSMgr		
Stopped	Manual	Own Process	
c:\windows\system32\sessmgr.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	Running	
Auto	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	LocalSystem	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	Running	Auto
Share Process			
c:\windows\system32\svchost.exe	-k netsvcs		
Normal	LocalSystem	0	
Secondary Logon	seclogon	Running	Auto
Share Process			
c:\windows\system32\svchost.exe	-k netsvcs		
Ignore	LocalSystem	0	

System Event Notification	SENS	Running	
Auto	Share Process		
c:\windows\system32\svchost.exe	-k netsvcs		
Normal	LocalSystem	0	
Internet Connection Firewall (ICF) / Internet	SharedAccess		
Connection Sharing (ICS)	Share Process		
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	Running	Auto
Process			Own
c:\windows\system32\spoolsv.exe			
Normal	LocalSystem	0	
SQLSERVERAGENT	SQLSERVERAGENT	Stopped	
Manual	Own Process		
c:\program-1\micro-1\mssql\binn\sqlagent.ex			
e	Normal	LocalSystem	0
Windows Image Acquisition (WIA)	stisvc		
Stopped	Disabled	Share Process	
c:\windows\system32\svchost.exe	-k imgsvc		
Normal	NT AUTHORITY\LocalService	0	
Surveyor	Surveyor	Stopped	Disabled
c:\compaq\survey\surveyor.exe	Normal		
LocalSystem	0		
Microsoft Software Shadow Copy Provider	swprv		
Stopped	Manual	Own Process	
c:\windows\system32\svchost.exe	-k swprv		
Normal	LocalSystem	0	
HP ProLiant System Shutdown Service	sysdown		
Stopped	Disabled	Own Process	
c:\windows\system32\sysdown.exe			
Normal	LocalSystem	0	
Performance Logs and Alerts	SysmonLog	Stopped	
Manual	Own Process		
c:\windows\system32\smlogsvc.exe			
Normal	NT Authority\NetworkService	0	
Telephony TAPI Srv	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 termsvcs		
Normal	LocalSystem	0	
Themes	Themes	Stopped	Disabled
c:\windows\system32\svchost.exe	-k netsvcs		
Normal	LocalSystem	0	
Telnet	TlntSvr	Stopped	Disabled
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		
Running	Auto	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	
Upload Manager	uploadmgr	Stopped	Manual
Share Process			
c:\windows\system32\svchost.exe	-k netsvcs		
Normal	LocalSystem	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
Process			Own
c:\windows\system32\vssvc.exe	Normal		
LocalSystem	0		
Windows Time	W32Time	Running	Auto
Share Process			
c:\windows\system32\svchost.exe	-k netsvcs		
Normal	LocalSystem	0	
WebClient	WebClient	Stopped	Disabled
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
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	wuauclt	Running	Auto
Share Process			
c:\windows\system32\svchost.exe	-k netsvcs		
Normal	LocalSystem	0	
Wireless Configuration	WZCSVC	Running	
Auto	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

Compaq System Tools All Users:Compaq System Tools All Users

Compaq System Tools\hp Array Configuration Utility All Users:Compaq System Tools\hp Array Configuration Utility All Users

Microsoft SQL Server All Users:Microsoft SQL Server 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 APARK2\Administrator:Accessories APARK2\Administrator

Accessories\Accessibility APARK2\Administrator:Accessories\Accessibility APARK2\Administrator

Accessories\Entertainment APARK2\Administrator:Accessories\Entertainment APARK2\Administrator

Administrative Tools APARK2\Administrator:Administrative Tools APARK2\Administrator

Startup APARK2\Administrator:Startup APARK2\Administrator

[Startup Programs]

Program	Command	User Name	Location
desktop	desktop.ini	NT AUTHORITY\SYSTEM	Startup
desktop	desktop.ini	APARK2\Administrator	Startup
desktop	desktop.ini	.DEFAULT	Startup
desktop	desktop.ini	All Users	Common

ACIntUsr c:\program files\altiris\aclient\acIntusr.exe All Users
HKLM\SOFTWARE\Microsoft\Windows\CurrentVers

ion\Run cpqteam.exe All Users
HKLM\SOFTWARE\Microsoft\Windows\CurrentVers

ion\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
WordPad Document	"%programfiles%\windows nt\accessories\wordpad.exe"
Windows Media Services DRM Storage object	Not Available
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	6.0.3790.0
Build	63790
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
actxprxy.dll	6.0.3790.0	95 KB	3/25/2003 1:00:00 AM	C:\WINDOWS\system32
actxprxy.dll	6.0.3790.0	95 KB	3/25/2003 1:00:00 AM	Microsoft Corporation
advpack.dll	6.0.3790.0	94 KB	3/25/2003 1:00:00 AM	

C:\WINDOWS\system32 Microsoft Corporation

File	Version	Size	Date	Path
advpack.dll	6.0.3790.0	94 KB	3/25/2003 1:00:00 AM	Microsoft Corporation
asctrls.ocx	6.0.3790.0	90 KB	3/25/2003 1:00:00 AM	C:\WINDOWS\system32
asctrls.ocx	6.0.3790.0	90 KB	3/25/2003 1:00:00 AM	Microsoft Corporation
browsecl.dll	6.0.3790.0	62 KB	3/25/2003 1:00:00 AM	C:\WINDOWS\system32
browsecl.dll	6.0.3790.0	62 KB	3/25/2003 1:00:00 AM	Microsoft Corporation
browseui.dll	6.0.3790.0	1,033 KB	3/25/2003 1:00:00 AM	C:\WINDOWS\system32
browseui.dll	6.0.3790.0	1,033 KB	3/25/2003 1:00:00 AM	Microsoft Corporation
cdfview.dll	6.0.3790.0	144 KB	3/25/2003 1:00:00 AM	C:\WINDOWS\system32
cdfview.dll	6.0.3790.0	144 KB	3/25/2003 1:00:00 AM	Microsoft Corporation
comctl32.dll	5.82.3790.0	561 KB	3/25/2003 1:00:00 AM	C:\WINDOWS\system32
comctl32.dll	5.82.3790.0	561 KB	3/25/2003 1:00:00 AM	Microsoft Corporation
dxtmsft.dll	6.3.3790.0	198 KB	3/25/2003 1:00:00 AM	C:\WINDOWS\system32
dxtmsft.dll	6.3.3790.0	198 KB	3/25/2003 1:00:00 AM	Microsoft Corporation
dxtrans.dll	6.3.3790.0	344 KB	3/25/2003 1:00:00 AM	C:\WINDOWS\system32
dxtrans.dll	6.3.3790.0	344 KB	3/25/2003 1:00:00 AM	Microsoft Corporation
iecont.dll	<File Missing>	Not Available	Not Available	Not Available
iecontlc.dll	<File Missing>	Not Available	Not Available	Not Available
iedkcs32.dll	16.0.3790.0	300 KB	3/25/2003 1:00:00 AM	

```

C:\WINDOWS\system32 Microsoft Corporation
iedkcs32.dll      16.0.3790.0      300 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
iepeers.dll       6.0.3790.0        230 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
iepeers.dll       6.0.3790.0        230 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
iesetup.dll       6.0.3790.0        59 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
iesetup.dll       6.0.3790.0        59 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
ieuinit.inf       Not Available      20 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Not Available
ieuinit.inf       Not Available      20 KB
3/25/2003 1:00:00 AM
Available
iexplore.exe      6.0.3790.0        90 KB
3/25/2003 1:00:00 AM
C:\Program
Files\Internet Explorer Microsoft Corporation
imgutil.dll       5.2.3790.0        35 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
imgutil.dll       5.2.3790.0        35 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
inetcpl.cpl       6.0.3790.0        303 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
inetcpl.cpl       6.0.3790.0        303 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
inetcplc.dll      6.0.3790.0        109 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
inetcplc.dll      6.0.3790.0        109 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
inseng.dll        6.0.3790.0        72 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
inseng.dll        6.0.3790.0        72 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
mlang.dll 6.0.3790.0 570 KB 3/25/2003
1:00:00 AM C:\WINDOWS\system32 Microsoft
Corporation
mlang.dll 6.0.3790.0 570 KB 3/25/2003
1:00:00 AM Microsoft Corporation

```

```

msencode.dll      2002.10.4.0      112 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Not Available
msencode.dll      2002.10.4.0      112 KB
3/25/2003 1:00:00 AM
Available
mshta.exe 6.0.3790.0 26 KB 3/25/2003
1:00:00 AM C:\WINDOWS\system32 Microsoft
Corporation
mshta.exe 6.0.3790.0 26 KB 3/25/2003
1:00:00 AM Microsoft Corporation
mshtml.dll        6.0.3790.0        2,848 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
mshtml.dll        6.0.3790.0        2,848 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
mshtml.tlb        6.0.3790.0        1,319 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
mshtml.tlb        6.0.3790.0        1,319 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
mshtml.dll        6.0.3790.0        444 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
mshtml.dll        6.0.3790.0        444 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
mshtml.dll        6.0.3790.0        55 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
mshtml.dll        6.0.3790.0        55 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
msident.dll       6.0.3790.0        47 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
msident.dll       6.0.3790.0        47 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
msident.dll       6.0.3790.0        15 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
msident.dll       6.0.3790.0        15 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
msieftp.dll       6.0.3790.0        230 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
msieftp.dll       6.0.3790.0        230 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
msrating.dll      6.0.3790.0        132 KB
3/25/2003 1:00:00 AM

```

```

C:\WINDOWS\system32 Microsoft Corporation
msrating.dll      6.0.3790.0        132 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
mstime.dll        6.0.3790.0        491 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
mstime.dll        6.0.3790.0        491 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
occache.dll       6.0.3790.0        89 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
occache.dll       6.0.3790.0        89 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
proctexe.ocx      6.3.3790.0        78 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Intel Corporation
proctexe.ocx      6.3.3790.0        78 KB
3/25/2003 1:00:00 AM
Intel Corporation
sendmail.dll      6.0.3790.0        52 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
sendmail.dll      6.0.3790.0        52 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
shdoclc.dll       6.0.3790.0        589 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
shdoclc.dll       6.0.3790.0        589 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
shdocvw.dll       6.0.3790.0        1,361 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
shdocvw.dll       6.0.3790.0        1,361 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
shfolder.dll      6.0.3790.0        23 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
shfolder.dll      6.0.3790.0        23 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
shlwapi.dll       6.0.3790.0        281 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
shlwapi.dll       6.0.3790.0        281 KB
3/25/2003 1:00:00 AM
Microsoft Corporation
tdc.ocx 1.3.0.3130 58 KB 3/25/2003
1:00:00 AM C:\WINDOWS\system32 Microsoft
Corporation

```

```

tdc.ocx 1.3.0.3130 58 KB 3/25/2003
1:00:00 AM . Microsoft Corporation

url.dll 6.0.3790.0 36 KB 3/25/2003
1:00:00 AM C:\WINDOWS\system32 Microsoft
Corporation
url.dll 6.0.3790.0 36 KB 3/25/2003
1:00:00 AM . Microsoft Corporation

urlmon.dll 6.0.3790.0 502 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation

urlmon.dll 6.0.3790.0 502 KB
3/25/2003 1:00:00 AM
. Microsoft Corporation
webcheck.dll 6.0.3790.0 262 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation

webcheck.dll 6.0.3790.0 262 KB
3/25/2003 1:00:00 AM
. Microsoft Corporation
wininet.dll 6.0.3790.0 609 KB
3/25/2003 1:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation

wininet.dll 6.0.3790.0 609 KB
3/25/2003 1:00:00 AM
. Microsoft Corporation

```

[Connectivity]

Item	Value
Connection Preference	Never dial

LAN Settings

```

AutoConfigProxy Not Available
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\NetworkService\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	Medium-low
Trusted sites	Low
Internet Medium	
Restricted sites	High

Client Summary

System Information report written at: 05/02/2003 04:23:21 PM

[System Information]

System Information report written at: 05/25/2003 05:01:52 PM

[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 2 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	CL74

System Manufacturer	HP
System Model	ProLiant DL360 G3
System Type	X86-based PC
Processor x86 Family	15 Model 2 Stepping 7
GenuineIntel ~37426 Mhz	
Processor x86 Family	15 Model 2 Stepping 7
GenuineIntel ~37426 Mhz	
Processor x86 Family	15 Model 2 Stepping 7
GenuineIntel ~37426 Mhz	
Processor x86 Family	15 Model 2 Stepping 7
GenuineIntel ~37426 Mhz	
BIOS Version	02/04/03
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	CL74\Administrator
Time Zone	Central Daylight Time
Total Physical Memory	1,572,372 KB
Available Physical Memory	1,198,756 KB
Total Virtual Memory	3,778,284 KB
Available Virtual Memory	3,173,004 KB
Page File Space	2,205,912 KB
Page File	C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
No conflicted/shared resources	

[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
0x0000-0x0CFF	PCI bus	OK
0x0000-0x0CFF	PCI bus	OK
0x0000-0x0CFF	Direct memory access controller	
OK		
0x03B0-0x03BB	PCI bus	OK
0x03B0-0x03BB	ATI Technologies Inc. RAGE XL PCI	
OK		
0x03C0-0x03DF	PCI bus	OK
0x03C0-0x03DF	ATI Technologies Inc. RAGE XL PCI	
OK		
0x2400-0x24FF	ATI Technologies Inc. RAGE XL PCI	
OK		
0x2800-0x28FF	Smart Array 5i	OK

0x1800-0x18FF HP ProLiant iLO Advanced System Management Controller OK
 0x2C00-0x2CFF HP iLO Management Interface Driver OK
 0x0A79-0x0A79 ISAPNP Read Data Port OK
 0x0279-0x0279 ISAPNP Read Data Port OK
 0x02F4-0x02F7 ISAPNP Read Data Port OK
 0x0F50-0x0F58 Motherboard resources OK
 0x0408-0x040F Motherboard resources OK
 0x0092-0x0092 Motherboard resources OK
 0x0900-0x0903 Motherboard resources OK
 0x0910-0x0911 Motherboard resources OK
 0x0920-0x0923 Motherboard resources OK
 0x0930-0x0937 Motherboard resources OK
 0x0940-0x0947 Motherboard resources OK
 0x0950-0x0957 Motherboard resources OK
 0x0C06-0x0C08 Motherboard resources OK
 0x0C14-0x0C14 Motherboard resources OK
 0x0C49-0x0C4A Motherboard resources OK
 0x0C50-0x0C52 Motherboard resources OK
 0x0C6C-0x0C6F Motherboard resources OK
 0x0010-0x001F Motherboard resources OK
 0x0230-0x0233 Motherboard resources OK
 0x0260-0x0267 Motherboard resources OK
 0x04D0-0x04D1 Motherboard resources OK
 0x0700-0x070F Motherboard resources OK
 0x0800-0x081F Motherboard resources OK
 0x0C80-0x0C83 Motherboard resources OK
 0x0CD4-0x0CD7 Motherboard resources OK
 0x0CF9-0x0CF9 Motherboard resources OK
 0x0020-0x0021 Programmable interrupt controller OK
 0x00A0-0x00A1 Programmable interrupt controller OK
 0x0C00-0x0C01 Programmable interrupt controller OK
 0x0040-0x0043 System timer OK
 0x0080-0x008F Direct memory access controller OK
 0x00C0-0x00DF Direct memory access controller OK
 0x040B-0x040B Direct memory access controller OK
 0x04D6-0x04D6 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
 0x0220-0x0223 Extended IO Bus OK
 0x0240-0x025F Extended IO Bus OK
 0x0070-0x0073 Extended IO Bus OK
 0x03F8-0x03FF Communications Port (COM1) OK
 0x03F2-0x03F5 Standard floppy disk controller OK
 0x03F7-0x03F7 Standard floppy disk controller OK
 0x2000-0x200F 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
31	Smart Array 5i
23	HP ProLiant iLO Advanced System Management Controller
22	HP iLO Management Interface Driver
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
7	Standard OpenHCD USB Host Controller
30	Compaq NC7781 Gigabit Server Adapter #2
29	Compaq NC7781 Gigabit Server Adapter

[Memory]

Range	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	ATI Technologies Inc. RAGE XL PCI	OK
0xF5D00000-0xF6FFFFFF	PCI bus	OK
0xF6000000-0xF6FFFFFF	ATI Technologies Inc. RAGE XL PCI	OK
0xF5FF0000-0xF5FF0FFF	ATI Technologies Inc. RAGE XL PCI	OK
0xF5F80000-0xF5FBFFFF	Smart Array 5i	OK
0xF5DF0000-0xF5DF3FFF	Smart Array 5i	OK
0xF5F70000-0xF5F70FFF	HP ProLiant iLO Advanced System Management Controller	OK
0xF5F60000-0xF5F60FFF	HP iLO Management Interface Driver	OK
0xF5F50000-0xF5F51FFF	HP iLO Management Interface Driver	OK
0xF5E80000-0xF5EFFFFF	HP iLO Management Interface Driver	OK
0xF5E70000-0xF5E70FFF	Standard OpenHCD USB Host Controller	OK
0xF7E00000-0xF7EFFFFF	PCI bus	OK
0xF7EF0000-0xF7EFFFFF	Compaq NC7781 Gigabit Server Adapter #2	OK
0xF7F00000-0xF7FFFFFF	PCI bus	OK
0xF7FF0000-0xF7FFFFFF	Compaq NC7781 Gigabit Server Adapter	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
c:\winnt\system32\tsssoft32.acm	INC.	DSP GROUP, OK		C:\WINNT\System32\TSSOFT32.ACM	1.01	9.27 KB (9,488 bytes)
c:\winnt\system32\lhacm.acm	Microsoft Corporation	OK		C:\WINNT\System32\LHACM.ACM	33.27 KB (34,064 bytes)	4.4.3385 9/13/2002
c:\winnt\system32\msgsm32.acm	Microsoft Corporation	OK		C:\WINNT\System32\MSGSM32.ACM	22.27 KB (22,800 bytes)	5.00.2134.1 12/7/1999
c:\winnt\system32\msadp32.acm	Microsoft Corporation	OK		C:\WINNT\System32\MSADP32.ACM	14.77 KB (15,120 bytes)	5.00.2134.1 12/7/1999
c:\winnt\system32\iac25_32.ax	Intel Corporation	Indeo® audio software	OK	C:\WINNT\System32\IAC25_32.AX	195.00 KB (199,680 bytes)	2.05.53 12/7/1999
c:\winnt\system32\msg711.acm	Microsoft Corporation	OK		C:\WINNT\System32\MSG711.ACM	10.27 KB (10,512 bytes)	5.00.2134.1 12/7/1999
c:\winnt\system32\imaadp32.acm	Microsoft Corporation	OK		C:\WINNT\System32\IMAADP32.ACM	5.00.2134.1	16.27 KB (16,656 bytes) 12/7/1999 7:00:00 AM
c:\winnt\system32\msg723.acm	Microsoft Corporation	OK		C:\WINNT\System32\MSG723.ACM	106.77 KB (109,328 bytes)	4.4.3385 9/13/2002
[Video Codecs]						
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\msh261.drv	Microsoft Corporation	OK		C:\WINNT\System32\MSH261.DRV	163.77 KB (167,696 bytes)	4.4.3385 9/13/2002
c:\winnt\system32\msh263.drv	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\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\msrle32.dll Microsoft Corporation OK
 C:\WINNT\System32\MSRLE32.DLL 5.00.2134.1
 10.77 KB (11,024 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 CRN-8245B
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	Not Available
SCSI Target ID	0
PNP Device ID	IDE\CDROMCOMPAQ_CRN-8245B
	2.19 \5&FB0C83D&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\3&267A616A&0&18
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&35118DFF&0
NumberOfFunctionKeys	12

[Pointing Device]

Item	Value
Hardware Type	PS/2 Compatible Mouse
Number of Buttons	5
Status	OK
PNP Device ID	ACPI\PNP0F13\4&35118DFF&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	5/25/2003 9:37:19 AM
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	Value
Not Available	

Name	Value
[00000001] WAN Miniport (L2TP)	
Adapter Type	Not Available
Product Name	WAN Miniport (L2TP)
Installed True	
PNP Device ID	ROOT\MS_L2TPMINIPORT\0000
Last Reset	5/25/2003 9:37:19 AM
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 (50800, 5.00.2179.1)

Name	Value
[00000002] WAN Miniport (PPTP)	
Adapter Type	Wide Area Network (WAN)
Product Name	WAN Miniport (PPTP)
Installed True	
PNP Device ID	ROOT\MS_PPTPMINIPORT\0000
Last Reset	5/25/2003 9:37:19 AM
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 (47856, 5.00.2160.1)

Name	Value
[00000003] Direct Parallel	
Adapter Type	Not Available
Product Name	Direct Parallel
Installed True	
PNP Device ID	ROOT\MS_PTMINIPORT\0000
Last Reset	5/25/2003 9:37:19 AM
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	Value
[00000004] WAN Miniport (IP)	

Adapter Type Not Available
 Product Name WAN Miniport (IP)
 Installed True
 PNP Device ID ROOT\MS_NDISWANIP\0000
 Last Reset 5/25/2003 9:37:19 AM
 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
 (90096, 5.00.2195.2779)

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 5/25/2003 9:37:19 AM
 Index 5
 Service Name q57w2k
 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 [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 5/25/2003 9:37:19 AM
 Index 6
 Service Name q57w2k
 IP Address 130.172.11.74
 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 00:08:02:A0:2C:B2
 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 5/25/2003 9:37:19 AM
 Index 7
 Service Name N100
 IP Address 130.172.11.74
 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 00:08:02:A0:2C:B2
 Service Name Not Available

 Name [00000008] Compaq NC7781 Gigabit Server
 Adapter
 Adapter Type Ethernet 802.3
 Product Name Compaq NC7781 Gigabit Server
 Adapter
 Installed True
 PNP Device ID PCI\VEN_14E4&DEV_16A7&SUBSYS_00CB0E11&REV_0
 2\3&1070020&0&10
 Last Reset 5/25/2003 9:37:19 AM
 Index 8
 Service Name q57w2k
 IP Address 130.172.11.74
 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 00:08:02:A0:2C:B2
 Service Name q57w2k
 IRQ Number 29
 Driver c:\winnt\system32\drivers\q57w2k.sys
 (79336, 2.90.0.0)

Name [00000009] Compaq NC7781 Gigabit Server
 Adapter
 Adapter Type Ethernet 802.3
 Product Name Compaq NC7781 Gigabit Server
 Adapter
 Installed True
 PNP Device ID PCI\VEN_14E4&DEV_16A7&SUBSYS_00CB0E11&REV_0
 2\3&13C0B0C5&0&10
 Last Reset 5/25/2003 9:37:19 AM
 Index 9
 Service Name q57w2k
 IP Address 130.168.40.74
 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 00:08:02:A0:2C:A1
 Service Name q57w2k
 IRQ Number 30
 Driver c:\winnt\system32\drivers\q57w2k.sys
 (79336, 2.90.0.0)

[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_{B9DB1569-B176-421B-86D8-AC02F735A9F4}] SEQPACKE 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_{B9DB1569-B176-421B-86D8-AC02F735A9F4}] 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_{73E91EA5-C772-4AC3-BB0F-9E3546589E66}] SEQPACKE 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_{73E91EA5-C772-4AC3-BB0F-9E3546589E66}] 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}] SEQPACKE 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_{3B09DD87-7EB8-4941-8121-52DC6359F5A6}] 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_{3B09DD87-7EB8-4941-8121-52DC6359F5A6}] 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}] SEQUENCE 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}] SEQUENCE 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 5.00.2195.2871
 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
 (62416, 5.00.2195.2780)

[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 (36,414,734,336 bytes)
 Free Space 31.58 GB (33,905,115,136 bytes)
 Volume Name
 Volume Serial Number C8B488FA
 Partition Disk #0, Partition #0
 Partition Size 33.91 GB (36,414,734,336 bytes)
 Starting Offset 16384 bytes
 Drive Description Disk drive
 Drive Manufacturer (Standard disk drives)
 Drive Model COMPAQ LOGICAL VOLUME SCSI Disk Device
 Drive BytesPerSector 512
 Drive MediaLoaded True
 Drive MediaType Fixed hard disk media
 Drive Partitions 1
 Drive SCSIbus 0
 Drive SCSILogicalUnit 0
 Drive SCSIPort 2
 Drive SCISITargetID 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 5i
 Caption Smart Array 5i
 Driver cpqcissm
 Status OK
 PNP Device ID
 PCI\VEN_0E11&DEV_B178&SUBSYS_40800E11&REV_01\3&267A616A&0&20
 Device ID
 PCI\VEN_0E11&DEV_B178&SUBSYS_40800E11&REV_01\3&267A616A&0&20
 Device Map Not Available
 Index Not Available
 Max Number Controlled Not Available
 IRQ Number 31
 I/O Port 0x2800-0x28FF
 Driver c:\winnt\system32\drivers\cpqcissm.sys (15568, 5.42.2.32 Build 1)

[Printing]

Name Port Name Server Name
 No printing information

[Problem Devices]

Device PNP Device ID Error Code
 No Problem Devices

[USB]

Device PNP Device ID
 Standard OpenHCD USB Host Controller
 PCI\VEN_1166&DEV_0220&SUBSYS_02201166&REV_05\3&267A616A&0&7A
 USB Root Hub USB\ROOT_HUB\4&AF5358C&0

[Software Environment]

[Following are sub-categories of this main category]

[Drivers]

Name	Description	File	Type
	Started Start Mode	Status Error Control	State 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		
aha154x	Aha154x 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
	Normal	False	OK
asc	asc	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	OK
asc3350p	asc3350p	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	OK
asc3550	asc3550	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	OK
asynmac	RAS Asynchronous Media Driver	c:\winnt\system32\drivers\asynmac.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
	Ignore	False	False
atirage3	atirage3	c:\winnt\system32\drivers\atirapab.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
	Normal	False	False
cd20xrnt	cd20xrnt	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
cdaudio	Cdaudio	c:\winnt\system32\drivers\cdaudio.sys	
	Kernel Driver	False	System
	Stopped OK	Ignore	False
	False		
cdfs	Cdfs	c:\winnt\system32\drivers\cdfs.sys	
	File System Driver	True	Disabled
	Running OK	Normal	False
	True		
cdrom	CD-ROM Driver	c:\winnt\system32\drivers\cdrom.sys	

	Kernel Driver	True	System			Stopped	OK	Normal	False			intelide	IntelIde	Not Available	Kernel Driver
	Running	OK	Normal	False									False	Disabled	Stopped
	True												Normal	False	OK
changer	Changer	Not Available	Kernel Driver		dmio	Logical Disk Manager Driver						ipfilterdriver	IP Traffic Filter Driver		
	False	System	Stopped	OK		c:\winnt\system32\drivers\dmio.sys							c:\winnt\system32\drivers\ipfltdrv.sys		
	Ignore	False	False			Kernel Driver	True	Boot					Kernel Driver	False	Manual
cpqarray	Cpqarray	Not Available	Kernel Driver			Running	OK	Normal	False				Stopped	OK	Normal
	False	Disabled	Stopped	OK		True							False		False
	Normal	False	False		dmload	c:\winnt\system32\drivers\dmload.sys						ipinip	IP in IP Tunnel Driver		
cpqarry2	cpqarry2	Not Available	Kernel Driver			Kernel Driver	True	Boot					c:\winnt\system32\drivers\ipinip.sys		
	False	Disabled	Stopped	OK		Running	OK	Normal	False				Kernel Driver	False	Manual
	Normal	False	False			True							Stopped	OK	Normal
cpqasm2	HP ProLiant iLO Advanced System Management				efs	EFS	c:\winnt\system32\drivers\efs.sys						False		False
Controller						File System Driver	True	Disabled				ipnat	IP Network Address Translator		
	c:\winnt\system32\drivers\cpqasm2.sys					Running	OK	Normal	False				c:\winnt\system32\drivers\ipnat.sys		
	Kernel Driver	True	Manual			True							Kernel Driver	False	Manual
	Running	OK	Normal	False	fastfat	Fastfat							Stopped	OK	Normal
	True					c:\winnt\system32\drivers\fastfat.sys							False		False
cpqcidrv	HP Integrated Lights-Out					File System Driver	True	Disabled				ipsec	IPSEC driver		
	c:\winnt\system32\drivers\cpqcidrv.sys					Running	OK	Normal	False				c:\winnt\system32\drivers\ipsec.sys		
	Kernel Driver	True	Manual			True							Kernel Driver	True	Manual
	Running	OK	Normal	False	fd16_700	Fd16_700	Not Available		Kernel Driver				Running	OK	Normal
	True					False	Disabled	Stopped	OK				True		False
cpqcisse	CPQCISSE					Normal	False	False				ipsraidn	ipsraidn	Not Available	Kernel Driver
	c:\winnt\system32\drivers\cpqcisse.sys					Floppy Disk Controller Driver							False	Disabled	Stopped
	Kernel Driver	True	Manual		fdc	c:\winnt\system32\drivers\fdc.sys							Normal	False	False
	Running	OK	Normal	False		Kernel Driver	True	Manual				isapnp	PnP ISA/EISA Bus Driver		
	True					Running	OK	Normal	False				c:\winnt\system32\drivers\isapnp.sys		
cpqcissm	cpqcissm					True							Kernel Driver	True	Boot
	c:\winnt\system32\drivers\cpqcissm.sys				fips	Fips							Running	OK	Critical
	Kernel Driver	True	Boot			c:\winnt\system32\drivers\fips.sys						kbdclass	Keyboard Class Driver		
	Running	OK	Normal	False		Kernel Driver	True	Auto					c:\winnt\system32\drivers\kbdclass.sys		
	True					Running	OK	Normal	False				Kernel Driver	True	System
cpqfcalm	cpqfcalm	Not Available	Kernel Driver		fireport	fireport	Not Available		Kernel Driver				Running	OK	Normal
	False	Disabled	Stopped	OK		False	Disabled	Stopped	OK			ksecdd	KSecDD		
	Normal	False	False			Normal	False	False					c:\winnt\system32\drivers\ksecdd.sys		
cpqfws2e	cpqfws2e	Not Available	Kernel Driver		flashpnt	flashpnt	Not Available		Kernel Driver				Kernel Driver	True	Boot
	False	Disabled	Stopped	OK		False	Disabled	Stopped	OK				Running	OK	Normal
	Normal	False	False			Normal	False	False					True		False
dac960nt	dac960nt	Not Available	Kernel Driver		flpydisk	Floppy Disk Driver						lbrtfdc	lbrtfdc	Not Available	Kernel Driver
	False	Disabled	Stopped	OK		c:\winnt\system32\drivers\flpydisk.sys							False	System	Stopped
	Normal	False	False			Kernel Driver	True	Manual					Ignore	False	False
deckzpsx	deckzpsx	Not Available	Kernel Driver			Running	OK	Normal	False			lp6nds35	lp6nds35	Not Available	Kernel Driver
	False	Disabled	Stopped	OK	ftdisk	Volume Manager Driver							False	Disabled	Stopped
	Normal	False	False			c:\winnt\system32\drivers\ftdisk.sys							Normal	False	False
dfsdriver	DfsDriver	c:\winnt\system32\drivers\dfs.sys				Kernel Driver	True	Boot				mnmd	mnmd		
	File System Driver	True	Boot			Running	OK	Normal	False				c:\winnt\system32\drivers\mnmd.sys		
	Running	OK	Normal	False		True							Kernel Driver	True	System
	True				gpc	Generic Packet Classifier							Running	OK	Ignore
disk	Disk Driver					c:\winnt\system32\drivers\msgpc.sys						modem	Modem		
	c:\winnt\system32\drivers\disk.sys					Kernel Driver	True	Manual					c:\winnt\system32\drivers\modem.sys		
	Kernel Driver	True	Boot			Running	OK	Normal	False				Kernel Driver	False	Manual
	Running	OK	Normal	False		True							Stopped	OK	Ignore
diskperf	Diskperf	c:\winnt\system32\drivers\diskperf.sys			i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver							False		False
	c:\winnt\system32\drivers\diskperf.sys					c:\winnt\system32\drivers\i8042prt.sys						mouclass	Mouse Class Driver		
	Kernel Driver	True	Boot			Kernel Driver	True	System					c:\winnt\system32\drivers\mouclass.sys		
	Running	OK	Normal	False		Running	OK	Normal	False				Kernel Driver	True	System
	True				ini910u	ini910u	Not Available		Kernel Driver				Running	OK	Normal
dmboot	dmboot					False	Disabled	Stopped	OK				True		False
	c:\winnt\system32\drivers\dmboot.sys					Normal	False	False							
	Kernel Driver	False	Disabled												

mountmgr	MountMgr	c:\winnt\system32\drivers\mountmgr.sys	Kernel Driver	True	Boot	Running	OK	Normal	False
mraid35x	mraid35x	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
mrxsmbr	MRXSMB	c:\winnt\system32\drivers\mrxsmbr.sys	File System Driver	True	System	Running	OK	Normal	False
msfs	Msfs	c:\winnt\system32\drivers\msfs.sys	File System Driver	True	System	Running	OK	Normal	False
mksrsv	Microsoft Streaming Service Proxy	c:\winnt\system32\drivers\mksrsv.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	False
mspclock	Microsoft Streaming Clock Proxy	c:\winnt\system32\drivers\mspclock.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	False
mspqm	Microsoft Streaming Quality Manager Proxy	c:\winnt\system32\drivers\mspqm.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	False
mup	Mup	c:\winnt\system32\drivers\mup.sys	File System Driver	True	Boot	Running	OK	Normal	False
n100 Driver	Compaq Ethernet or Fast Ethernet NIC NT	c:\winnt\system32\drivers\n100nt5.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	False
ncrc710	Ncr710	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
ndis	NDIS System Driver	c:\winnt\system32\drivers\ndis.sys	Kernel Driver	True	Boot	Running	OK	Normal	False
ndistapi	Remote Access NDIS TAPI Driver	c:\winnt\system32\drivers\ndistapi.sys	Kernel Driver	True	Manual	Running	OK	Normal	False
ndiswan	Remote Access NDIS WAN Driver	c:\winnt\system32\drivers\ndiswan.sys	Kernel Driver	True	Manual	Running	OK	Normal	False
ndproxy	NDIS Proxy	c:\winnt\system32\drivers\ndproxy.sys	Kernel Driver	True	Manual				

netbios	NetBIOS Interface	c:\winnt\system32\drivers\netbios.sys	File System Driver	True	System	Running	OK	Normal	False
netbt	NetBios over Tcpip	c:\winnt\system32\drivers\netbt.sys	Kernel Driver	True	System	Running	OK	Normal	False
netdetect	NetDetect	c:\winnt\system32\drivers\netdetect.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	False
npfs	Npfs	c:\winnt\system32\drivers\npfs.sys	File System Driver	True	System	Running	OK	Normal	False
ntfs	Ntfs	c:\winnt\system32\drivers\ntfs.sys	File System Driver	True	Disabled	Running	OK	Normal	False
null	Null	c:\winnt\system32\drivers\null.sys	Kernel Driver	True	System	Running	OK	Normal	False
nwlkflt	IPX Traffic Filter Driver	c:\winnt\system32\drivers\nwlkflt.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	False
nwlkfwd	IPX Traffic Forwarder Driver	c:\winnt\system32\drivers\nwlkfwd.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	False
openhci	Microsoft USB Open Host Controller Driver	c:\winnt\system32\drivers\openhci.sys	Kernel Driver	True	Manual	Running	OK	Normal	False
parallel	Parallel	c:\winnt\system32\drivers\parallel.sys	Kernel Driver	False	Manual	Stopped	OK	Ignore	False
parport	Parport	c:\winnt\system32\drivers\parport.sys	Kernel Driver	False	Manual	Stopped	OK	Ignore	False
partmgr	PartMgr	c:\winnt\system32\drivers\partmgr.sys	Kernel Driver	True	Boot	Running	OK	Normal	False

parvdm	ParVdm	c:\winnt\system32\drivers\parvdm.sys	Kernel Driver	False	Manual	Stopped	OK	Ignore	False
pci	PCI Bus Driver	c:\winnt\system32\drivers\pci.sys	Kernel Driver	True	Boot	Running	OK	Critical	False
pcidump	PCIDump	Not Available	Kernel Driver	False	System	Stopped	OK	Ignore	False
pciide	PCIIDE	c:\winnt\system32\drivers\pciide.sys	Kernel Driver	True	Boot	Running	OK	Normal	False
pcmcia	Pcmcia	c:\winnt\system32\drivers\pcmcia.sys	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
pdcomp	PDCOMP	Not Available	Kernel Driver	False	Manual	Stopped	OK	Ignore	False
pdframe	PDFRAME	Not Available	Kernel Driver	False	Manual	Stopped	OK	Ignore	False
pdreli	PDRELI	Not Available	Kernel Driver	False	Manual	Stopped	OK	Ignore	False
pdrframe	PDFRAME	Not Available	Kernel Driver	False	Manual	Stopped	OK	Ignore	False
pptpminiport	WAN Miniport (PPTP)	c:\winnt\system32\drivers\rasppptp.sys	Kernel Driver	True	Manual	Running	OK	Normal	False
ptilink	Direct Parallel Link Driver	c:\winnt\system32\drivers\ptilink.sys	Kernel Driver	True	Manual	Running	OK	Normal	False
q57w2k	Compaq NC7781 Gigabit Server Adapter	c:\winnt\system32\drivers\q57w2k.sys	Kernel Driver	True	Manual	Running	OK	Normal	False
ql1080	ql1080	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
ql10wnt	ql10wnt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
ql1240	ql1240	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
ql2100	ql2100	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False

rasacd	Remote Access Auto Connection Driver	c:\winnt\system32\drivers\rasacd.sys	Kernel Driver	True	System
	Running	OK	Normal	False	True
rasl2tp	WAN Miniport (L2TP)	c:\winnt\system32\drivers\rasl2tp.sys	Kernel Driver	True	Manual
	Running	OK	Normal	False	True
raspti	Direct Parallel	c:\winnt\system32\drivers\raspti.sys	Kernel Driver	True	Manual
	Running	OK	Normal	False	True
rca	Microsoft Streaming Network Raw Channel	c:\winnt\system32\drivers\rca.sys	Kernel Driver	False	Manual
Access	Stopped	OK	Normal	False	False
rdbss	Rdbss	c:\winnt\system32\drivers\rdbss.sys	File System Driver	True	System
	Running	OK	Normal	False	True
rdpdr	Terminal Server Device Redirector Driver	c:\winnt\system32\drivers\rdpdr.sys	Kernel Driver	True	Manual
	Running	OK	Normal	False	True
rdpwd	RDPWD	c:\winnt\system32\drivers\rdpwd.sys	Kernel Driver	True	Manual
	Running	OK	Ignore	False	True
redbook	Digital CD Audio Playback Filter Driver	c:\winnt\system32\drivers\redbook.sys	Kernel Driver	False	System
	Stopped	OK	Normal	False	False
serenum	Serenum Filter Driver	c:\winnt\system32\drivers\serenum.sys	Kernel Driver	True	Manual
	Running	OK	Normal	False	True
serial	Serial port driver	c:\winnt\system32\drivers\serial.sys	Kernel Driver	True	Manual
	Running	OK	Ignore	False	True
sfloppy	SFloppy	c:\winnt\system32\drivers\sfloppy.sys	Kernel Driver	False	System
	Stopped	OK	Ignore	False	False
sglfb	sglfb	Not Available	Kernel Driver	False	System
	Stopped	OK	Normal	False	False
simbad	Simbad	Not Available	Kernel Driver	False	System
	Stopped	OK	Normal	False	False

sparrow	Sparrow	Not Available	Kernel Driver	False	Disabled	Stopped	OK
	Normal	False	False	True			
spud	Special Purpose Utility Driver	c:\winnt\system32\drivers\spud.sys	Kernel Driver	True	Manual	Running	OK
	Running	OK	Normal	False	True		
srv	Srv	c:\winnt\system32\drivers\srv.sys	File System Driver	True	Manual	Running	OK
	Running	OK	Normal	False	True		
swenum	Software Bus Driver	c:\winnt\system32\drivers\swenum.sys	Kernel Driver	True	Manual	Running	OK
	Running	OK	Normal	False	True		
symc810	symc810	Not Available	Kernel Driver	False	Disabled	Stopped	OK
	Normal	False	False	True			
symc8xx	symc8xx	Not Available	Kernel Driver	False	Disabled	Stopped	OK
	Normal	False	False	True			
sym_hi	sym_hi	Not Available	Kernel Driver	False	Disabled	Stopped	OK
	Normal	False	False	True			
sysmgmt	HP ProLiant System Management Interface	c:\winnt\system32\drivers\sysmgmt.sys	Kernel Driver	True	Manual	Running	OK
Driver	Running	OK	Normal	False	True		
tcpip	TCP/IP Protocol Driver	c:\winnt\system32\drivers\tcpip.sys	Kernel Driver	True	System	Running	OK
	Running	OK	Normal	False	True		
tdasync	TDASync	c:\winnt\system32\drivers\tdasync.sys	Kernel Driver	False	Manual	Stopped	OK
	Stopped	OK	Ignore	False	False		
tdipx	TDIPX	c:\winnt\system32\drivers\tdipx.sys	Kernel Driver	False	Manual	Stopped	OK
	Stopped	OK	Ignore	False	False		
tdnetb	TDNETB	c:\winnt\system32\drivers\tdnetb.sys	Kernel Driver	False	Manual	Stopped	OK
	Stopped	OK	Ignore	False	False		
tdpipe	TDPIPE	c:\winnt\system32\drivers\tdpipe.sys	Kernel Driver	False	Manual	Stopped	OK
	Stopped	OK	Ignore	False	False		
tdspx	TDSPX	c:\winnt\system32\drivers\tdspx.sys	Kernel Driver	False	Manual	Stopped	OK
	Stopped	OK	Ignore	False	False		
tdtcp	TDTCP	c:\winnt\system32\drivers\tdtcp.sys					

	Kernel Driver	True	Manual
	Running	OK	Ignore
	True		False
termdd	Terminal Device Driver	c:\winnt\system32\drivers\termdd.sys	Kernel Driver
	Running	OK	Normal
	True		False
tga	tga	Not Available	Kernel Driver
	False	System	Stopped
	Ignore	False	False
udfs	Udfs	c:\winnt\system32\drivers\udfs.sys	File System Driver
	Stopped	OK	Normal
	False		Disabled
	False		False
ultra66	ultra66	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
update	Microcode Update Driver	c:\winnt\system32\drivers\update.sys	Kernel Driver
	Running	OK	Normal
	True		Manual
usbhub	Microsoft USB Standard Hub Driver	c:\winnt\system32\drivers\usbhub.sys	Kernel Driver
	Running	OK	Normal
	True		False
vgasave	VgaSave	c:\winnt\system32\drivers\vga.sys	Kernel Driver
	Running	OK	Ignore
	True		False
wanarp	Remote Access IP ARP Driver	c:\winnt\system32\drivers\wanarp.sys	Kernel Driver
	Running	OK	Normal
	True		Manual
wdica	WDICA	Not Available	Kernel Driver
	False	Manual	Stopped
	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\80\Tools\BINN	<SYSTEM>
windir	%SystemRoot%	<SYSTEM>
OS	Windows_NT	<SYSTEM>
PROCESSOR_ARCHITECTURE	x86	<SYSTEM>
PROCESSOR_LEVEL	15	<SYSTEM>
PROCESSOR_IDENTIFIER	x86 Family 15 Model 2	<SYSTEM>
Stepping 7, GenuineIntel		<SYSTEM>
PROCESSOR_REVISION	0207	<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>
 TEMP %USERPROFILE%\Local Settings\Temp
 CL74\Administrator
 TMP %USERPROFILE%\Local Settings\Temp
 CL74\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	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	File Date	Start Time	
Version	Size			
system idle process	Not Available	0	0	
Available	Not Available	Not Available	Not	
system	Unknown	Unknown	Unknown	
	Not Available	8	8	0
	1413120	Not Available	Unknown	
	Unknown	Unknown		
smss.exe	c:\winnt\system32\smss.exe	236	11	
	204800	1413120	5/25/2003 2:37:32 PM	
	5.00.2195.2901	44.27 KB	(45,328 bytes)	
	12/7/1999 7:00:00 AM			
csrss.exe	Not Available	264	13	Not
Available	Not Available	5/25/2003 2:37:33 PM		
	Unknown	Unknown	Unknown	
winlogon.exe	c:\winnt\system32\winlogon.exe			
	260	13	204800	1413120
	5/25/2003 2:37:34 PM			
	5.00.2195.2953	173.77 KB	(177,936 bytes)	
	12/7/1999 7:00:00 AM			
services.exe	c:\winnt\system32\services.exe			
	312	9	204800	1413120
	5/25/2003 2:37:35 PM			
	5.00.2195.2780	86.77 KB	(88,848 bytes)	
	12/7/1999 7:00:00 AM			

lsass.exe	c:\winnt\system32\lsass.exe	324	9	
	204800	1413120	5/25/2003 2:37:35 PM	
	5.00.2195.2964	32.77 KB	(33,552 bytes)	
	12/7/1999 7:00:00 AM			
termsrv.exe	c:\winnt\system32\termsrv.exe	424		
	10	204800	1413120	5/25/2003
	2:37:35 PM	5.00.2195.2342	137.27 KB	
	(140,560 bytes)	9/13/2002 6:09:44 PM		
aclient.exe	c:\altiris\aclient\aclient.exe			
	524	8	204800	1413120
	5/25/2003 2:37:36 PM		5.6.72	
	3.83 MB	(4,014,156 bytes)	1/10/2003	
	3:29:39 PM			
cpqrcmc.exe	c:\winnt\system32\cpqrcmc.exe	548		
	8	204800	1413120	5/25/2003
	2:37:37 PM	5.0.2.0	96.27 KB	(98,576 bytes)
	2/7/2001 4:40:24 PM			
rsys.exe	c:\benchcraft\rsys.exe	584	8	
	204800	1413120	5/25/2003 2:37:37 PM	
	Not Available	32.00 KB	(32,768 bytes)	
	9/13/2002 6:30:57 PM			
svchost.exe	c:\winnt\system32\svchost.exe	604		
	8	204800	1413120	5/25/2003
	2:37:37 PM	5.00.2134.1	7.77 KB	
	(7,952 bytes)	12/7/1999 7:00:00 AM		
sysdown.exe	c:\winnt\system32\sysdown.exe	636		
	8	204800	1413120	5/25/2003
	2:37:38 PM	5.27.2195.0	34.77 KB	
	(35,600 bytes)	11/25/2002 2:41:41 AM		
svchost.exe	c:\winnt\system32\svchost.exe	736		
	8	204800	1413120	5/25/2003
	2:37:41 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	764		
	8	204800	1413120	5/25/2003
	2:37:41 PM	4.71.2195.1	115.27 KB	
	(118,032 bytes)	9/13/2002 6:09:32 PM		
winmgmt.exe	c:\winnt\system32\wbem\winmgmt.exe	784		
	8	204800	1413120	5/25/2003
	2:37:41 PM	1.50.1085.0029	192.08 KB	
	(196,685 bytes)	9/13/2002 6:09:52 PM		
inetinfo.exe	c:\winnt\system32\inetrv\inetinfo.exe	820		
	8	204800	1413120	5/25/2003
	2:37:41 PM	5.00.0984.14.27 KB	(14,608 bytes)	
	9/13/2002 6:10:42 PM			
dfssvc.exe	c:\winnt\system32\dfssvc.exe	808		
	8	204800	1413120	5/25/2003
	2:37:46 PM	5.00.2195.2841	88.27 KB	
	(90,384 bytes)	9/13/2002 6:09:18 PM		
explorer.exe	c:\winnt\explorer.exe			
	1072	8	204800	1413120
	5/25/2003 2:39:33 PM			
	5.00.3315.2846	237.27 KB	(242,960 bytes)	
	9/13/2002 6:09:47 PM			
acIntusr.exe	c:\altiris\aclient\acIntusr.exe			
	1112	8	204800	1413120
	5/25/2003 2:39:33 PM		5, 6, 0, 50	
	176.00 KB	(180,224 bytes)	4/29/2003	
	2:47:15 PM			
tardis.exe	c:\program files\tardis 2000			
v1.4\tardis.exe	1144	8	204800	

	1413120	5/25/2003 2:39:33 PM	5,
0, 1, 4	308.00 KB	(315,392 bytes)	9/13/2002
6:21:25 PM			
svchost.exe	c:\winnt\system32\svchost.exe		
	1236	8	204800
	5/25/2003 2:39:35 PM		5.00.2134.1
	7.77 KB	(7,952 bytes)	12/7/1999
7:00:00 AM			
dllhost.exe	Not Available	740	8
	Not Available	Not Available	
	5/25/2003 3:11:46 PM		Unknown
	Unknown	Unknown	
mmc.exe	c:\winnt\system32\mmc.exe	5544	8
	204800	1413120	5/25/2003 5:01:07 PM
	5.00.2195.2301	589.27 KB	(603,408 bytes)
	9/13/2002 6:09:26 PM		
rsvp.exe	c:\winnt\system32\rsvp.exe	2316	8
	204800	1413120	5/25/2003 5:01:40 PM
	5.00.2167.1	172.77 KB	(176,912 bytes)
	12/7/1999 7:00:00 AM		

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer
Path				
traffic.dll	5.00.2139.1	30.77 KB		
(31,504 bytes)	12/7/1999 7:00:00 AM			
	Microsoft Corporation			
rsvp.exe	5.00.2167.1	172.77 KB	(176,912 bytes)	
	12/7/1999 7:00:00 AM			
	Microsoft Corporation			
faxshell.dll	5.00.2134.1	8.27 KB		
(8,464 bytes)	12/7/1999 7:00:00 AM			
	Microsoft Corporation			
msacm32.dll	5.00.2134.1	65.27 KB		
(66,832 bytes)	12/7/1999 7:00:00 AM			
	Microsoft Corporation			
avifil32.dll	5.00.2134.1	76.27 KB		
(78,096 bytes)	12/7/1999 7:00:00 AM			
	Microsoft Corporation			
msvfw32.dll	5.00.2134.1	113.77 KB		
(116,496 bytes)	12/7/1999 7:00:00 AM			
	Microsoft Corporation			
docprop2.dll	5.00.2178.1	297.77 KB		
(304,912 bytes)	12/7/1999 7:00:00 AM			
	Microsoft Corporation			
wbemprox.dll	1.50.1085.0045	40.08 KB		
(41,040 bytes)	9/13/2002 6:09:52 PM			
	Microsoft Corporation			
mlang.dll	5.00.3103.1000	510.77 KB	(523,024 bytes)	
	9/13/2002 6:09:26 PM			
	Microsoft Corporation			
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 5.00.2177.1 312.27 KB
(319,760 bytes) 9/13/2002 5:46:00 PM
Microsoft Corporation c:\program
files\common files\microsoft
shared\msinfo\msinfo32.dll
mmcndmgr.dll 5.00.2178.1 815.27 KB
(834,832 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\mmcndmgr.dll
msvcp50.dll 5.00.7051 552.50 KB (565,760
bytes) 12/7/1999 7:00:00 AM Microsoft
Corporation c:\winnt\system32\msvcp50.dll
mmc.exe 5.00.2195.2301 589.27 KB (603,408
bytes) 9/13/2002 6:09:26 PM Microsoft
Corporation c:\winnt\system32\mmc.exe
tapisrv.dll 5.00.2195.2955 169.27 KB
(173,328 bytes) 9/13/2002 6:09:44 PM
Microsoft Corporation
c:\winnt\system32\tapisrv.dll
rapilib.dll 5.00.2195.2717 24.77 KB
(25,360 bytes) 9/13/2002 6:09:39 PM
Microsoft Corporation
c:\winnt\system32\rapilib.dll
rsvsp.dll 5.00.2195.2749 74.77 KB
(76,560 bytes) 9/13/2002 6:09:40 PM
Microsoft Corporation
c:\winnt\system32\rsvsp.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 5, 6, 0, 50 176.00 KB
(180,224 bytes) 4/29/2003 2:47:15 PM
c:\altiris\aclient\aclntusr.exe
shdoclc.dll 5.00.3315.2879 324.50 KB
(332,288 bytes) 9/13/2002 6:09:41 PM
Microsoft Corporation
c:\winnt\system32\shdoclc.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 5.00.3103.1000 13.27 KB
(13,584 bytes) 9/13/2002 6:09:38 PM
Microsoft Corporation
c:\winnt\system32\powrprof.dll
batmeter.dll 5.00.3103.1000 20.27 KB
(20,752 bytes) 9/13/2002 6:09:14 PM
Microsoft Corporation
c:\winnt\system32\batmeter.dll
stobject.dll 5.00.2195.2780 79.27 KB
(81,168 bytes) 9/13/2002 6:09:43 PM
Microsoft Corporation
c:\winnt\system32\stobject.dll
webcheck.dll 5.00.3315.1000 251.77 KB
(257,808 bytes) 9/13/2002 6:09:45 PM
Microsoft Corporation
c:\winnt\system32\webcheck.dll
msi.dll 1.11.2405.0 1.69 MB (1,767,184
bytes) 9/13/2002 6:09:29 PM Microsoft
Corporation c:\winnt\system32\msi.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 5.00.2920.0000 55.77 KB
(57,104 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\mydocs.dll
browseui.dll 5.00.3315.2846 788.77 KB
(807,696 bytes) 9/13/2002 6:09:14 PM
Microsoft Corporation
c:\winnt\system32\browseui.dll
shdocvw.dll 5.00.3315.2879 1.05 MB
(1,104,144 bytes) 9/13/2002 6:09:42 PM
Microsoft Corporation
c:\winnt\system32\shdocvw.dll
explorer.exe 5.00.3315.2846 237.27 KB
(242,960 bytes) 9/13/2002 6:09:47 PM
Microsoft Corporation
c:\winnt\explorer.exe
dfssvc.exe 5.00.2195.2841 88.27 KB
(90,384 bytes) 9/13/2002 6:09:18 PM
Microsoft Corporation
c:\winnt\system32\dfssvc.exe
adslldp.dll 5.00.2195.2778 119.77 KB
(122,640 bytes) 9/13/2002 6:09:12 PM
Microsoft Corporation
c:\winnt\system32\adslldp.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.dll 2000.080.0194.00 88.00 KB
(90,112 bytes) 9/13/2002 6:19:44 PM
Microsoft Corporation
c:\winnt\system32\sqlsrv32.dll
mtxdm.dll 2000.2.3471.1 23.27 KB (23,824 bytes)
9/13/2002 6:09:33 PM Microsoft
Corporation c:\winnt\system32\mtxdm.dll
tpcc_com_all.dll 1, 0, 0, 1 80.00 KB
(81,920 bytes) 9/13/2002 6:29:44 PM
c:\inetpub\wwwroot\tpcc_c-2.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_odbc.dll Not Available 28.00 KB
(28,672 bytes) 9/13/2002 6:29:42 PM Not
Available c:\inetpub\wwwroot\tpcc_odbc.dll
mfc42.dll 6.00.8665.0 972.05 KB (995,383
bytes) 12/7/1999 7:00:00 AM Microsoft
Corporation c:\winnt\system32\mfc42.dll
wam.dll 5.00.0984 70.77 KB (72,464 bytes)
9/13/2002 6:10:44 PM Microsoft
Corporation c:\winnt\system32\inetsrv\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 2000.2.3471.1 101.77 KB
(104,208 bytes) 9/13/2002 6:09:33 PM
Microsoft Corporation
c:\winnt\system32\mtxoci.dll
resutils.dll 5.00.2195.2787 39.77 KB
(40,720 bytes) 9/13/2002 6:09:40 PM
Microsoft Corporation
c:\winnt\system32\resutils.dll
clusapi.dll 5.00.2195.2104 54.27 KB
(55,568 bytes) 9/13/2002 6:09:16 PM
Microsoft Corporation
c:\winnt\system32\clusapi.dll
mtxclu.dll 2000.2.3471.1 51.27 KB
(52,496 bytes) 9/13/2002 6:09:33 PM
Microsoft Corporation
c:\winnt\system32\mtxclu.dll
msdtcprx.dll 2000.2.3471.1 665.77 KB
(681,744 bytes) 9/13/2002 6:09:27 PM
Microsoft Corporation
c:\winnt\system32\msdtcprx.dll
comsvcs.dll 2000.2.3471.1 1.35 MB
(1,417,488 bytes) 9/13/2002 6:09:17 PM
Microsoft Corporation
c:\winnt\system32\comsvcs.dll
iislog.dll 5.00.0984 75.27 KB (77,072 bytes)
9/13/2002 6:10:42 PM Microsoft
Corporation c:\winnt\system32\inetsrv\iislog.dll
inetsloc.dll 5.00.0984 20.27 KB (20,752 bytes)
9/13/2002 6:09:24 PM Microsoft
Corporation c:\winnt\system32\inetsloc.dll
isatq.dll 5.00.0984 60.27 KB (61,712 bytes)
9/13/2002 6:10:43 PM Microsoft
Corporation c:\winnt\system32\inetsrv\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
svceext.dll 5.00.0984 39.77 KB (40,720 bytes)
9/13/2002 6:10:44 PM Microsoft
Corporation c:\winnt\system32\inetsrv\svceext.dll
admexs.dll 5.00.0984 27.77 KB (28,432 bytes)
9/13/2002 6:10:41 PM Microsoft
Corporation c:\winnt\system32\inetsrv\admexs.dll
wamreg.dll 5.00.0984 45.77 KB (46,864 bytes)
9/13/2002 6:10:44 PM Microsoft
Corporation c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll 5.00.0984 68.77 KB (70,416 bytes)
9/13/2002 6:10:43 PM Microsoft
Corporation c:\winnt\system32\inetsrv\metadata.dll
iismap.dll 5.00.0984 55.77 KB (57,104 bytes)
9/13/2002 6:09:23 PM Microsoft
Corporation c:\winnt\system32\iismap.dll
nsepm.dll 5.00.0984 43.27 KB (44,304 bytes)
9/13/2002 6:10:43 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 39.27 KB (40,208 bytes)
 9/13/2002 6:10:41 PM Microsoft Corporation
 c:\winnt\system32\inetsrv\coadmin.dll
 iisadmin.dll 5.00.0984 15.27 KB (15,632 bytes)
 9/13/2002 6:10:42 PM Microsoft Corporation
 c:\winnt\system32\inetsrv\iisadmin.dll
 rpcrref.dll 5.00.0984 4.27 KB (4,368 bytes)
 9/13/2002 6:10:43 PM Microsoft Corporation
 c:\winnt\system32\inetsrv\rpcrref.dll
 iisrtrl.dll 5.00.0984 119.77 KB (122,640 bytes)
 9/13/2002 6:09:23 PM Microsoft Corporation
 c:\winnt\system32\iisrtrl.dll
 inetinfo.exe 5.00.0984 14.27 KB (14,608 bytes)
 9/13/2002 6:10:42 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 5.00.2134.1 70.27 KB (71,952 bytes)
 12/7/1999 7:00:00 AM Microsoft Corporation
 c:\winnt\system32\netui0.dll
 ntlanman.dll 5.00.2157.1 35.27 KB (36,112 bytes)
 12/7/1999 7:00:00 AM Microsoft Corporation
 c:\winnt\system32\ntlanman.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 1.50.1085.0000 192.06 KB (196,669 bytes)
 12/7/1999 7:00:00 AM 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 1.50.1085.0000 164.05 KB (167,992 bytes)
 12/7/1999 7:00:00 AM Microsoft Corporation
 c:\winnt\system32\wbem\framedyn.dll
 cimwin32.dll 1.50.1085.0038 1.02 MB (1,073,232 bytes)
 9/13/2002 6:09:50 PM Microsoft Corporation
 c:\winnt\system32\wbem\cimwin32.dll
 wbemsvc.dll 1.50.1085.0007 40.07 KB (41,036 bytes)
 9/13/2002 6:09:52 PM Microsoft Corporation
 c:\winnt\system32\wbem\wbemsvc.dll
 wbemess.dll 1.50.1085.0039 364.07 KB (372,804 bytes)
 9/13/2002 6:09:52 PM Microsoft Corporation
 c:\winnt\system32\wbem\wbemess.dll

fastprox.dll 1.50.1085.0037 144.08 KB (147,536 bytes)
 9/13/2002 6:09:51 PM Microsoft Corporation
 c:\winnt\system32\wbem\fastprox.dll
 wbemcore.dll 1.50.1085.0036 628.07 KB (643,140 bytes)
 9/13/2002 6:09:52 PM Microsoft Corporation
 c:\winnt\system32\wbem\wbemcore.dll
 wbemcomn.dll 1.50.1085.0021 692.07 KB (708,675 bytes)
 9/13/2002 6:09:51 PM Microsoft Corporation
 c:\winnt\system32\wbem\wbemcomn.dll
 winmgmt.exe 1.50.1085.0029 192.08 KB (196,685 bytes)
 9/13/2002 6:09:52 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 4.71.2195.1 115.27 KB (118,032 bytes)
 9/13/2002 6:09:32 PM Microsoft Corporation
 c:\winnt\system32\mstask.exe
 rasdlg.dll 5.00.2195.2671 514.27 KB (526,608 bytes)
 12/7/1999 7:00:00 AM Microsoft Corporation
 c:\winnt\system32\rasdlg.dll
 netcfgx.dll 5.00.2195.2228 534.77 KB (547,600 bytes)
 9/13/2002 6:09:34 PM Microsoft Corporation
 c:\winnt\system32\netcfgx.dll
 rasmans.dll 5.00.2195.2728 147.27 KB (150,800 bytes)
 9/13/2002 6:09:39 PM Microsoft Corporation
 c:\winnt\system32\rasmans.dll
 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 5.00.2195.2779 457.27 KB (468,240 bytes)
 9/13/2002 6:09:34 PM Microsoft Corporation
 c:\winnt\system32\netshell.dll
 netman.dll 5.00.2195.2779 89.27 KB (91,408 bytes)
 9/13/2002 6:09:34 PM Microsoft Corporation
 c:\winnt\system32\netman.dll
 ntmsdba.dll 5.00.2195.2779 167.27 KB (171,280 bytes)
 9/13/2002 6:09:35 PM Microsoft Corporation
 c:\winnt\system32\ntmsdba.dll
 sens.dll 5.00.2163.1 36.77 KB (37,648 bytes)
 12/7/1999 7:00:00 AM Microsoft Corporation
 c:\winnt\system32\sens.dll
 ntmsvc.dll 5.00.2195.2779 391.27 KB (400,656 bytes)
 9/13/2002 6:09:35 PM Microsoft Corporation
 c:\winnt\system32\ntmsvc.dll
 txfaux.dll 2000.2.3471.1 374.27 KB (383,248 bytes)
 9/13/2002 6:09:44 PM Microsoft Corporation
 c:\winnt\system32\txfaux.dll

es.dll 2000.2.3471.1 222.27 KB (227,600 bytes)
 9/13/2002 6:09:21 PM Microsoft Corporation
 c:\winnt\system32\es.dll
 util.dll 5.00.2153.1 25.77 KB (26,384 bytes)
 12/7/1999 7:00:00 AM 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
 sysdown.exe 5.27.2195.0 34.77 KB (35,600 bytes)
 11/25/2002 2:41:41 AM Compaq Computer Corporation
 c:\winnt\system32\sysdown.exe
 rpcss.dll 5.00.2195.2815 231.27 KB (236,816 bytes)
 9/13/2002 6:09:40 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:\benchmark\rsys.exe
 cpqrcmc.exe 5.0.2.0 96.27 KB (98,576 bytes)
 2/7/2001 4:40:24 PM Compaq
 c:\winnt\system32\cpqrcmc.exe
 ntmarta.dll 5.00.2195.2862 98.77 KB (101,136 bytes)
 9/13/2002 6:09:35 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 5.30.23.1205 421.27 KB (431,376 bytes)
 9/13/2002 6:09:40 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 5.00.3103.1000 236.77 KB (242,448 bytes)
 12/7/1999 7:00:00 AM Microsoft Corporation
 c:\winnt\system32\comdlg32.dll
 aclient.exe 5.6.72 3.83 MB (4,014,156 bytes)
 1/10/2003 3:29:39 PM Altiris, Inc.
 c:\altiris\aclient\aclient.exe
 rdpswx.dll 5.00.2180.1 94.40 KB (96,664 bytes)
 9/13/2002 5:45:10 PM Microsoft Corporation
 c:\winnt\system32\rdpswx.dll
 ntlapi.dll 5.00.2134.1 6.77 KB (6,928 bytes)
 12/7/1999 7:00:00 AM Microsoft Corporation
 c:\winnt\system32\ntlapi.dll
 mstlsapi.dll 5.00.2181.1 24.77 KB (25,360 bytes)
 12/7/1999 7:00:00 AM Microsoft Corporation
 c:\winnt\system32\mstlsapi.dll

icaapi.dll	5.00.2134.1	118.77 KB
(121,616 bytes)	9/13/2002 5:45:09 PM	
Microsoft Corporation		
c:\winnt\system32\icaapi.dll		
regapi.dll	5.00.2155.1	35.27 KB
(36,112 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\regapi.dll		
termsrv.exe	5.00.2195.2342	137.27 KB
(140,560 bytes)	9/13/2002 6:09:44 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	5.00.2195.2228	142.77 KB
(146,192 bytes)	9/13/2002 6:10:37 PM	
Microsoft Corporation		
c:\winnt\system32\dsenh.dll		
oakley.dll	5.00.2195.2785	378.77 KB
(387,856 bytes)	9/13/2002 6:09:36 PM	
Microsoft Corporation		
c:\winnt\system32\oakley.dll		
mfc42u.dll	6.00.8665.0	972.05 KB
(995,384 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\mfc42u.dll		
polagent.dll	5.00.2183.1	108.27 KB
(110,864 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\polagent.dll		
scecli.dll	5.00.2195.2780	105.27 KB
(107,792 bytes)	9/13/2002 6:09:41 PM	
Microsoft Corporation		
c:\winnt\system32\scecli.dll		
atl.dll	3.00.8449 57.56 KB	(58,938 bytes)
	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\atl.dll		
certcli.dll	5.00.2195.2778	130.77 KB
(133,904 bytes)	9/13/2002 6:09:16 PM	
Microsoft Corporation		
c:\winnt\system32\certcli.dll		
esent.dll	6.0.3940.13	1.08 MB (1,135,376 bytes)
	9/13/2002 6:09:21 PM	
Microsoft Corporation		
c:\winnt\system32\esent.dll		
ntdsatq.dll	5.00.2195.2878	31.27 KB
(32,016 bytes)	9/13/2002 6:09:35 PM	
Microsoft Corporation		
c:\winnt\system32\ntdsatq.dll		
ntdsa.dll	5.00.2195.2899	990.77 KB (1,014,544 bytes)
	9/13/2002 6:09:34 PM	
Microsoft Corporation		
c:\winnt\system32\ntdsa.dll		
kdcsvc.dll	5.00.2195.2878	137.77 KB
(141,072 bytes)	9/13/2002 6:09:26 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	5.00.2195.2671	21.27 KB
(21,776 bytes)	9/13/2002 6:09:39 PM	

	Microsoft Corporation	
	c:\winnt\system32\rassfm.dll	
mpr.dll	5.00.2195.2779	53.27 KB (54,544 bytes)
	9/13/2002 6:09:27 PM	
Microsoft Corporation		
c:\winnt\system32\mpr.dll		
rsabase.dll	5.00.2195.2228	128.27 KB
(131,344 bytes)	5/4/2001 12:05:02 PM	
Microsoft Corporation		
c:\winnt\system32\rsabase.dll		
schannel.dll	5.00.2195.2922	138.27 KB
(141,584 bytes)	5/4/2001 12:05:02 PM	
Microsoft Corporation		
c:\winnt\system32\schannel.dll		
netlogon.dll	5.00.2195.2865	357.77 KB
(366,352 bytes)	9/13/2002 6:09:34 PM	
Microsoft Corporation		
c:\winnt\system32\netlogon.dll		
kerberos.dll	5.00.2195.2913	198.77 KB
(203,536 bytes)	9/13/2002 6:09:26 PM	
Microsoft Corporation		
c:\winnt\system32\kerberos.dll		
msprivs.dll	5.00.2154.1	41.50 KB
(42,496 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\msprivs.dll		
samsrv.dll	5.00.2195.2918	369.77 KB
(378,640 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\samsrv.dll		
lsasrv.dll	5.00.2195.2964	492.77 KB
(504,592 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\lsasrv.dll		
lsass.exe	5.00.2195.2964	32.77 KB (33,552 bytes)
	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\lsass.exe		
xactsrv.dll	5.00.2134.1	90.27 KB
(92,432 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\xactsrv.dll		
wmicore.dll	5.00.2195.2842	72.27 KB
(74,000 bytes)	9/13/2002 6:09:46 PM	
Microsoft Corporation		
c:\winnt\system32\wmicore.dll		
browser.dll	5.00.2195.2778	48.27 KB
(49,424 bytes)	9/13/2002 6:09:14 PM	
Microsoft Corporation		
c:\winnt\system32\browser.dll		
trkwks.dll	5.00.2166.1	88.77 KB
(90,896 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\trkwks.dll		
psbase.dll	5.00.2195.2779	111.77 KB
(114,448 bytes)	9/13/2002 6:09:39 PM	
Microsoft Corporation		
c:\winnt\system32\psbase.dll		
cryptsvc.dll	5.00.2181.1	61.77 KB
(63,248 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\cryptsvc.dll		
seclogon.dll	5.00.2135.1	15.77 KB
(16,144 bytes)	12/7/1999 7:00:00 AM	

	Microsoft Corporation	
	c:\winnt\system32\seclogon.dll	
cryptdll.dll	5.00.2135.1	41.27 KB
(42,256 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\cryptdll.dll		
wkssvc.dll	5.00.2195.2780	95.27 KB
(97,552 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\wkssvc.dll		
srvsvc.dll	5.00.2195.2904	79.27 KB
(81,168 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\srvsvc.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		
dmserver.dll	2195.2778.297.3	11.77 KB
(12,048 bytes)	9/13/2002 6:09:19 PM	
VERITAS Software Corp.		
c:\winnt\system32\dmserver.dll		
lmhsvc.dll	5.00.2195.2778	9.77 KB
(10,000 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\lmhsvc.dll		
eventlog.dll	5.00.2178.1	43.77 KB
(44,816 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\eventlog.dll		
scsersv.dll	5.00.2195.2780	226.27 KB
(231,696 bytes)	9/13/2002 6:09:41 PM	
Microsoft Corporation		
c:\winnt\system32\scsersv.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	5.00.2195.2780	86.77 KB
(88,848 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\services.exe		
cscui.dll	5.00.2195.2959	228.27 KB (233,744 bytes)
	9/13/2002 6:09:17 PM	
Microsoft Corporation		
c:\winnt\system32\cscui.dll		
wininet.dll	5.00.3315.1000	456.77 KB
(467,728 bytes)	9/13/2002 6:09:46 PM	
Microsoft Corporation		
c:\winnt\system32\wininet.dll		
cryptnet.dll	5.131.2157.1	41.77 KB
(42,768 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\cryptnet.dll		
msv1_0.dll	5.00.2195.2900	111.77 KB
(114,448 bytes)	12/7/1999 7:00:00 AM	
Microsoft Corporation		
c:\winnt\system32\msv1_0.dll		
ntdsapi.dll	5.00.2195.2661	55.77 KB
(57,104 bytes)	9/13/2002 6:09:35 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
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
winrrr.dll 5.00.2160.1 18.77 KB
(19,216 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\winrrr.dll
clbcatq.dll 2000.2.3471.1 496.77 KB
(508,688 bytes) 9/13/2002 6:09:16 PM
Microsoft Corporation
c:\winnt\system32\clbcatq.dll
dhcpcsvc.dll 5.00.2195.2778 88.77 KB
(90,896 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\dhcpcsvc.dll
tapi32.dll 5.00.2182.1 123.27 KB
(126,224 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\tapi32.dll
rasman.dll 5.00.2195.2780 54.77 KB
(56,080 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\rasman.dll
rasapi32.dll 5.00.2195.2671 189.77 KB
(194,320 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
adslldpc.dll 5.00.2195.2842 127.27 KB
(130,320 bytes) 9/13/2002 6:09:12 PM
Microsoft Corporation
c:\winnt\system32\adslldpc.dll
activeds.dll 5.00.2195.2778 174.77 KB
(178,960 bytes) 9/13/2002 6:09:09 PM
Microsoft Corporation
c:\winnt\system32\activeds.dll
oleaut32.dll 2.40.4517 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 5.00.2173.2 67.77 KB
(69,392 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\iphlpapi.dll
rnr20.dll 5.00.2195.2871 35.77 KB (36,624 bytes)
9/13/2002 6:09:40 PM Microsoft
Corporation c:\winnt\system32\rnr20.dll
wshtcpip.dll 5.00.2195.2104 17.27 KB
(17,680 bytes) 9/13/2002 6:09:46 PM
Microsoft Corporation
c:\winnt\system32\wshtcpip.dll

msafd.dll 5.00.2195.2779 106.77 KB (109,328
bytes) 9/13/2002 6:09:27 PM Microsoft
Corporation c:\winnt\system32\msafd.dll
mswsock.dll 5.00.2195.2871 62.77 KB
(64,272 bytes) 9/13/2002 6:09:33 PM
Microsoft Corporation
c:\winnt\system32\mswsock.dll
winspool.drv 5.00.2195.2780 109.77 KB
(112,400 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\winspool.drv
winscard.dll 5.00.2134.1 77.27 KB
(79,120 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\winscard.dll
wlnotify.dll 5.00.2195.2780 53.77 KB
(55,056 bytes) 9/13/2002 6:09:46 PM
Microsoft Corporation
c:\winnt\system32\wlnotify.dll
csd.dll 5.00.2195.2401 98.27 KB
(100,624 bytes) 9/13/2002 6:09:17 PM
Microsoft Corporation
c:\winnt\system32\csd.dll
lz32.dll 5.00.2134.1 9.77 KB (10,000 bytes)
12/7/1999 7:00:00 AM Microsoft
Corporation c:\winnt\system32\lz32.dll
version.dll 5.00.2134.1 15.77 KB
(16,144 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\version.dll
rsaenh.dll 5.00.2195.2228 130.77 KB
(133,904 bytes) 9/13/2002 6:10:37 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 5.00.2195.2887 969.77 KB (993,040
bytes) 9/13/2002 6:09:38 PM Microsoft
Corporation c:\winnt\system32\ole32.dll
imagehlp.dll 5.00.2195.2778 125.77 KB
(128,784 bytes) 5/4/2001 12:05:02 PM
Microsoft Corporation
c:\winnt\system32\imagehlp.dll
msasn1.dll 5.00.2134.1 51.27 KB
(52,496 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\msasn1.dll
crypt32.dll 5.131.2195.2833 451.27 KB
(462,096 bytes) 9/13/2002 6:09:17 PM
Microsoft Corporation
c:\winnt\system32\crypt32.dll
wintrust.dll 5.131.2195.2779 162.27 KB
(166,160 bytes) 9/13/2002 6:09:46 PM
Microsoft Corporation
c:\winnt\system32\wintrust.dll
shlwapi.dll 5.00.3315.1000 282.77 KB
(289,552 bytes) 9/13/2002 6:09:42 PM
Microsoft Corporation
c:\winnt\system32\shlwapi.dll
shell32.dll 5.00.3315.2902 2.25 MB
(2,359,056 bytes) 9/13/2002 6:09:42 PM

Microsoft Corporation
c:\winnt\system32\shell32.dll
msgina.dll 5.00.2195.2779 324.27 KB
(332,048 bytes) 12/7/1999 7:00:00 AM
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 5.00.2195.2663 555.77 KB
(569,104 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 5.00.2195.2386 36.77 KB
(37,648 bytes) 9/13/2002 6:09:46 PM
Microsoft Corporation
c:\winnt\system32\winsta.dll
wsck32.dll 5.00.2195.2871 21.27 KB
(21,776 bytes) 9/13/2002 6:09:46 PM
Microsoft Corporation
c:\winnt\system32\wsck32.dll
dnsapi.dll 5.00.2195.2785 130.77 KB
(133,904 bytes) 9/13/2002 6:09:19 PM
Microsoft Corporation
c:\winnt\system32\dnsapi.dll
wldap32.dll 5.00.2195.2797 125.27 KB
(128,272 bytes) 9/13/2002 6:09:46 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 5.00.2195.2780 67.77 KB
(69,392 bytes) 9/13/2002 6:09:46 PM
Microsoft Corporation
c:\winnt\system32\ws2_32.dll
samlib.dll 5.00.2195.2780 49.77 KB
(50,960 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 5.00.2195.2808 303.77 KB
(311,056 bytes) 9/13/2002 6:09:34 PM
Microsoft Corporation
c:\winnt\system32\netapi32.dll
profmap.dll 5.00.2181.1 29.27 KB
(29,968 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\profmap.dll
secur32.dll 5.00.2195.2862 46.77 KB
(47,888 bytes) 9/13/2002 6:09:41 PM
Microsoft Corporation
c:\winnt\system32\secur32.dll
sfc.dll 5.00.2195.2896 92.11 KB (94,320 bytes)
9/13/2002 6:09:41 PM Microsoft
Corporation c:\winnt\system32\sfc.dll

```

nddeapi.dll          5.00.2137.1      15.27 KB
(15,632 bytes)      12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\nddeapi.dll
userenv.dll          5.00.2195.2780   361.77 KB
(370,448 bytes)     12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\userenv.dll
user32.dll            5.00.2195.2821   392.77 KB
(402,192 bytes)     12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\user32.dll
gdi32.dll             5.00.2195.2778   228.77 KB (234,256
bytes)              12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\gdi32.dll
rpcrt4.dll            5.00.2195.2832   437.27 KB
(447,760 bytes)     9/13/2002 6:09:40 PM
Microsoft Corporation
c:\winnt\system32\rpcrt4.dll
advapi32.dll          5.00.2195.2867   351.77 KB
(360,208 bytes)     12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\advapi32.dll
kernel32.dll           5.00.2195.2778   714.77 KB
(731,920 bytes)     12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\kernel32.dll
msvcrt.dll            6.10.8924.0      284.05 KB
(290,869 bytes)     5/4/2001 12:05:02 PM
Microsoft Corporation
c:\winnt\system32\msvcrt.dll
winlogon.exe          5.00.2195.2953   173.77 KB
(177,936 bytes)     12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\winlogon.exe
sfcfiles.dll          5.00.2195.2967   948.27 KB
(971,024 bytes)     9/13/2002 6:09:41 PM
Microsoft Corporation
c:\winnt\system32\sfcfiles.dll
ntdll.dll             5.00.2195.2779   478.77 KB (490,256
bytes)              5/4/2001 12:05:02 PM
Microsoft Corporation
c:\winnt\system32\ntdll.dll
smss.exe              5.00.2195.2901   44.27 KB (45,328 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	AltClient	Running	
Auto	Own Process		
c:\altiris\aclient\aclient.exe	-service		
Normal	LocalSystem	0	
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	

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		
Compaq Remote Monitor Service	CpqRcmc	Running		
Auto	Own Process			
c:\winnt\system32\cpqrcmc.exe				
Normal	LocalSystem	0		
Distributed File System	Dfs	Running		
Auto	Own Process			
c:\winnt\system32\dfssvc.exe				
Normal	LocalSystem	0		
DHCP Client	Dhcp	Running	Auto	
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
c:\winnt\system32\faxsvc.exe				
Normal	LocalSystem	0		
IIS Admin Service	IISADMIN	Running	Auto	
Share Process				
c:\winnt\system32\inetnsrv\inetinfo.exe				
Normal	LocalSystem	0		
Intersite Messaging IsmServ	Stopped	Disabled	Own	
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	Running	
Auto	Share Process		
c:\winnt\system32\services.exe			
Normal	LocalSystem	0	
Messenger	Messenger	Stopped	Manual
c:\winnt\system32\services.exe			
Normal	LocalSystem	0	
NetMeeting Remote Desktop Sharing	nmmsrv		
Stopped	Manual	Own Process	
c:\winnt\system32\nmmsrv.exe			
Normal	LocalSystem	0	
Distributed Transaction Coordinator	MSDTC		
Stopped	Manual	Own Process	
c:\winnt\system32\msdtc.exe			
Normal	LocalSystem	0	
Windows Installer	MSIServer	Stopped	Manual
Share Process			
c:\winnt\system32\msiexec.exe	/v		
Normal	LocalSystem	0	
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
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
Process	c:\winnt\system32\ntfrs.exe		Own
Normal	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
Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Routing and Remote Access RemoteAccess
Stopped Disabled Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Remote Registry Service RemoteRegistry
Stopped Manual Own Process
c:\winnt\system32\regsvc.exe Normal
LocalSystem 0
Remote Command Service RMSYS Running
Auto Own Process
c:\benchmark\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

```

```

HP ProLiant System Shutdown Service sysdown
Running Auto Own Process
c:\winnt\system32\sysdown.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\inetssrv\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

```

[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
Compaq System Tools All Users:Compaq System Tools All
Users
Microsoft SQL Server All Users:Microsoft SQL
Server All Users
Startup All Users:Startup All Users
Tardis All Users:Tardis All Users
Accessories CL74\Administrator:Accessories
CL74\Administrator
Accessories\Accessibility
CL74\Administrator:Accessories\Accessibilit
y
CL74\Administrator
Accessories\Entertainment
CL74\Administrator:Accessories\Entertainmen
t
CL74\Administrator
Accessories\System Tools
CL74\Administrator:Accessories\System Tools
CL74\Administrator
Administrative Tools
CL74\Administrator:Administrative Tools
CL74\Administrator
Compaq System Tools CL74\Administrator:Compaq System
Tools
CL74\Administrator
Startup CL74\Administrator:Startup
CL74\Administrator

```

[Startup Programs]

Program	Command	User Name	Location
Tardis 2000	c:\progra-1\tardis-1.4\tardis.exe		
AClntUsr	c:\altiris\aclient\acIntusr.exe		All Users
ion\Run	HKLM\SOFTWARE\Microsoft\Windows\CurrentVers		

[OLE Registration]

Object	Local Server
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid
Sound	Not Available
Media Clip	Not Available
Image Document	"C:\Program Files\Windows NT\Accessories\ImageVue\KodakImg.exe"
WordPad Document	"%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"
Windows Media Services DRM Storage object	Not Available

```

Bitmap Image      mspaint.exe

[Internet Explorer 5]

[ Following are sub-categories of this main category
]

[Summary]

Item      Value
Version   5.00.3315.1000
Build     53315.1000
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
Company
advapi32.dll      5.0.2195.2867      352 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
advpack.dll      5.0.3103.1000      87 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
browsec.dll      5.0.3315.2846      35 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
browseui.dll      5.0.3315.2846      789 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
ckcnv.exe 5.0.2189.1      9 KB
8:00:00 AM C:\WINNT\system32 Microsoft Corporation
comctl32.dll      5.81.3103.1000      538 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
crypt32.dll      5.131.2195.2833      451 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
enhsig.dll      <File Missing>      Not Available
Not Available      Not Available      Not Available
Available
iemigrat.dll      <File Missing>      Not Available
Not Available      Not Available      Not Available
Available
iesetup.dll      5.0.3103.1000      57 KB
5/4/2001 12:05:02 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      5.0.2195.2778      126 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation

```

```

imghelp.dll      <File Missing>      Not Available
Not Available      Not Available      Not Available
Available
inseng.dll      5.0.3103.1000      72 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
jobexec.dll      5.0.0.1      47 KB
8:00:00 AM C:\WINNT\system32 Microsoft Corporation
jscript.dll      5.1.0.5907      476 KB
5/4/2001 12:05:02 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      Not Available
Available
mshtml.dll      5.0.3315.2870      2290 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
msjava.dll      5.0.3802.0      923 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
msoss.dll <File Missing>      Not Available      Not Available
Available Not Available Not Available
msxml.dll 8.0.5718.1      493 KB
12:05:02 PM C:\WINNT\system32 Microsoft Corporation
occache.dll      5.0.3103.1000      86 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
ole32.dll 5.0.2195.2887      970 KB
12:05:02 PM C:\WINNT\system32 Microsoft Corporation
oleaut32.dll      2.40.4517.0      612 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
olepro32.dll      5.0.4517.0      160 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
rsabase.dll      5.0.2195.2228      128 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
rsaenh.dll      5.0.2195.2228      131 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
rsapi32.dll      <File Missing>      Not Available
Not Available      Not Available      Not Available
Available
rsasig.dll      <File Missing>      Not Available
Not Available      Not Available      Not Available
Available
schannel.dll      5.1.2195.0      138 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
shdoc401.dll      <File Missing>      Not Available
Not Available      Not Available      Not Available
Available
shdocvw.dll      5.0.3315.2879      1078 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation

```

```

shell32.dll      5.0.3315.2902      2304 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
shlwapi.dll      5.0.3315.1000      283 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
url.dll 5.0.2920.0      82 KB
8:00:00 AM C:\WINNT\system32 Microsoft Corporation
urlmon.dll      5.0.3315.1000      441 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
vbscript.dll      5.1.0.5907      428 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
webcheck.dll      5.0.3315.1000      252 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
win.com 5.0.2134.1      24 KB
8:00:00 AM C:\WINNT\system32 Microsoft Corporation
wininet.dll      5.0.3315.1000      457 KB
5/4/2001 12:05:02 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      5.131.2195.2779      162 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
wsock.vxd <File Missing>      Not Available      Not Available
Available Not Available Not Available
wsock32.dll      5.0.2195.2871      21 KB
5/4/2001 12:05:02 PM
C:\WINNT\system32 Microsoft Corporation
wsock32n.dll      <File Missing>      Not Available
Not Available      Not Available      Not Available
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 34727 MB
Available Disk Space 32334 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 2000 Installation Procedures

Microsoft SQL Server 2000 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 SQL Collation binary
sort order/Latin_1_General

Microsoft COM Component Configuration Parameters

The component services tool in Windows 2000 Server 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 the client was 268. 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	3,520				TpmC	43,326.36
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	3520	376	32	20		428
District	35200	3912	32	197		4141
Customer	105600000	76800000	4931600	4,086,580		85818180
History	105600000	5866680	24		1,168,106	5866704
New_order	31680000	500872	1360	25,112		527344
Orders	105600000	3236784	1787752		7,971,734	5024536
Order_line	1056005900	66000376	164360		15,105,715	66164736
Item	100000	9528	56	479		10063
Stock	352000000	112640000	252272	5,644,614		118536886
Total		265,058,528	7,137,488	9,757,002	24,245,555	281,953,018
MB						
Dynamic Space	73,344	Sum of Data for Order, Orderline and History				
Static Space	202,001	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)				
Daily Growth	14,444	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed				
60 Day Space MB	1,068,650					
60 Day Space GB	1,043.60	GB				
Log Size	122,495.99	MB				
KB Per New Order	4.89	KB				
8 hr log MB	99,300	MB				
8 hr log GB	96.9730	GB				
Space Usage	GB Needed	Disks Measured	GB Priced	Disk Size	Formatted Size	
60 Day Space DB	1,043.60	126	2136.71	18.2GB	16.958	OK
			0.00			
			0.00			
Total DB			2136.71			
8-hr log + mirror	193.9460	4	271.34	72.8GB	67.836	OK
OS, Swap	3	0	0.00	36.4GB	33.917	
Total Storage	1,240.55	GB	2,408.05	GB		

	MSSQL_misc_fg	MSSQL_cs_fg
	428	
	4141	
		85818180
	7034810	
	527344	
	12996270	
	81270451	
	10063	
		118536886
	101,843,508	204,355,066
files=	3	3
size=	4,354,176	8,979,200
Total=	13,062,528	26,937,600
8K blocks	104,500,224	215,500,800
	OK	OK

tpmC		43,326.36									
		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		5,866,680	24	6228296	64	361,616	40	361,656	0.0562	1,168,105.92	1,140.73
Order		3,236,784	1,787,752	3909864	3582792	673,080	1,795,040	2,468,120	0.3833	7,971,734.41	7,784.90
Order-Line		66,000,376	164,360	70512664	328936	4,512,288	164,576	4,676,864	0.7264	15,105,715.15	14,751.67
											23,677.30
		sum(*) Before		sum(*) After		Num New-					
d_next_o_id		105,635,200		112,074,029		6,438,829					
		Before MB		After MB		Grow MB				8-Hr Growth MB	8-Hr Growth GB
Log		1364.23		32108.49		30744.27			4.8894	99,300.33	96.97
									5,006.7644	bytes	
122495.99		1.11369		26.211872							
Database tpcc log used (%)											

Appendix E:

Third Party Letters

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
<http://www.microsoft.com/>

Microsoft

May 23, 2003

Hewlett-Packard Company
James Barrett
MS150402
20555 SH 249
Houston, TX 77070

Mr. Barrett:

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
810-00845	SQL Server 2000 Enterprise Edition <i>Per processor licensing</i> <i>Discount Schedule: Open Program Level B</i> <i>Unit Price reflects a 14% discount from the retail unit price of \$19,999.</i>	\$17,279	2	\$34,558
C11-00821	Windows 2000 Server <i>Server license only - No CALs</i> <i>Discount Schedule: Open Program - No Level</i> <i>Unit Price reflects a 8% discount from the retail unit price of \$799.</i>	\$738	1	\$738
P72-00264	Windows Server 2003, Enterprise Server <i>Server license only - No CALs</i> <i>Discount Schedule: Open Program - No Level</i> <i>Unit Price reflects a 40% discount from the retail unit price of \$3,999.</i>	\$2,399	1	\$2,399
254-00170	Visual C++ Standard <i>No discounts applied</i>	\$109	1	\$109
PRO-PRORS-16U-01	Database Server Support Package <i>1 Year Term</i>	\$1,950	3	\$5,850

Some products may not be currently orderable but will be available through Microsoft's normal distribution channels by April 2, 2003.

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.

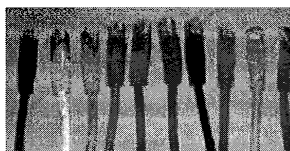
Reference ID: PCjaba0323056591

Please include this Reference ID in any correspondence regarding this price quote.

Home
Network Cards
Network Cables & MISC Cat5e
Crossover Cables
Print Servers
Barcode Readers
Extension Cables
Miscellaneous
TEST
WE ARE ANTI SPAM
Blacklisted Brands
gaming
Cables - Misc
SCSI Cables & devices
Boneyard Cables
6ft 4 wire black molded As low as 34 cents each
network patch cable supports 10 / 100 mbps networks *Order quantities over 5 ONLY*

Show Order
Privacy Policy
Info & Shipping Notes & Ways to delay Processing of order
Search
Index
Y! SHOPPING

LanAdapters.com



7ft Cat 5e Network Patch Cables. (compatible with cat 5)

7ft Category 5 and (Cat5e)Enhanced Network patch cables MOLDED. 350MHZ RJ45/RJ45 Twist Pair supports fast ethernet. These cat 5 e cables are backwards compatible with cat 5

green purple come with booted snagless ends

blue light gray white black red yellow orange come with compact molded soft snagless

Availability: Usually ships the same business day.

CBLC57 \$1.00, 125/\$118.75 Color: ANYshipASAP



Order