Itautec Philco S.A.

TPC Benchmark™ C
Full Disclosure Report
for
Servidor Itautec 3254 1P
using
Microsoft SQL Server 2000
Enterprise Edition SP3
and
Microsoft Windows Server 2003
Standard Edition

Second Edition
Submitted Jan 5, 2005

Second Edition – Jan 5, 2005

Itautec Philco S.A. believes that the information included in this document is accurate as of the publication date. The information in this document is subject to change without notice. Furthermore, Itautec Philco S.A. is not responsible for any errors contained within this document. The pricing information given in this FDR is accurate as of the publication date, May 10, 2004 , but Itautec Philco S.A. cannot guarantee that all sources will offer the same pricing.

Benchmark results are highly dependent upon workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result for these and other factors. Therefore, TPC Benchmark 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 his report were obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. Itautec Philco S.A. 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.

Itautec	**Servidor Itautec 3254 1P C/S with Servidor Itautec 3040HU**	**TPC-C Rev. 5.2** **Report Date:** 05/10/2004

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
R$ 393.899,26	21,197	R$ 18,58	05/10/2004

Processors	Database Manager	Operating System	Other Software	Number of Users
1 Intel Xeon DP Processor 3.06GHz – Server 2 Intel Pentium III Xeon Processor 1.0 GHz - Client	Microsoft SQL Server 2000 Enterprise Edition SP3	Windows Server 2003 Standard Edition	Microsoft Visual C++ .NET Microsoft COM+	**17,000**

Storage
4 x Itautec Storage S320
56 36.4GB 15 Krpm drives

Target Server
Servidor Itautec 3254 1P, 4 GB RAM,
1 x Intel Xeon DP 3.06Ghz
3 x Mylex eXtremeRAID 2000 Disk Array Controllers
5 36.4GB 15 Krpm drive in internal bay

Client
1 x Servidor Itautec 3040HU
2 x Intel Pentium III Xeon 1Ghz
1GB RAM

System Components	Server		Each Client	
	Quantity	Description	Quantity	Description
Processor	1	3.06 GHz Intel Xeon DP w/ 512 KB cache / 533 MHz frontside BUS	2	1.0 GHz Pentium III Xeon w/ 256 KB cache / 133 MHz frontside BUS
Memory	2	2GB Register DDR ECC 266MHz	2	512 MB Register SDRAM ECC DIMM
Disk Controllers	3	Mylex eXtremeRAID 2000	1	Adaptec AIC-7899 Ultra2
Disk Drives	61	36.4 GB SCSI Drive 15 Krpm	1	18.2 GB 10Krpm SCSI Drive
Total Storage	2220 GB		18.2 GB	
Tape Drives	1	20/40 GB DAT		

Numerical Quantities Sumary

MQTh, Computed Maximum Qualified Throughput **21,197 tpmC**

Response Times (in seconds)

	Average	90 th	Max
New Order	0.46	0.72	5.78
Payment	0.25	0.43	5.11
Delivery	0.11	0.12	0.98
Stock Level	3.11	4.37	9.87
Order Status	0.34	0.55	4.98
Delivery (def)	0.68	1.15	2.28
Menu	0.11	0.12	3.18

Response time delay added for emulated components

Menu 0.1
Resp. 0.1

Transaction Mix, in percent of total transactions

New-Order	44.95%
Payment	43.02%
Delivery	4.01%
Stock Level	4.01%
Order Status	4.01%

Keying/Think Times (in seconds)

	Min		Average		Max	
New Order	18.00	0.00	18.01	12.07	18.04	120.71
Payment	3.00	0.00	3.01	12.07	3.04	120.71
Delivery	2.00	0.00	2.01	5.09	2.04	50.71
Stock Level	2.00	0.00	2.01	5.08	2.04	50.71
Order Status	2.00	0.00	2.01	10.09	2.04	100.70

Test Duration

Ramp-up time	24 minutes
Measurement interval	120 minutes
Number of checkpoints	4
Checkpoint interval	30 minutes
Number of transactions (all types) completed in measurement interval	5,885,130

Itautec Philco	Servidor Itautec 3254 1P	TPC-C Rev. 5.2 Report Date: May 10, 2004

Description	Part Number	Third Party Brand	Third Party Pricing	Unit Price R$	Quantity	Extended Price	3 yr. Maint. Price
Server Hardware							
Servidor Itautec 3254 1P							
Base System with 1 x Xeon DP 3.06Ghz/512KB	B5WVD	Itautec	1	15.650	1	15.650	-
2 GB DDR200 PC1600 ECC Reg memory	F1563	Itautec	1	7.242	2	14.484	-
36 GB 15K HDD	F4637	Itautec	1	1.932	5	9.660	-
On-Board Intel PRO1000XT LAN	Included	Itautec	1	-	1	-	-
On-Board Intel PRO100+ LAN	Included	Itautec	1	-	1	-	-
CD-ROM, Internal SCSI Adapter	Included	Itautec	1	-	1	-	-
Upgrade to 3-year / 4-hour response / 7days - 24hrs	1994/2003_02	Itautec	1	-	1	-	12.563
Monitor 15" SYNC MASTER 551S	E9801	Itautec	1	410	1	410	-
UPS SMART APC 1	92759	Itautec	1	3.469	1	3.469	-
					Subtotal	43.673	12.563
Disk Subsystem							
Extreme RAID 2000 4 Channel Controller	A4805	Itautec	1	7.798	3	23.393	-
Itautec Storage Ultrab SC2100CTR-AC	F1390	Itautec	1	21.902	4	87.607	-
36GB Ultra320 15K rpm	F6853	Itautec	1	2.613	56	146.300	-
36GB Ultra320 15K rpm (10% spare)	F6853	Itautec	1	2.613	6	15.675	-
EXT SCSI Cable Eurou VHDCI68P	E8158	Itautec	1	747	4	2.986	-
Kit Rack for Storage	F1781	Itautec	1	775	4	3.101	-
Rack Itautec F2400	F2400	Itautec	1	5.175	1	5.175	-
					Subtotal	284.238	-
Server Software							
Microsoft Windows 2003 Standard Edition	P72-00264	Microsoft	1	3.648	1	3.648	-
SQL SVR 2000 ENT/1 PROCES LICENSE CD ING	H0138	Microsoft	1	70.557	1	69.975	-
					Subtotal	73.623	-
Client Hardware							
Servidor Itautec 3040HU							
Base System with 1 x Pentium III 1Ghz/256KB	B4G6Q	Itautec	1	6.729	1	6.729	-
1 x Pentium III 1GHz/256KB BTO Option	A9347	Itautec	1	1.415	1	1.415	-
2 x 512MB Memory	F0251	Itautec	1	1.344	1	1.344	-
1 x 18GB 10K rpm	F0555	Itautec	1	641	1	641	-
Intel Pro/1000 T Server Adapter	F3608	Itautec	1	1.912	1	1.912	-
CD-ROM, On-Board LAN	Included	Itautec	1	-	1	-	-
Upgrade to 3-year / 4-hour response / 7days - 24hrs	1994/2003	Itautec	1	-	1	-	2.726
Monitor 15" SYNC MASTER 551S	E9801	Itautec	1	410	1	410	-
					Subtotal	12.451	2.726
Client Software							
Microsoft Windows 2000 Server	C11-00821	Microsoft	1	3.648	1	3.648	-
Visual C++ .NET Standard	254-00170	Microsoft	1	389	1	389	-
					Subtotal	4.037	-
Network components							
PATCH CORD CATSE 2,5M	88602	Itautec	1	2	1	2	-
					Subtotal	2	-
					TOTAL	418.024	15.289
Large volume purchase with cash in advance Discount on Itautec Hardware (11.58%)						39.414	
					TOTAL	378.610	15.289
					3-Yr. Cost of Ownership:	R$	393.899,26
Pricing: 1 - Itautec					tpmC Rating:		21197
					R$ / tpmC:		18,58

Audited By Tom Sawyer of Performance Metrics Inc.

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 specifications. If you find that the stated prices are not available according to these items, please inform the TPC at pricing@tpc.org.

Table of Contents

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted on the Servidor Itautec 3254 1P. The tests were conducted by Centro de Informática / UFPE in Brazil. The tests were run in a client/server configuration using a Servidor Itautec 3040HU as client. The operating system used for the benchmark was Microsoft Windows Server 2003 Standard Edition for server and Microsoft Windows 2000 Server Standard Edition for clients. The database was Microsoft SQL Server 2000 SP3.

All tests were done in compliance with Revision 5.2 of the Transaction Processing Council's TPC Benchmark C Standard Specification. Two standard TPC Benchmark ™ C metrics, transactions per minute (tpmC) and price per tpmC (R$/tpmC) are reported and referred to in this document. The results from the tests are summarized below.

Hardware	Software	Total System Cost	tpmC	R$/tpmC	Total Solution Availability Date
Servidor Itautec 3254 1P	**Microsoft SQL Server 2000 Enterprise Edition** **Microsoft Windows Server 2003 Standard Edition**	**R$ 393.899,26**	**21197**	**18.58**	**May 10, 2004**

Auditor

The benchmark configuration, methodology, and results were audited by Tom Sawyer of Performance Metrics, Inc. to validate compliance with the TPC specifications.

Preface

The Transaction Processing Performance Council (TPC) developed The TPC Benchmark™ C. 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 Specification Version 5.2.

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

Introduction

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.

General Items

Benchmark Sponsor

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

Itautec Philco S.A. was the benchmark sponsor for this TPC Benchmark™ C. The benchmark was developed and engineered by Centro de Informática / UFPE in Brazil.

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 but not limited to:

- *Database tuning options.*
- *Recovery/commit options.*
- *Consistency/locking options.*
- *Operating system and application configuration parameters.*

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

Configuration Diagrams

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

The following pages contain the diagrams for both the tested and priced configurations.

Measured Configuration

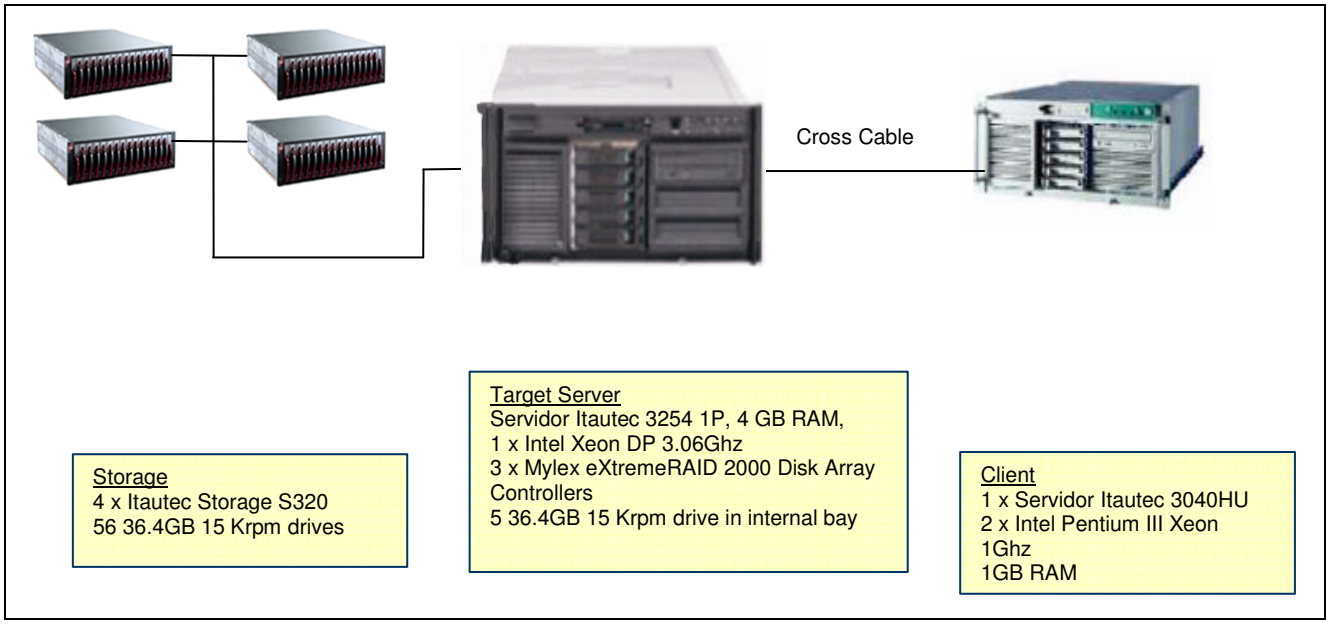


Figure 0.1 – Measured Configuration

Priced Configuration

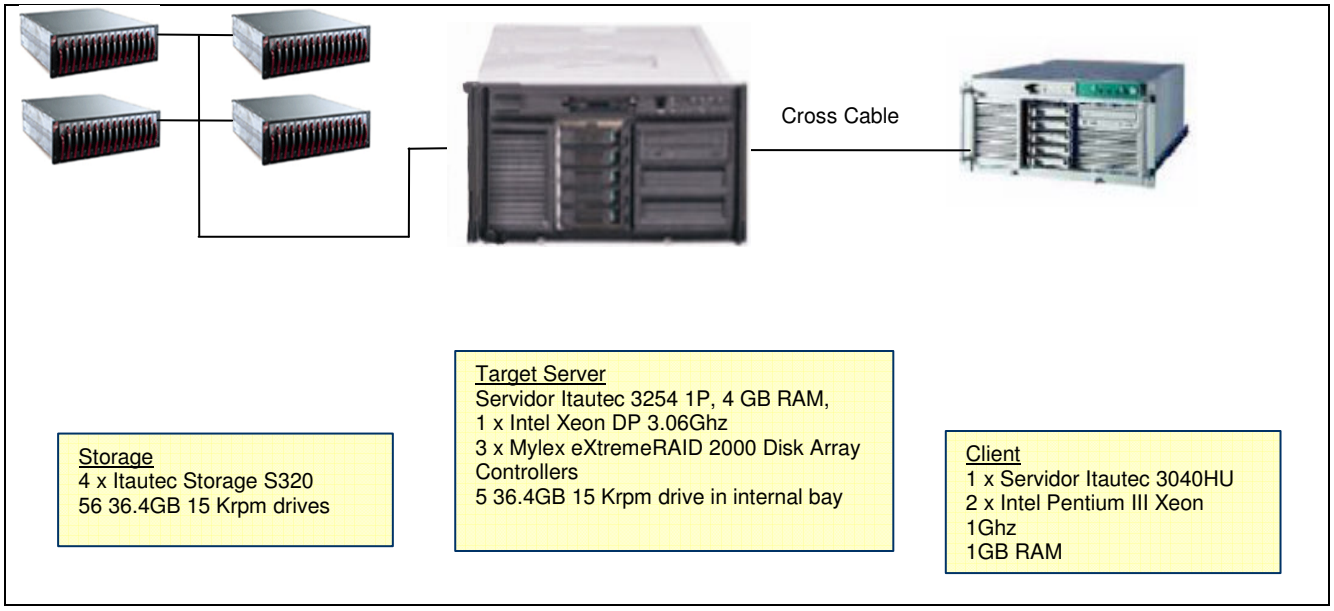


Figure 0.2 – Priced Configuration

Clause 1 - Logical Database Design Related Items

Table Definitions

Listings 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 the Database

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

The tested database configuration used 60 disk drives. The physical organization is documented in Table 4.2: Data Distribution.

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 restriction 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 maximum key value for these new rows.

Insert and delete functions were fully operational during the running of the benchmark.

Horizontal or Vertical Partitioning

While there are few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark (see Clause 1.6), any such partitioning must be disclosed.

Partitioning was not used in this benchmark.

Replication

Replication tables, if used, must be disclosed (see Clause 1.4.6).

Replication was not used in this benchmark.

Table Attributes

Additional and/or duplicated attributes in any table must be disclosed, along with a statement on the impact on performance (see Clause 1.4.7).

No additional attributes were used in this benchmark.

Clause 2 - Transaction and Terminal Profiles Related Items

Random Number Generation

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

The random number generation was handled internally in the Microsoft BenchCraft RTE program. The independent auditing process verified this.

Screen Layout

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

All screen layouts followed the Standard Specifications.

Terminal 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 auditor with a thorough execution of the five transaction types, using Microsoft Internet Explorer, verified the terminal features.

Intelligent Terminals

Any usage of presentation managers or intelligent terminals must be explained.
Comment 1: *The intent of this clause is to describe any special manipulations performed by a local terminal or workstation to off-load work from the SUT. This includes, but is not limited to: screen presentations; message bundling, and local storage of TPC-C rows.*
Comment 2: *This disclosure also requires that all data manipulation functions performed by the local terminal to provide navigational aids for transaction(s) must also be described. Within this disclosure, the purpose of such additional function(s) must be explained.*

The application code responsible for processing the data was executed on the clients. HTML Screen manipulation commands were downloaded to the web browser, which controlled the input and output graphics. This code is documented in Appendix A. IIS (Microsoft Internet Information Server) was involved in processing and presenting this data.

Transaction Profiles

The percentage of home and remote order-lines in the New-Order transactions must be disclosed.
The percentage of New-Order transactions that was rolled back as a result of an unused item number must be disclosed.
The number of items per orders entered by New-Order transactions must be disclosed.
The percentage of home and remote Payment transactions must be disclosed.
The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the database must be disclosed.
The percentage of Delivery transactions that were skipped as a result of an insufficient number of rows in the NEW-ORDER table must be disclosed.

Table 2.1: Transaction Statistics

Transaction	Function	Value
New Order	Home Warehouse Order Lines	99.00%
	Remote Warehouse Order Lines	1.00%
	Rolled Back Transactions	1.00%
	Average Lines Per Order	10.00
Payment	Home Warehouse Transactions	85%
	Remote Warehouse Transactions	15%
	Non-Primary Key Access	60.01%
Order Status	Non-Primary Key Access	59.96%
Delivery	Delivery Transactions Skipped	0

Transaction Mix

The mix (i.e., percentages) of transaction types seen by the SUT must be disclosed. (8.1.3.11)

Table 2.2: Transaction Mix

Transaction	Percentage
New Order	44.95
Payment	43.02
Order Status	4.01
Delivery	4.01
Stock Level	4.01

Deferred Delivery Mechanism

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

Microsoft COM+ components installed on each Web Client create a pool of threads connected to the database, responsible for processing delivery transactions. When the application on the Web Client side receives a request for a delivery transaction, one of the threads is assigned for processing it while the control is immediately returned to the user (RTE). Upon completion, the thread writes an entry in the delivery log and returns to the thread pool.

The source code is listed in Appendix A.

Clause 3 - Transaction and System Properties Related Items

The results of the ACID test 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 tests were conducted successfully according to specification.

Atomicity

The system under test must guarantee that 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 Transaction

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 Transaction

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 including several checkpoints. The script was re-executed and 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 auditor to demonstrate the required isolation had been met reviewed and verified the captured files.

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.

Loss of Data / Loss of Log

To demonstrate recovery from a permanent failure of durable media containing TPC-C tables, the following steps were executed:

1) A 1700 warehouse database was generated.
2) The database was backed up using SQL scripts.
3) A sum of D_NEXT_O_ID was recorded.
4) The RTEs were started with 10% of the benchmark users.
5) The system was run in steady state for 10 minutes.

6) One log disk was removed from the drive cabinet. No interruption occurred.
7) Keep running more 10 minutes.
8) One data disk was removed, causing an SQL Server error.
9) The RTE was stopped.
10) SQL Server was stopped and restarted.
11) The transaction log was dumped to disk.
12) SQL Server was stopped, the Windows was shutdown and the machine powered off.
13) The failed disk was replaced.
14) The machine was powered on.
15) SQL Server was started.
16) The database and the transaction log were restored from backup.
17) The sum of D_NEXT_O_ID was taken.
18) This number was compared with the number of new orders reported by the RTE.
19) Consistency test #3 was executed and verified.

Instantaneous Interruption and Loss of Memory

To validate system recovery an instantaneous interruption was caused by removing power to the Server, the following steps were executed:

1) A sum of D_NEXT_O_ID was taken.
2) 17,000 users were started via the RTEs.
3) The system was run is steady state for 5 minutes.
4) The power supply cord was removed from the server, causing instantaneous interruption.
5) The RTE's were stopped.
6) Power was reconnected and the system was rebooted.
7) SQL Server was started and the recovery process completed successfully.
8) A new count of D_NEXT_O_ID was taken.
9) This number was validated against the calculated number reported by the RTEs.

Clause 4: Scaling and Database Population Related Items

Cardinality of the Tables

The cardinality (e.g., the number of rows) of each table, as it existed at the start of the benchmark run (see Clause 4.2), must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted (see Clause 4.2.2), the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

The database was generated with 1700 warehouses, and the audited performance run used 1700 warehouses.

Table 4.1: Table Cardinality

Table	Initial Cardinality
Warehouse	1700
District	17000
Customer	51000000
New Order	15300000
Orders	51000000
History	51000000
Order Line	510007366
Item	100000
Stock	170000000
Deleted Warehouses	0

Distribution of Database Tables and Logs

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

The system was configured with 60 36.4GB 15Krpm SCSI disks for the database. There were 56 disks connected to 2 controllers configured as RAID 0 and 4 disks connected to one controller and configured as RAID 0+1. The operating system was stored in one 36.4GB 15Krpm disk. The OS disk was connected to the same controller as the log disks. Most logical data drives contained 3 partitions for miscellaneous, customer and stock, and backup data. Raw file systems were used except for the NTFS formatted backup partitions. The configuration is further detailed below in Table 4.2.

Table 4.2: Data Distribution

Controller	DB Components	Partition	Size	Disks
0	Miscellaneous Customer and Stock Backup	O:/ R:/ W:/	26.01 GB 50.01 GB 880.63 GB	28 – 36 GB 15K
1	Miscellaneous Customer and Stock Backup	P:/ S:/ X:/	26.01 GB 50.01 GB 880.63 GB	28 – 36 GB 15K
2	Transaction Log	M:/	68.33 GB	4 – 36 GB 15K

Database Model

A statement must be provided that describes:

1. *The data model implemented by the DBMS used (e.g., relational, network, hierarchical)*
2. *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 transactions. 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 using the Microsoft ODBC interface.

Mapping Partitions/Replication

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

No partitioning or replication was used.

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 (see Clause 4.2.3).

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

1. The current log space was recorded by running *dbcc sqlperf(logspace)*
2. Transactions were run against the database with a full user load.
3. The final log space usage was recorded by running *dbcc sqlperf(logspace)*
4. The space used was calculated as the difference between the first and second queries.
5. The number of NEW-ORDERS was retrieved from the RTE report generated for the entire run.
6. The total space used was divided by the number of NEW-ORDERS producing a size per NEW-ORDER.
7. The NEW-ORDER size was multiplied by the measured tpmC rate and multiplied by 480 minutes.

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

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

Clause 5: Performance Metrics and Response Time Related Items

Measured tpmC

Measured tpmC must be reported.

Measured tpmC: **21,197** tpmC
Price per tpmC: **R$ 18.58** 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 5.1: Response Times

Transaction	Average	90%	Maximum
New order	0.46	0.72	5.78
Payment	0.25	0.43	5.11
Delivery	0.11	0.12	0.98
Stock Level	3.11	4.37	9.87
Order Status	0.34	0.55	4.98
Delivery (def)	0.68	1.15	2.28
Menu	0.11	0.12	3.18

Keying and Think Times

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

Table 5.2: Keying Times

Transaction	Average	Minimum	Maximum
New Order	18.01	18.00	18.04
Payment	3.01	3.00	3.04
Delivery	2.01	2.00	2.04
Stock Level	2.01	2.00	2.04
Order Status	2.01	2.00	2.04

Table 5.3: Think Times

Transaction	Average	Minimum	Maximum
New Order	12.07	0.00	120.71
Payment	12.07	0.00	120.71
Delivery	5.09	0.00	50.71
Stock Level	5.08	0.00	50.71
Order Status	10.09	0.00	100.70

Response Time Distribution Curves

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

Figure 5.1 - New-Order Transaction Response Time Distribution

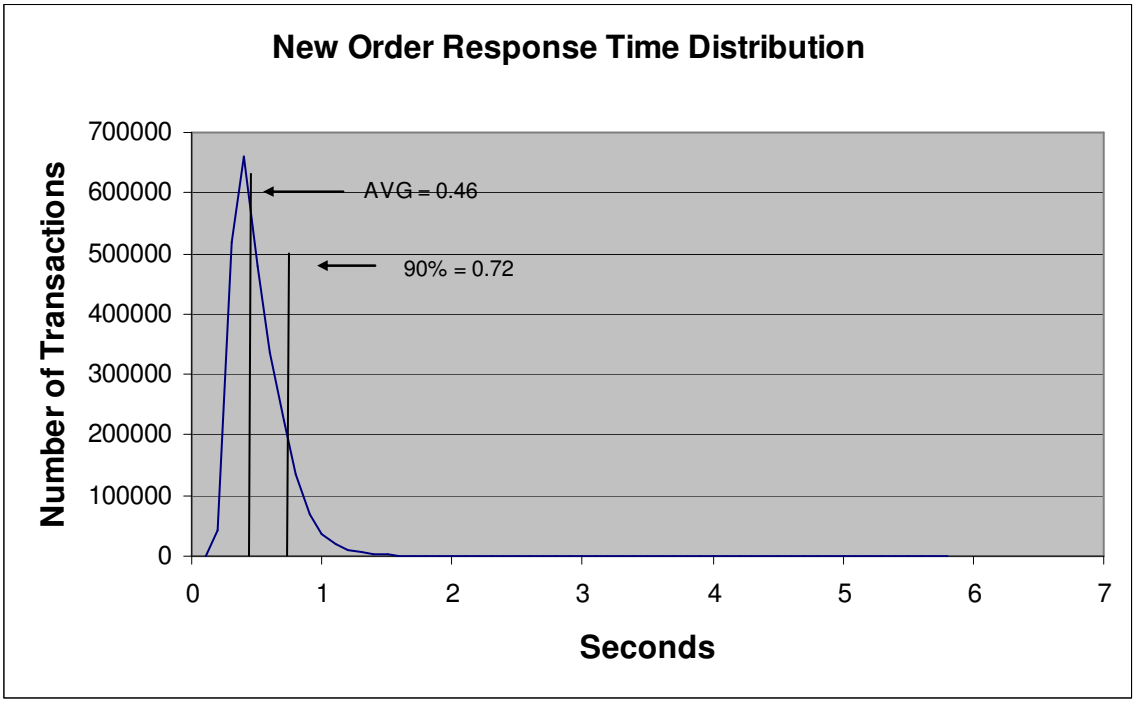


Figure 5.2 - Payment Transaction Response Time Distribution

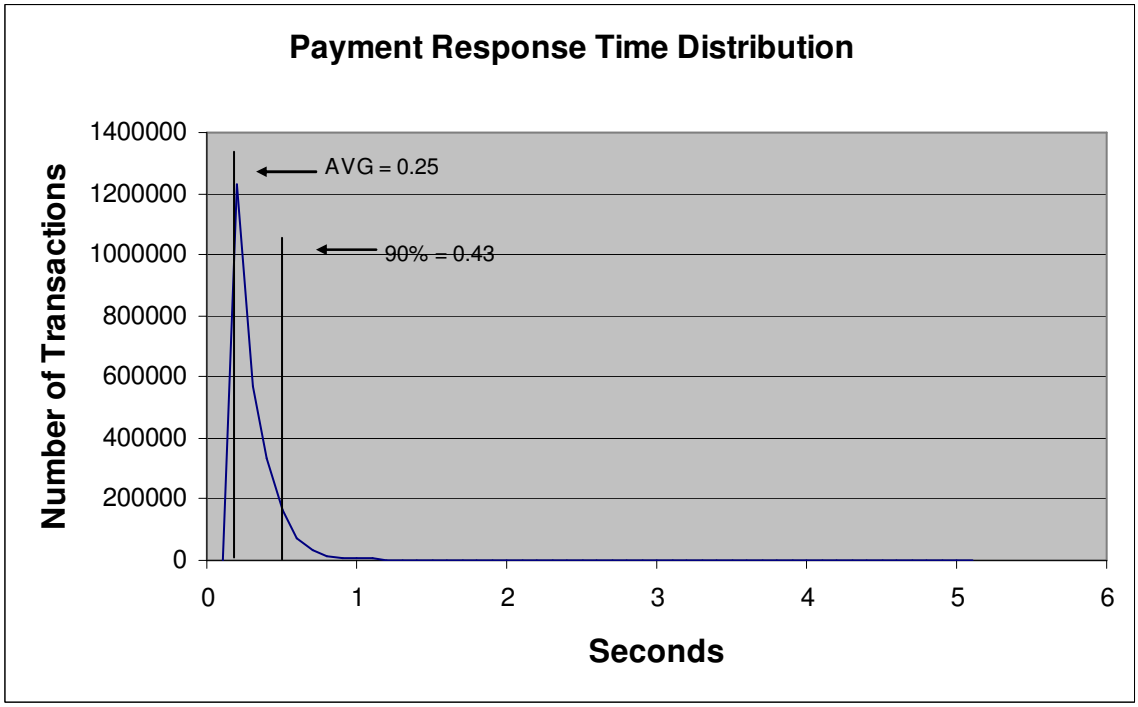


Figure 5.3 – Stock Level Transaction Response Time Distribution

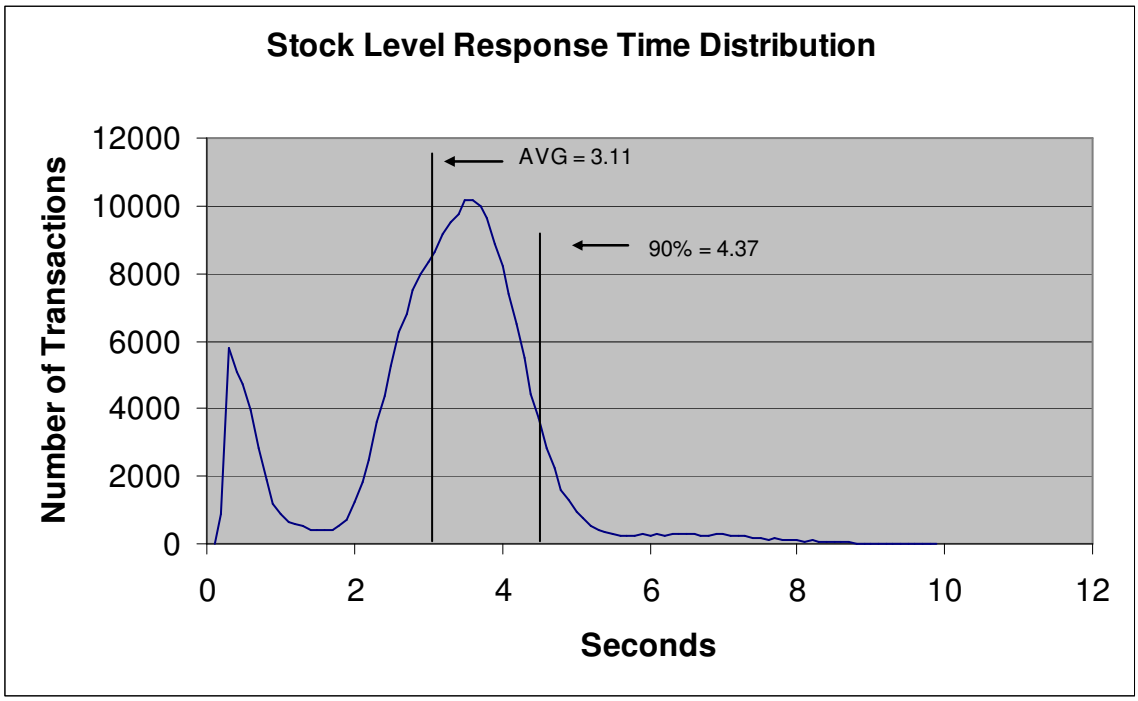


Figure 5.4 – Order Status Transaction Response Time Distribution

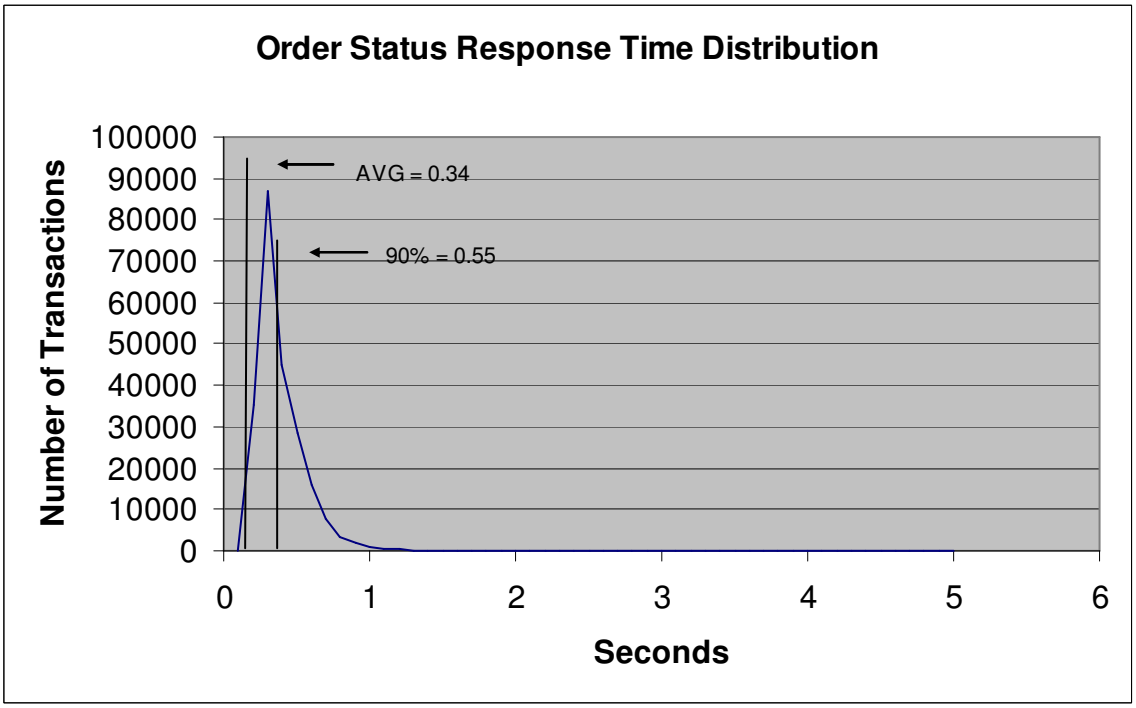
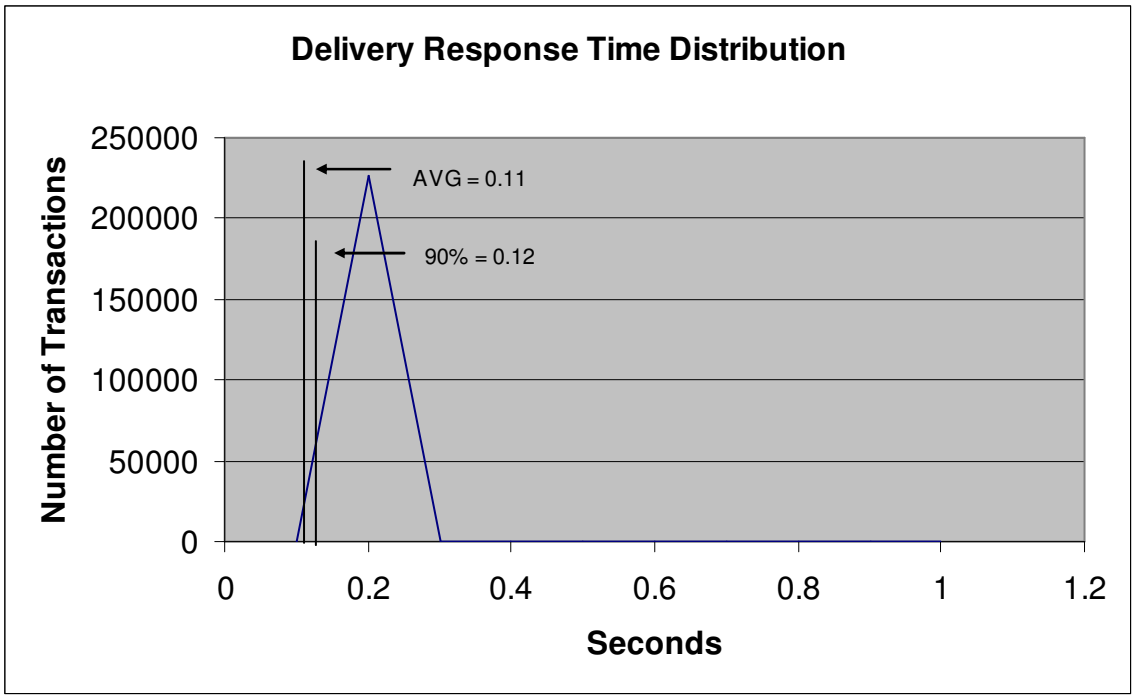


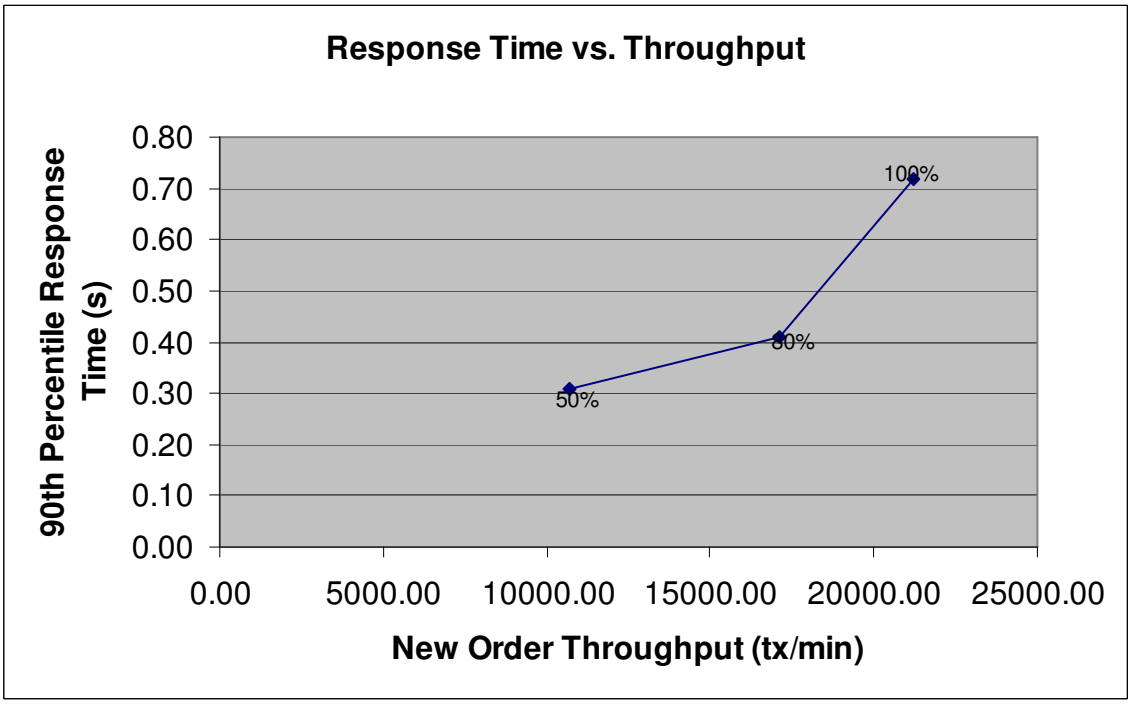
Figure 5.5 – Delivery Transaction Response Time Distribution



New Order Response Time vs. Throughput Performance

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

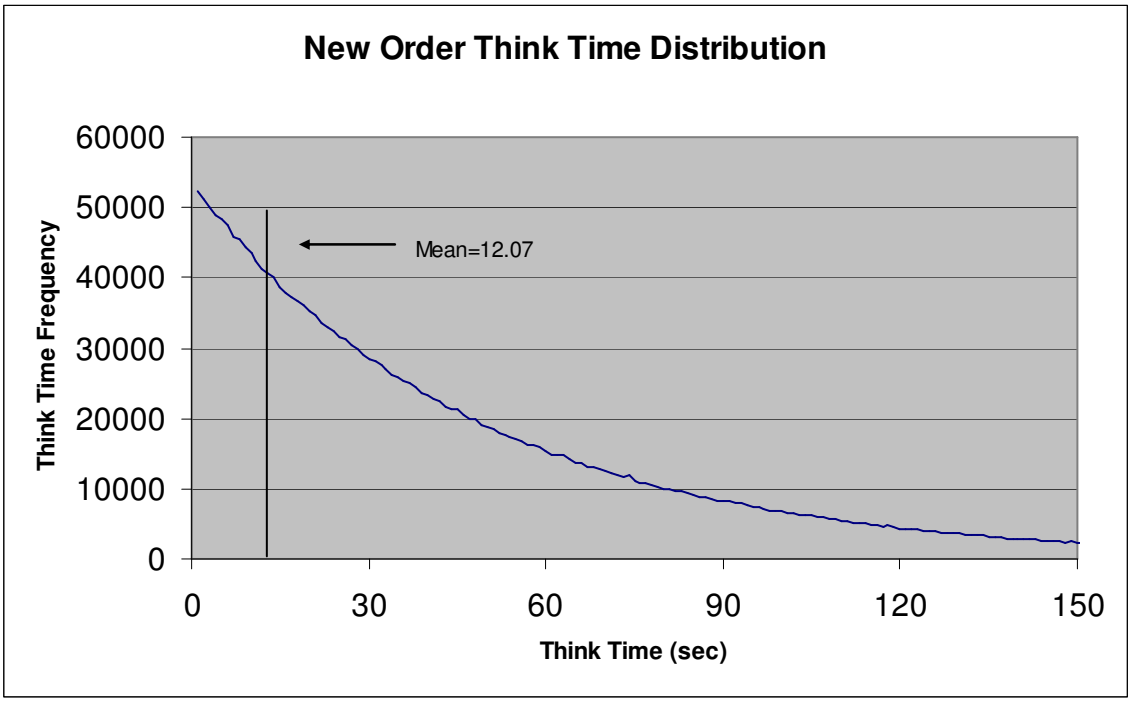
Figure 5.6 – New Order Response Time vs. Throughput



New Order Think Time Distribution

Think Time frequency distribution curves (see Clause 5.6.3) must be reported for the New-Order transaction.

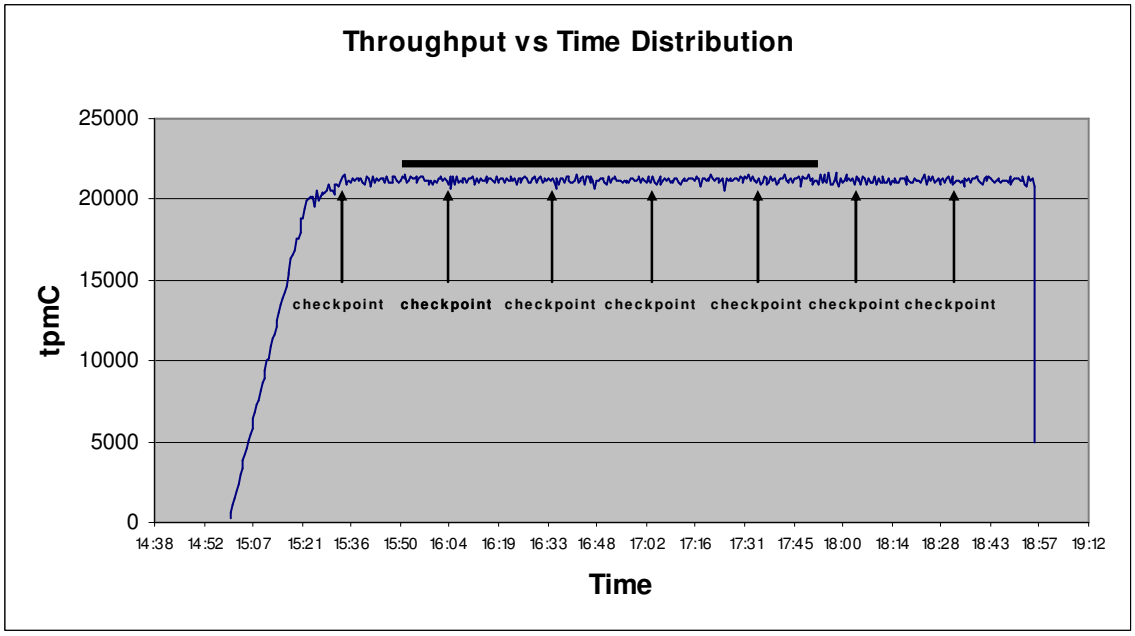
Figure 5.7 – New Order Think Time Distribution



New Order Throughput vs. Elapsed Time

A graph of throughput versus elapsed time (see Clause 5.6.4) must be reported for the New-Order transaction.

Figure 5.8 – Throughput vs. Time Distribution



Steady State Methodology

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval (see Clause 5.5) must be described.

By using the monitoring tools on the RTE, a steady state was determined. Figure 5.8 further supports the level chosen by the utilities used.

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.

A checkpoint in Microsoft SQL Server writes to disk all updated memory pages that have not been yet actually written to disk. SQL Server recovery interval parameter was set to 60 to perform checkpoint at specific intervals. A checkpoint script, which issues specified number of checkpoint at specified (30 minutes) intervals, was started after all users logged in and sending transactions.

Measurement Period Duration and Checkpoint Duration

The start time and duration in seconds of at least the four (4) longest checkpoints during the Measurement Interval must be disclosed (see Clause 5.5.2.2 (2)).[Clause 8.1.6.11]
A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC) must be included.[Clause 8.1.6.12]

Table 5.9: Checkpoints

	Start	End	Duration (in seconds)
Measurement Interval	15:50:47	17:50:47	7200
1st Checkpoint	16:04:51	16:15:42	651
2nd Checkpoint	16:34:46	16:46:26	700
3rd Checkpoint	17:04:40	17:17:15	755
4th Checkpoint	17:34:36	17:46:26	710

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 that could not be 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.

The above statistics are disclosed in Table 2.1.

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.

There was one checkpoint before measurement and four checkpoints during measurement. The time of the first checkpoint during the measurement interval is 651 seconds from the start of the measurement, and the checkpoint interval is 30 minutes.

RTE Parameters

The RTE input parameters, code fragments, functions, etc. used to generate each transaction input field must be disclosed. (8.1.7.1)
***Comment:** The intent is to demonstrate the RTE was configured to generate transaction input data as specified in Clause 2.*

The RTE input parameters are listed in Appendix C - Tunable Parameters.

Lost Terminal Connections

The number of terminal connections lost during the Measurement Interval must be disclosed.

No terminal connections were lost.

Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to that of the priced system. The results of the test described in Clause 6.6.3.4 must be disclosed. (8.1.7.2)

No components were emulated.

Benchmarked and Targeted System Configuration 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 software and hardware functionality being performed on the Driver System, and its interface to the SUT must be disclosed (see Clause 6.6.3.6). (8.1.7.3)

The driver system performed transaction data generation and communication to the client through the standard Web browser (HTTP) protocol. It also captured and timestamped the SUT output data for post-processing of the reported metrics. No other functionality was included on the driver system.

Figures 0.1 and 0.2 of this report contain detailed diagrams of both the benchmark configuration and the priced configuration.

Network Configuration

The network configurations of both the tested services and the proposed (target) services which are being represented and a thorough explanation of exactly which parts of the proposed configuration are being replaced on the Driver System must be disclosed (see Clause 6.6.4). (8.1.7.4)

The network configurations of the benchmarked and priced configurations were identical.

Network Bandwidth

The bandwidth of the network(s) used in the tested/priced configuration must be disclosed. (8.1.7.5)

The bandwidth of the tested and priced networks were as follows:
• 100 BaseT (100 Mbit/sec) network segments between the RTE and the clients
• 1000 BaseT (1000 Mbit/sec) between the Clients and Server.

Operator Intervention

If the configuration requires operator intervention (see Clause 6.6.6), the mechanism and the frequency of this intervention must be disclosed. (8.1.7.6)

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

Clause 7 - Pricing Related Items

Hardware and Software List

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 date. If package-pricing is used, vendor part number of the package and a description uniquely dentifying each of the components of the package must be disclosed. Pricing source(s) and effective date(s) of price(s) must also be reported. (8.1.8.1)

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. (8.1.8.2)

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 Date

The committed delivery date for general availability (availability date) of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the reported vailability date for the priced system must be the date at which all components are committed to be available. (8.1.8.3)

Hardware Availability Date: 05/10/2004
Software Availability Date: 05/10/2004

Measured TpmC

A statement of the measured tpmC, as well as the respective calculations for 5-year pricing, price/performance (price/tpmC), and the availability date must be included. (8.1.8.4)

Maximum Qualified Throughput: 21,197 tpmC
Price Performance Metric: R$/tpmC 18.58

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. (8.1.8.5)

This system is priced for Brazil.

Usage Pricing

For any usage pricing, the sponsor must disclose (8.1.8.6):
• Usage level at which the component was priced.
• A statement of the company policy allowing such pricing.
***Comment**: Usage pricing may include, but is not limited to, the operating system and database management software.*

The component pricing based on usage is shown below:
- 1 Microsoft Windows 2000 Server Licenses
- 1 Microsoft Windows Server 2003 Standard Edition License
- 1 Microsoft SQL Server 2000 Enterprise Edition Licenses (per processor)
- 1 Microsoft Visual C++ .NET Standard Edition
- 3 Year Support for Hardware Components.

System Pricing

System pricing should include subtotals for the following components: Server Hardware, Server Software, Client Hardware, Client Software, and Network Components used for terminal connection (see Clause 7.2.2.3). Clause 6.1 describes the Server and Client components. An example of the standard pricing sheet is shown in Appendix B. (8.1.8.7)
System pricing must include line item indication where non-sponsoring companies' brands are used. System pricing must also include line item indication of third party pricing. See example in Appendix B. (8.1.8.8)

The details of the hardware and software are reported in the front of this report as part of the Executive Summary.

Clause 9 - Audit Related Items

Auditor

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. (8.1.9.1)
A review of the pricing model is required to ensure that all components required are priced (see Clause 9.2.8). The auditor is not required to review the final Full Disclosure Report or the final pricing prior to issuing the attestations letter. (8.1.9.2)

This TPC-C benchmark has been audited by Tom Sawyer of Performance Metrics.

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 Adminstrator, TPC
Presidio of San Francisco
Bldg 572B Ruger St.
San Francisco, CA 94129-0920
Phone: (415) 561-6272, fax (415) 561-6120
www.tpc.org

or:

Itautec Philco S.A.
Rua Santa Catarina, 1
030860-025 – São Paulo – SP
Phone: +55 (11) 6097-3000, fax +55 (11) 6097-4284

May 5, 2004

Mr. Fábio Ávila Rêgo Pessoa
Itautec Performance Lab
Centro de Informática – UFPE
Recife, Brazil

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

Platform: Servidor Itautec 3254 1P
Database Manager: Microsoft SQL Server 2000 Enterprise Edition 32-bit SP3
Operating System: Microsoft Windows 2003 Server, Standard Edition
Transaction Monitor: Microsoft COM+

Server: Servidor Itautec 3254 1P				
CPU's	Memory	Disks	90% Response	TpmC
1 Intel Xeon DP @ 3.06 GHz	Main: 4 GB CPU cache: 512 KB	61 @ 36GB	**0.72 sec**	**21,197**
Client: Servidor Itautec 3040HU				
2 Intel Xeon @ 1 GHz	Main: 1 GB CPU cache: 256 KB	1 @ 18GB	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 was properly sized and populated.
- The database was properly scaled with 1,700 warehouses, all of which were active during the Measured Interval.

- The ACID properties were successfully demonstrated.
- The durability data loss and log loss tests were performed on the fully scaled database with 3.400 users and 340 active warehouses, running at over 10% load.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was configured on the measured system.
- Eight hours of dynamic table growth space was configured on the measured system.
- The 60-day space calculation was verified. The measured configuration has sufficient storage to satisfy this requirement.
- Measurement cycle times included a 0.1 second menu and a 0.1 second response time delays for emulated Web browsers.
- There were 17,000 user contexts present on the system.
- Each emulated user started with a different random number seed.
- The NURand constants used for database load and at run time were 123 and 233 respectively.
- The steady state portion of the test was 2 hours (7200 seconds).
- One checkpoint was taken before the measured interval and after steady state was reached.
- Four checkpoints were taken during the measured interval.
- The system pricing was checked for major components and maintenance.

Auditor Notes:

None

Sincerely,

Tom Sawyer
Auditor

Appendix A – Source Code

isapi_dll/src/tpcc.def

```
LIBRARY TPCC.DLL

EXPORTS

          GetExtensionVersion @1
          HttpExtensionProc   @2
          TerminateExtension  @3
```

Isapi_dll/src/tpcc.h

```
/*        FILE:               TPCC.H
 *                                  Microsoft TPC-C
Kit Ver. 4.20.000
 *                                  Copyright
Microsoft, 1999
 *              All Rights Reserved
 *
 *                                  Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
 *
 *        PURPOSE:  Header file for ISAPI TPCC.DLL,
defines structures and functions used in the isapi
tpcc.dll.
 *
 */

//VERSION RESOURCE DEFINES
#define _APS_NEXT_RESOURCE_VALUE
                    101
#define _APS_NEXT_COMMAND_VALUE
                    40001
#define _APS_NEXT_CONTROL_VALUE
                    1000
#define _APS_NEXT_SYMED_VALUE
                    101

#define TP_MAX_RETRIES
                                  50


//note that the welcome form must be processed first as
terminal ids assigned here, once the
//terminal id is assigned then the forms can be
processed in any order.
#define WELCOME_FORM
                    1
          //beginning form no term id assigned, form id
#define MAIN_MENU_FORM
                    2
          //term id assigned main menu form id
#define NEW_ORDER_FORM
                    3                   //new
order form id
#define PAYMENT_FORM
                              4
          //payment form id
#define DELIVERY_FORM
                              5
          //delivery form id
#define ORDER_STATUS_FORM
                    6                   //order status
id
#define STOCK_LEVEL_FORM
                    7                   //stock level
form id

//This macro is used to prevent the compiler error
unused formal parameter
#define UNUSEDPARAM(x) (x = x)


//This structure defines the data necessary to keep
distinct for each terminal or client connection.
typedef struct _CLIENTDATA
{
          int                         iNextFree;
                                      //index of 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;
                                //syncronization 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;

          //syncronization 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(long w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);
```

isapi_dll/src/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)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif //_WIN32

#ifndef _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
//////////////////////////////////////////////////////////
//////////////////////


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


//////////////////////////////////////////////////////////
//////////////////////
#endif    // not APSTUDIO_INVOKED
```

isapi_dll/src/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
definations 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 occured 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(&TermCriticalSectio
n);

                                    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");

         hLibInstanceTm = LoadLibrary( szDllName );
                                                      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");

         hLibInstanceTm = LoadLibrary( szDllName );
                                                      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");

         hLibInstanceTm = LoadLibrary( szDllName );
                                                      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");

         hLibInstanceDb = LoadLibrary( szDllName );

         if (hLibInstanceDb == NULL)

                  throw new CWEBCLNT_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 CWEBCLNT_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 CWEBCLNT_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 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(&DelBuffCriticalSec
tion);

          hWorkerSemaphore = CreateSemaphore( NULL, 0,
dwDelBuffSize, NULL );

          dwDelBuffFreeCount = dwDelBuffSize;


          InitJulianTime(NULL);

                                                //
create unique log file name based on delilog-yymmdd-
hhmm.log

          SYSTEMTIME Time;

          GetLocalTime( &Time );

          wsprintf( szLogFile, "%sdelivery-
%2.2d%2.2d%2.2d-%2.2d%2.2d.log",

                    Reg.szPath, Time.wYear % 100,
Time.wMonth, Time.wDay, Time.wHour, Time.wMinute );

          txnDelilog = new CTxnLog(szLogFile,
TXN_LOG_WRITE);


          //write event into txn log for START

          txnDelilog-
>WriteCtrlRecToLog(TXN_EVENT_START, szMyComputerName,
sizeof(szMyComputerName));

                                                //
allocate structures for delivery buffers and thread
mgmt

          pDeliHandles = new
HANDLE[dwNumDeliveryThreads];

          pDelBuff = new
DELIVERY_TRANSACTION[dwDelBuffSize];
                                                //
launch DeliveryWorkerThread to perform actual delivery
txns

          for(i=0; i<dwNumDeliveryThreads; i++)
                                                {

          pDeliHandles[i] = (HANDLE) _beginthread(
DeliveryWorkerThread, 0, NULL );

          if (pDeliHandles[i] == INVALID_HANDLE_VALUE)

                    throw new CWEBCLNT_ERR(
ERR_DELIVERY_THREAD_FAILED );
                                                }
                                        }

                                        break;

                                case DLL_PROCESS_DETACH:
                                        if
(dwNumDeliveryThreads)
                                        {
                                                if
(txnDelilog != NULL)
                                                {

          //write event into txn log for STOP

          txnDelilog->WriteCtrlRecToLog(TXN_EVENT_STOP,
szMyComputerName, sizeof(szMyComputerName));


          // This will do a clean shutdown of the
delivery log file

          CTxnLog   *txnDelilogLocal = txnDelilog;

          txnDelilog= NULL;

          delete txnDelilogLocal;
                                                }


          delete [] pDeliHandles;

          delete [] pDelBuff;


          CloseHandle( hWorkerSemaphore );

          CloseHandle( hDoneEvent );

          DeleteCriticalSection(&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);
          lstrcpyn(pVer->lpszExtensionDesc, "TPC-C
Server.", HSE_MAX_EXT_DLL_NAME_LEN);

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

          return TRUE;
}

/* FUNCTION: TerminateExtension
 *
 * PURPOSE:         This function is called by the inet
service when the DLL is about to be unloaded.
 *                          Release all resources in
anticipation of being unloaded.
 *
 * RETURNS:                    TRUE      inet service
expected return value.
 */

BOOL WINAPI TerminateExtension( DWORD dwFlags )
{
          if (pDeliHandles)
          {
                    SetEvent( hDoneEvent );
                    for(DWORD i=0;
i<dwNumDeliveryThreads; i++)
                    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, Reg.szSPPrefix );
                else if (Reg.eDB_Protocol == DBLIB)
                                pTxn = pCTPCC_DBLIB_new(
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );
                pDeliveryData = pTxn-
>BuffAddr_Delivery();
        }
        catch (CBaseErr *e)
        {
                char szTmp[1024];
                wsprintf( szTmp, "Error in Delivery
Txn thread.  Could not connect to database.  "
                                                "%s.
Server=%s, User=%s, Password=%s, Database=%s",
                                                e->ErrorText(),
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
Reg.szDbName );
                WriteMessageToEventLog( szTmp );
                delete e;
                goto ErrorExit;
        }
        catch (...)
        {

        WriteMessageToEventLog(TEXT("Unhandled
exception caught in DeliveryWorkerThread."));
                goto ErrorExit;
        }

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

        txnDeliRec.TxnType =
TXN_REC_TYPE_TPCC_DELIV_DEF;

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

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

        dwDelBuffFreeCount++;

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

        dwDelBuffBusyIndex = 0;


        LeaveCriticalSection(&DelBuffCriticalSection)
;

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

                                                txnDeliRec.w_id
= pDeliveryData->w_id;

        txnDeliRec.o_carrier_id = pDeliveryData-
>o_carrier_id;

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

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

                                                //log txn

        txnDeliRec.TxnStatus = ERR_SUCCESS;
                                                for (int i=0;
i<10; i++)

        txnDeliRec.o_id[i] = pDeliveryData->o_id[i];

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

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

                                                if (txnDelilog
!= NULL)

        txnDelilog->WriteToLog(&txnDeliRec);
                                        }
                                }
                                catch (CBaseErr *e)
```

```
                    {
                              char szTmp[1024];
                              wsprintf( szTmp, "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 successfully
 *
          TRUE      error cannot post delivery info
 */

BOOL PostDeliveryInfo(long 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 relevent
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>"

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

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

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

          "Source:   "__FILE__" ("__TIMESTAMP__") <BR>"

          "</PRE></font>"

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

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

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

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

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

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

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

          sprintf( szTmp,     "Configuration Settings:
<BR><font face=\"Courier New\" color=\"blue\"><PRE>"
                                                  "Txn
Monitor            = <B>%s</B><BR>"

          "Database protocol      = <B>%s</B><BR>"
                                                  "Max
Connections        = <B>%d</B><BR>"
                                                  "# of
Delivery Threads  = <B>%d</B><BR>"
```

```
                                                  "Max
Pending Deliveries = <B>%d</B><BR>"
                        ,
szTxnMonNames[Reg.eTxnMon],
szDBNames[Reg.eDB_Protocol],
                                Reg.dwMaxConnections,
dwNumDeliveryThreads, dwDelBuffSize );
          strcat( szBuffer, szTmp);

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

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

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

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

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

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

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

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

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

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

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

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

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

          "</PRE></font>"
                                        ,
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
Reg.szDbName );
          strcat( szBuffer, szTmp);

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

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

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

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

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

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

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

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

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

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

          EnterCriticalSection(&TermCriticalSection);

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

```
        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>"
                        , iTotal );
}

char *CWEBCLNT_ERR::ErrorText()
{
        static SERRORMSG errorMsgs[] =
        {
                {       ERR_COMMAND_UNDEFINED,

        "Command undefined."

                                                },
                {       ERR_D_ID_INVALID,

        "Invalid District ID Must be 1 to 10."

                },
                {
        ERR_DELIVERY_CARRIER_ID_RANGE,
                "Delivery Carrier ID out of range
must be 1 - 10."
        },
                {
        ERR_DELIVERY_CARRIER_INVALID,
        "Delivery Carrier ID invalid must be numeric
1 - 10."                        },
                {
        ERR_DELIVERY_MISSING_OCD_KEY,
        "Delivery missing Carrier ID key \"OCD*\"."

                },
                {
        ERR_DELIVERY_THREAD_FAILED,
                "Could not start delivery worker
thread."
                },
                {       ERR_GETPROCADDR_FAILED,

        "Could not map proc in DLL.  GetProcAddr
error.  DLL="                                   },
                {       ERR_HTML_ILL_FORMED,

        "Required key field is missing from HTML
string."                                        },
                {
        ERR_INVALID_SYNC_CONNECTION,
        "Invalid Terminal Sync ID."

                                        },
                {       ERR_INVALID_TERMID,

        "Invalid Terminal ID."

                                                },
                {       ERR_LOADDLL_FAILED,
                                                "Load
of DLL failed.  DLL="

                        },
                {
        ERR_MAX_CONNECTIONS_EXCEEDED,
        "No connections available.  Max Connections
is probably too low."   },
                {
        ERR_MISSING_REGISTRY_ENTRIES,
        "Required registry entries are missing.
Rerun INSTALL to correct."    },
                {
        ERR_NEWORDER_CUSTOMER_INVALID,
                "New Order customer id invalid data
type, range = 1 to 3000."               },
                {
        ERR_NEWORDER_CUSTOMER_KEY,
                "New Order missing Customer key
\"CID*\"."
                        },
                {
        ERR_NEWORDER_DISTRICT_INVALID,
                "New Order District ID Invalid
range 1 - 10."
                },
                {
        ERR_NEWORDER_FORM_MISSING_DID,
                "New Order missing District key
\"DID*\"."
                        },
                {
        ERR_NEWORDER_ITEMID_INVALID,
        "New Order Item Id is wrong data type, must
be numeric."                            },
                {
        ERR_NEWORDER_ITEMID_RANGE,
                "New Order Item Id is out of range.
Range = 1 to 999999."                           },
                {
        ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
        "New Order Item_Id field entered without a
corresponding Supp_W."          },
                {
        ERR_NEWORDER_MISSING_IID_KEY,
        "New Order missing Item Id key \"IID*\"."

                },
                {
        ERR_NEWORDER_MISSING_QTY_KEY,
        "New Order Missing Qty key \"Qty##*\"."

                },
                {
        ERR_NEWORDER_MISSING_SUPPW_KEY,
                "New Order missing Supp_W key
\"SP###*\"."
                        },
                {
        ERR_NEWORDER_NOITEMS_ENTERED,
        "New Order No order lines entered."

                                },
                {       ERR_NEWORDER_QTY_INVALID,
                                        "New Order Qty
invalid must be numeric range 1 - 99."
                },
                {       ERR_NEWORDER_QTY_RANGE,
                                                "New
Order Qty is out of range. Range = 1 to 99."
                                        },
                {
        ERR_NEWORDER_QTY_WITHOUT_SUPPW,
                "New Order Qty field entered
without a corresponding Supp_W."                },
                {
        ERR_NEWORDER_SUPPW_INVALID,
                "New Order Supp_W invalid data type
must be numeric."                               },
                {       ERR_NO_SERVER_SPECIFIED,
                                        "No Server name
specified."

                },
                {
        ERR_ORDERSTATUS_CID_AND_CLT,
        "Order Status Only Customer ID or Last Name
may be entered, not both."      },
                {
        ERR_ORDERSTATUS_CID_INVALID,
        "Order Status Customer ID invalid, range must
be numeric 1 - 3000."           },
                {
        ERR_ORDERSTATUS_CLT_RANGE,
                "Order Status Customer last name
longer than 16 characters."             },
                {
        ERR_ORDERSTATUS_DID_INVALID,
        "Order Status District invalid, value must be
numeric 1 - 10."                },
                {
        ERR_ORDERSTATUS_MISSING_CID_CLT,
        "Order Status Either Customer ID or Last Name
must be entered."               },
                {
        ERR_ORDERSTATUS_MISSING_CID_KEY,
        "Order Status missing Customer key \"CID*\"."

        },
                {
        ERR_ORDERSTATUS_MISSING_CLT_KEY,
        "Order Status missing Customer Last Name key
\"CLT*\"."                                      },
                {
        ERR_ORDERSTATUS_MISSING_DID_KEY,
        "Order Status missing District key \"DID*\"."

        },
                {       ERR_PAYMENT_CDI_INVALID,
                                        "Payment
Customer district invalid must be numeric."
                        },
                {       ERR_PAYMENT_CID_AND_CLT,
                                        "Payment Only
Customer ID or Last Name may be entered, not both."
        },
                {
        ERR_PAYMENT_CUSTOMER_INVALID,
        "Payment Customer data type invalid, must be
numeric."                                       },
                {       ERR_PAYMENT_CWI_INVALID,
                                        "Payment
Customer Warehouse invalid, must be numeric."
                        },
                {
        ERR_PAYMENT_DISTRICT_INVALID,
        "Payment District ID is invalid, must be 1 -
10."                                            },
                {       ERR_PAYMENT_HAM_INVALID,
                                        "Payment Amount
invalid data type must be numeric."
                        },
                {       ERR_PAYMENT_HAM_RANGE,

        "Payment Amount out of range, 0 - 9999.99."

                },
                {
        ERR_PAYMENT_LAST_NAME_TO_LONG,
                "Payment Customer last name longer
```

```
than 16 characters."
          },
                    {
          ERR_PAYMENT_MISSING_CDI_KEY,
          "Payment missing Customer district key
\"CDI*\"."
          },
                    {
          ERR_PAYMENT_MISSING_CID_CLT,
          "Payment Either Customer ID or Last Name must
be entered."                              },
                    {
          ERR_PAYMENT_MISSING_CID_KEY,
          "Payment missing Customer Key \"CID*\"."

                    },
                    {
          ERR_PAYMENT_MISSING_CLT_KEY,
          "Payment missing Customer Last Name key
\"CLT*\"."
          },
                    {
          ERR_PAYMENT_MISSING_CWI_KEY,
          "Payment missing Customer Warehouse key
\"CWI*\"."
          },
                    {
          ERR_PAYMENT_MISSING_DID_KEY,
          "Payment missing District Key \"DID*\"."

                    },
                    {
          ERR_PAYMENT_MISSING_HAM_KEY,
          "Payment missing Amount key \"HAM*\"."

                    },
                    {
          ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
          "Stock Level; missing Threshold key \"TT*\"."

          },
                    {
          ERR_STOCKLEVEL_THRESHOLD_INVALID,
          "Stock Level; Threshold value must be in the
range = 1 - 99."          },
                    {
          ERR_STOCKLEVEL_THRESHOLD_RANGE,
                    "Stock Level Threshold out of
range, range must be 1 - 99."                         },
                    {         ERR_VERSION_MISMATCH,

          "Invalid version field.  RTE and Web Client
are probably out of sync." },
                    {         ERR_W_ID_INVALID,

          "Invalid Warehouse ID."

                                                  },
                    {         0,

                              ""



                                                  }
          };

          char szTmp[256];
          int i = 0;
          while (TRUE)
          {
                    if (errorMsgs[i].szMsg[0] == 0)
                    {
                              strcpy( szTmp, "Unknown
error number." );
                              break;
                    }
                    if (m_Error == errorMsgs[i].iError)
                    {
                              strcpy( szTmp,
errorMsgs[i].szMsg );
                              break;
                    }
                    i++;
          }

          if (m_szTextDetail)
                    strcat( szTmp, m_szTextDetail );
          if (m_SystemErr)
                    wsprintf( szTmp+strlen(szTmp), "
Error=%d", m_SystemErr );

          m_szErrorText = new char[strlen(szTmp)+1];
          strcpy( m_szErrorText, szTmp );
          return m_szErrorText;

}

/* FUNCTION: GetKeyValue
 *
 * PURPOSE:         This function parses a http
formatted string for specific key values.
 *
 * ARGUMENTS:       char
          *pQueryString       http string from client
browser
 *                                   char
                    *pKey                         key
value to look for
 *                                   char
                    *pValue
          character array into which to place key's
value
 *                                   int
                    iMax
          maximum length of key value array.
 *                                   WEBERROR
          err                                     error
value to throw
 *
 * RETURNS:                  nothing.
 *
 * ERROR:           if (the pKey value is not found)
then
 *                                             if
(err == 0)
 *
          return (empty string)
 *                                             else
 *
          throw CWEBCLNT_ERR(err)
 *
 * COMMENTS:        http keys are formatted either
KEY=value& or KEY=value\0. This DLL formats
 *                                   TPC-C input
fields in such a manner that the keys can be extracted
in the
 *                                   above manner.
 */

void GetKeyValue(char **pQueryString, char *pKey, char
*pValue, int iMax, WEBERROR err)
{
          char *ptr;

          if ( !(ptr=strstr(*pQueryString, pKey)) )
                    goto ErrorExit;
          ptr += strlen(pKey);
          if ( *ptr != '=' )
                    goto ErrorExit;
          ptr++;

          iMax--;  // one position is for terminating
null
          while( *ptr && *ptr != '&' && iMax)
          {
                    *pValue++ = *ptr++;
                    iMax--;
          }
          *pValue = 0;  // terminating null

          *pQueryString = ptr;
          return;

ErrorExit:
          if (err != NO_ERR)
                    throw new CWEBCLNT_ERR( err );
          *pValue = 0;  // return empty result string
}

/* FUNCTION: GetIntKeyValue
 *
 * PURPOSE:         This function parses a http
formatted string for a specific key value.
 *
 * ARGUMENTS:       char
          *pQueryString       http string from client
browser
 *                                   char
                    *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=\"TERMID\" 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=\"TERMID\" 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=\"TERMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                "<PRE><font face=\"Courier\">
Stock-Level<BR>"
                "Warehouse: %6.6d   District:
%2.2d<BR> <BR>",
                STOCK_LEVEL_FORM, iTermId,
Term.pClientData[iTermId].iSyncId,
                Term.pClientData[iTermId].w_id,
Term.pClientData[iTermId].d_id);

        if ( bInput )
        {
                strcpy(szForm+c,
                                "Stock Level Threshold:
<INPUT NAME=\"TT*\" SIZE=2><BR> <BR>"
                                "low stock:
</font><BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR>"
                                " <BR> <BR> <BR> <BR>
<BR> <BR> <BR></PRE><HR>"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
                                "</FORM></HTML>" );
        }
        else
        {
                wsprintf(szForm+c,
                                "Stock Level Threshold:
%2.2d<BR> <BR>"
                                "low stock: %3.3d</font>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                                " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR></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>
<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=\"TERMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                "<PRE><font face=\"Courier\">
New Order<BR>"
```

```
                    , bValid ? 0 : ERR_BAD_ITEM_ID,
NEW_ORDER_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);

          if ( bInput )
          {
                    c += wsprintf(szForm+c, "Warehouse:
%6.6d   ", Term.pClientData[iTermId].w_id );

                    strcpy( szForm+c,
                              "District: <INPUT
NAME=\"DID*\" SIZE=1>                         Date:<BR>"
                              "Customer:  <INPUT
NAME=\"CID*\" SIZE=4>   Name:
Credit:      %Disc:<BR>"
                              "Order Number:
Number of Lines:           W_tax:         D_tax:<BR>
<BR>"
                              " Supp_W  Item_Id  Item
Name                  Qty  Stock  B/G  Price
Amount<BR>"
                              "  <INPUT NAME=\"SP00*\"
SIZE=4>  <INPUT NAME=\"IID00*\" SIZE=6>
<INPUT NAME=\"Qty00*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP01*\"
SIZE=4>  <INPUT NAME=\"IID01*\" SIZE=6>
<INPUT NAME=\"Qty01*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP02*\"
SIZE=4>  <INPUT NAME=\"IID02*\" SIZE=6>
<INPUT NAME=\"Qty02*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP03*\"
SIZE=4>  <INPUT NAME=\"IID03*\" SIZE=6>
<INPUT NAME=\"Qty03*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP04*\"
SIZE=4>  <INPUT NAME=\"IID04*\" SIZE=6>
<INPUT NAME=\"Qty04*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP05*\"
SIZE=4>  <INPUT NAME=\"IID05*\" SIZE=6>
<INPUT NAME=\"Qty05*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP06*\"
SIZE=4>  <INPUT NAME=\"IID06*\" SIZE=6>
<INPUT NAME=\"Qty06*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP07*\"
SIZE=4>  <INPUT NAME=\"IID07*\" SIZE=6>
<INPUT NAME=\"Qty07*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP08*\"
SIZE=4>  <INPUT NAME=\"IID08*\" SIZE=6>
<INPUT NAME=\"Qty08*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP09*\"
SIZE=4>  <INPUT NAME=\"IID09*\" SIZE=6>
<INPUT NAME=\"Qty09*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP10*\"
SIZE=4>  <INPUT NAME=\"IID10*\" SIZE=6>
<INPUT NAME=\"Qty10*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP11*\"
SIZE=4>  <INPUT NAME=\"IID11*\" SIZE=6>
<INPUT NAME=\"Qty11*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP12*\"
SIZE=4>  <INPUT NAME=\"IID12*\" SIZE=6>
<INPUT NAME=\"Qty12*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP13*\"
SIZE=4>  <INPUT NAME=\"IID13*\" SIZE=6>
<INPUT NAME=\"Qty13*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP14*\"
SIZE=4>  <INPUT NAME=\"IID14*\" SIZE=6>
<INPUT NAME=\"Qty14*\" SIZE=1><BR>"
                              "Execution Status:
Total:<BR>"
                              "</font></PRE><HR>"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
                              "</FORM></HTML>"
                              );
          }
          else
          {
                    c += wsprintf(szForm+c, "Warehouse:
%6.6d   District: %2.2d                    Date: ",
                              pNewOrderData->w_id,
                              pNewOrderData->d_id);

                    if ( bValid )
                    {
                              c += wsprintf(szForm+c,
"%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
                                        pNewOrderData-
>o_entry_d.day,
                                        pNewOrderData-
>o_entry_d.month,
                                        pNewOrderData-
>o_entry_d.year,
                                        pNewOrderData-
>o_entry_d.hour,
                                        pNewOrderData-
>o_entry_d.minute,
                                        pNewOrderData-
>o_entry_d.second);
                    }

                    c += wsprintf(szForm+c,
"<BR>Customer:  %4.4d   Name: %-16s   Credit: %-2s   ",
                              pNewOrderData->c_id,
pNewOrderData->c_last, pNewOrderData->c_credit);

                    if ( bValid )
                    {
                              c += sprintf(szForm+c,

          "%%Disc: %5.2f            <BR>"

          "Order Number: %8.8d  Number of Lines: %2.2d
W_tax: %5.2f   D_tax: %5.2f  <BR> <BR>"

          " Supp_W  Item_Id  Item Name
Qty  Stock  B/G  Price    Amount<BR>",

          100.0*pNewOrderData->c_discount,
                              pNewOrderData-
>o_id,
                              pNewOrderData-
>o_ol_cnt,
                              100.0 *
pNewOrderData->w_tax,
                              100.0 *
pNewOrderData->d_tax);

                              for(i=0; i<pNewOrderData-
>o_ol_cnt; i++)
                              {
                                        c +=
sprintf(szForm+c, "%6.6d   %6.6d   %-24s  %2.2d
%3.3d    %1.1s   $%6.2f  $%7.2f  <BR>",

          pNewOrderData->OL[i].ol_supply_w_id,

          pNewOrderData->OL[i].ol_i_id,

          pNewOrderData->OL[i].ol_i_name,

          pNewOrderData->OL[i].ol_quantity,

          pNewOrderData->OL[i].ol_stock,

          pNewOrderData->OL[i].ol_brand_generic,

          pNewOrderData->OL[i].ol_i_price,

          pNewOrderData->OL[i].ol_amount );
                              }
                    }
                    else
                    {
                              c += wsprintf(szForm+c,
                                        "%Disc:<BR>"
                                        "Order Number:
%8.8d  Number of Lines:              W_tax:
D_tax:<BR> <BR>"
                                        " Supp_W
Item_Id  Item Name                 Qty  Stock  B/G
Price    Amount<BR>"
                                        ,
pNewOrderData->o_id);

                              i = 0;
                    }

                    strncpy( szForm+c, szBR, (15-i)*5
);
                    c += (15-i)*5;

                    if ( bValid )
                              c += sprintf(szForm+c,
"Execution Status: Transaction commited.
Total:  $%8.2f  ",
                                        pNewOrderData-
>total_amount);
                    else
                              c += wsprintf(szForm+c,
"Execution Status: Item number is not valid.
Total:");

                    strcpy(szForm+c,
                              " <BR></font></PRE><HR>"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
                              "</FORM></HTML>"
                              );
          }
}

/* FUNCTION: MakePaymentForm
 *
 * COMMENTS:        The internal client buffer is
created when the terminal id is assigned and should not
 *                                        be freed except
when the client terminal id is no longer needed.
 */

void MakePaymentForm(int iTermId, PAYMENT_DATA
*pPaymentData, BOOL bInput, char *szForm)
{
          int c;
```

```
        c = wsprintf(szForm,
                        "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>"
                        "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%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 += wsprintf(szForm+c, "%2.2d-
%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
                                pPaymentData->h_date.day,
                                pPaymentData-
>h_date.month,
                                pPaymentData-
>h_date.year,
                                pPaymentData-
>h_date.hour,
                                pPaymentData-
>h_date.minute,
                                pPaymentData-
>h_date.second);
        }

        if ( bInput )
        {
                c += wsprintf(szForm+c,
                                "<BR> <BR>Warehouse:
%6.6d"
                                "
District: <INPUT NAME=\"DID*\" SIZE=1><BR> <BR> <BR>
<BR> <BR>"
                                "Customer: <INPUT
NAME=\"CID*\" SIZE=4>"
                                "Cust-Warehouse: <INPUT
NAME=\"CWI*\" SIZE=4>  "
                                "Cust-District: <INPUT
NAME=\"CDI*\" SIZE=1><BR>"
                                "Name:
<INPUT NAME=\"CLT*\" SIZE=16>
Since:<BR>"
                                "
Credit:<BR>"
                                "
Disc:<BR>"
                                "
Phone:<BR> <BR>"
                                "Amount Paid:
$<INPUT NAME=\"HAM*\" SIZE=7>      New Cust-
Balance:<BR>"
                                "Credit Limit:<BR>
<BR>Cust-Data: <BR> <BR> <BR> <BR>
<BR></font></PRE><HR>"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\"><INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
                                "</BODY></FORM></HTML>"
Term.pClientData[iTermId].w_id);
        }
        else
        {
                c += wsprintf(szForm+c,
                                "<BR> <BR>Warehouse:
%6.6d                           District: %2.2d<BR>"
                                "%-20s
%-20s<BR>"
                                "%-20s
%-20s<BR>"
                                "%-20s %-2s %5.5s-%4.4s
%-20s %-2s %5.5s-%4.4s<BR> <BR>"
                                "Customer: %4.4d  Cust-
Warehouse: %6.6d  Cust-District: %2.2d<BR>"
                                "Name:   %-16s %-2s %-16s
Since:  %2.2d-%2.2d-%4.4d<BR>"
                                "       %-20s
Credit: %-2s<BR>"

Term.pClientData[iTermId].w_id, pPaymentData->d_id
                                , pPaymentData-
>w_street_1, pPaymentData->d_street_1
                                , pPaymentData-
>w_street_2, pPaymentData->d_street_2
                                , pPaymentData->w_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>            %-
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                     i, c;
        static char szBR[] = " <BR> <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=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
```

```
                    "<PRE><font face=\"Courier\">
Order-Status<BR>"
                    "Warehouse: %6.6d   ",
                    ORDER_STATUS_FORM, iTermId,
Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id);

          if ( bInput )
          {
                    strcpy(szForm+c,
                              "District: <INPUT
NAME=\"DID*\" SIZE=1><BR>"
                              "Customer: <INPUT
NAME=\"CID*\" SIZE=4>   Name:
<INPUT NAME=\"CLT*\" SIZE=23><BR>"
                              "Cust-Balance:<BR> <BR>"
                              "Order-Number:
Entry-Date:                       Carrier-Number:<BR>"
                              "Supply-W     Item-Id
Qty      Amount       Delivery-Date<BR> <BR> <BR> <BR>
<BR>"
                              " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR></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 += wsprintf(szForm+c,
                              "Order-Number: %8.8d
Entry-Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d
Carrier-Number: %2.2d<BR>"
                              "Supply-W     Item-Id
Qty      Amount       Delivery-Date<BR>",
                              pOrderStatusData->o_id,
                              pOrderStatusData-
>o_entry_d.day,
                              pOrderStatusData-
>o_entry_d.month,
                              pOrderStatusData-
>o_entry_d.year,
                              pOrderStatusData-
>o_entry_d.hour,
                              pOrderStatusData-
>o_entry_d.minute,
                              pOrderStatusData-
>o_entry_d.second,
                              pOrderStatusData-
>o_carrier_id);

                    for(i=0; i< pOrderStatusData-
>o_ol_cnt; i++)
                    {
                              c += sprintf(szForm+c, "
%6.6d     %6.6d     %2.2d     $%8.2f      %2.2d-%2.2d-
%4.4d<BR>",

                    pOrderStatusData->OL[i].ol_supply_w_id,

                    pOrderStatusData->OL[i].ol_i_id,

                    pOrderStatusData->OL[i].ol_quantity,

                    pOrderStatusData->OL[i].ol_amount,

                    pOrderStatusData->OL[i].ol_delivery_d.day,

                    pOrderStatusData->OL[i].ol_delivery_d.month,

                    pOrderStatusData->OL[i].ol_delivery_d.year);
                              }

                              strncpy( szForm+c, szBR, (15-i)*5
);
                              c += (15-i)*5;

                              strcpy(szForm+c,
                                        "</font></PRE><HR><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
                                        "</BODY></FORM></HTML>"
);
          }
}

/* FUNCTION: MakeDeliveryForm
 *
 * COMMENTS:        The internal client buffer is
created when the terminal id is assigned and should not
 *                                        be freed except
when the client terminal id is no longer needed.
 */

void MakeDeliveryForm(int iTermId, DELIVERY_DATA
*pDeliveryData, BOOL bInput, char *szForm)
{
          int       c;

          c = wsprintf(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: %6.6d<BR> <BR>",
                    (!bInput && (pDeliveryData-
>exec_status_code != eOK)) ? ERR_TYPE_DELIVERY_POST :
0,
                    DELIVERY_FORM, iTermId,
Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id);

          if ( bInput )
          {
                    strcpy( szForm+c,
                              "Carrier Number: <INPUT
NAME=\"OCD*\" SIZE=1><BR> <BR>"
                              "Execution Status: <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                              " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> </font></PRE><HR>"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
                              "</BODY></FORM></HTML>"
);
          }
          else
          {
                    wsprintf( 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> <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 CWEBCLNT_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
CWEBCLNT_ERR( ERR_NEWORDER_SUPPW_INVALID );
                              pNewOrderData-
>OL[items].ol_supply_w_id = atoi(szTmp);

                              ol_i_id = pNewOrderData-
>OL[items].ol_i_id =

          GetIntKeyValue(&ptr, szIID[i],
ERR_NEWORDER_MISSING_IID_KEY,
ERR_NEWORDER_ITEMID_INVALID);
                              if ( ol_i_id > 999999 ||
ol_i_id < 1 )
                                        throw new
CWEBCLNT_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
CWEBCLNT_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
CWEBCLNT_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_ol_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;
}


isapi_dll/src/resource.h

//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by tpcc.rc
//
#define IDD_DIALOG1                     101

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        102
#define _APS_NEXT_COMMAND_VALUE         40001
#define _APS_NEXT_CONTROL_VALUE         1000
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

common/src/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 ( !stricmp(szTmp,
szDBNames[ODBC]) )
                        pReg->eDB_Protocol =
ODBC;
                else if ( !stricmp(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 ( !stricmp(szTmp,
szTxnMonNames[TUXEDO]) )
```

```
                        pReg->eTxnMon = TUXEDO;
                else if ( !stricmp(szTmp,
szTxnMonNames[ENCINA]) )
                        pReg->eTxnMon = ENCINA;
                else if ( !stricmp(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 ( !stricmp(szTmp, "YES") )
                        pReg->bCOM_SinglePool =
TRUE;
        }

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

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

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

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

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

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

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

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

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

        RegCloseKey(hKey);

        return FALSE;
}
```

common/src/ReadRegistry.h

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

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

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

//This structure defines the data necessary to keep
distinct for each terminal or client connection.
typedef struct _TPCCREGISTRYDATA
{
        enum DBPROTOCOL eDB_Protocol;
        enum TXNMON eTxnMon;
        BOOL bCOM_SinglePool;
        DWORD dwMaxConnections;
        DWORD dwMaxPendingDeliveries;
        DWORD dwNumberOfDeliveryThreads;
        char szPath[128];
        char szDbServer[32];
        char szDbName[32];
        char szDbUser[32];
        char szDbPassword[32];
        wchar_t szSPPrefix[32];       //tpcc_odbc.dll
stored procedures prefix
} TPCCREGISTRYDATA, *PTPCCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg
);
```

common/src/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 message
        char    szMsg[256];
        //message to sent to browser
} 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
#define ERR_TYPE_SOAP_HTTP
                                26
        //HTTP/SOAP dll generated error
// 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_TPCW_ODBC
                                56
#define ERR_TYPE_SCHANNEL
                                57
#define ERR_TYPE_THINK_LIST
                                58

#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 receive HTML pages."


class CBaseErr
{
public:

          CBaseErr(LPCTSTR szLoc = NULL)
          {
                    m_idMsg             =
GetLastError();     //take the error code immediatelly
before it is reset by other functions


                    if (szLoc)
                    {
                              m_szLoc = new
char[strlen(szLoc)+1/*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[strlen(szLoc)+1/*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,
                    eWSAGetOverlappedResult,
                    eWSARecv,
                    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,
                    eReleaseSemaphore,
                    eFSeek,
                    eFRead,
                    eFWrite,
                    eTmpFile,
                    eSetFilePointer,
                    eNew,
                    eCloseHandle,
          };

                              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,LPTSTR);

          int ErrorType() {return ERR_BUF_OVERFLOW;}

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

common/src/trans.h

```
/*        FILE:               TRANS.H
 *                                        Microsoft TPC-C
Kit Ver. 4.42.000
 *                                        Copyright
Microsoft, 2002
 *             All Rights Reserved
 *
 *                                        Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
 *
 *        PURPOSE:  Header file for TPC-C structure
templates.
 *
 *  Change history:
 *                  4.42.000 - changed w_id fields from
short to long to support >32K warehouses
 *                  4.20.000 - updated rev number to
match kit
 */
#pragma once

// String length constants
#define SERVER_NAME_LEN           20
#define DATABASE_NAME_LEN         20
#define USER_NAME_LEN             20
#define PASSWORD_LEN              20
#define TABLE_NAME_LEN            20
#define I_DATA_LEN                50
#define I_NAME_LEN                24
#define BRAND_LEN                  1
#define LAST_NAME_LEN             16
#define W_NAME_LEN                10
#define ADDRESS_LEN               20
#define STATE_LEN                  2
#define ZIP_LEN                    9
#define S_DIST_LEN                24
#define S_DATA_LEN                50
#define D_NAME_LEN                10
#define FIRST_NAME_LEN            16
#define MIDDLE_NAME_LEN            2
#define PHONE_LEN                 16
#define DATETIME_LEN              30
#define CREDIT_LEN                 2
#define C_DATA_LEN               250
#define H_DATA_LEN                24
#define DIST_INFO_LEN             24
#define MAX_OL_NEW_ORDER_ITEMS    15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN                25
#define OL_DIST_INFO_LEN                    24

// TIMESTAMP_STRUCT is provided by the ODBC header file
sqltypes.h, but is not available
// when compiling with dblib, so redefined here.  Note:
we are using the symbol "__SQLTYPES"
// (declared in sqltypes.h) as a way to determine if
TIMESTAMP_STRUCT has been declared.
#ifndef __SQLTYPES
          typedef struct
          {
                              short
          /* SQLSMALLINT */   year;
                              unsigned short      /*
SQLUSMALLINT */     month;
                              unsigned short      /*
SQLUSMALLINT */     day;
                              unsigned short      /*
SQLUSMALLINT */     hour;
                              unsigned short      /*
SQLUSMALLINT */     minute;
                              unsigned short      /*
SQLUSMALLINT */     second;
                              unsigned long       /*
SQLUINTEGER */      fraction;
          } TIMESTAMP_STRUCT;
#endif

// possible values for exec_status_code after
transaction completes
enum EXEC_STATUS
{
          eOK,                          // 0
          "Transaction commited."
          eInvalidItem,       // 1      "Item number is
not valid."
          eDeliveryFailed     // 2      "Delivery Post
Failed."
};

// transaction structures
typedef struct
{
          // input params
          long
          ol_supply_w_id;
          long
          ol_i_id;
          short
          ol_quantity;

          // output params
```

```
        char
        ol_i_name[I_NAME_LEN+1];
        char
        ol_brand_generic[BRAND_LEN+1];
        double
        ol_i_price;
        double
        ol_amount;
        short
        ol_stock;
} OL_NEW_ORDER_DATA;

typedef struct
{
        // input params
        long                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
        long
w_id;
        short
d_id;
        long
c_id;
        short
c_d_id;
        long
c_w_id;
        double
h_amount;
        char
c_last[LAST_NAME_LEN+1];

        // output params
        EXEC_STATUS
exec_status_code;
        TIMESTAMP_STRUCT     h_date;
        char
w_street_1[ADDRESS_LEN+1];
        char
w_street_2[ADDRESS_LEN+1];
        char
w_city[ADDRESS_LEN+1];
        char
w_state[STATE_LEN+1];
        char
w_zip[ZIP_LEN+1];
        char
d_street_1[ADDRESS_LEN+1];
        char
d_street_2[ADDRESS_LEN+1];
        char
d_city[ADDRESS_LEN+1];
        char
d_state[STATE_LEN+1];
        char
d_zip[ZIP_LEN+1];
        char
c_first[FIRST_NAME_LEN+1];
        char
c_middle[MIDDLE_NAME_LEN + 1];
        char
c_street_1[ADDRESS_LEN+1];
        char
c_street_2[ADDRESS_LEN+1];
        char
c_city[ADDRESS_LEN+1];
        char
c_state[STATE_LEN+1];
        char
c_zip[ZIP_LEN+1];
        char
c_phone[PHONE_LEN+1];
        TIMESTAMP_STRUCT     c_since;
        char
c_credit[CREDIT_LEN+1];
        double
c_credit_lim;
        double
c_discount;
        double
c_balance;
        char
c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

typedef struct
{
        long
        ol_i_id;
        long                            ol_supply_w_id;
        short
        ol_quantity;
        double
        ol_amount;
        TIMESTAMP_STRUCT    ol_delivery_d;
} OL_ORDER_STATUS_DATA;

typedef struct
{       // input params
        long                w_id;
        short               d_id;
        long                c_id;
        char                c_last[LAST_NAME_LEN+1];

        // output params
        EXEC_STATUS
exec_status_code;
        char
c_first[FIRST_NAME_LEN+1];
        char
c_middle[MIDDLE_NAME_LEN+1];
        double              c_balance;
        long                o_id;
        TIMESTAMP_STRUCT    o_entry_d;
        short               o_carrier_id;
        OL_ORDER_STATUS_DATA
OL[MAX_OL_ORDER_STATUS_ITEMS];
        short               o_ol_cnt;
} ORDER_STATUS_DATA, *PORDER_STATUS_DATA;

typedef struct
{       // input params
        long                            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 for districts 1 to 10
} DELIVERY_DATA, *PDELIVERY_DATA;

//This structure is used for posting delivery
transactions and for writing them to the delivery
server.
typedef struct _DELIVERY_TRANSACTION
{
        SYSTEMTIME          queue;
        //time delivery transaction queued
        long                w_id;
        //delivery warehouse
        short               o_carrier_id;
        //carrier id
} DELIVERY_TRANSACTION;

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

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

common/src/txn_base.h

```
/*      FILE:               TXN_BASE.H
 *                                      Microsoft TPC-C
Kit Ver. 4.20.000
 *                                      Copyright
Microsoft, 1999
 *              All Rights Reserved
 *
 *                                      Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
 *
 *      PURPOSE:  Header file for TPC-C txn class
implementation.
 *
 *  Change history:
 *                  4.20.000 - updated rev number to
match kit
 */

#pragma once

// need to declare functions for import, unless define
has already been created
```

```
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class DllDecl CTPCC_BASE
{
          public:
                    CTPCC_BASE(void) {};
                    virtual ~CTPCC_BASE(void) {};

                    virtual PNEW_ORDER_DATA
          BuffAddr_NewOrder()           = 0;
                    virtual PPAYMENT_DATA
          BuffAddr_Payment()            = 0;
                    virtual PDELIVERY_DATA
          BuffAddr_Delivery()           = 0;
                    virtual PSTOCK_LEVEL_DATA
          BuffAddr_StockLevel()         = 0;
                    virtual PORDER_STATUS_DATA
          BuffAddr_OrderStatus()        = 0;

                    virtual void NewOrder
          () = 0;
                    virtual void Payment
          () = 0;
                    virtual void Delivery
          () = 0;
                    virtual void StockLevel
          () = 0;
                    virtual void OrderStatus      () =
0;
};
```

db_dblib_dll/src/tpcc_dblib.cpp

```
/*        FILE:               TPCC_DBLIB.CPP
 *                                  Microsoft TPC-C
Kit Ver. 4.42.000
 *                                  Copyright
Microsoft, 2002
 *              All Rights Reserved
 *
 *                                  Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
 *
 *        PURPOSE:  Implements dblib calls for TPC-C
txns.
 *        Contact:  Charles Levine
(clevine@microsoft.com)
 *
 *  Change history:
 *                  4.42.000 - changed w_id fields from
short to long to support >32K warehouses
 *                  4.20.000 - updated rev number to
match kit
 *                  4.10.001 - not deleting error class
in catch handler on deadlock retry;
 *                                    not a
functional bug, but a memory leak
 *                                  - had to tweak
some declarations to compile with latest SDK; no
functional change
 */

#include <windows.h>
#include <stdio.h>
#include <assert.h>

#define DBNTWIN32
#include <sqlfront.h>
#include <sqldb.h>

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

// need to declare functions for export
#define DllDecl  __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_dblib.h"

#define   DEFCLPACKSIZE                           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 iErrOleDbProvider = 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
                    *msgtext            printable
message description
 *
 * 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 SERRORMSG errorMsgs[] =
          {
                    { ERR_WRONG_SP_VERSION,
          "Wrong version of stored procs on database
server"   },
                    { ERR_INVALID_CUST,
          "Invalid Customer id,name."
                                                  },
                    { ERR_NO_SUCH_ORDER,
          "No orders found for customer."
                                                  },
                    { ERR_RETRIED_TRANS,
          "Retries before transaction succeeded."
                              },
                    { 0,      ""



          }
          };

          static char szNotFound[] = "Unknown error
number.";

          for(i=0; errorMsgs[i].szMsg[0]; i++)
          {
                    if ( m_errno == errorMsgs[i].iError
)
                              break;
          }
          if ( !errorMsgs[i].szMsg[0] )
                    return szNotFound;
          else
                    return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_DBLIB* CTPCC_DBLIB_new(
          LPCSTR szServer,                  // name of SQL
server
          LPCSTR szUser,                              //
user name for login
          LPCSTR szPassword,                // password for
login
          LPCSTR szHost,                              //
workstation name; shows up in sp_who; max 30 chars,
only first 10 kept by SQL Server
          LPCSTR szDatabase )               // name of
database to use
{
          return new CTPCC_DBLIB( szServer, szUser,
szPassword, szHost, szDatabase );
}


CTPCC_DBLIB::CTPCC_DBLIB (
          LPCSTR szServer,                  // name of SQL
server
          LPCSTR szUser,                              //
user name for login
          LPCSTR szPassword,                // password for
login
          LPCSTR szHost,                              //
workstation name; shows up in sp_who; max 30 chars,
only first 10 kept by SQL Server
          LPCSTR szDatabase )               // name of
database to use
{
          LOGINREC  *login;
          const BYTE          *pData;

          // initialization
          m_dbproc = NULL;
          m_DbLibErr = (CDBLIBERR*)NULL;
          m_SqlErr = (CSQLERR*)NULL;

          m_MaxRetries = 10;                // how many
retries on deadlock

          // increase max number of connections if
getting close
          if ( dbgetmaxprocs() < (iConnectionCount+5) )
          {
                    if (
dbsetmaxprocs(iConnectionCount+10) == FAIL )

          ThrowError(CDBLIBERR::eDbSetMaxProcs);
          }

          // allocate a login structure
          login = dblogin();
          if (login == NULL)
                    ThrowError(CDBLIBERR::eLogin);
          InterlockedIncrement( &iConnectionCount );

          // register error and message handler
functions
          if (dbprocerrhandle(login, err_handler) ==
NULL)

          ThrowError(CDBLIBERR::eDbProcHandler);

          if (dbprocmsghandle(login, msg_handler) ==
NULL)

          ThrowError(CDBLIBERR::eDbProcHandler);


          DBSETLUSER(login, szUser);
          DBSETLPWD(login, szPassword);
          DBSETLHOST(login, szHost);
          DBSETLPACKET(login, (unsigned
short)DEFCLPACKSIZE);
          DBSETLVERSION(login, DBVER60);
          // use dblib ver 6.0 client behavior

          // set time to wait for login
          if (dbsetlogintime(60) == FAIL)
                    ThrowError(CDBLIBERR::eDbSet);

          // set time to wait for statement execution
          if (dbsettime(180) == FAIL)
                    ThrowError(CDBLIBERR::eDbSet);

          m_dbproc = dbopen(login, szServer);

          // deallocate login structure before checking
for success
          dbfreelogin( login );

          if (m_dbproc == NULL)
                    ThrowError(CDBLIBERR::eDbOpen);

          // save address of class instance so that the
message and error handler
          // can get to data.
          dbsetuserdata(m_dbproc, (LPVOID)this);

          // Use the the right database
          if (dbuse(m_dbproc, szDatabase) == FAIL)
                    ThrowError(CDBLIBERR::eDbUse);

          dbcmd(m_dbproc, "set nocount on ");
                    // do not return row counts
          dbcmd(m_dbproc, "set XACT_ABORT ON");
          // rollback transaction on abort

          if (dbsqlexec(m_dbproc) == FAIL)
                    ThrowError(CDBLIBERR::eDbSqlExec);

          DiscardNextResults(2);

          // verify that version of stored procs on
server is correct
          dbrpcinit(m_dbproc, "tpcc_version", 0);

          if (dbrpcexec(m_dbproc) == FAIL)
                    ThrowError(CDBLIBERR::eDbRpcExec);

          if (dbresults(m_dbproc) != SUCCEED)
                    ThrowError(CDBLIBERR::eDbResults);

          if (dbnextrow(m_dbproc) != REG_ROW)
                    ThrowError(CDBLIBERR::eDbNextRow);

          char szSrvVersion[16];
          pData=dbdata(m_dbproc, 1);
          if (pData)
                    UtilStrCpy(szSrvVersion, pData,
dbdatlen(m_dbproc, 1));
          else
                    szSrvVersion[0]=0;
          if (strcmp(szSrvVersion,sVersion))
                    throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_WRONG_SP_VERSION );


          DiscardNextRows(0);
          DiscardNextResults(0);
}


CTPCC_DBLIB::~CTPCC_DBLIB( void )
{
```

```
        // close db connection and deallocate
resources
        dbclose(m_dbproc);
        InterlockedDecrement( &iConnectionCount );
        if (m_DbLibErr != NULL)
                delete m_DbLibErr;
        if (m_SqlErr != NULL)
                delete m_SqlErr;
}

void CTPCC_DBLIB::SetDbLibError(int severity, int
dberr, int oserr, LPCSTR dberrstr, LPCSTR oserrstr)
{
        delete m_DbLibErr;
        m_DbLibErr = new
CDBLIBERR(CDBLIBERR::eUnknown, severity, dberr, oserr);

        if (dberrstr != NULL)
        {
                m_DbLibErr->m_dberrstr = new char[
strlen(dberrstr)+1 ];
                strcpy( m_DbLibErr->m_dberrstr,
dberrstr );
        }

        if (oserrstr != NULL)
        {
                m_DbLibErr->m_oserrstr = new char[
strlen(oserrstr)+1 ];
                strcpy( m_DbLibErr->m_oserrstr,
oserrstr );
        }

}

void CTPCC_DBLIB::SetSqlError( int /*DBINT*/ msgno, int
msgstate, int severity, LPCSTR msgtext )
{
        if (m_SqlErr == NULL)
                m_SqlErr = new CSQLERR();

        m_SqlErr->m_msgno = msgno;
        m_SqlErr->m_msgstate = msgstate;
        m_SqlErr->m_severity = severity;

        delete [] m_SqlErr->m_msgtext;
        if (msgtext != NULL)
        {
                m_SqlErr->m_msgtext = new char[
strlen(msgtext)+1 ];
                strcpy( m_SqlErr->m_msgtext,
msgtext );
        }
}

void CTPCC_DBLIB::ThrowError( CDBLIBERR::ACTION eAction
)
{
        // discard anything still in return buffer
        DiscardNextRows(-1);
        DiscardNextResults(-1);

        // check for SQL Server error first;  if yes,
throw it and ignore any DBLib error.
        if (m_SqlErr != NULL)
        {
                CSQLERR             *pSqlErr;
                pSqlErr = m_SqlErr;
                m_SqlErr = NULL;    // clear our
pointer to instance; catch handler will delete
                throw pSqlErr;
        }
        CDBLIBERR *pDbLibErr;
        if (m_DbLibErr == NULL)
                // this case isn't expected to
happen, since it means that an error was returned
                // 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, SQLINT4, -1, -1, (BYTE *)
&m_txn.StockLevel.w_id);            // @w_id int
                        dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.StockLevel.d_id);            // @d_id
tinyint
                        dbrpcparam(m_dbproc,
NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.StockLevel.threshold);   // @threshhold 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)) &&
                                        (++iTryCount <=
iMaxRetries))
                              {
                                        // hit
deadlock; backoff for increasingly longer period
                                        delete e;
                                        Sleep(10 *
iTryCount);
                              }
                              else
                                        throw;
                    }
          }         // while (TRUE)

          //if (iTryCount)
          //        throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}


void CTPCC_DBLIB::NewOrder()
{
          int                       i;
          DBINT               commit_flag;
          DBDATETIME          datetime;
          DBDATEREC daterec;

          int                       iTryCount = 0;
          const BYTE          *pData;

          ResetError();

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

                              dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.w_id);
                              dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.d_id);
                              dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.c_id);
                              dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.o_ol_cnt);

                              // check whether any
order lines are for a remote warehouse

          m_txn.NewOrder.o_all_local = 1;
                              for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; i++)
                              {
                                        if
(m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
                                        {
          m_txn.NewOrder.o_all_local = 0;  // at least
one remote warehouse

          break;
                                        }
                              }
                              dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.o_all_local);

                              for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; i++)
                              {

          dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -
1, (BYTE *) &m_txn.NewOrder.OL[i].ol_i_id);

          dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -
1, (BYTE *) &m_txn.NewOrder.OL[i].ol_supply_w_id);

          dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -
1, (BYTE *) &m_txn.NewOrder.OL[i].ol_quantity);
                              }

                              if (dbrpcexec(m_dbproc)
== FAIL)

          ThrowError(CDBLIBERR::eDbRpcExec);


                              // Get order line results

          m_txn.NewOrder.total_amount = 0;
                              for (i = 0;
i<m_txn.NewOrder.o_ol_cnt; i++)
                              {
                                        if
(dbresults(m_dbproc) != SUCCEED)

          ThrowError(CDBLIBERR::eDbResults);

                                        if
(dbnumcols(m_dbproc) != 5)

          ThrowError(CDBLIBERR::eWrongNumCols);

                                        if
(dbnextrow(m_dbproc) != REG_ROW)

          ThrowError(CDBLIBERR::eDbNextRow);


          if(pData=dbdata(m_dbproc, 1))

          UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name,
pData, dbdatlen(m_dbproc, 1));

          if(pData=dbdata(m_dbproc, 2))

          m_txn.NewOrder.OL[i].ol_stock = (*(DBSMALLINT
*) pData);

          if(pData=dbdata(m_dbproc, 3))

          UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_gene
ric, 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   =
daterec.year;

          m_txn.NewOrder.o_entry_d.month  =
daterec.month;

          m_txn.NewOrder.o_entry_d.day    =
daterec.day;

          m_txn.NewOrder.o_entry_d.hour   =
daterec.hour;

          m_txn.NewOrder.o_entry_d.minute =
daterec.minute;

          m_txn.NewOrder.o_entry_d.second =
daterec.second;
                        }
                        if
(pData=dbdata(m_dbproc, 8))
                                  commit_flag =
(*(DBTINYINT *) pData);

                        DiscardNextRows(0);
                        DiscardNextResults(0);


                        if (commit_flag == 1)
                        {

          m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

          m_txn.NewOrder.exec_status_code = eOK;
                        }
                        else

          m_txn.NewOrder.exec_status_code =
eInvalidItem;

                        return;
                }
                catch (CSQLERR *e)
                {
                        if ((e->m_msgno == 1205
||
                                  (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::Payment()
{
          DBDATETIME          datetime;
          DBDATEREC daterec;

          int                 iTryCount = 0;
          const BYTE          *pData;

          ResetError();

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

                        dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.Payment.w_id);
                        dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.Payment.c_w_id);
                        dbrpcparam(m_dbproc,
NULL, 0, SQLFLT8, -1, -1, (BYTE *)
&m_txn.Payment.h_amount);
                        dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.Payment.d_id);
                        dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.Payment.c_d_id);
                        dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.Payment.c_id);

                        // if customer id is
zero, then payment is by name
                        if (m_txn.Payment.c_id ==
0)

          dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
strlen(m_txn.Payment.c_last), (unsigned char
*)m_txn.Payment.c_last);

                        if (dbrpcexec(m_dbproc)
== FAIL)

          ThrowError(CDBLIBERR::eDbRpcExec);

                        if (dbresults(m_dbproc)
!= 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;

                        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::OrderStatus()
{
          int                       i;
          DBDATETIME          datetime;
          DBDATEREC daterec;

          int                       iTryCount = 0;
          RETCODE             rc;
          const BYTE          *pData;

          ResetError();

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

                        dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.OrderStatus.w_id);
                        dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.OrderStatus.d_id);
                        dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.OrderStatus.c_id);

                        // if customer id is
zero, then order status is by name
                        if
(m_txn.OrderStatus.c_id == 0)

          dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
strlen(m_txn.OrderStatus.c_last), (unsigned char
*)m_txn.OrderStatus.c_last);


                        if (dbrpcexec(m_dbproc)
== FAIL)

          ThrowError(CDBLIBERR::eDbRpcExec);

                        // Get order lines
                        if (dbresults(m_dbproc)
!= SUCCEED)
                        {
                                  if ((m_DbLibErr
== NULL) && (m_SqlErr == NULL))
                                            throw
new CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER
);
                                  else

          ThrowError(CDBLIBERR::eDbResults);
                        }

                        if (dbnumcols(m_dbproc)
!= 5)

          ThrowError(CDBLIBERR::eWrongNumCols);

                        i = 0;
```

```
                              while (TRUE)
                              {
                                        rc =
dbnextrow(m_dbproc);
                                        if (rc ==
NO_MORE_ROWS)
          break;
                                        if (rc !=
REG_ROW)
          ThrowError(CDBLIBERR::eDbNextRow);


          if(pData=dbdata(m_dbproc, 1))
          m_txn.OrderStatus.OL[i].ol_supply_w_id =
(*(DBSMALLINT *) pData);
          if(pData=dbdata(m_dbproc, 2))
          m_txn.OrderStatus.OL[i].ol_i_id = (*(DBINT *)
pData);
          if(pData=dbdata(m_dbproc, 3))
          m_txn.OrderStatus.OL[i].ol_quantity =
(*(DBSMALLINT *) pData);
          if(pData=dbdata(m_dbproc, 4))
          dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,4),
                    SQLFLT8, (BYTE
*)&m_txn.OrderStatus.OL[i].ol_amount, 8);
          if(pData=dbdata(m_dbproc, 5))
                                        {
          datetime = *((DBDATETIME *) pData);
          dbdatecrack(m_dbproc, &daterec, &datetime);
          m_txn.OrderStatus.OL[i].ol_delivery_d.year
= daterec.year;
          m_txn.OrderStatus.OL[i].ol_delivery_d.month
= daterec.month;
          m_txn.OrderStatus.OL[i].ol_delivery_d.day
= daterec.day;
          m_txn.OrderStatus.OL[i].ol_delivery_d.hour
= daterec.hour;
          m_txn.OrderStatus.OL[i].ol_delivery_d.minute
= daterec.minute;
          m_txn.OrderStatus.OL[i].ol_delivery_d.second
= daterec.second;
                                        }
                                        i++;
                              }
          m_txn.OrderStatus.o_ol_cnt = i;


                              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))
          m_txn.OrderStatus.c_id = (*(DBINT *) pData);
                              if(pData=dbdata(m_dbproc,
2))
          UtilStrCpy(m_txn.OrderStatus.c_last, pData,
dbdatlen(m_dbproc,2));
                              if(pData=dbdata(m_dbproc,
3))
          UtilStrCpy(m_txn.OrderStatus.c_first, pData,
dbdatlen(m_dbproc,3));
                              if(pData=dbdata(m_dbproc,
4))
          UtilStrCpy(m_txn.OrderStatus.c_middle, pData,
dbdatlen(m_dbproc, 4));
                              if(pData=dbdata(m_dbproc,
5))
                              {
                                        datetime =
*((DBDATETIME *) pData);
          dbdatecrack(m_dbproc, &daterec, &datetime);
          m_txn.OrderStatus.o_entry_d.year   =
daterec.year;
          m_txn.OrderStatus.o_entry_d.month  =
daterec.month;
          m_txn.OrderStatus.o_entry_d.day    =
daterec.day;
          m_txn.OrderStatus.o_entry_d.hour   =
daterec.hour;
          m_txn.OrderStatus.o_entry_d.minute =
daterec.minute;
          m_txn.OrderStatus.o_entry_d.second =
daterec.second;
                              }
                              if(pData=dbdata(m_dbproc,
6))
          m_txn.OrderStatus.o_carrier_id =
(*(DBSMALLINT *) pData);
                              if(pData=dbdata(m_dbproc,
7))
          dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,7),
                    SQLFLT8, (BYTE
*)&m_txn.OrderStatus.c_balance, 8);
                              if(pData=dbdata(m_dbproc,
8))
          m_txn.OrderStatus.o_id = (*(DBINT *) pData);
                              DiscardNextRows(0);
                              DiscardNextResults(0);

                              if
(m_txn.OrderStatus.o_ol_cnt == 0)
                                        throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
                              else if
(m_txn.OrderStatus.c_id == 0 &&
m_txn.OrderStatus.c_last[0] == 0)
                                        throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
                              else

          m_txn.OrderStatus.exec_status_code = eOK;

                              return;
                    }
                    catch (CSQLERR *e)
                    {
                              if ((e->m_msgno == 1205
||
                                        (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::Delivery()
{
          int                       i;
          int                       iTryCount = 0;
          const BYTE          *pData;

          ResetError();

          while (TRUE)
          {
                    try
                    {
                              dbrpcinit(m_dbproc,
"tpcc_delivery", 0);
                              dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.Delivery.w_id);
                              dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.Delivery.o_carrier_id);
```

```
                        if (dbrpcexec(m_dbproc)
== FAIL)
                ThrowError(CDBLIBERR::eDbRpcExec);

                        if (dbresults(m_dbproc)
!= SUCCEED)
                ThrowError(CDBLIBERR::eDbResults);

                        if (dbnextrow(m_dbproc)
!= REG_ROW)
                ThrowError(CDBLIBERR::eDbNextRow);

                        if (dbnumcols(m_dbproc)
!= 10)
                ThrowError(CDBLIBERR::eWrongNumCols);

                        for (i=0; i<10; i++)
                        {
                                if (pData =
dbdata(m_dbproc, i+1))
                m_txn.Delivery.o_id[i] = *((DBINT *)pData);
                        }
                        DiscardNextRows(0);
                        DiscardNextResults(0);


                m_txn.Delivery.exec_status_code = eOK;
                        return;
                }
                catch (CSQLERR *e)
                {
                        if ((e->m_msgno == 1205
||
                                (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;
}
```

db_dblib_dll/src/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 dblogin
                        eDbOpen,
        // error from dbopen
                        eDbUse,
                // error from dbuse
                        eDbSqlExec,
                // error from dbsqlexec
                        eDbSet,
                // error from one of the dbset*
routines
                        eDbNextRow,
                // error from dbnextrow
                        eWrongRowCount,
        // more or less rows returned than expected
                        eWrongNumCols,
        // more or less columns returned than
expected
                        eDbResults,
                // error from dbresults
                        eDbRpcExec,
                // error from dbrpcexec
                        eDbSetMaxProcs,
        // error from dbsetmaxprocs
                        eDbProcHandler
        // error from either dbprocerrhandle or
dbprocmsghandle
                };

                CDBLIBERR(ACTION eAction, int
severity = 0, int dberror = 0, int oserr = 0)
                {
                        m_eAction = eAction;
                        m_severity = severity;
                        m_dberror = dberror;
                        m_oserr = oserr;

                        m_dberrstr = NULL;
                        m_oserrstr = NULL;
                };

                ~CDBLIBERR()
                {
                        delete [] m_dberrstr;
                        delete [] m_oserrstr;
                };

                ACTION    m_eAction;
                int                m_severity;
                int                m_dberror;
                int                m_oserr;
                char   *m_dberrstr;
                char   *m_oserrstr;

                int ErrorType() {return
ERR_TYPE_DBLIB;};
                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_hangler
                    // outside of the class
                    void SetDbLibError(int severity,
int dberr, int oserr, LPCSTR dberrstr, LPCSTR
oserrstr);
                    void SetSqlError( int msgno, int
msgstate, int severity, LPCSTR msgtext );

};

extern "C" DllDecl CTPCC_DBLIB* CTPCC_DBLIB_new
          ( LPCSTR szServer, LPCSTR szUser, LPCSTR
szPassword, LPCSTR szHost, LPCSTR szDatabase );

typedef CTPCC_DBLIB* (TYPE_CTPCC_DBLIB)(LPCSTR, LPCSTR,
LPCSTR, LPCSTR, LPCSTR);


tm_com_dll/src/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
definations 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 );
}
```

tm_com_dll/src/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.
#ifndef 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;

                    VARIANT m_vTxn;
          public:
                    CTPCC_COM(BOOL bSinglePool);
                    ~CTPCC_COM(void);

                    inline PNEW_ORDER_DATA
          BuffAddr_NewOrder()           { return
&m_pTxn->u.NewOrder;    };
                    inline PPAYMENT_DATA
          BuffAddr_Payment()            { return
&m_pTxn->u.Payment;     };
                    inline PDELIVERY_DATA
          BuffAddr_Delivery()           { return
&m_pTxn->u.Delivery;    };
                    inline PSTOCK_LEVEL_DATA
          BuffAddr_StockLevel()         { return
&m_pTxn->u.StockLevel;  };
                    inline PORDER_STATUS_DATA
          BuffAddr_OrderStatus()        { return
&m_pTxn->u.OrderStatus; };

                    void NewOrder                 ();
                    void Payment                  ();
                    void StockLevel               ();
                    void OrderStatus    ();
                    void Delivery                 () {
throw new CCOMERR(E_NOTIMPL); }  // not supported
};


inline void ReleaseInterface(IUnknown *pUnk)
{
          if (pUnk)
          {
                    pUnk->Release();
                    pUnk = NULL;
          }
}


// wrapper routine for class constructor
extern "C" __declspec(dllexport) CTPCC_COM*
CTPCC_COM_new(BOOL);

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);
```

tpcc_com_all/src/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<CComSingleThreadModel>)
        COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

};


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

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

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


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

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

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


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

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

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


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

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

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

tpcc_com_all/src/resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by tpcc_com_all.rc
//
#define IDS_PROJNAME                    100
#define IDR_TPCC                        101
#define IDR_NEWORDER                    102
#define IDR_ORDERSTATUS                 103
#define IDR_PAYMENT                     104
#define IDR_STOCKLEVEL                  105

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        202
#define _APS_NEXT_COMMAND_VALUE         32768
#define _APS_NEXT_CONTROL_VALUE         201
#define _APS_NEXT_SYMED_VALUE           106
#endif
#endif
```

tpcc_com_all/src/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 defination 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.  GetProcAddr
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)
                wsprintf( 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, Reg.szSPPrefix );
                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,

                    txn_in.parray->rgsabound-
>cElements,

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

                    memcpy( &pData->u.NewOrder,
pNewOrder, sizeof(NEW_ORDER_DATA));

                    pData->retval = ERR_SUCCESS;
                    pData->error = 0;
                    return S_OK;
          }
          catch (CBaseErr *e)
          {
                    // check for lost database
connection; if yes, component is toast
                    if ( ((e->ErrorType() ==
ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                              ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                         m_bCanBePooled = FALSE;

                    pData->retval = e->ErrorType();
                    pData->error = e->ErrorNum();
                    delete e;
                    return E_FAIL;
          }
          catch (...)
          {

          WriteMessageToEventLog(TEXT("Unhandled
exception."));
                    pData->retval = ERR_TYPE_LOGIC;
                    pData->error = 0;
                    m_bCanBePooled = FALSE;
                    return E_FAIL;
          }
}

HRESULT CTPCC_Common::Payment(VARIANT txn_in, VARIANT*
txn_out)
{
          PPAYMENT_DATA       pPayment;
          COM_DATA            *pData;
          try
          {
                    pData = (COM_DATA*)txn_in.parray-
>pvData;
                    pPayment = m_pTxn-
>BuffAddr_Payment();

                    memcpy(pPayment, &pData->u.Payment,
sizeof(PAYMENT_DATA));

                    m_pTxn->Payment();            // do
the actual txn

                    VariantInit(txn_out);
                    txn_out->vt = VT_SAFEARRAY;
                    txn_out->parray =
SafeArrayCreateVector( VT_UI1,

                    txn_in.parray->rgsabound-
>cElements,

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

                    memcpy( &pData->u.Payment,
pPayment, sizeof(PAYMENT_DATA));

                    pData->retval = ERR_SUCCESS;
                    pData->error = 0;
                    return S_OK;
          }
          catch (CBaseErr *e)
          {
                    // check for lost database
connection; if yes, component is toast
                    if ( ((e->ErrorType() ==
ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                              ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                         m_bCanBePooled = FALSE;

                    pData->retval = e->ErrorType();
                    pData->error = e->ErrorNum();
                    delete e;
                    return E_FAIL;
          }
          catch (...)
          {

          WriteMessageToEventLog(TEXT("Unhandled
exception."));
                    pData->retval = ERR_TYPE_LOGIC;
                    pData->error = 0;
                    m_bCanBePooled = FALSE;
                    return E_FAIL;
          }
}

HRESULT CTPCC_Common::StockLevel(VARIANT txn_in,
VARIANT* txn_out)
{
          PSTOCK_LEVEL_DATA   pStockLevel;
          COM_DATA                      *pData;

          try
          {
                    pData = (COM_DATA*)txn_in.parray-
>pvData;
                    pStockLevel = m_pTxn-
>BuffAddr_StockLevel();

                    memcpy(pStockLevel, &pData-
>u.StockLevel, sizeof(STOCK_LEVEL_DATA));

                    m_pTxn->StockLevel();

                    VariantInit(txn_out);
                    txn_out->vt = VT_SAFEARRAY;
                    txn_out->parray =
SafeArrayCreateVector( VT_UI1,

                    txn_in.parray->rgsabound-
>cElements,

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

                    memcpy( &pData->u.StockLevel,
pStockLevel, sizeof(STOCK_LEVEL_DATA));

                    pData->retval = ERR_SUCCESS;
                    pData->error = 0;
                    return S_OK;
          }
          catch (CBaseErr *e)
          {
                    // check for lost database
connection; if yes, component is toast
                    if ( ((e->ErrorType() ==
ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                              ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                         m_bCanBePooled = FALSE;

                    pData->retval = e->ErrorType();
                    pData->error = e->ErrorNum();
                    delete e;
                    return E_FAIL;
          }
          catch (...)
          {

          WriteMessageToEventLog(TEXT("Unhandled
exception."));
                    pData->retval = ERR_TYPE_LOGIC;
                    pData->error = 0;
                    m_bCanBePooled = FALSE;
                    return E_FAIL;
          }
}

HRESULT CTPCC_Common::OrderStatus(VARIANT txn_in,
VARIANT* txn_out)
{
          PORDER_STATUS_DATA  pOrderStatus;
          COM_DATA                      *pData;
          try
          {
                    pData = (COM_DATA*)txn_in.parray-
>pvData;
                    pOrderStatus = m_pTxn-
>BuffAddr_OrderStatus();

                    memcpy(pOrderStatus, &pData-
>u.OrderStatus, sizeof(ORDER_STATUS_DATA));

                    m_pTxn->OrderStatus();

                    VariantInit(txn_out);
                    txn_out->vt = VT_SAFEARRAY;
                    txn_out->parray =
SafeArrayCreateVector( VT_UI1,

                    txn_in.parray->rgsabound-
>cElements,

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

                    memcpy( &pData->u.OrderStatus,
pOrderStatus, sizeof(ORDER_STATUS_DATA));

                    pData->retval = ERR_SUCCESS;
                    pData->error = 0;
```

```
                return S_OK;
        }
        catch (CBaseErr *e)
        {
                // check for lost database
connection; if yes, component is toast
                if ( ((e->ErrorType() ==
ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                        ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                        m_bCanBePooled = FALSE;

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {

        WriteMessageToEventLog(TEXT("Unhandled
exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}
```

tpcc_com_all/src/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/src/tpcc_com_all.h

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

/* this ALWAYS GENERATED file contains the definitions
for the interfaces */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Aug 01 10:56:27 2003
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum
stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany),
__declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

/* verify that the <rpcndr.h> version is high enough to
compile this file*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

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

#ifndef __tpcc_com_all_h__
#define __tpcc_com_all_h__

#if defined(_MSC_VER) && (_MSC_VER >= 1020)
#pragma once
#endif

/* Forward Declarations */

#ifndef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__

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

#endif    /* __TPCC_FWD_DEFINED__ */


#ifndef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

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

#endif    /* __NewOrder_FWD_DEFINED__ */


#ifndef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

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

#endif    /* __OrderStatus_FWD_DEFINED__ */


#ifndef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

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

#endif    /* __Payment_FWD_DEFINED__ */


#ifndef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

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

#endif    /* __StockLevel_FWD_DEFINED__ */


/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"
#include "tpcc_com_ps.h"

#ifdef __cplusplus
extern "C"{
#endif

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

/* interface __MIDL_itf_tpcc_com_all_0000 */
/* [local] */




extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;


#ifndef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

/* library TPCCLib */
/* [helpstring][version][uuid] */


EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus

class DECLSPEC_UUID("122A3128-2520-11D3-BA71-
00C04FBFE08B")
TPCC;
#endif

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus

class DECLSPEC_UUID("975BAABF-84A7-11D2-BA47-
00C04FBFE08B")
NewOrder;
#endif

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus

class DECLSPEC_UUID("266836AD-A50D-11D2-BA4E-
00C04FBFE08B")
OrderStatus;
#endif

EXTERN_C const CLSID CLSID_Payment;
```

```
#ifdef __cplusplus

class DECLSPEC_UUID("CD02F7EF-A4FA-11D2-BA4E-
00C04FBFE08B")
Payment;
#endif

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus

class DECLSPEC_UUID("2668369E-A50D-11D2-BA4E-
00C04FBFE08B")
StockLevel;
#endif
#endif /* __TPCCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_all/src/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 "oaidl.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/src/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"

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

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

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

#ifdef APSTUDIO_INVOKED
////////////////////////////////////////////////////
////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE DISCARDABLE
BEGIN
    "resource.h\0"
END

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

3 TEXTINCLUDE DISCARDABLE
BEGIN
    "1 TYPELIB ""tpcc_com_all.tlb""\r\n"
    "\0"
END

#endif    // APSTUDIO_INVOKED


#ifndef _MAC
////////////////////////////////////////////////////
////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
 FILEVERSION 1,0,0,1
 PRODUCTVERSION 1,0,0,1
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x4L
 FILETYPE 0x2L
 FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
```

```
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "tpcc_com_all
Module\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "TPCCNEWORDER\0"
            VALUE "LegalCopyright", "Copyright 1997\0"
            VALUE "OriginalFilename",
"tpcc_com_all.DLL\0"
            VALUE "ProductName", "tpcc_com_all
Module\0"
            VALUE "ProductVersion", "1, 0, 0, 1\0"
            VALUE "OLESelfRegister", "\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

#endif    // !_MAC


/////////////////////////////////////////////////////
//////////////////////
//
// REGISTRY
//

IDR_TPCC                REGISTRY DISCARDABLE
"tpcc_com_all.rgs"
IDR_NEWORDER            REGISTRY DISCARDABLE
"tpcc_com_no.rgs"
IDR_ORDERSTATUS         REGISTRY DISCARDABLE
"tpcc_com_os.rgs"
IDR_PAYMENT             REGISTRY DISCARDABLE
"tpcc_com_pay.rgs"
IDR_STOCKLEVEL          REGISTRY DISCARDABLE
"tpcc_com_sl.rgs"

/////////////////////////////////////////////////////
//////////////////////
//
// String Table
//

STRINGTABLE DISCARDABLE
BEGIN
    IDS_PROJNAME            "tpcc_com_all"
END

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



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

/////////////////////////////////////////////////////
//////////////////////
#endif    // not APSTUDIO_INVOKED
```

tpcc_com_all/src/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/src/tpcc_com_all_i.c

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

/* this ALWAYS GENERATED file contains the IIDs and
CLSIDs */

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Aug 01 10:56:27 2003
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum
stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany),
__declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if !defined(_M_IA64) && !defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif
#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7
,b8) \

DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7
,b8) \
        const type name =
{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,0x71,0x00,0
xC0,0x4F,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,0x
00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0
xC0,0x4F,0xBF,0xE0,0x8B);
```

```
MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x0
0,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif


#endif /* !defined(_M_IA64) && !defined(_M_AMD64)*/
```

tpcc_com_all/src/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_all/src/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_all/src/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_all/src/tpcc_com_ps.h

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

/* this ALWAYS GENERATED file contains the definitions
for the interfaces */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Aug 01 10:56:14 2003
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum
stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany),
__declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


/* verify that the <rpcndr.h> version is high enough to
compile this file*/
#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__

#if defined(_MSC_VER) && (_MSC_VER >= 1020)
#pragma once
#endif

/* 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"

#ifdef __cplusplus
extern "C"{
#endif

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

/* interface __MIDL_itf_tpcc_com_ps_0000 */
/* [local] */



extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#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 *txn_out) = 0;

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

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

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

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

        virtual HRESULT __stdcall CallSetComplete(
void) = 0;

    };

#else     /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

        HRESULT ( STDMETHODCALLTYPE *QueryInterface )(
            ITPCC * This,
            /* [in] */ REFIID riid,
            /* [iid_is][out] */ void **ppvObject);

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

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

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

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

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

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

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

        HRESULT ( __stdcall *CallSetComplete )(
            ITPCC * This);

        END_INTERFACE
    } ITPCCVtbl;

    interface ITPCC
    {
        CONST_VTBL struct ITPCCVtbl *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 __stdcall ITPCC_NewOrder_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


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


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


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


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


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


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


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


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


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


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC * This);


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


#endif    /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces */

unsigned long             __RPC_USER  VARIANT_UserSize(
unsigned long *, unsigned long            , VARIANT *
);
```

```
unsigned char * __RPC_USER  VARIANT_UserMarshal(
unsigned long *, unsigned char *, VARIANT * );
unsigned char * __RPC_USER
VARIANT_UserUnmarshal(unsigned long *, unsigned char *,
VARIANT * );
void                      __RPC_USER  VARIANT_UserFree(
unsigned long *, VARIANT * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_all/src/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_com_ps/src/dlldata.c

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

        DO NOT ALTER THIS FILE

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

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

*********************************************************
**/


#include <rpcproxy.h>

#ifdef __cplusplus
extern "C"   {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )


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


DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

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

/* end of generated dlldata file */


tpcc_com_ps/src/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/src/tpcc_com_ps.h


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

/* this ALWAYS GENERATED file contains the definitions
for the interfaces */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Aug 01 10:56:14 2003
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum
stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany),
__declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


/* verify that the <rpcndr.h> version is high enough to
compile this file*/
#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__

#if defined(_MSC_VER) && (_MSC_VER >= 1020)
#pragma once
#endif

/* 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"

#ifdef __cplusplus
extern "C"{
#endif

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

/* interface __MIDL_itf_tpcc_com_ps_0000 */
/* [local] */



extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#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 *txn_out) = 0;

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

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

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

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

        virtual HRESULT __stdcall CallSetComplete(
void) = 0;

    };

#else     /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

        HRESULT ( STDMETHODCALLTYPE *QueryInterface )(
            ITPCC * This,
            /* [in] */ REFIID riid,
            /* [iid_is][out] */ void **ppvObject);

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

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

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

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

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

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

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

        HRESULT ( __stdcall *CallSetComplete )(
            ITPCC * This);

        END_INTERFACE
    } ITPCCVtbl;

    interface ITPCC
    {
        CONST_VTBL struct ITPCCVtbl *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 __stdcall ITPCC_NewOrder_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


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


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


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


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


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


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


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


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


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


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC * This);


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


#endif    /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces */

unsigned long             __RPC_USER  VARIANT_UserSize(
unsigned long *, unsigned long            , VARIANT *
);
unsigned char * __RPC_USER  VARIANT_UserMarshal(
unsigned long *, unsigned char *, VARIANT * );
unsigned char * __RPC_USER
VARIANT_UserUnmarshal(unsigned long *, unsigned char *,
VARIANT * );
```

```
void                    __RPC_USER  VARIANT_UserFree(
unsigned long *, VARIANT * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif



tpcc_com_ps/src/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 "oaidl.idl";
import "ocidl.idl";

          [
                    object,
                    oleautomation,
                    uuid(FEEE6AA2-84B1-11d2-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/src/tpcc_com_ps_i.c


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

/* this ALWAYS GENERATED file contains the IIDs and
CLSIDs */

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Aug 01 10:56:14 2003
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum
stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany),
__declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if !defined(_M_IA64) && !defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#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_AMD64)*/


tpcc_com_ps/src/tpcc_com_ps_p.c


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

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


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Aug 01 10:56:14 2003
 */
```

```
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum
stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany),
__declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if !defined(_M_IA64) && !defined(_M_AMD64)
#define USE_STUBLESS_PROXY


/* verify that the <rpcproxy.h> version is high enough
to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 440
#endif


#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of
<rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__


#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE   1023
#define PROC_FORMAT_STRING_SIZE   193
#define TRANSMIT_AS_TABLE_SIZE    0
#define WIRE_MARSHAL_TABLE_SIZE   1

typedef struct _MIDL_TYPE_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ TYPE_FORMAT_STRING_SIZE ];
    } MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ PROC_FORMAT_STRING_SIZE ];
    } MIDL_PROC_FORMAT_STRING;


static RPC_SYNTAX_IDENTIFIER  _RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,0x
10,0x48,0x60}},{2,0}};


extern const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString;


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo;


extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

#if !defined(__RPC_WIN32__)
#error  Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this
stub because it uses these features:
#error   -Oif or -Oicf, [wire_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 */
/*  8 */  NdrFcShort( 0x1c ), /* x86 Stack size/offset
= 28 */
/* 10 */  NdrFcShort( 0x0 ),  /* 0 */
/* 12 */  NdrFcShort( 0x8 ),  /* 8 */
/* 14 */  0x7,                /* Oi2 Flags:  srv must
size, clt must size, has return, */
                                0x3,                /* 3
*/

            /* Parameter txn_in */

/* 16 */  NdrFcShort( 0x8b ), /* Flags:  must size,
must free, in, by val, */
/* 18 */  NdrFcShort( 0x4 ),  /* x86 Stack size/offset
= 4 */
/* 20 */  NdrFcShort( 0x3e2 ),          /* Type
Offset=994 */

            /* Parameter txn_out */

/* 22 */  NdrFcShort( 0x4113 ),         /* Flags:  must
size, must free, out, simple ref, srv alloc size=16 */
/* 24 */  NdrFcShort( 0x14 ), /* x86 Stack size/offset
= 20 */
/* 26 */  NdrFcShort( 0x3f4 ),          /* Type
Offset=1012 */

            /* Return value */

/* 28 */  NdrFcShort( 0x70 ), /* Flags:  out, return,
base type, */
/* 30 */  NdrFcShort( 0x18 ), /* x86 Stack size/offset
= 24 */
/* 32 */  0x8,                /* FC_LONG */
                                0x0,                /* 0
*/

            /* Procedure Payment */

/* 34 */  0x33,               /* FC_AUTO_HANDLE */
                                0x6c,               /*
Old Flags:  object, Oi2 */
/* 36 */  NdrFcLong( 0x0 ),   /* 0 */
/* 40 */  NdrFcShort( 0x4 ),  /* 4 */
/* 42 */  NdrFcShort( 0x1c ), /* x86 Stack size/offset
= 28 */
/* 44 */  NdrFcShort( 0x0 ),  /* 0 */
/* 46 */  NdrFcShort( 0x8 ),  /* 8 */
/* 48 */  0x7,                /* Oi2 Flags:  srv must
size, clt must size, has return, */
                                0x3,                /* 3
*/

            /* Parameter txn_in */

/* 50 */  NdrFcShort( 0x8b ), /* Flags:  must size,
must free, in, by val, */
/* 52 */  NdrFcShort( 0x4 ),  /* x86 Stack size/offset
= 4 */
/* 54 */  NdrFcShort( 0x3e2 ),          /* Type
Offset=994 */

            /* Parameter txn_out */

/* 56 */  NdrFcShort( 0x4113 ),         /* Flags:  must
size, must free, out, simple ref, srv alloc size=16 */
/* 58 */  NdrFcShort( 0x14 ), /* x86 Stack size/offset
= 20 */
/* 60 */  NdrFcShort( 0x3f4 ),          /* Type
Offset=1012 */

            /* Return value */

/* 62 */  NdrFcShort( 0x70 ), /* Flags:  out, return,
base type, */
/* 64 */  NdrFcShort( 0x18 ), /* x86 Stack size/offset
= 24 */
/* 66 */  0x8,                /* FC_LONG */
                                0x0,                /* 0
*/

            /* Procedure Delivery */

/* 68 */  0x33,               /* FC_AUTO_HANDLE */
                                0x6c,               /*
Old Flags:  object, Oi2 */
/* 70 */  NdrFcLong( 0x0 ),   /* 0 */
/* 74 */  NdrFcShort( 0x5 ),  /* 5 */
/* 76 */  NdrFcShort( 0x1c ), /* x86 Stack size/offset
= 28 */
/* 78 */  NdrFcShort( 0x0 ),  /* 0 */
/* 80 */  NdrFcShort( 0x8 ),  /* 8 */
/* 82 */  0x7,                /* Oi2 Flags:  srv must
size, clt must size, has return, */
                                0x3,                /* 3
*/

            /* Parameter txn_in */

/* 84 */  NdrFcShort( 0x8b ), /* Flags:  must size,
must free, in, by val, */
/* 86 */  NdrFcShort( 0x4 ),  /* x86 Stack size/offset
= 4 */
/* 88 */  NdrFcShort( 0x3e2 ),          /* Type
Offset=994 */

            /* Parameter txn_out */

/* 90 */  NdrFcShort( 0x4113 ),         /* Flags:  must
size, must free, out, simple ref, srv alloc size=16 */
/* 92 */  NdrFcShort( 0x14 ), /* x86 Stack size/offset
= 20 */
/* 94 */  NdrFcShort( 0x3f4 ),          /* Type
Offset=1012 */
```

```
        /* Return value */

/* 96 */  NdrFcShort( 0x70 ), /* Flags:  out, return,
base type, */
/* 98 */  NdrFcShort( 0x18 ), /* x86 Stack size/offset
= 24 */
/* 100 */ 0x8,               /* FC_LONG */
                              0x0,                /* 0
*/

        /* Procedure StockLevel */

/* 102 */ 0x33,              /* FC_AUTO_HANDLE */
                              0x6c,                /*
Old Flags:  object, Oi2 */
/* 104 */ NdrFcLong( 0x0 ),   /* 0 */
/* 108 */ NdrFcShort( 0x6 ),  /* 6 */
/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset
= 28 */
/* 112 */ NdrFcShort( 0x0 ),  /* 0 */
/* 114 */ NdrFcShort( 0x8 ),  /* 8 */
/* 116 */ 0x7,               /* Oi2 Flags:  srv must
size, clt must size, has return, */
                              0x3,                /* 3
*/

        /* Parameter txn_in */

/* 118 */ NdrFcShort( 0x8b ), /* Flags:  must size,
must free, in, by val, */
/* 120 */ NdrFcShort( 0x4 ),  /* x86 Stack size/offset
= 4 */
/* 122 */ NdrFcShort( 0x3e2 ),          /* Type
Offset=994 */

        /* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ),        /* Flags:  must
size, must free, out, simple ref, srv alloc size=16 */
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset
= 20 */
/* 128 */ NdrFcShort( 0x3f4 ),          /* Type
Offset=1012 */

        /* Return value */

/* 130 */ NdrFcShort( 0x70 ), /* Flags:  out, return,
base type, */
/* 132 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset
= 24 */
/* 134 */ 0x8,               /* FC_LONG */
                              0x0,                /* 0
*/

        /* Procedure OrderStatus */

/* 136 */ 0x33,              /* FC_AUTO_HANDLE */
                              0x6c,                /*
Old Flags:  object, Oi2 */
/* 138 */ NdrFcLong( 0x0 ),   /* 0 */
/* 142 */ NdrFcShort( 0x7 ),  /* 7 */
/* 144 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset
= 28 */
/* 146 */ NdrFcShort( 0x0 ),  /* 0 */
/* 148 */ NdrFcShort( 0x8 ),  /* 8 */
/* 150 */ 0x7,               /* Oi2 Flags:  srv must
size, clt must size, has return, */
                              0x3,                /* 3
*/

        /* Parameter txn_in */

/* 152 */ NdrFcShort( 0x8b ), /* Flags:  must size,
must free, in, by val, */
/* 154 */ NdrFcShort( 0x4 ),  /* x86 Stack size/offset
= 4 */
/* 156 */ NdrFcShort( 0x3e2 ),          /* Type
Offset=994 */

        /* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ),        /* Flags:  must
size, must free, out, simple ref, srv alloc size=16 */
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset
= 20 */
/* 162 */ NdrFcShort( 0x3f4 ),          /* Type
Offset=1012 */

        /* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags:  out, return,
base type, */
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset
= 24 */
/* 168 */ 0x8,               /* FC_LONG */
                              0x0,                /* 0
*/

        /* Procedure CallSetComplete */

/* 170 */ 0x33,              /* FC_AUTO_HANDLE */
                              0x6c,                /*
Old Flags:  object, Oi2 */
/* 172 */ NdrFcLong( 0x0 ),   /* 0 */
/* 176 */ NdrFcShort( 0x8 ),  /* 8 */
/* 178 */ NdrFcShort( 0x8 ),  /* x86 Stack size/offset
= 8 */
/* 180 */ NdrFcShort( 0x0 ),  /* 0 */
/* 182 */ NdrFcShort( 0x8 ),  /* 8 */
/* 184 */ 0x4,               /* Oi2 Flags:  has
return, */
                              0x1,                /* 1
*/

        /* Return value */

/* 186 */ NdrFcShort( 0x70 ), /* Flags:  out, return,
base type, */
/* 188 */ NdrFcShort( 0x4 ),  /* x86 Stack size/offset
= 4 */
/* 190 */ 0x8,               /* FC_LONG */
                              0x0,                /* 0
*/

                              0x0
        }
    };

static const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString =
    {
        0,
        {
                              NdrFcShort( 0x0 ),  /* 0
*/
/*  2 */
                              0x12, 0x0,          /*
FC_UP */
/*  4 */  NdrFcShort( 0x3ca ),          /* Offset= 970
(974) */
/*  6 */
                              0x2b,               /*
FC_NON_ENCAPSULATED_UNION */
                              0x9,                /*
FC_ULONG */
/*  8 */  0x7,                /* Corr desc: FC_USHORT
*/
                              0x0,                /*
*/
/* 10 */  NdrFcShort( 0xfff8 ),         /* -8 */
/* 12 */  NdrFcShort( 0x2 ),  /* Offset= 2 (14) */
/* 14 */  NdrFcShort( 0x10 ), /* 16 */
/* 16 */  NdrFcShort( 0x2f ), /* 47 */
/* 18 */  NdrFcLong( 0x14 ),  /* 20 */
/* 22 */  NdrFcShort( 0x800b ),         /* Simple arm
type: FC_HYPER */
/* 24 */  NdrFcLong( 0x3 ),   /* 3 */
/* 28 */  NdrFcShort( 0x8008 ),         /* Simple arm
type: FC_LONG */
/* 30 */  NdrFcLong( 0x11 ),  /* 17 */
/* 34 */  NdrFcShort( 0x8001 ),         /* Simple arm
type: FC_BYTE */
/* 36 */  NdrFcLong( 0x2 ),   /* 2 */
/* 40 */  NdrFcShort( 0x8006 ),         /* Simple arm
type: FC_SHORT */
/* 42 */  NdrFcLong( 0x4 ),   /* 4 */
/* 46 */  NdrFcShort( 0x800a ),         /* Simple arm
type: FC_FLOAT */
/* 48 */  NdrFcLong( 0x5 ),   /* 5 */
/* 52 */  NdrFcShort( 0x800c ),         /* Simple arm
type: FC_DOUBLE */
/* 54 */  NdrFcLong( 0xb ),   /* 11 */
/* 58 */  NdrFcShort( 0x8006 ),         /* Simple arm
type: FC_SHORT */
/* 60 */  NdrFcLong( 0xa ),   /* 10 */
/* 64 */  NdrFcShort( 0x8008 ),         /* Simple arm
type: FC_LONG */
/* 66 */  NdrFcLong( 0x6 ),   /* 6 */
/* 70 */  NdrFcShort( 0xe8 ), /* Offset= 232 (302) */
/* 72 */  NdrFcLong( 0x7 ),   /* 7 */
/* 76 */  NdrFcShort( 0x800c ),         /* Simple arm
type: FC_DOUBLE */
/* 78 */  NdrFcLong( 0x8 ),   /* 8 */
/* 82 */  NdrFcShort( 0xe2 ), /* Offset= 226 (308) */
/* 84 */  NdrFcLong( 0xd ),   /* 13 */
/* 88 */  NdrFcShort( 0xf4 ), /* Offset= 244 (332) */
/* 90 */  NdrFcLong( 0x9 ),   /* 9 */
/* 94 */  NdrFcShort( 0x100 ),          /* Offset= 256
(350) */
/* 96 */  NdrFcLong( 0x2000 ),          /* 8192 */
/* 100 */ NdrFcShort( 0x10c ),         /* Offset= 268
(368) */
/* 102 */ NdrFcLong( 0x24 ),  /* 36 */
/* 106 */ NdrFcShort( 0x31a ),         /* Offset= 794
(900) */
/* 108 */ NdrFcLong( 0x4024 ),         /* 16420 */
/* 112 */ NdrFcShort( 0x314 ),         /* Offset= 788
(900) */
/* 114 */ NdrFcLong( 0x4011 ),         /* 16401 */
/* 118 */ NdrFcShort( 0x312 ),         /* Offset= 786
(904) */
/* 120 */ NdrFcLong( 0x4002 ),         /* 16386 */
/* 124 */ NdrFcShort( 0x310 ),         /* Offset= 784
(908) */
/* 126 */ NdrFcLong( 0x4003 ),         /* 16387 */
/* 130 */ NdrFcShort( 0x30e ),         /* Offset= 782
(912) */
/* 132 */ NdrFcLong( 0x4014 ),         /* 16404 */
/* 136 */ NdrFcShort( 0x30c ),         /* Offset= 780
(916) */
/* 138 */ NdrFcLong( 0x4004 ),         /* 16388 */
/* 142 */ NdrFcShort( 0x30a ),         /* Offset= 778
(920) */
/* 144 */ NdrFcLong( 0x4005 ),         /* 16389 */
```

```
/* 148 */ NdrFcShort( 0x308 ),          /* Offset= 776
(924) */
/* 150 */ NdrFcLong( 0x400b ),          /* 16395 */
/* 154 */ NdrFcShort( 0x2f2 ),          /* Offset= 754
(908) */
/* 156 */ NdrFcLong( 0x400a ),          /* 16394 */
/* 160 */ NdrFcShort( 0x2f0 ),          /* Offset= 752
(912) */
/* 162 */ NdrFcLong( 0x4006 ),          /* 16390 */
/* 166 */ NdrFcShort( 0x2fa ),          /* Offset= 762
(928) */
/* 168 */ NdrFcLong( 0x4007 ),          /* 16391 */
/* 172 */ NdrFcShort( 0x2f0 ),          /* Offset= 752
(924) */
/* 174 */ NdrFcLong( 0x4008 ),          /* 16392 */
/* 178 */ NdrFcShort( 0x2f2 ),          /* Offset= 754
(932) */
/* 180 */ NdrFcLong( 0x400d ),          /* 16397 */
/* 184 */ NdrFcShort( 0x2f0 ),          /* Offset= 752
(936) */
/* 186 */ NdrFcLong( 0x4009 ),          /* 16393 */
/* 190 */ NdrFcShort( 0x2ee ),          /* Offset= 750
(940) */
/* 192 */ NdrFcLong( 0x6000 ),          /* 24576 */
/* 196 */ NdrFcShort( 0x2ec ),          /* Offset= 748
(944) */
/* 198 */ NdrFcLong( 0x400c ),          /* 16396 */
/* 202 */ NdrFcShort( 0x2ea ),          /* Offset= 746
(948) */
/* 204 */ NdrFcLong( 0x10 ),  /* 16 */
/* 208 */ NdrFcShort( 0x8002 ),         /* Simple arm
type: FC_CHAR */
/* 210 */ NdrFcLong( 0x12 ),  /* 18 */
/* 214 */ NdrFcShort( 0x8006 ),         /* Simple arm
type: FC_SHORT */
/* 216 */ NdrFcLong( 0x13 ),  /* 19 */
/* 220 */ NdrFcShort( 0x8008 ),         /* Simple arm
type: FC_LONG */
/* 222 */ NdrFcLong( 0x15 ),  /* 21 */
/* 226 */ NdrFcShort( 0x800b ),         /* Simple arm
type: FC_HYPER */
/* 228 */ NdrFcLong( 0x16 ),  /* 22 */
/* 232 */ NdrFcShort( 0x8008 ),         /* Simple arm
type: FC_LONG */
/* 234 */ NdrFcLong( 0x17 ),  /* 23 */
/* 238 */ NdrFcShort( 0x8008 ),         /* Simple arm
type: FC_LONG */
/* 240 */ NdrFcLong( 0xe ),   /* 14 */
/* 244 */ NdrFcShort( 0x2c8 ),          /* Offset= 712
(956) */
/* 246 */ NdrFcLong( 0x400e ),          /* 16398 */
/* 250 */ NdrFcShort( 0x2cc ),          /* Offset= 716
(966) */
/* 252 */ NdrFcLong( 0x4010 ),          /* 16400 */
/* 256 */ NdrFcShort( 0x2ca ),          /* Offset= 714
(970) */
/* 258 */ NdrFcLong( 0x4012 ),          /* 16402 */
/* 262 */ NdrFcShort( 0x286 ),          /* Offset= 646
(908) */
/* 264 */ NdrFcLong( 0x4013 ),          /* 16403 */
/* 268 */ NdrFcShort( 0x284 ),          /* Offset= 644
(912) */
/* 270 */ NdrFcLong( 0x4015 ),          /* 16405 */
/* 274 */ NdrFcShort( 0x282 ),          /* Offset= 642
(916) */
/* 276 */ NdrFcLong( 0x4016 ),          /* 16406 */
/* 280 */ NdrFcShort( 0x278 ),          /* Offset= 632
(912) */
/* 282 */ NdrFcLong( 0x4017 ),          /* 16407 */
/* 286 */ NdrFcShort( 0x272 ),          /* Offset= 626
(912) */
/* 288 */ NdrFcLong( 0x0 ),   /* 0 */
/* 292 */ NdrFcShort( 0x0 ),  /* Offset= 0 (292) */
/* 294 */ NdrFcLong( 0x1 ),   /* 1 */
/* 298 */ NdrFcShort( 0x0 ),  /* Offset= 0 (298) */
/* 300 */ NdrFcShort( 0xffffffff ),     /* Offset= -1
(299) */
/* 302 */
                                0x15,               /*
FC_STRUCT */
                                0x7,                /* 7
*/
/* 304 */ NdrFcShort( 0x8 ),  /* 8 */
/* 306 */ 0xb,                 /* FC_HYPER */
                                0x5b,               /*
FC_END */
/* 308 */
                                0x12, 0x0,          /*
FC_UP */
/* 310 */ NdrFcShort( 0xc ),  /* Offset= 12 (322) */
/* 312 */
                                0x1b,               /*
FC_CARRAY */
                                0x1,                /* 1
*/
/* 314 */ NdrFcShort( 0x2 ),  /* 2 */
/* 316 */ 0x9,                 /* Corr desc: FC_ULONG */
                                0x0,                /*
*/
/* 318 */ NdrFcShort( 0xfffc ),         /* -4 */
/* 320 */ 0x6,                 /* FC_SHORT */
                                0x5b,               /*
FC_END */
/* 322 */
                                0x17,               /*
FC_CSTRUCT */
                                0x3,                /* 3
*/
/* 324 */ NdrFcShort( 0x8 ),  /* 8 */
/* 326 */ NdrFcShort( 0xfffffff2 ),     /* Offset= -14
(312) */
/* 328 */ 0x8,                 /* FC_LONG */
                                0x8,                /*
FC_LONG */
/* 330 */ 0x5c,                /* FC_PAD */
                                0x5b,               /*
FC_END */
/* 332 */
                                0x2f,               /*
FC_IP */
                                0x5a,               /*
FC_CONSTANT_IID */
/* 334 */ NdrFcLong( 0x0 ),   /* 0 */
/* 338 */ NdrFcShort( 0x0 ),  /* 0 */
/* 340 */ NdrFcShort( 0x0 ),  /* 0 */
/* 342 */ 0xc0,                /* 192 */
                                0x0,                /* 0
*/
/* 344 */ 0x0,                 /* 0 */
                                0x0,                /* 0
*/
/* 346 */ 0x0,                 /* 0 */
                                0x0,                /* 0
*/
/* 348 */ 0x0,                 /* 0 */
                                0x46,               /* 70
*/
/* 350 */
                                0x2f,               /*
FC_IP */
                                0x5a,               /*
FC_CONSTANT_IID */
/* 352 */ NdrFcLong( 0x20400 ),         /* 132096 */
/* 356 */ NdrFcShort( 0x0 ),  /* 0 */
/* 358 */ NdrFcShort( 0x0 ),  /* 0 */
/* 360 */ 0xc0,                /* 192 */
                                0x0,                /* 0
*/
/* 362 */ 0x0,                 /* 0 */
                                0x0,                /* 0
*/
/* 364 */ 0x0,                 /* 0 */
                                0x0,                /* 0
*/
/* 366 */ 0x0,                 /* 0 */
                                0x46,               /* 70
*/
/* 368 */
                                0x12, 0x10,         /*
FC_UP [pointer_deref] */
/* 370 */ NdrFcShort( 0x2 ),  /* Offset= 2 (372) */
/* 372 */
                                0x12, 0x0,          /*
FC_UP */
/* 374 */ NdrFcShort( 0x1fc ),          /* Offset= 508
(882) */
/* 376 */
                                0x2a,               /*
FC_ENCAPSULATED_UNION */
                                0x49,               /* 73
*/
/* 378 */ NdrFcShort( 0x18 ), /* 24 */
/* 380 */ NdrFcShort( 0xa ),  /* 10 */
/* 382 */ NdrFcLong( 0x8 ),   /* 8 */
/* 386 */ NdrFcShort( 0x58 ), /* Offset= 88 (474) */
/* 388 */ NdrFcLong( 0xd ),   /* 13 */
/* 392 */ NdrFcShort( 0x78 ), /* Offset= 120 (512) */
/* 394 */ NdrFcLong( 0x9 ),   /* 9 */
/* 398 */ NdrFcShort( 0x94 ), /* Offset= 148 (546) */
/* 400 */ NdrFcLong( 0xc ),   /* 12 */
/* 404 */ NdrFcShort( 0xbc ), /* Offset= 188 (592) */
/* 406 */ NdrFcLong( 0x24 ),  /* 36 */
/* 410 */ NdrFcShort( 0x114 ),          /* Offset= 276
(686) */
/* 412 */ NdrFcLong( 0x800d ),          /* 32781 */
/* 416 */ NdrFcShort( 0x130 ),          /* Offset= 304
(720) */
/* 418 */ NdrFcLong( 0x10 ),  /* 16 */
/* 422 */ NdrFcShort( 0x148 ),          /* Offset= 328
(750) */
/* 424 */ NdrFcLong( 0x2 ),   /* 2 */
/* 428 */ NdrFcShort( 0x160 ),          /* Offset= 352
(780) */
/* 430 */ NdrFcLong( 0x3 ),   /* 3 */
/* 434 */ NdrFcShort( 0x178 ),          /* Offset= 376
(810) */
/* 436 */ NdrFcLong( 0x14 ),  /* 20 */
/* 440 */ NdrFcShort( 0x190 ),          /* Offset= 400
(840) */
/* 442 */ NdrFcShort( 0xffffffff ),     /* Offset= -1
(441) */
/* 444 */
                                0x1b,               /*
FC_CARRAY */
                                0x3,                /* 3
*/
/* 446 */ NdrFcShort( 0x4 ),  /* 4 */
/* 448 */ 0x19,                /* Corr desc:  field
pointer, FC_ULONG */
                                0x0,                /*
*/
/* 450 */ NdrFcShort( 0x0 ),  /* 0 */
/* 452 */
                                0x4b,               /*
FC_PP */
```

```
                        0x5c,               /*
FC_PAD */
/* 454 */
                        0x48,               /*
FC_VARIABLE_REPEAT */
                        0x49,               /*
FC_FIXED_OFFSET */
/* 456 */ NdrFcShort( 0x4 ),  /* 4 */
/* 458 */ NdrFcShort( 0x0 ),  /* 0 */
/* 460 */ NdrFcShort( 0x1 ),  /* 1 */
/* 462 */ NdrFcShort( 0x0 ),  /* 0 */
/* 464 */ NdrFcShort( 0x0 ),  /* 0 */
/* 466 */ 0x12, 0x0,          /* FC_UP */
/* 468 */ NdrFcShort( 0xffffff6e ),     /* Offset= -146
(322) */
/* 470 */
                        0x5b,               /*
FC_END */

                        0x8,                /*
FC_LONG */
/* 472 */ 0x5c,               /* FC_PAD */
                        0x5b,               /*
FC_END */
/* 474 */
                        0x16,               /*
FC_PSTRUCT */
                        0x3,                /* 3
*/
/* 476 */ NdrFcShort( 0x8 ),  /* 8 */
/* 478 */
                        0x4b,               /*
FC_PP */
                        0x5c,               /*
FC_PAD */
/* 480 */
                        0x46,               /*
FC_NO_REPEAT */
                        0x5c,               /*
FC_PAD */
/* 482 */ NdrFcShort( 0x4 ),  /* 4 */
/* 484 */ NdrFcShort( 0x4 ),  /* 4 */
/* 486 */ 0x11, 0x0,          /* FC_RP */
/* 488 */ NdrFcShort( 0xfffffd4 ),     /* Offset= -44
(444) */
/* 490 */
                        0x5b,               /*
FC_END */

                        0x8,                /*
FC_LONG */
/* 492 */ 0x8,                /* FC_LONG */
                        0x5b,               /*
FC_END */
/* 494 */
                        0x21,               /*
FC_BOGUS_ARRAY */
                        0x3,                /* 3
*/
/* 496 */ NdrFcShort( 0x0 ),  /* 0 */
/* 498 */ 0x19,               /* Corr desc:  field
pointer, FC_ULONG */
                        0x0,                /*
*/
/* 500 */ NdrFcShort( 0x0 ),  /* 0 */
/* 502 */ NdrFcLong( 0xffffffff ),     /* -1 */
/* 506 */ 0x4c,               /* FC_EMBEDDED_COMPLEX */
                        0x0,                /* 0
*/
/* 508 */ NdrFcShort( 0xffffff50 ),     /* Offset= -176
(332) */
/* 510 */ 0x5c,               /* FC_PAD */
                        0x5b,               /*
FC_END */
/* 512 */
                        0x1a,               /*
FC_BOGUS_STRUCT */
                        0x3,                /* 3
*/
/* 514 */ NdrFcShort( 0x8 ),  /* 8 */
/* 516 */ NdrFcShort( 0x0 ),  /* 0 */
/* 518 */ NdrFcShort( 0x6 ),  /* Offset= 6 (524) */
/* 520 */ 0x8,                /* FC_LONG */
                        0x36,               /*
FC_POINTER */
/* 522 */ 0x5c,               /* FC_PAD */
                        0x5b,               /*
FC_END */
/* 524 */
                        0x11, 0x0,          /*
FC_RP */
/* 526 */ NdrFcShort( 0xfffffffe0 ),     /* Offset= -32
(494) */
/* 528 */
                        0x21,               /*
FC_BOGUS_ARRAY */
                        0x3,                /* 3
*/
/* 530 */ NdrFcShort( 0x0 ),  /* 0 */
/* 532 */ 0x19,               /* Corr desc:  field
pointer, FC_ULONG */
                        0x0,                /*
*/
/* 534 */ NdrFcShort( 0x0 ),  /* 0 */
/* 536 */ NdrFcLong( 0xffffffff ),     /* -1 */
/* 540 */ 0x4c,               /* FC_EMBEDDED_COMPLEX */
                        0x0,                /* 0
*/
/* 542 */ NdrFcShort( 0xffffff40 ),     /* Offset= -192
(350) */
/* 544 */ 0x5c,               /* FC_PAD */
                        0x5b,               /*
FC_END */
/* 546 */
                        0x1a,               /*
FC_BOGUS_STRUCT */
                        0x3,                /* 3
*/
/* 548 */ NdrFcShort( 0x8 ),  /* 8 */
/* 550 */ NdrFcShort( 0x0 ),  /* 0 */
/* 552 */ NdrFcShort( 0x6 ),  /* Offset= 6 (558) */
/* 554 */ 0x8,                /* FC_LONG */
                        0x36,               /*
FC_POINTER */
/* 556 */ 0x5c,               /* FC_PAD */
                        0x5b,               /*
FC_END */
/* 558 */
                        0x11, 0x0,          /*
FC_RP */
/* 560 */ NdrFcShort( 0xfffffffe0 ),     /* Offset= -32
(528) */
/* 562 */
                        0x1b,               /*
FC_CARRAY */
                        0x3,                /* 3
*/
/* 564 */ NdrFcShort( 0x4 ),  /* 4 */
/* 566 */ 0x19,               /* Corr desc:  field
pointer, FC_ULONG */
                        0x0,                /*
*/
/* 568 */ NdrFcShort( 0x0 ),  /* 0 */
/* 570 */
                        0x4b,               /*
FC_PP */
                        0x5c,               /*
FC_PAD */
/* 572 */
                        0x48,               /*
FC_VARIABLE_REPEAT */
                        0x49,               /*
FC_FIXED_OFFSET */
/* 574 */ NdrFcShort( 0x4 ),  /* 4 */
/* 576 */ NdrFcShort( 0x0 ),  /* 0 */
/* 578 */ NdrFcShort( 0x1 ),  /* 1 */
/* 580 */ NdrFcShort( 0x0 ),  /* 0 */
/* 582 */ NdrFcShort( 0x0 ),  /* 0 */
/* 584 */ 0x12, 0x0,          /* FC_UP */
/* 586 */ NdrFcShort( 0x184 ),          /* Offset= 388
(974) */
/* 588 */
                        0x5b,               /*
FC_END */

                        0x8,                /*
FC_LONG */
/* 590 */ 0x5c,               /* FC_PAD */
                        0x5b,               /*
FC_END */
/* 592 */
                        0x1a,               /*
FC_BOGUS_STRUCT */
                        0x3,                /* 3
*/
/* 594 */ NdrFcShort( 0x8 ),  /* 8 */
/* 596 */ NdrFcShort( 0x0 ),  /* 0 */
/* 598 */ NdrFcShort( 0x6 ),  /* Offset= 6 (604) */
/* 600 */ 0x8,                /* FC_LONG */
                        0x36,               /*
FC_POINTER */
/* 602 */ 0x5c,               /* FC_PAD */
                        0x5b,               /*
FC_END */
/* 604 */
                        0x11, 0x0,          /*
FC_RP */
/* 606 */ NdrFcShort( 0xfffffd4 ),     /* Offset= -44
(562) */
/* 608 */
                        0x2f,               /*
FC_IP */
                        0x5a,               /*
FC_CONSTANT_IID */
/* 610 */ NdrFcLong( 0x2f ),  /* 47 */
/* 614 */ NdrFcShort( 0x0 ),  /* 0 */
/* 616 */ NdrFcShort( 0x0 ),  /* 0 */
/* 618 */ 0xc0,               /* 192 */
                        0x0,                /* 0
*/
/* 620 */ 0x0,                /* 0 */
                        0x0,                /* 0
*/
/* 622 */ 0x0,                /* 0 */
                        0x0,                /* 0
*/
/* 624 */ 0x0,                /* 0 */
                        0x46,               /* 70
*/
/* 626 */
                        0x1b,               /*
FC_CARRAY */
                        0x0,                /* 0
*/
/* 628 */ NdrFcShort( 0x1 ),  /* 1 */
```

```
/* 630 */ 0x19,             /* Corr desc:  field
pointer, FC_ULONG */
                           0x0,                /*
*/
/* 632 */ NdrFcShort( 0x4 ),  /* 4 */
/* 634 */ 0x1,             /* FC_BYTE */
                           0x5b,               /*
FC_END */
/* 636 */
                           0x1a,               /*
FC_BOGUS_STRUCT */
                           0x3,                /* 3
*/
/* 638 */ NdrFcShort( 0x10 ), /* 16 */
/* 640 */ NdrFcShort( 0x0 ),  /* 0 */
/* 642 */ NdrFcShort( 0xa ),  /* Offset= 10 (652) */
/* 644 */ 0x8,             /* FC_LONG */
                           0x8,                /*
FC_LONG */
/* 646 */ 0x4c,            /* FC_EMBEDDED_COMPLEX */
                           0x0,                /* 0
*/
/* 648 */ NdrFcShort( 0xffffffd8 ),     /* Offset= -40
(608) */
/* 650 */ 0x36,            /* FC_POINTER */
                           0x5b,               /*
FC_END */
/* 652 */
                           0x12, 0x0,          /*
FC_UP */
/* 654 */ NdrFcShort( 0xfffffffe4 ),     /* Offset= -28
(626) */
/* 656 */
                           0x1b,               /*
FC_CARRAY */
                           0x3,                /* 3
*/
/* 658 */ NdrFcShort( 0x4 ),  /* 4 */
/* 660 */ 0x19,            /* Corr desc:  field
pointer, FC_ULONG */
                           0x0,                /*
*/
/* 662 */ NdrFcShort( 0x0 ),  /* 0 */
/* 664 */
                           0x4b,               /*
FC_PP */
                           0x5c,               /*
FC_PAD */
/* 666 */
                           0x48,               /*
FC_VARIABLE_REPEAT */
                           0x49,               /*
FC_FIXED_OFFSET */
/* 668 */ NdrFcShort( 0x4 ),  /* 4 */
/* 670 */ NdrFcShort( 0x0 ),  /* 0 */
/* 672 */ NdrFcShort( 0x1 ),  /* 1 */
/* 674 */ NdrFcShort( 0x0 ),  /* 0 */
/* 676 */ NdrFcShort( 0x0 ),  /* 0 */
/* 678 */ 0x12, 0x0,          /* FC_UP */
/* 680 */ NdrFcShort( 0xffffffd4 ),     /* Offset= -44
(636) */
/* 682 */
                           0x5b,               /*
FC_END */

                           0x8,                /*
FC_LONG */
/* 684 */ 0x5c,            /* FC_PAD */
                           0x5b,               /*
FC_END */
/* 686 */
                           0x1a,               /*
FC_BOGUS_STRUCT */
                           0x3,                /* 3
*/
/* 688 */ NdrFcShort( 0x8 ),  /* 8 */
/* 690 */ NdrFcShort( 0x0 ),  /* 0 */
/* 692 */ NdrFcShort( 0x6 ),  /* Offset= 6 (698) */
/* 694 */ 0x8,             /* FC_LONG */
                           0x36,               /*
FC_POINTER */
/* 696 */ 0x5c,            /* FC_PAD */
                           0x5b,               /*
FC_END */
/* 698 */
                           0x11, 0x0,          /*
FC_RP */
/* 700 */ NdrFcShort( 0xffffffd4 ),     /* Offset= -44
(656) */
/* 702 */
                           0x1d,               /*
FC_SMFARRAY */
                           0x0,                /* 0
*/
/* 704 */ NdrFcShort( 0x8 ),  /* 8 */
/* 706 */ 0x1,             /* FC_BYTE */
                           0x5b,               /*
FC_END */
/* 708 */
                           0x15,               /*
FC_STRUCT */
                           0x3,                /* 3
*/
/* 710 */ NdrFcShort( 0x10 ), /* 16 */
/* 712 */ 0x8,             /* FC_LONG */
                           0x6,                /*
FC_SHORT */
/* 714 */ 0x6,             /* FC_SHORT */
                           0x4c,               /*
FC_EMBEDDED_COMPLEX */
/* 716 */ 0x0,             /* 0 */
                           NdrFcShort( 0xffffffd1 ),
          /* Offset= -15 (702) */
                           0x5b,               /*
FC_END */
/* 720 */
                           0x1a,               /*
FC_BOGUS_STRUCT */
                           0x3,                /* 3
*/
/* 722 */ NdrFcShort( 0x18 ), /* 24 */
/* 724 */ NdrFcShort( 0x0 ),  /* 0 */
/* 726 */ NdrFcShort( 0xa ),  /* Offset= 10 (736) */
/* 728 */ 0x8,             /* FC_LONG */
                           0x36,               /*
FC_POINTER */
/* 730 */ 0x4c,            /* FC_EMBEDDED_COMPLEX */
                           0x0,                /* 0
*/
/* 732 */ NdrFcShort( 0xfffffffe8 ),     /* Offset= -24
(708) */
/* 734 */ 0x5c,            /* FC_PAD */
                           0x5b,               /*
FC_END */
/* 736 */
                           0x11, 0x0,          /*
FC_RP */
/* 738 */ NdrFcShort( 0xffffff0c ),     /* Offset= -244
(494) */
/* 740 */
                           0x1b,               /*
FC_CARRAY */
                           0x0,                /* 0
*/
/* 742 */ NdrFcShort( 0x1 ),  /* 1 */
/* 744 */ 0x19,            /* Corr desc:  field
pointer, FC_ULONG */
                           0x0,                /*
*/
/* 746 */ NdrFcShort( 0x0 ),  /* 0 */
/* 748 */ 0x1,             /* FC_BYTE */
                           0x5b,               /*
FC_END */
/* 750 */
                           0x16,               /*
FC_PSTRUCT */
                           0x3,                /* 3
*/
/* 752 */ NdrFcShort( 0x8 ),  /* 8 */
/* 754 */
                           0x4b,               /*
FC_PP */
                           0x5c,               /*
FC_PAD */
/* 756 */
                           0x46,               /*
FC_NO_REPEAT */
                           0x5c,               /*
FC_PAD */
/* 758 */ NdrFcShort( 0x4 ),  /* 4 */
/* 760 */ NdrFcShort( 0x4 ),  /* 4 */
/* 762 */ 0x12, 0x0,          /* FC_UP */
/* 764 */ NdrFcShort( 0xfffffff8 ),     /* Offset= -24
(740) */
/* 766 */
                           0x5b,               /*
FC_END */

                           0x8,                /*
FC_LONG */
/* 768 */ 0x8,             /* FC_LONG */
                           0x5b,               /*
FC_END */
/* 770 */
                           0x1b,               /*
FC_CARRAY */
                           0x1,                /* 1
*/
/* 772 */ NdrFcShort( 0x2 ),  /* 2 */
/* 774 */ 0x19,            /* Corr desc:  field
pointer, FC_ULONG */
                           0x0,                /*
*/
/* 776 */ NdrFcShort( 0x0 ),  /* 0 */
/* 778 */ 0x6,             /* FC_SHORT */
                           0x5b,               /*
FC_END */
/* 780 */
                           0x16,               /*
FC_PSTRUCT */
                           0x3,                /* 3
*/
/* 782 */ NdrFcShort( 0x8 ),  /* 8 */
/* 784 */
                           0x4b,               /*
FC_PP */
                           0x5c,               /*
FC_PAD */
/* 786 */
                           0x46,               /*
FC_NO_REPEAT */
                           0x5c,               /*
FC_PAD */
/* 788 */ NdrFcShort( 0x4 ),  /* 4 */
```

```
/* 790 */ NdrFcShort( 0x4 ),  /* 4 */
/* 792 */ 0x12, 0x0,          /* FC_UP */
/* 794 */ NdrFcShort( 0xffffffe8 ),     /* Offset= -24
(770) */
/* 796 */
                               0x5b,                /*
FC_END */

                               0x8,                 /*
FC_LONG */
/* 798 */ 0x8,                 /* FC_LONG */
                               0x5b,                /*
FC_END */
/* 800 */
                               0x1b,                /*
FC_CARRAY */
                               0x3,                 /* 3
*/
/* 802 */ NdrFcShort( 0x4 ),  /* 4 */
/* 804 */ 0x19,               /* Corr desc:  field
pointer, FC_ULONG */
                               0x0,                 /*
*/
/* 806 */ NdrFcShort( 0x0 ),  /* 0 */
/* 808 */ 0x8,                /* FC_LONG */
                               0x5b,                /*
FC_END */
/* 810 */
                               0x16,                /*
FC_PSTRUCT */
                               0x3,                 /* 3
*/
/* 812 */ NdrFcShort( 0x8 ),  /* 8 */
/* 814 */
                               0x4b,                /*
FC_PP */
                               0x5c,                /*
FC_PAD */
/* 816 */
                               0x46,                /*
FC_NO_REPEAT */
                               0x5c,                /*
FC_PAD */
/* 818 */ NdrFcShort( 0x4 ),  /* 4 */
/* 820 */ NdrFcShort( 0x4 ),  /* 4 */
/* 822 */ 0x12, 0x0,          /* FC_UP */
/* 824 */ NdrFcShort( 0xffffffe8 ),     /* Offset= -24
(800) */
/* 826 */
                               0x5b,                /*
FC_END */

                               0x8,                 /*
FC_LONG */
/* 828 */ 0x8,                 /* FC_LONG */
                               0x5b,                /*
FC_END */
/* 830 */
                               0x1b,                /*
FC_CARRAY */
                               0x7,                 /* 7
*/
/* 832 */ NdrFcShort( 0x8 ),  /* 8 */
/* 834 */ 0x19,               /* Corr desc:  field
pointer, FC_ULONG */
                               0x0,                 /*
*/
/* 836 */ NdrFcShort( 0x0 ),  /* 0 */
/* 838 */ 0xb,                /* FC_HYPER */
                               0x5b,                /*
FC_END */
/* 840 */
                               0x16,                /*
FC_PSTRUCT */
                               0x3,                 /* 3
*/
/* 842 */ NdrFcShort( 0x8 ),  /* 8 */
/* 844 */
                               0x4b,                /*
FC_PP */
                               0x5c,                /*
FC_PAD */
/* 846 */
                               0x46,                /*
FC_NO_REPEAT */
                               0x5c,                /*
FC_PAD */
/* 848 */ NdrFcShort( 0x4 ),  /* 4 */
/* 850 */ NdrFcShort( 0x4 ),  /* 4 */
/* 852 */ 0x12, 0x0,          /* FC_UP */
/* 854 */ NdrFcShort( 0xffffffe8 ),     /* Offset= -24
(830) */
/* 856 */
                               0x5b,                /*
FC_END */

                               0x8,                 /*
FC_LONG */
/* 858 */ 0x8,                 /* FC_LONG */
                               0x5b,                /*
FC_END */
/* 860 */
                               0x15,                /*
FC_STRUCT */
                               0x3,                 /* 3
*/
/* 862 */ NdrFcShort( 0x8 ),  /* 8 */
/* 864 */ 0x8,                /* FC_LONG */
                               0x8,                 /*
FC_LONG */
/* 866 */ 0x5c,               /* FC_PAD */
                               0x5b,                /*
FC_END */
/* 868 */
                               0x1b,                /*
FC_CARRAY */
                               0x3,                 /* 3
*/
/* 870 */ NdrFcShort( 0x8 ),  /* 8 */
/* 872 */ 0x7,                /* Corr desc: FC_USHORT
*/
                               0x0,                 /*
*/
/* 874 */ NdrFcShort( 0xffd8 ),         /* -40 */
/* 876 */ 0x4c,               /* FC_EMBEDDED_COMPLEX */
                               0x0,                 /* 0
*/
/* 878 */ NdrFcShort( 0xfffffffe ),     /* Offset= -18
(860) */
/* 880 */ 0x5c,               /* FC_PAD */
                               0x5b,                /*
FC_END */
/* 882 */
                               0x1a,                /*
FC_BOGUS_STRUCT */
                               0x3,                 /* 3
*/
/* 884 */ NdrFcShort( 0x28 ), /* 40 */
/* 886 */ NdrFcShort( 0xfffffffee ),    /* Offset= -18
(868) */
/* 888 */ NdrFcShort( 0x0 ),  /* Offset= 0 (888) */
/* 890 */ 0x6,                /* FC_SHORT */
                               0x6,                 /*
FC_SHORT */
/* 892 */ 0x8,                /* FC_LONG */
                               0x8,                 /*
FC_LONG */
/* 894 */ 0x4c,               /* FC_EMBEDDED_COMPLEX */
                               0x0,                 /* 0
*/
/* 896 */ NdrFcShort( 0xfffffdf8 ),     /* Offset= -520
(376) */
/* 898 */ 0x5c,               /* FC_PAD */
                               0x5b,                /*
FC_END */
/* 900 */
                               0x12, 0x0,           /*
FC_UP */
/* 902 */ NdrFcShort( 0xfffffef6 ),     /* Offset= -266
(636) */
/* 904 */
                               0x12, 0x8,           /*
FC_UP [simple_pointer] */
/* 906 */ 0x1,                /* FC_BYTE */
                               0x5c,                /*
FC_PAD */
/* 908 */
                               0x12, 0x8,           /*
FC_UP [simple_pointer] */
/* 910 */ 0x6,                /* FC_SHORT */
                               0x5c,                /*
FC_PAD */
/* 912 */
                               0x12, 0x8,           /*
FC_UP [simple_pointer] */
/* 914 */ 0x8,                /* FC_LONG */
                               0x5c,                /*
FC_PAD */
/* 916 */
                               0x12, 0x8,           /*
FC_UP [simple_pointer] */
/* 918 */ 0xb,                /* FC_HYPER */
                               0x5c,                /*
FC_PAD */
/* 920 */
                               0x12, 0x8,           /*
FC_UP [simple_pointer] */
/* 922 */ 0xa,                /* FC_FLOAT */
                               0x5c,                /*
FC_PAD */
/* 924 */
                               0x12, 0x8,           /*
FC_UP [simple_pointer] */
/* 926 */ 0xc,                /* FC_DOUBLE */
                               0x5c,                /*
FC_PAD */
/* 928 */
                               0x12, 0x0,           /*
FC_UP */
/* 930 */ NdrFcShort( 0xfffffd8c ),     /* Offset= -628
(302) */
/* 932 */
                               0x12, 0x10,          /*
FC_UP [pointer_deref] */
/* 934 */ NdrFcShort( 0xfffffd8e ),     /* Offset= -626
(308) */
/* 936 */
                               0x12, 0x10,          /*
FC_UP [pointer_deref] */
/* 938 */ NdrFcShort( 0xfffffda2 ),     /* Offset= -606
(332) */
/* 940 */
                               0x12, 0x10,          /*
FC_UP [pointer_deref] */
/* 942 */ NdrFcShort( 0xfffffdb0 ),     /* Offset= -592
(350) */
```

```
/* 944 */
                        0x12, 0x10,         /*
FC_UP [pointer_deref] */
/* 946 */ NdrFcShort( 0xfffffdbe ),     /* Offset= -578
(368) */
/* 948 */
                        0x12, 0x10,         /*
FC_UP [pointer_deref] */
/* 950 */ NdrFcShort( 0x2 ),  /* Offset= 2 (952) */
/* 952 */
                        0x12, 0x0,          /*
FC_UP */
/* 954 */ NdrFcShort( 0x14 ), /* Offset= 20 (974) */
/* 956 */
                        0x15,               /*
FC_STRUCT */
                        0x7,                /* 7
*/
/* 958 */ NdrFcShort( 0x10 ), /* 16 */
/* 960 */ 0x6,                /* FC_SHORT */
                        0x1,                /*
FC_BYTE */
/* 962 */ 0x1,                /* FC_BYTE */
                        0x8,                /*
FC_LONG */
/* 964 */ 0xb,                /* FC_HYPER */
                        0x5b,               /*
FC_END */
/* 966 */
                        0x12, 0x0,          /*
FC_UP */
/* 968 */ NdrFcShort( 0xfffffff4 ),     /* Offset= -12
(956) */
/* 970 */
                        0x12, 0x8,          /*
FC_UP [simple_pointer] */
/* 972 */ 0x2,                /* FC_CHAR */
                        0x5c,               /*
FC_PAD */
/* 974 */
                        0x1a,               /*
FC_BOGUS_STRUCT */
                        0x7,                /* 7
*/
/* 976 */ NdrFcShort( 0x20 ), /* 32 */
/* 978 */ NdrFcShort( 0x0 ),  /* 0 */
/* 980 */ NdrFcShort( 0x0 ),  /* Offset= 0 (980) */
/* 982 */ 0x8,                /* FC_LONG */
                        0x8,                /*
FC_LONG */
/* 984 */ 0x6,                /* FC_SHORT */
                        0x6,                /*
FC_SHORT */
/* 986 */ 0x6,                /* FC_SHORT */
                        0x6,                /*
FC_SHORT */
/* 988 */ 0x4c,               /* FC_EMBEDDED_COMPLEX */
                        0x0,                /* 0
*/
/* 990 */ NdrFcShort( 0xfffffc28 ),     /* Offset= -984
(6) */
/* 992 */ 0x5c,               /* FC_PAD */
                        0x5b,               /*
FC_END */
/* 994 */ 0xb4,               /* FC_USER_MARSHAL */
                        0x83,               /*
131 */
/* 996 */ NdrFcShort( 0x0 ),  /* 0 */
/* 998 */ NdrFcShort( 0x10 ), /* 16 */
/* 1000 */          NdrFcShort( 0x0 ),  /* 0 */
/* 1002 */          NdrFcShort( 0xfffffc18 ),     /*
Offset= -1000 (2) */
/* 1004 */
                        0x11, 0x4,          /*
FC_RP [alloced_on_stack] */
/* 1006 */          NdrFcShort( 0x6 ),  /* Offset= 6
(1012) */
/* 1008 */
                        0x13, 0x0,          /*
FC_OP */
/* 1010 */          NdrFcShort( 0xffffffdc ),     /*
Offset= -36 (974) */
/* 1012 */          0xb4,               /*
FC_USER_MARSHAL */
                        0x83,               /*
131 */
/* 1014 */          NdrFcShort( 0x0 ),  /* 0 */
/* 1016 */          NdrFcShort( 0x10 ), /* 16 */
/* 1018 */          NdrFcShort( 0x0 ),  /* 0 */
/* 1020 */          NdrFcShort( 0xfffffff4 ),     /*
Offset= -12 (1008) */

                        0x0
        }
    };

static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
        {

            {
            VARIANT_UserSize
            ,VARIANT_UserMarshal
            ,VARIANT_UserUnmarshal
            ,VARIANT_UserFree
            }

        };



/* Standard interface: __MIDL_itf_tpcc_com_ps_0000,
ver. 0.0,

GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00}} */


/* Object interface: IUnknown, ver. 0.0,

GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x0
0,0x00,0x00,0x46}} */


/* Object interface: ITPCC, ver. 0.0,

GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4
F,0xBF,0xE0,0x8B}} */

#pragma code_seg(".orpc")
static const unsigned short
ITPCC_FormatStringOffsetTable[] =
    {
    0,
    34,
    68,
    102,
    136,
    170
    };

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


static const MIDL_SERVER_INFO ITPCC_ServerInfo =
    {
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0};
CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *) (INT_PTR) -1 /* ITPCC::NewOrder */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Payment */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Delivery */ ,
    (void *) (INT_PTR) -1 /* ITPCC::StockLevel */ ,
    (void *) (INT_PTR) -1 /* ITPCC::OrderStatus */ ,
    (void *) (INT_PTR) -1 /* ITPCC::CallSetComplete */
};

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

static const MIDL_STUB_DESC Object_StubDesc =
    {
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x600015b, /* MIDL Version 6.0.347 */
    0,
    UserMarshalRoutines,
    0,  /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0,   /* proxy/server info */
    0   /* Reserved5 */
    };

const CInterfaceProxyVtbl *
_tpcc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};
```

```
const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[]
=
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[]
=
{
    "ITPCC",
    0
};


#define _tpcc_com_ps_CHECK_IID(n)
          IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID, n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID *
pIID, int * pIndex )
{

    if(!_tpcc_com_ps_CHECK_IID(0))
        {
        *pIndex = 0;
        return 1;
        }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) &
_tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) &
_tpcc_com_ps_StubVtblList,
    (const PCInterfaceName * ) &
_tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    & _tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0  /* Filler3 */
};

#endif /* !defined(_M_IA64) && !defined(_M_AMD64)*/
```

common/txnlog/include/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 );
}
```

common/txnlog/include/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.
 */

#ifndef _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
```

common/txnlog/include/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
 *
 */
#include <stdio.h>          //needed for FILE

#define DRIVER_NAME_LEN
                    32      //max length of the
driver engine name - must be the same as in engstut.h!
#define TXN_LOG_INCORRECTLY_SHUT_DOWN   100
          //ctrl rec subtype generated by the txn log
when reading an abruptly shut down log

#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;
        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

          #define TXN_REC_TYPE_TPCW
          4         // replaces TRANSACTION_TYPE_TPCW

          typedef struct _TXN_RECORD_HEADER
          {
                    JULIAN_TIME         TxnStartT0;
                    // start of txn
                    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 txn
                    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;
        //

                bool IsSuccessRecord() { return
(TxnStatus == ERR_SUCCESS || TxnStatus ==
ERR_BAD_ITEM_ID || TxnStatus ==
ERR_TYPE_DELIVERY_POST); }
        } 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

                bool IsSuccessRecord() { return
(TxnStatus == ERR_SUCCESS || TxnStatus ==
ERR_BAD_ITEM_ID || TxnStatus ==
ERR_TYPE_DELIVERY_POST); }
        } TXN_RECORD_TPCC_DELIV_DEF,
*PTXN_RECORD_TPCC_DELIV_DEF;

        //
        //TPC-W records.
        //
        typedef struct _TXN_RECORD_TPCW
        {
                // common header; must exactly
match TXN_RECORD_HEADER
                JULIAN_TIME         TxnStartT0;
                // start of txn
                BYTE        TxnType;
        // = TXN_REC_TYPE_TPCW
                BYTE        TxnSubType;
                // depends on TxnType
                // end of common header

                int     ThinkTime;          //
think time (ms)
                int     WIRT;
        // response time (ms)
                int                 TxnError;
        // error code providing more detail for
TxnStatus
                BYTE        TxnStatus;
        // completion status for txn to indicate
errors
                //This field below depends on the
txn sub type:
                //- for Home interaction:     it
indicates whether the user was a new customer (or
returning)
                //- for Buy Confirm:
        it indicates whether the shipping address was
updated
                //- for Search Request:
        it indicates the search type (Author, Title,
or Subject)
                //This statistics needs to be
reported according to 5.5.5.1 clause in the specs.
                //Because this field occupies 1
byte, the record structure is already aligned on word
boundary.
                union       {
                        BYTE        newCustomer;
                        BYTE        addrUpdated;
                        BYTE        searchType;
                }           intrDetails;

                //This field is mostly for
informational/debugging purposes.
                //It indicates what user performed
this web interaction and what instance (session) of
that use it was.
                //The first 22 bits indicate the
user #, and the top 10 bits indicate instance (session)
#.
                unsigned __int32    uiUser;

                bool IsSuccessRecord() { return
(TxnStatus == ERR_SUCCESS); }
        } TXN_RECORD_TPCW, *PTXN_RECORD_TPCW;

        //
        //      Data part of a control record
written when a user is created (or it's new session) -
to record USMD
        typedef struct _TXN_RECORD_TPCW_USER_DATA
        {
                unsigned __int32    uiUser;
                                    // user number
                JULIAN_TIME
        USMD;                                   //
USMD for this user
                BYTE
        bRetCust;                       // returning
customer?
        } TXN_RECORD_TPCW_USER_DATA,
*PTXN_RECORD_TPCW_USER_DATA;

        //The entire TPCW User control record
structure
        typedef struct _TXN_RECORD_TPCW_USER
        {
                // common header; must exactly
match TXN_RECORD_HEADER
                JULIAN_TIME         TxnStartT0;
                // start of txn
                BYTE        TxnType;
        // = TXN_REC_TYPE_CONTROL
                BYTE        TxnSubType;
                // depends on TxnType
                // end of common header

                DWORD       Len;
                // number of bytes after this field
                //The fields above must exactly
match TXN_RECORD_CONTROL

                //The fields below must exactly
match TXN_RECORD_TPCW_USER_DATA
                unsigned __int32    uiUser;
                                    // user number
                JULIAN_TIME
        USMD;                                   //
USMD for this user
                BYTE
        bRetCust;                       // returning
customer?
        } TXN_RECORD_TPCW_USER,
*PTXN_RECORD_TPCW_USER;


#define             USER_INDEX_NBITS    22
#define             USER_INDEX_MASK
        0x003fffff                      //lower 22 bits
mask for user field in TPCW record
#define             USER_SESSION_MASK   0xffc00000
                    //upper 10 bits mask for user field
in TPCW record
#define             USER_CREATE_REC             254
                    //subtype for the control record
written when a user is created

#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 bytes;
should always be "BC"
                int
        LogVersion;                 // set to
TXN_LOG_VERSION
                JULIAN_TIME
        BeginTxnTS;                 // timestamp of
first (lowest) txn start
                JULIAN_TIME
        EndTxnTS;           // timestamp of last
(highest) txn completion time
```

```
                int
        iRecCount;                      // number of
records in log file
                BOOL
        bLogSorted;
                int
        iFileSize;                      // file size in
bytes

                // driver engine that created this
log file
                char
        szDriverEngineName[DRIVER_NAME_LEN];
                // the record map provides a fast
way to get close to a particular timestamp in a sorted
log file.
//              struct
//              {
//                      JULIAN_TIME
        TS;                             // timestamp of
record
//                      int
                iPos;                   // byte
position in file
//              }
        RecMap[RecMapSize];
//#define           RecMapSize
        200

        } 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   WRITE_BUFFER_SIZE              128*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.
                JULIAN_TIME
        SavePtTime;
//              int
        iSavePtFilePointer;
                LARGE_INTEGER
        lSavePtFilePointer;
                int
        iSavePtNextRec;

                JULIAN_TIME         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 input and output
                JULIAN_TIME
        BeginTxnTS;                                 //
timestamp of first (lowest) txn start
                JULIAN_TIME
        EndTxnTS;                   // timestamp of
last (highest) txn completion time
                int
        iRecCount;                                  //
number of records in log file
                // To write a checkpoint
information into the header, need to know the EndTxnTS
for the
                // last record written to the disk.
It is not necessarily the last record in the
                // last written buffer, as the last
record may be only partially in the buffer.
                // So remember the timestamps for 2
last records that begin in the buffer - one of
                // them will the last complete
record written to disk.
                JULIAN_TIME
        PrevEndTxnTS;               // timestamp of
the previous to last record
                union {
                        TXN_LOG_HEADER
        HeaderForCheckpoint;// header written on
every checkpoint
                        char
                szHeaderBuffer[512];          //
512 bytes is the minimum we can write to the disk

                } HeaderBuffer;     //need the
union because can't write sizeof(TXN_LOG_HEADER) - too
few bytes

                // Control record returned from
GetNextRecord if the file
                // currently opened for read was
not properly shut down
                struct
                {
                        TXN_RECORD_CONTROL
        RecHeader;
                        char
                szDriverName[DRIVER_NAME_LEN];

                } IncorrectShutDownRec;

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

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

                DWORD               dwError;
                DWORD
        dwCheckpointError;          //error in
checkpoint thread
                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
                HANDLE
        hStopCheckpointThread;      //event to
signal the checkpoint thread to exit

                Spinlock  Spin;
                                    //spin lock to protect
the txn log file buffers
                Spinlock  WriteSpin;
                                    //spin lock to protect
the WriteFile operation between IO and Checkpoint
threads

                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()

                    BOOL      bIncorrectShutDown;
                    // indicates whether the log opened
for read was not correctly shut down

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

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

                    static void
CheckpointThread(CTxnLog *);  // checkpointing thread

          public:

                    CTxnLog(LPCTSTR szFileName, DWORD
dwOpts, char *szDriver = NULL);
                    ~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 WriteToLog(PTXN_RECORD_TPCW
pTxnRcrd);          //support for TPC-W

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

                    void CloseTransactionLogFile(void);

                    PTXN_RECORD_HEADER
GetNextRecord(BOOL bSkipCtrlRecs = FALSE);
                    PTXN_RECORD_HEADER
GetNextRecord(JULIAN_TIME SeekTimeTO, 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 *ErrorTypeStr() { return "TXN
LOG"; }

                    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

Build Scripts

setup.cmd

```
::-------------------------------------------------------
-----------------------------------
::---  FILE:       RUNSQLCFG.CMD
::---              Microsoft TPC-C Kit Ver. 4.41
::---              Copyright Microsoft, 2001
::---              All Rights Reserved
::---
::---  PURPOSE:    Calls RunSQLCfg.sql to configure SQL
Server
::---
::---  ARGUMENTS:  Optionally, the user can pass the
following positional arguments:
::---                  Server Name
::---                  sa SQL Server account password
::---                  Number of Warehouses
::---                  Build Option
::---
{full,builddb,objects,objectsfull,bulkload,bulkloadfull
,backup}
::---                  Database Type
::---                     {normal or scale_down}
::---
::---              If they are not passed, then the
user will be prompted by the VBS file.
::---
::-------------------------------------------------------
-----------------------------------
@cscript SetupScripts\setup.vbs //H:CScript //I %1 %2
%3 %4 %5
```

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  = "O:",
          SIZE                = 26000MB,
          FILEGROWTH          = 0),
(         NAME                = MSSQL_misc2,
          FILENAME  = "P:",
          SIZE                = 26000MB,
          FILEGROWTH          = 0),
FILEGROUP MSSQL_cs_fg
(         NAME                = MSSQL_cs1,
          FILENAME  = "R:",
          SIZE                = 50000MB,
          FILEGROWTH          = 0),
(         NAME                = MSSQL_cs2,
          FILENAME  = "S:",
          SIZE                = 50000MB,
          FILEGROWTH          = 0)
LOG ON
(         NAME                = MSSQL_tpcc_log,
          FILENAME  = "L:",
          SIZE                = 65000MB,
          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
```

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
```

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
```

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
```

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
```

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
```

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 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
```

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 with
replace, stats = 1

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

go
```

Stored Procedures

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,

          @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),
        @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 availible order id and
update
--  plus initialize local variables

          update    district
          set       @d_tax       = d_tax,
                    @o_id        = d_next_o_id,
                    d_next_o_id  = d_next_o_id + 1,
                    @o_entry_d   = getdate(),
                    @li_no       = 0,
                    @commit_flag = 1
          where     d_w_id       = @w_id and
                    d_id         = @d_id

--  process orderlines

          while (@li_no < @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

--  all that work for nuthin!!!

                rollback transaction n

--  return order data to client

        select  @w_tax,
                @d_tax,
                @o_id,
                @c_last,
                @c_discount,
                @c_credit,
                @o_entry_d,
                @commit_flag

end

go
```

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
```

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
                    order     by c_w_id, c_d_id,
c_last, c_first

                    set       rowcount 0
          end

          else

          begin

--  get customer info if by id

                    select    @c_balance          =
c_balance,
                              @c_first  = c_first,
                              @c_middle = c_middle,
                              @c_last             =
c_last
                    from      customer (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
```

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
updlock)
                    where     no_w_id   = @w_id and
                              no_d_id   = @d_id
                    order     by no_o_id asc

                  if (@@rowcount <> 0)
                    begin

--  claim the order for this district

                              delete    new_order
                              where     no_w_id   =
@w_id and
                                        no_d_id   =
@d_id and
                                        no_o_id   =
@o_id

--  set carrier_id on this order (and get customer id)

                              update    orders
                              set       o_carrier_id
          = @o_carrier_id,
                                        @c_id
          = o_c_id
                              where     o_w_id
          = @w_id and
                                        o_d_id
          = @d_id and
                                        o_id
          = @o_id

--  set date in all lineitems for this order (and sum
amounts)

                              update    order_line
                              set       ol_delivery_d
          = getdate(),
                                        @total
          = @total + ol_amount
                              where     ol_w_id
          = @w_id and
                                        ol_d_id
          = @d_id and
                                        ol_o_id
          = @o_id

--  accummulate 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
```

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,
                                    @threshhold
          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          <  @threshhold

go
```

Loader Source Code

tpcc.h

```
//        File:               TPCC.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 DEFLDPACKSIZE                           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;                             //
set if loading all tables
          BOOL
          table_item;                             //
set if loading ITEM table specifically
          BOOL
          table_warehouse;    // set if loading
WAREHOUSE, DISTRICT, and STOCK
          BOOL
          table_customer;               // set if
loading CUSTOMER and HISTORY
          BOOL
          table_orders;                 // set if
loading NEW-ORDER, ORDERS, ORDER-LINE
    long
          num_warehouses;
    long
          batch;
    long
          verbose;
          long
          pack_size;
          char
          *loader_res_file;
          char
          *log_path;
          char
          *synch_servername;
          long
          case_sensitivity;
          long
          starting_warehouse;
          long
          build_index;
          long
          index_order;
          long
          scale_down;
          char
          *index_script_path;
} TPCCLDR_ARGS;

// String length constants
          #define SERVER_NAME_LEN            20
          #define DATABASE_NAME_LEN          20
          #define USER_NAME_LEN              20
          #define PASSWORD_LEN               20
          #define TABLE_NAME_LEN             20
          #define I_DATA_LEN                 50
          #define I_NAME_LEN                 24
          #define BRAND_LEN                   1
          #define LAST_NAME_LEN              16
          #define W_NAME_LEN                 10
          #define ADDRESS_LEN                20
          #define STATE_LEN                   2
          #define ZIP_LEN                     9
          #define S_DIST_LEN                 24
          #define S_DATA_LEN                 50
          #define D_NAME_LEN                 10
          #define FIRST_NAME_LEN             16
          #define MIDDLE_NAME_LEN             2
          #define PHONE_LEN                  16
          #define CREDIT_LEN                  2
          #define C_DATA_LEN                500
          #define H_DATA_LEN                 24
          #define DIST_INFO_LEN              24
          #define MAX_OL_NEW_ORDER_ITEMS     15
          #define MAX_OL_ORDER_STATUS_ITEMS  15
          #define STATUS_LEN                 25
          #define OL_DIST_INFO_LEN                    24

          #define C_SINCE_LEN
             23
          #define H_DATE_LEN
             23
          #define OL_DELIVERY_D_LEN                   23
          #define O_ENTRY_D_LEN              23
```

```
// Functions in random.c
void      seed();
long      irand();
double    drand();
void      WUCreate();
short     WURand();
long      RandomNumber(long lower, long upper);

// Functions in getargs.c;
void      GetArgsLoader();
void      GetArgsLoaderUsage();

// Functions in time.c
long      TimeNow();

// Functions in strings.c
void      MakeAddress();
void      LastName();
int       MakeAlphaString();
int       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*  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]);

                    if (hThread[0] == NULL)
                    {
                              printf("Error, failed in
creating creating thread = 0.\n");
                              exit(-1);
                    }
          }

          if (aptr->tables_all || aptr-
>table_warehouse)
          {         fprintf(fLoader, "Starting loader
threads for: warehouse\n");

                    hThread[1] = CreateThread(NULL,

                                        0,


(LPTHREAD_START_ROUTINE) LoadWarehouse,

                                        NULL,

                                        0,

                                        &dwThreadID[1]);

                    if (hThread[1] == NULL)
                    {
                              printf("Error, failed in
creating creating thread = 1.\n");
                              exit(-1);
                    }
          }

          if (aptr->tables_all || aptr->table_customer)
          {         fprintf(fLoader, "Starting loader
threads for: customer\n");

                    hThread[2] = CreateThread(NULL,

                                        0,


(LPTHREAD_START_ROUTINE) LoadCustomer,

                                        NULL,

                                        0,

                                        &dwThreadID[2]);

                    if (hThread[2] == NULL)
                    {
                              printf("Error, failed in
creating creating main thread = 2.\n");
                              exit(-1);
                    }
          }

          if (aptr->tables_all || aptr->table_orders)
          {         fprintf(fLoader, "Starting loader
threads for: orders\n");
```

```
                    hThread[3] = CreateThread(NULL,
                                        0,
(LPTHREAD_START_ROUTINE) LoadOrders,
                                        NULL,
                                        0,
                                        &dwThreadID[3]);
                    if (hThread[3] == NULL)
                    {
                              printf("Error, failed in
creating creating main thread = 3.\n");
                              exit(-1);
                    }
          }

          // Wait for threads to finish...
          for (i=0; i<MAX_MAIN_THREADS; i++)
          {
                    if (hThread[i] != NULL)
                    {
                              WaitForSingleObject(
hThread[i], INFINITE );
                              CloseHandle(hThread[i]);
                              hThread[i] = NULL;
                    }
          }

          main_time_end = (TimeNow() / MILLI);

    sprintf(buffer,"\nTPC-C load completed successfully
in %ld minutes.\n",
                              (main_time_end -
main_time_start)/60);

          printf("%s",buffer);
          fprintf(fLoader, "%s", buffer);

          fclose(fLoader);

          SQLFreeEnv(henv);

    exit(0);

          return 0;
}


//================================================
================
//
// Function name: LoadItem
//
//================================================
================

void LoadItem()
{
    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);

          rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 10);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 11);
          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;
    short                       d_id;
          DWORD
          dwThreadID[MAX_ORDER_THREADS];
          HANDLE
          hThread[MAX_ORDER_THREADS];
          char                          name[20];
          RETCODE
          rc;
          char
          bcphint[128];
          char
          err_log_path_ord[256];
          char
          err_log_path_nord[256];
          char
          err_log_path_ordl[256];

          // seed with unique number
          seed(6);

          printf("Loading orders...\n");

          // if build index before load...
          if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
          {
                              BuildIndex("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_deli
very_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 )
          {

                    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 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]
= {"000000000"};
          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_UINTEGER );
          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;

                    // interate 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')
                                        {

          printf("The History table is missing or
damaged.\n");

          ExitFlag = 1;
                                        }
                                        break;
                              case 4:
                                        if
(TablesBitMap[i] == '0')
                                        {

          printf("The New_Order table is missing or
damaged.\n");

          ExitFlag = 1;
                                        }
                                        break;
                              case 5:
                                        if
(TablesBitMap[i] == '0')
                                        {

          printf("The Orders table is missing or
damaged.\n");

          ExitFlag = 1;
                                        }
                                        break;
                              case 6:
                                        if
(TablesBitMap[i] == '0')
                                        {

          printf("The Order_Line table is missing or
damaged.\n");

          ExitFlag = 1;
                                        }
                                        break;
                              case 7:
                                        if
(TablesBitMap[i] == '0')
                                        {

          printf("The Item table is missing or
damaged.\n");

          ExitFlag = 1;
                                        }
                                        break;
                              case 8:
                                        if
(TablesBitMap[i] == '0')
                                        {

          printf("The Stock table is missing or
damaged.\n");

          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;
}

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, TPCCLDR_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                        =
DEFLDPACKSIZE;
        pargs->starting_warehouse     =
DEF_STARTING_WAREHOUSE;
        pargs->build_index                      =
BUILD_INDEX;
        pargs->index_order                      =
INDEX_ORDER;
        pargs->index_script_path      =
INDEX_SCRIPT_PATH;
        pargs->scale_down                       =
SCALE_DOWN;

    /* check for zero command line args */
    if ( argc == 1 )
                GetArgsLoaderUsage();

    for (i = 1; i < argc; ++i)
    {
                if (argv[i][0] != '-' && argv[i][0]
!= '/')
                {
                printf("\nUnrecognized command");
                GetArgsLoaderUsage();
                exit(1);
                }

                ptr = argv[i];

                switch (ptr[1])
                {
                case '?':     /* Fall throught */

        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) DEFLDPACKSIZE);
        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);
}



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;
}


#if 0

//Orginal 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;
}



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[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqr
stuvwxyz";
          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 prercentage is valid
    if ((percent < 0) || (percent > 100))
    {
                    printf("MakeOrigianlAlphaString:
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;
}
```

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;
}
```

Appendix C: Tunable Parameters

Microsoft SQL Server 2000 Startup Parameters

```
Microsoft SQL Server was started with the following
command-line options:

"c:\program files\microsoft sql
server\mssql\binn\sqlservr.exe" -e"c:\program
files\microsoft sql server\mssql\log\errorlog" -c -x -
t3502 -g100


Boot.ini

[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows
Server 2003, Standard" /fastdetect
```

Microsoft Windows Server 2003 SE Configuration (Enabled Services)

```
These Windows services are started:

COM+ Event System
Event Log
Logical Disk Manager
Network Connections
Network Location Awareness
Plug and Play
Remote Procedure Call
Security Accounts Manager
System Event Notification
Terminal Services
Windows Management Instrumentation
```

Microsoft SQL Server 2000 Configuration Parameters

```
name     minimum     maximum     config_value run_value
-------------------------------------------------
affinity mask -2147483648 2147483647  3            3
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  704          704
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
274          274
media retention  0           365         0            0
min memory per query (KB)           512
2147483647  512          512
min server memory (MB)              0
2147483647  0            0
nested triggers  0           1           1            1
network packet size (B)             512         65536
2048         2048
open objects     0           2147483647  0            0
priority boost   0           1           1            1
query governor cost limit           0
2147483647  0            0
query wait (s)  -1          2147483647  -1           -1
recovery interval (min)             0           32767
32767        32767
remote access    0           1           1            1
remote login timeout (s)            0
2147483647  20           20
remote proc trans                   0           1
0            0
remote query timeout (s)            0
2147483647  600          600
scan for startup procs              0           1
0            0
set working set size                0           1
0            0
show advanced options               0           1
1            1
two digit year cutoff               1753        9999
2049         2049
user connections 0           32767       0            0
user options     0           32767       0            0
```

Server Registry Parameters

```
Windows Registry Editor Version 5.2

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\T
CPIP/Parameters]
"MaxUserPort":dword:60000 (decimal)

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\d
ac2w2k\Parameters\Device]
"DriverParameter":string:"ConfigureSIR=16"


[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Se
ssionManager\I/O System]
"CountOperations":dword:0000 (decimal)
```

Disk Controller Driver Registry Parameters

```
Windows Registry Editor Version 5.2

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\da
c2w2k
Class Name:        <NO CLASS>
Last Write Time:   5/5/2004 - 12:55 PM
Value 0
  Name:            Group
  Type:            REG_SZ
  Data:            SCSI miniport

Value 1
  Name:            Start
  Type:            REG_DWORD
  Data:            0

Value 2
  Name:            Tag
  Type:            REG_DWORD
  Data:            0x20

Value 3
  Name:            Type
  Type:            REG_DWORD
  Data:            0x1

Value 4
  Name:            ErrorControl
  Type:            REG_DWORD
  Data:            0x1

Value 5
  Name:            ImagePath
  Type:            REG_EXPAND_SZ
  Data:            system32\DRIVERS\dac2w2k.sys


Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\da
c2w2k\Parameters
Class Name:        <NO CLASS>
Last Write Time:   4/22/2004 - 10:54 AM
Value 0
  Name:            BusType
  Type:            REG_DWORD
  Data:            0x8


Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\da
c2w2k\Parameters\Device
Class Name:        <NO CLASS>
Last Write Time:   4/22/2004 - 3:31 PM
Value 0
  Name:            DriverParameter
```

```
  Type:             REG_SZ
  Data:             ConfigureSIR=24


Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\da
c2w2k\Parameters\PnpInterface
Class Name:       <NO CLASS>
Last Write Time:  4/20/2004 - 10:23 AM
Value 0
  Name:           2
  Type:           REG_DWORD
  Data:           0x1

Value 1
  Name:           5
  Type:           REG_DWORD
  Data:           0x1


Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\da
c2w2k\Enum
Class Name:       <NO CLASS>
Last Write Time:  5/5/2004 - 12:55 PM
Value 0
  Name:           0
  Type:           REG_SZ
  Data:
PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\6&175a51b6
&0&4010E818

Value 1
  Name:           Count
  Type:           REG_DWORD
  Data:           0x3

Value 2
  Name:           NextInstance
  Type:           REG_DWORD
  Data:           0x3

Value 3
  Name:           1
  Type:           REG_SZ
  Data:
PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\6&3b7ff379
&0&4008F818

Value 4
  Name:           2
  Type:           REG_SZ
  Data:
PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\6&17f0941a
&0&4010F818

Windows Registry Editor Version 5.2

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ma
c2w2k
Class Name:       <NO CLASS>
Last Write Time:  5/5/2004 - 12:55 PM
Value 0
  Name:           Type
  Type:           REG_DWORD
  Data:           0x1

Value 1
  Name:           Start
  Type:           REG_DWORD
  Data:           0

Value 2
  Name:           ErrorControl
  Type:           REG_DWORD
  Data:           0x1

Value 3
  Name:           Tag
  Type:           REG_DWORD
  Data:           0xf

Value 4
  Name:           ImagePath
  Type:           REG_EXPAND_SZ
  Data:           system32\DRIVERS\mac2w2k.sys

Value 5
  Name:           Group
  Type:           REG_SZ
  Data:           System Bus Extender


Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ma
c2w2k\Security
Class Name:       <NO CLASS>
Last Write Time:  4/22/2004 - 10:48 AM
Value 0
  Name:           Security
  Type:           REG_BINARY
  Data:
00000000   01 00 14 80 90 00 00 00 - 9c 00 00 00 14 00
00 00  ..............
00000010   30 00 00 00 02 00 1c 00 - 01 00 00 00 02 80
14 00  0.............
00000020   ff 01 0f 00 01 01 00 00 - 00 00 00 01 00 00
00 00  ÿ.............
00000030   02 00 60 00 04 00 00 00 - 00 00 14 00 fd 01
02 00  ..`.........ý...
00000040   01 01 00 00 00 00 00 05 - 12 00 00 00 00 00
18 00  ..............
00000050   ff 01 0f 00 01 02 00 00 - 00 00 00 05 20 00
00 00  ÿ........... ...
00000060   20 02 00 00 00 00 14 00 - 8d 01 02 00 01 01
00 00   ...............
00000070   00 00 00 05 0b 00 00 00 - 00 00 18 00 fd 01
02 00  ............ý...
00000080   01 02 00 00 00 00 00 05 - 20 00 00 00 23 02
00 00  ........ ...#...
00000090   01 01 00 00 00 00 00 05 - 12 00 00 00 01 01
00 00  ..............
00 00 00 05 12 00 00 00 -
........


Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ma
c2w2k\Enum
Class Name:       <NO CLASS>
Last Write Time:  5/5/2004 - 12:55 PM
Value 0
  Name:           0
  Type:           REG_SZ
  Data:           Root\LEGACY_MAC2W2K\0000

Value 1
  Name:           Count
  Type:           REG_DWORD
  Data:           0x5

Value 2
  Name:           NextInstance
  Type:           REG_DWORD
  Data:           0x5

Value 3
  Name:           1
  Type:           REG_SZ
  Data:
SCSI\Disk&Ven_MYLEX&Prod_eXtremeRAID_2000&Rev_0702\7&14
e2b2c9&0&400

Value 4
  Name:           2
  Type:           REG_SZ
  Data:
SCSI\Disk&Ven_MYLEX&Prod_eXtremeRAID_2000&Rev_0702\7&30
08e2bd&0&400

Value 5
  Name:           3
  Type:           REG_SZ
  Data:
SCSI\Disk&Ven_MYLEX&Prod_eXtremeRAID_2000&Rev_0702\7&2c
a21e49&0&400

Value 6
  Name:           4
  Type:           REG_SZ
  Data:
SCSI\Disk&Ven_MYLEX&Prod_eXtremeRAID_2000&Rev_0702\7&2c
a21e49&0&410
```

System Sumary

```
System Information report written at: 04/27/04 12:53:06
System Name: SQL02
[System Summary]

Item      Value
OS Name   Microsoft(R) Windows(R) Server 2003, Standard
Edition
Version   5.2.3790  Build 3790
OS Manufacturer     Microsoft Corporation
Activation Status   Activation Pending (54 days
remaining)
System Name         SQL02
System Manufacturer Intel
System Model        S7501HG0
System Type         X86-based PC
Processor x86 Family 15 Model 2 Stepping 7 GenuineIntel
~3056 Mhz
Processor x86 Family 15 Model 2 Stepping 7 GenuineIntel
~3056 Mhz
BIOS Version/Date   Intel Corporation
S7501HG0.86B.0047.P09.0307011125, 1/7/2003
SMBIOS Version      2.3
Windows Directory   C:\WINDOWS
System Directory    C:\WINDOWS\system32
Boot Device         \Device\HarddiskVolume4
Locale    United States
Hardware Abstraction Layer    Version = "5.2.3790.0
(srv03_rtm.030324-2048)"
User Name SQL02\Administrator
Time Zone E. South America Standard Time
Total Physical Memory         4.096,00 MB
Available Physical Memory     157,33 MB
Total Virtual Memory          8,35 GB
Available Virtual Memory      2,30 GB
```

```
Page File Space     5,10 GB
Page File C:\pagefile.sys

[Hardware Resources]



[Conflicts/Sharing]

Resource  Device
I/O Port 0x00000000-0x00000CF7          PCI bus
I/O Port 0x00000000-0x00000CF7          Direct memory
access controller

Memory Address 0xFFC00000-0xFFFFFFFF    PCI bus
Memory Address 0xFFC00000-0xFFFFFFFF    Intel(r) 82802
Firmware Hub Device

I/O Port 0x00002000-0x00006FFF          Intel(R) E7000
Series Hub Interface C PCI-to-PCI Bridge - 2545
I/O Port 0x00002000-0x00006FFF          Intel(r) 82870
Hub Interface to PCI Bridges
I/O Port 0x00002000-0x00006FFF          DEC 21154 PCI
to PCI bridge
I/O Port 0x00002000-0x00006FFF          Mylex
eXtremeRAID 2000 Disk Array Controller (Accelerated)


Memory Address 0xE8000000-0xEFFFFFFF    Intel(r) 82870
Hub Interface to PCI Bridges
Memory Address 0xE8000000-0xEFFFFFFF    DEC 21154 PCI
to PCI bridge

Memory Address 0xD8000000-0xEFFFFFFF    Intel(R) E7000
Series Hub Interface C PCI-to-PCI Bridge - 2545
Memory Address 0xD8000000-0xEFFFFFFF    Intel(r) 82870
Hub Interface to PCI Bridges
Memory Address 0xD8000000-0xEFFFFFFF    DEC 21154 PCI
to PCI bridge

I/O Port 0x00003000-0x00003FFF          DEC 21154 PCI
to PCI bridge
I/O Port 0x00003000-0x00003FFF          Mylex
eXtremeRAID 2000 Disk Array Controller (Accelerated)


I/O Port 0x00005000-0x00006FFF          Intel(r) 82870
Hub Interface to PCI Bridges
I/O Port 0x00005000-0x00006FFF          DEC 21154 PCI
to PCI bridge
I/O Port 0x00005000-0x00006FFF          Mylex
eXtremeRAID 2000 Disk Array Controller (Accelerated)


Memory Address 0xA0000-0xBFFFF          PCI bus
Memory Address 0xA0000-0xBFFFF          RAGE XL PCI
Family (Microsoft Corporation)

Memory Address 0xFA800000-0xFEAFFFFF    Intel(r) 82870
Hub Interface to PCI Bridges
Memory Address 0xFA800000-0xFEAFFFFF    DEC 21154 PCI
to PCI bridge

Memory Address 0xF2100000-0xFEBFFFFF    Intel(R) E7000
Series Hub Interface C PCI-to-PCI Bridge - 2545
Memory Address 0xF2100000-0xFEBFFFFF    Intel(r) 82870
Hub Interface to PCI Bridges
Memory Address 0xF2100000-0xFEBFFFFF    DEC 21154 PCI
to PCI bridge


[DMA]

Resource  Device    Status
Channel 4 Direct memory access controller         OK

Channel 2 Standard floppy disk controller         OK

Channel 3 ECP Printer Port (LPT1)       OK

[Forced Hardware]

Device    PNP Device ID

[I/O]

Resource  Device    Status
0x00000000-0x00000CF7          PCI bus   OK
0x00000000-0x00000CF7          Direct memory access
controller          OK
0x00000D00-0x0000FFFF          PCI bus   OK
0x00002000-0x00006FFF          Intel(R) E7000 Series Hub
Interface C PCI-to-PCI Bridge - 2545    OK
0x00002000-0x00006FFF          Intel(r) 82870 Hub
Interface to PCI Bridges      OK
0x00002000-0x00006FFF          DEC 21154 PCI to PCI
bridge    OK
0x00002000-0x00006FFF          Mylex eXtremeRAID 2000
Disk Array Controller (Accelerated)     OK
0x00005000-0x00006FFF          Intel(r) 82870 Hub
Interface to PCI Bridges      OK
0x00005000-0x00006FFF          DEC 21154 PCI to PCI
bridge    OK
0x00005000-0x00006FFF          Mylex eXtremeRAID 2000
Disk Array Controller (Accelerated)     OK
0x00006040-0x0000607F          Intel(R) PRO/1000 MT Dual
Port Network Connection       OK
0x00006000-0x0000603F          Intel(R) PRO/1000 MT Dual
Port Network Connection #2    OK
0x00003000-0x00003FFF          DEC 21154 PCI to PCI
bridge    OK
0x00003000-0x00003FFF          Mylex eXtremeRAID 2000
Disk Array Controller (Accelerated)     OK
0x00004800-0x000048FF          Adaptec AIC-7902-based
Ultra320 SCSI       OK
0x00004C00-0x00004CFF          Adaptec AIC-7902-based
Ultra320 SCSI       OK
0x00004000-0x000040FF          Adaptec AIC-7902-based
Ultra320 SCSI       OK
0x00004400-0x000044FF          Adaptec AIC-7902-based
Ultra320 SCSI       OK
0x00001000-0x000010FF          RAGE XL PCI Family
(Microsoft Corporation)       OK
0x000003B0-0x000003BB          RAGE XL PCI Family
(Microsoft Corporation)       OK
0x000003C0-0x000003DF          RAGE XL PCI Family
(Microsoft Corporation)       OK
0x00000A79-0x00000A79          ISAPNP Read Data Port
          OK
0x00000279-0x00000279          ISAPNP Read Data Port
          OK
0x00000274-0x00000277          ISAPNP Read Data Port
          OK
0x00000092-0x00000092          Motherboard resources
          OK
0x000000B2-0x000000B3          Motherboard resources
          OK
0x000003F0-0x000003F1          Motherboard resources
          OK
0x00000400-0x000004BF          Motherboard resources
          OK
0x000004D0-0x000004D1          Motherboard resources
          OK
0x00000010-0x0000001F          Motherboard resources
          OK
0x0000040B-0x0000040B          Motherboard resources
          OK
0x000004D6-0x000004D6          Motherboard resources
          OK
0x00000C14-0x00000C14          Motherboard resources
          OK
0x00000C49-0x00000C49          Motherboard resources
          OK
0x00000C52-0x00000C52          Motherboard resources
          OK
0x00000C6C-0x00000C6C          Motherboard resources
          OK
0x00000C6F-0x00000C6F          Motherboard resources
          OK
0x00000F50-0x00000F57          Motherboard resources
          OK
0x00000C00-0x00000C01          Motherboard resources
          OK
0x00000C98-0x00000C98          Motherboard resources
          OK
0x00000CD6-0x00000CD7          Motherboard resources
          OK
0x0000002E-0x0000002F          Motherboard resources
          OK
0x00000530-0x00000531          Motherboard resources
          OK
0x00000500-0x0000052F          Motherboard resources
          OK
0x00000532-0x0000053F          Motherboard resources
          OK
0x00000540-0x0000055F          Motherboard resources
          OK
0x00000560-0x0000057F          Motherboard resources
          OK
0x000005A0-0x000005BF          Motherboard resources
          OK
0x00000CA2-0x00000CA5          Motherboard resources
          OK
0x00000020-0x00000021          Programmable interrupt
controller          OK
0x000000A0-0x000000A1          Programmable interrupt
controller          OK
0x00000024-0x00000025          Programmable interrupt
controller          OK
0x00000028-0x00000029          Programmable interrupt
controller          OK
0x0000002C-0x0000002D          Programmable interrupt
controller          OK
0x00000030-0x00000031          Programmable interrupt
controller          OK
0x00000034-0x00000035          Programmable interrupt
controller          OK
0x00000038-0x00000039          Programmable interrupt
controller          OK
0x0000003C-0x0000003D          Programmable interrupt
controller          OK
0x00000050-0x00000052          Programmable interrupt
controller          OK
0x000000A4-0x000000A5          Programmable interrupt
controller          OK
0x000000A8-0x000000A9          Programmable interrupt
controller          OK
0x000000AC-0x000000AD          Programmable interrupt
controller          OK
0x000000B0-0x000000B1          Programmable interrupt
controller          OK
0x000000B4-0x000000B5          Programmable interrupt
controller          OK
0x000000B8-0x000000B9          Programmable interrupt
controller          OK
```

```
0x000000BC-0x000000BD         Programmable interrupt
controller          OK
0x00000080-0x00000080         Direct memory access
controller          OK
0x00000081-0x00000083         Direct memory access
controller          OK
0x00000084-0x00000086         Direct memory access
controller          OK
0x00000087-0x00000087         Direct memory access
controller          OK
0x00000088-0x00000088         Direct memory access
controller          OK
0x00000089-0x0000008B         Direct memory access
controller          OK
0x0000008C-0x0000008E         Direct memory access
controller          OK
0x0000008F-0x0000008F         Direct memory access
controller          OK
0x00000090-0x00000091         Direct memory access
controller          OK
0x00000093-0x0000009F         Direct memory access
controller          OK
0x000000C0-0x000000DF         Direct memory access
controller          OK
0x00000040-0x00000043         System timer        OK

0x00000070-0x00000071         System CMOS/real time
clock     OK
0x00000072-0x00000073         System CMOS/real time
clock     OK
0x00000074-0x00000075         System CMOS/real time
clock     OK
0x00000076-0x00000077         System CMOS/real time
clock     OK
0x00000061-0x00000061         System speaker      OK

0x000000F0-0x000000FF         Numeric data processor
          OK
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
0x00000CA6-0x00000CA6         Microsoft ACPI-Compliant
Embedded Controller OK
0x00000CA7-0x00000CA7         Microsoft ACPI-Compliant
Embedded Controller OK
0x000003F2-0x000003F3         Standard floppy disk
controller          OK
0x000003F4-0x000003F5         Standard floppy disk
controller          OK
0x000003F7-0x000003F7         Standard floppy disk
controller          OK
0x000003F8-0x000003FF         Communications Port
(COM1)    OK
0x000002F8-0x000002FF         Communications Port
(COM2)    OK
0x00000378-0x0000037F         ECP Printer Port (LPT1)
          OK
0x00000778-0x0000077F         ECP Printer Port (LPT1)
          OK
0x000003A0-0x000003AF         Intel(r) 82801CA Ultra
ATA Storage Controller-248B   OK
0x000001F0-0x000001F7         Primary IDE Channel OK

0x000003F6-0x000003F6         Primary IDE Channel OK

0x00000170-0x00000177         Secondary IDE Channel
          OK
0x00000376-0x00000376         Secondary IDE Channel
          OK
0x00000580-0x0000059F         Intel(R) 82801CA/CAM
SMBus Controller - 2483       OK
```

```
[IRQs]

Resource  Device    Status
IRQ 9     Microsoft ACPI-Compliant System         OK

IRQ 52    Mylex eXtremeRAID 2000 Disk Array Controller
(Accelerated)       OK
IRQ 58    Intel(R) PRO/1000 MT Dual Port Network
Connection          OK
IRQ 59    Intel(R) PRO/1000 MT Dual Port Network
Connection #2       OK
IRQ 28    Mylex eXtremeRAID 2000 Disk Array Controller
(Accelerated)       OK
IRQ 24    Mylex eXtremeRAID 2000 Disk Array Controller
(Accelerated)       OK
IRQ 30    Adaptec AIC-7902-based Ultra320 SCSI    OK

IRQ 31    Adaptec AIC-7902-based Ultra320 SCSI    OK

IRQ 23    RAGE XL PCI Family (Microsoft Corporation)
          OK
IRQ 0     System timer        OK
IRQ 8     System CMOS/real time clock   OK
IRQ 13    Numeric data processor        OK
IRQ 12    Logitech PS/2 Port Mouse      OK
IRQ 1     Standard 101/102-Key or Microsoft Natural
PS/2 Keyboard       OK
IRQ 6     Standard floppy disk controller         OK

IRQ 4     Communications Port (COM1)    OK
IRQ 3     Communications Port (COM2)    OK
IRQ 14    Primary IDE Channel OK
IRQ 15    Secondary IDE Channel         OK

[Memory]

Resource  Device    Status
0xA0000-0xBFFFF     PCI bus   OK
0xA0000-0xBFFFF     RAGE XL PCI Family (Microsoft
Corporation)        OK
0xD0000000-0xFEBFFFFF         PCI bus   OK
0xFFC00000-0xFFFFFFFF         PCI bus   OK
0xFFC00000-0xFFFFFFFF         Intel(r) 82802 Firmware
Hub Device          OK
0xF2100000-0xFEBFFFFF         Intel(R) E7000 Series Hub
Interface C PCI-to-PCI Bridge - 2545    OK
0xF2100000-0xFEBFFFFF         Intel(r) 82870 Hub
Interface to PCI Bridges      OK
0xF2100000-0xFEBFFFFF         DEC 21154 PCI to PCI
bridge    OK
0xD8000000-0xEFFFFFFF         Intel(R) E7000 Series Hub
Interface C PCI-to-PCI Bridge - 2545    OK
0xD8000000-0xEFFFFFFF         Intel(r) 82870 Hub
Interface to PCI Bridges      OK
0xD8000000-0xEFFFFFFF         DEC 21154 PCI to PCI
bridge    OK
0xFEBE0000-0xFEBE0FFF         Intel(r) 82870 I/OxAPIC
Interrupt Controller          OK
0xFA800000-0xFEAFFFFF         Intel(r) 82870 Hub
Interface to PCI Bridges      OK
0xFA800000-0xFEAFFFFF         DEC 21154 PCI to PCI
bridge    OK
0xE8000000-0xEFFFFFFF         Intel(r) 82870 Hub
Interface to PCI Bridges      OK
0xE8000000-0xEFFFFFFF         DEC 21154 PCI to PCI
bridge    OK
0xFC000000-0xFDFFFFFF         Mylex eXtremeRAID 2000
Disk Array Controller (Accelerated)     OK
0xEC000000-0xEFFFFFFF         Mylex eXtremeRAID 2000
Disk Array Controller (Accelerated)     OK
```

```
0xFEAC0000-0xFEADFFFF         Intel(R) PRO/1000 MT Dual
Port Network Connection       OK
0xFEAE0000-0xFEAFFFFF         Intel(R) PRO/1000 MT Dual
Port Network Connection #2    OK
0xFEBF0000-0xFEBF0FFF         Intel(r) 82870 I/OxAPIC
Interrupt Controller          OK
0xF4000000-0xF5FFFFFF         Mylex eXtremeRAID 2000
Disk Array Controller (Accelerated)     OK
0xDC000000-0xDFFFFFFF         Mylex eXtremeRAID 2000
Disk Array Controller (Accelerated)     OK
0xF6300000-0xFA4FFFFF         DEC 21154 PCI to PCI
bridge    OK
0xE0000000-0xE7FFFFFF         DEC 21154 PCI to PCI
bridge    OK
0xF8000000-0xF9FFFFFF         Mylex eXtremeRAID 2000
Disk Array Controller (Accelerated)     OK
0xE4000000-0xE7FFFFFF         Mylex eXtremeRAID 2000
Disk Array Controller (Accelerated)     OK
0xFA7E0000-0xFA7E1FFF         Adaptec AIC-7902-based
Ultra320 SCSI       OK
0xFA7F0000-0xFA7F1FFF         Adaptec AIC-7902-based
Ultra320 SCSI       OK
0xF1000000-0xF1FFFFFF         RAGE XL PCI Family
(Microsoft Corporation)       OK
0xF20F0000-0xF20F0FFF         RAGE XL PCI Family
(Microsoft Corporation)       OK
0xFFF00000-0xFFFFFFFF         Intel(r) 82802 Firmware
Hub Device          OK
0xFFE00000-0xFFEFFFFF         Intel(r) 82802 Firmware
Hub Device          OK
0xFFD00000-0xFFDFFFFF         Intel(r) 82802 Firmware
Hub Device          OK
0xF20FFC00-0xF20FFFFF         Intel(r) 82801CA Ultra
ATA Storage Controller-248B   OK

[Components]



[Multimedia]



[Audio Codecs]

CODEC     Manufacturer        Description
          Status    File      Version   Size
          Creation Date
c:\windows\system32\tssoft32.acm        DSP GROUP, INC.
          OK
          C:\WINDOWS\system32\TSSOFT32.ACM        1.01
          9,50 KB (9.728 bytes)         25/3/2003 09:00

c:\windows\system32\msg711.acm          Microsoft
Corporation               OK
          C:\WINDOWS\system32\MSG711.ACM
          5.2.3790.0 (srv03_rtm.030324-2048)      10,00
KB (10.240 bytes)   25/3/2003 09:00
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)   25/3/2003 09:00
c:\windows\system32\msg723.acm          Microsoft
Corporation               OK
          C:\WINDOWS\system32\MSG723.ACM
          4.4.4000  116,00 KB (118.784 bytes)
          20/4/2004 13:46
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)
          25/3/2003 09:00
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)       25/3/2003 09:00

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)   25/3/2003 09:00
c:\windows\system32\msadp32.acm         Microsoft
Corporation                 OK
          C:\WINDOWS\system32\MSADP32.ACM
          5.2.3790.0 (srv03_rtm.030324-2048)      14,50
KB (14.848 bytes)   25/3/2003 09:00
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)     25/3/2003 09:00


[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)
          20/4/2004 13:46
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)    24/3/2003 22:50
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)   25/3/2003 09:00
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)
          24/3/2003 22:49
c:\windows\system32\msh263.drv          Microsoft
Corporation                 OK
          C:\WINDOWS\system32\MSH263.DRV
          4.4.4000  284,00 KB (290.816 bytes)
          24/3/2003 22:46
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)   24/3/2003 22:49
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)   25/3/2003 09:00

[CD-ROM]

Item      Value
Drive     K:
Description         CD-ROM Drive
Media Loaded        Yes
Media Type          CD-ROM
Name      LITEON CD-ROM LTN526S
Manufacturer        (Standard CD-ROM drives)
Status    OK
Transfer Rate       Not Available
SCSI Target ID      0
PNP Device ID       IDE\CDROMLITEON_CD-
ROM_LTN526S_____________________YSOJ____\5&13055604&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), 25/3/2003 09:00)

[Sound Device]

Item      Value

[Display]

Item      Value
Name      RAGE XL PCI Family (Microsoft Corporation)

PNP Device ID
          PCI\VEN_1002&DEV_4752&SUBSYS_34248086&REV_27\
4&27A7C225&0&60F0
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  atiixpad.inf (ati2mpad section)
Color Planes        1
Color Table Entries 4294967296
Resolution          1024 x 768 x 60 hertz
Bits/Pixel          32
Memory Address      0xF1000000-0xF1FFFFFF
I/O Port  0x00001000-0x00001OFF
Memory Address      0xF20F0000-0xF20F0FFF
IRQ Channel         IRQ 23
I/O Port  0x000003B0-0x000003BB
I/O Port  0x000003C0-0x000003DF
Memory Address      0xA0000-0xBFFFF
Driver    c:\windows\system32\drivers\ati2mpad.sys
(5.10.3663.6013, 335,38 KB (343.424 bytes), 20/4/2004
10:24)

[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    00000416
PNP Device ID       ACPI\PNP0303\4&1E30281&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), 25/3/2003 09:00)

[Pointing Device]

Item      Value
Hardware Type       Logitech PS/2 Port Mouse
Number of Buttons   5
Status    OK
PNP Device ID       ACPI\PNP0F12\4&1E30281&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), 25/3/2003 09:00)

[Modem]

Item      Value

[Network]



[Adapter]

Item      Value
Name      [00000001] Intel(R) PRO/1000 MT Dual Port
Network Connection
Adapter Type        Ethernet 802.3
Product Type        Intel(R) PRO/1000 MT Dual Port
Network Connection
Installed Yes
PNP Device ID
          PCI\VEN_8086&DEV_1010&SUBSYS_34248086&REV_01\
5&36D4E837&0&28E818
Last Reset          23/4/2004 15:50
Index     1
Service Name        E1000
IP Address          192.168.10.22
IP Subnet 255.255.255.0
Default IP Gateway  Not Available
DHCP Enabled        No
DHCP Server         Not Available
DHCP Lease Expires  Not Available
DHCP Lease Obtained Not Available
MAC Address         00:03:47:32:07:72
Memory Address      0xFEAC0000-0xFEADFFFF
I/O Port  0x00006040-0x0000607F
IRQ Channel         IRQ 58
Driver    c:\windows\system32\drivers\e1000325.sys
(6.3.6.31 built by: WinDDK, 99,00 KB (101.376 bytes),
20/4/2004 10:24)

Name      [00000002] Intel(R) PRO/1000 MT Dual Port
Network Connection
Adapter Type        Ethernet 802.3
Product Type        Intel(R) PRO/1000 MT Dual Port
Network Connection
Installed Yes
PNP Device ID
          PCI\VEN_8086&DEV_1010&SUBSYS_34248086&REV_01\
5&36D4E837&0&29E818
Last Reset          23/4/2004 15:50
Index     2
Service Name        E1000
IP Address          0.0.0.0
IP Subnet 0.0.0.0
Default IP Gateway  Not Available
DHCP Enabled        Yes
DHCP Server
DHCP Lease Expires  Not Available
DHCP Lease Obtained Not Available
MAC Address         00:03:47:32:07:73
```

```
Memory Address      0xFEAE0000-0xFEAFFFFF
I/O Port  0x00006000-0x0000603F
IRQ Channel         IRQ 59
Driver    c:\windows\system32\drivers\e1000325.sys
(6.3.6.31 built by: WinDDK, 99,00 KB (101.376 bytes),
20/4/2004 10:24)

Name      [00000003] RAS Async Adapter
Adapter Type        Not Available
Product Type        RAS Async Adapter
Installed Yes
PNP Device ID       Not Available
Last Reset          23/4/2004 15:50
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          23/4/2004 15:50
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), 25/3/2003 09:00)

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          23/4/2004 15:50
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), 25/3/2003 09:00)

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          23/4/2004 15:50
Index     6
Service Name        RasPppoe
IP Address          Not Available
IP Subnet Not Available
Default IP Gateway  Not Available
DHCP Enabled        No
DHCP Server         Not Available
DHCP Lease Expires  Not Available
DHCP Lease Obtained Not Available
MAC Address         33:50:6F:45:30:30
Driver    c:\windows\system32\drivers\raspppoe.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 38,00 KB (38.912
bytes), 25/3/2003 09:00)

Name      [00000007] Direct Parallel
Adapter Type        Not Available
Product Type        Direct Parallel
Installed Yes
PNP Device ID       ROOT\MS_PTIMINIPORT\0000
Last Reset          23/4/2004 15:50
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), 25/3/2003 09:00)

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          23/4/2004 15:50
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), 25/3/2003 09:00)

[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_{E61F2C76-
9C01-4F6A-AE87-7E932C9C6387}] SEQPACKET 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_{E61F2C76-
9C01-4F6A-AE87-7E932C9C6387}] 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_{B6109A63-
3896-4ED0-8C79-43145F20E6AA}] SEQPACKET 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_{B6109A63-
3896-4ED0-8C79-43145F20E6AA}] 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_{8921B94F-
1723-4684-9866-F76C48EA50B5}] SEQPACKET 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_{8921B94F-
1723-4684-9866-F76C48EA50B5}] 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_{F3618BE3-
D332-421C-AFB8-9B23E565A035}] SEQPACKET 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_{F3618BE3-
D332-421C-AFB8-9B23E565A035}] 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\1
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
I/O Port  0x000003F8-0x000003FF
IRQ Channel         IRQ 4
Driver    c:\windows\system32\drivers\serial.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 76,00 KB (77.824
bytes), 25/3/2003 09:00)

Name      Communications Port (COM2)
Status    OK
PNP Device ID       ACPI\PNP0501\2
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
I/O Port  0x000002F8-0x000002FF
IRQ Channel         IRQ 3
Driver    c:\windows\system32\drivers\serial.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 76,00 KB (77.824
bytes), 25/3/2003 09:00)

[Parallel]

Item      Value
Name      LPT1
PNP Device ID       ACPI\PNP0401\4&1E30281&0
I/O Port  0x00000378-0x0000037F
I/O Port  0x00000778-0x0000077F
DMA Channel         Channel 3
Driver    c:\windows\system32\drivers\parport.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 76,50 KB (78.336
bytes), 24/3/2003 20:04)

[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      34,16 GB (36.676.489.216 bytes)
Free Space          30,60 GB (32.856.858.624 bytes)

Volume Name
Volume Serial Number          40BB8589

Drive     K:
Description         CD-ROM Disc

Drive     M:
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     O:
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     P:
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     R:
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     S:
Description         Local Fixed Disk
Compressed          Not Available
File System         Not Available
Size      Not Available
Free Space          Not Available
Volume Name         Not Available
Volume Serial Number          Not Available

Drive     W:
Description         Local Fixed Disk
Compressed          No
File System         NTFS
Size      880,63 GB (945.569.959.936 bytes)
Free Space          816,01 GB (876.182.900.736 bytes)

Volume Name         backup1
Volume Serial Number          88747289

Drive     X:
Description         Local Fixed Disk
Compressed          No
File System         NTFS
Size      880,63 GB (945.569.959.936 bytes)
Free Space          817,78 GB (878.083.129.344 bytes)

Volume Name         backup2
Volume Serial Number          A47F7269

[Disks]

Item      Value
Description         \\.\PHYSICALDRIVE0
Manufacturer        Not Available
Model     Not Available
Bytes/Sector        512
Media Loaded        Yes
Media Type          Fixed hard disk
Partitions          3
SCSI Bus  4
SCSI Logical Unit   0
SCSI Port 2
SCSI Target ID      0
Sectors/Track       63
Size      956,64 GB (1.027.189.416.960 bytes)
Total Cylinders     124.882
Total Sectors       2.006.229.330
Total Tracks        31.844.910
Tracks/Cylinder     255
Partition Disk #0, Partition #0
Partition Size      26,01 GB (27.924.793.344 bytes)

Partition Starting Offset     32.256 bytes
Partition Disk #0, Partition #1
Partition Size      50,01 GB (53.694.627.840 bytes)

Partition Starting Offset     27.924.825.600 bytes

Partition Disk #0, Partition #2
Partition Size      880,63 GB (945.569.963.520 bytes)

Partition Starting Offset     81.619.453.440 bytes


Description         \\.\PHYSICALDRIVE2
Manufacturer        Not Available
Model     Not Available
Bytes/Sector        512
Media Loaded        Yes
Media Type          Fixed hard disk
Partitions          1
SCSI Bus  4
SCSI Logical Unit   0
SCSI Port 4
SCSI Target ID      0
Sectors/Track       63
Size      34,17 GB (36.684.748.800 bytes)
Total Cylinders     4.460
Total Sectors       71.649.900
Total Tracks        1.137.300
Tracks/Cylinder     255
Partition Disk #2, Partition #0
Partition Size      34,16 GB (36.676.491.264 bytes)

Partition Starting Offset     32.256 bytes

Description         \\.\PHYSICALDRIVE3
Manufacturer        Not Available
Model     Not Available
Bytes/Sector        512
Media Loaded        Yes
Media Type          Fixed hard disk
Partitions          1
SCSI Bus  4
SCSI Logical Unit   0
SCSI Port 4
SCSI Target ID      1
Sectors/Track       63
Size      68,33 GB (73.369.497.600 bytes)
Total Cylinders     8.920
Total Sectors       143.299.800
Total Tracks        2.274.600
Tracks/Cylinder     255
Partition Disk #3, Partition #0
```

```
Partition Size      68,33 GB (73.369.465.344 bytes)

Partition Starting Offset     32.256 bytes

Description         \\.\PHYSICALDRIVE1
Manufacturer        Not Available
Model     Not Available
Bytes/Sector        512
Media Loaded        Yes
Media Type          Fixed hard disk
Partitions          3
SCSI Bus  4
SCSI Logical Unit   0
SCSI Port 3
SCSI Target ID      0
Sectors/Track       63
Size      956,64 GB (1.027.189.416.960 bytes)
Total Cylinders     124.882
Total Sectors       2.006.229.330
Total Tracks        31.844.910
Tracks/Cylinder     255
Partition Disk #1, Partition #0
Partition Size      26,01 GB (27.924.793.344 bytes)

Partition Starting Offset     32.256 bytes
Partition Disk #1, Partition #1
Partition Size      50,01 GB (53.694.627.840 bytes)

Partition Starting Offset     27.924.825.600 bytes

Partition Disk #1, Partition #2
Partition Size      880,63 GB (945.569.963.520 bytes)

Partition Starting Offset     81.619.453.440 bytes


[SCSI]

Item      Value
Name      Mylex eXtremeRAID 2000 Disk Array Controller
(Accelerated)
Manufacturer        Mylex
Status    OK
PNP Device ID
          PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\
6&175A51B6&0&4010E818
Memory Address      0xFC000000-0xFDFFFFFF
I/O Port  0x00005000-0x00006FFF
Memory Address      0xEC000000-0xEFFFFFFF
IRQ Channel         IRQ 52
Driver    c:\windows\system32\drivers\dac2w2k.sys
(7.00-14, 172,75 KB (176.896 bytes), 23/9/2003 10:46)


Name      Mylex eXtremeRAID 2000 Disk Array Controller
(Accelerated)
Manufacturer        Mylex
Status    OK
PNP Device ID
          PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\
6&3B7FF379&0&4008F818
Memory Address      0xF4000000-0xF5FFFFFF
I/O Port  0x00002000-0x00006FFF
Memory Address      0xDC000000-0xDFFFFFFF
IRQ Channel         IRQ 28
Driver    c:\windows\system32\drivers\dac2w2k.sys
(7.00-14, 172,75 KB (176.896 bytes), 23/9/2003 10:46)


Name      Mylex eXtremeRAID 2000 Disk Array Controller
(Accelerated)
Manufacturer        Mylex
Status    OK
PNP Device ID
          PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\
6&17F0941A&0&4010F818
Memory Address      0xF8000000-0xF9FFFFFF
I/O Port  0x00003000-0x00003FFF
Memory Address      0xE4000000-0xE7FFFFFF
IRQ Channel         IRQ 24
Driver    c:\windows\system32\drivers\dac2w2k.sys
(7.00-14, 172,75 KB (176.896 bytes), 23/9/2003 10:46)


Name      Adaptec AIC-7902-based Ultra320 SCSI
Manufacturer        Adaptec
Status    OK
PNP Device ID
          PCI\VEN_9005&DEV_801F&SUBSYS_34248086&REV_03\
5&279870C6&0&20F818
I/O Port  0x00004800-0x000048FF
Memory Address      0xFA7E0000-0xFA7E1FFF
I/O Port  0x00004C00-0x00004CFF
IRQ Channel         IRQ 30
Driver    c:\windows\system32\drivers\adpu320.sys
(6.0.001.000 (Lab01_N(portbld).020729-2000), 101,63 KB
(104.064 bytes), 25/3/2003 09:00)

Name      Adaptec AIC-7902-based Ultra320 SCSI
Manufacturer        Adaptec
Status    OK
PNP Device ID
          PCI\VEN_9005&DEV_801F&SUBSYS_34248086&REV_03\
5&279870C6&0&21F818
I/O Port  0x00004000-0x000040FF
Memory Address      0xFA7F0000-0xFA7F1FFF
I/O Port  0x00004400-0x000044FF
IRQ Channel         IRQ 31
Driver    c:\windows\system32\drivers\adpu320.sys
(6.0.001.000 (Lab01_N(portbld).020729-2000), 101,63 KB
(104.064 bytes), 25/3/2003 09:00)

[IDE]

Item      Value
Name      Intel(r) 82801CA Ultra ATA Storage
Controller-248B
Manufacturer        Intel
Status    OK
PNP Device ID
          PCI\VEN_8086&DEV_248B&SUBSYS_34248086&REV_02\
3&267A616A&0&F9
I/O Port  0x000003A0-0x000003AF
Memory Address      0xF20FFC00-0xF20FFFFF
Driver    c:\windows\system32\drivers\intelide.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 7,00 KB (7.168
bytes), 25/3/2003 09:00)

Name      Primary IDE Channel
Manufacturer        (Standard IDE ATA/ATAPI
controllers)
Status    OK
PNP Device ID       PCIIDE\IDECHANNEL\4&380A97AE&0&0

I/O Port  0x000001F0-0x000001F7
I/O Port  0x000003F6-0x000003F6
IRQ Channel         IRQ 14
Driver    c:\windows\system32\drivers\atapi.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 89,00 KB (91.136
bytes), 25/3/2003 09:00)

Name      Secondary IDE Channel
Manufacturer        (Standard IDE ATA/ATAPI
controllers)
Status    OK
PNP Device ID       PCIIDE\IDECHANNEL\4&380A97AE&0&1

I/O Port  0x00000170-0x00000177
I/O Port  0x00000376-0x00000376
IRQ Channel         IRQ 15
Driver    c:\windows\system32\drivers\atapi.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 89,00 KB (91.136
bytes), 25/3/2003 09:00)

[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
          Ignore    No        No
acpi      Microsoft ACPI Driver
          c:\windows\system32\drivers\acpi.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

acpiec    Microsoft Embedded Controller Driver
          c:\windows\system32\drivers\acpiec.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

adpu160m  adpu160m  Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
adpu320   adpu320
          c:\windows\system32\drivers\adpu320.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

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
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\atapi.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

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
cpqarry2  cpqarry2  Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No

cpqcissm  cpqcissm  Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
cpqfcalm  cpqfcalm  Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
crcdisk   CRC Disk Filter Driver
          c:\windows\system32\drivers\crcdisk.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

dac2w2k   dac2w2k
          c:\windows\system32\drivers\dac2w2k.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

dac960nt  dac960nt  Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
dellcerc  dellcerc  Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
dfsdriver DfsDriver c:\windows\system32\drivers\dfs.sys
          File System Driver  Yes       Boot
          Running   OK        Normal    No        Yes

disk      Disk Driver
          c:\windows\system32\drivers\disk.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

dmboot    dmboot
          c:\windows\system32\drivers\dmboot.sys
          Kernel Driver       No        Disabled
          Stopped   OK        Normal    No        No

dmio      Logical Disk Manager Driver
          c:\windows\system32\drivers\dmio.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

dmload    dmload
          c:\windows\system32\drivers\dmload.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

dpti2o    dpti2o    Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
e1000     Intel(R) PRO/1000 Device Driver
          c:\windows\system32\drivers\e1000325.sys
          Kernel Driver       Yes       Manual
          Running   OK        Normal    No        Yes

fastfat   Fastfat
          c:\windows\system32\drivers\fastfat.sys File
System Driver       Yes       Disabled  Running   OK
          Normal    No        Yes
fdc       Floppy Disk Controller Driver
          c:\windows\system32\drivers\fdc.sys
          Kernel Driver       Yes       Manual
          Running   OK        Normal    No        Yes

fips      Fips
          c:\windows\system32\drivers\fips.sys
          Kernel Driver       Yes       System
          Running   OK        Normal    No        Yes

flpydisk  Floppy Disk Driver
          c:\windows\system32\drivers\flpydisk.sys
          Kernel Driver       Yes       Manual
          Running   OK        Normal    No        Yes

ftdisk    Volume Manager Driver
          c:\windows\system32\drivers\ftdisk.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

gpc       Generic Packet Classifier
          c:\windows\system32\drivers\msgpc.sys
          Kernel Driver       Yes       Manual
          Running   OK        Normal    No        Yes

hpn       hpn       Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
hpt3xx    hpt3xx    Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
http      HTTP
          c:\windows\system32\drivers\http.sys
          Kernel Driver       No        Manual
          Stopped   OK        Normal    No        No

i2omgmt   i2omgmt   Not Available       Kernel Driver
          No        System    Stopped   OK
          Normal    No        No
i2omp     i2omp     Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
i8042prt  i8042 Keyboard and PS/2 Mouse Port Driver
          c:\windows\system32\drivers\i8042prt.sys
          Kernel Driver       Yes       System
          Running   OK        Normal    No        Yes

iirsp     iirsp     Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
imapi     CD-Burning Filter Driver
          c:\windows\system32\drivers\imapi.sys
          Kernel Driver       No        System
          Stopped   OK        Normal    No        No

intelide  IntelIde
          c:\windows\system32\drivers\intelide.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

ipfilterdriver      IP Traffic Filter Driver
          c:\windows\system32\drivers\ipfltdrv.sys
          Kernel Driver       No        Manual
          Stopped   OK        Normal    No        No

ipinip    IP in IP Tunnel Driver
          c:\windows\system32\drivers\ipinip.sys
          Kernel Driver       No        Manual
          Stopped   OK        Normal    No        No

ipnat     IP Network Address Translator
          c:\windows\system32\drivers\ipnat.sys
          Kernel Driver       No        Manual
          Stopped   OK        Normal    No        No

ipsec     IPSEC driver
          c:\windows\system32\drivers\ipsec.sys
          Kernel Driver       Yes       System
          Running   OK        Normal    No        Yes

ipsraidn  ipsraidn  Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
irenum    IR Enumerator Service
          c:\windows\system32\drivers\irenum.sys
```

```
          Kernel Driver       No        Manual
          Stopped   OK        Normal    No        No

isapnp    PnP ISA/EISA Bus Driver
          c:\windows\system32\drivers\isapnp.sys
          Kernel Driver       Yes       Boot
          Running   OK        Critical  No        Yes

kbdclass  Keyboard Class Driver
          c:\windows\system32\drivers\kbdclass.sys
          Kernel Driver       Yes       System
          Running   OK        Normal    No        Yes

ksecdd    KSecDD
          c:\windows\system32\drivers\ksecdd.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

lp6nds35  lp6nds35  Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
mac2w2k   mac2w2k
          c:\windows\system32\drivers\mac2w2k.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

mnmdd     mnmdd
          c:\windows\system32\drivers\mnmdd.sys
          Kernel Driver       Yes       System
          Running   OK        Ignore    No        Yes

modem     Modem
          c:\windows\system32\drivers\modem.sys
          Kernel Driver       No        Manual
          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       No        Manual
          Stopped   OK        Normal    No        No

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   Parallel port driver
          c:\windows\system32\drivers\parport.sys
          Kernel Driver       Yes       Manual
          Running   OK        Normal    No        Yes

partmgr   Partition Manager
          c:\windows\system32\drivers\partmgr.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

parvdm    Parvdm
          c:\windows\system32\drivers\parvdm.sys
          Kernel Driver       Yes       Auto
          Running   OK        Ignore    No        Yes

pci       PCI Bus Driver
          c:\windows\system32\drivers\pci.sys
          Kernel Driver       Yes       Boot
          Running   OK        Critical  No        Yes

pciide    PCIIde    Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
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   PDFRAME   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\raspptp.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

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
```

```
raspti    Direct Parallel
          c:\windows\system32\drivers\raspti.sys
          Kernel Driver       Yes       Manual
          Running   OK        Normal    No        Yes

rdbss     Rdbss
          c:\windows\system32\drivers\rdbss.sys   File
System Driver       Yes       System    Running   OK
          Normal    No        Yes
rdpcdd    RDPCDD
          c:\windows\system32\drivers\rdpcdd.sys
          Kernel Driver       Yes       System
          Running   OK        Ignore    No        Yes

rdpdr     Terminal Server Device Redirector Driver
          c:\windows\system32\drivers\rdpdr.sys
          Kernel Driver       Yes       Manual
          Running   OK        Normal    No        Yes

rdpwd     RDPWD
          c:\windows\system32\drivers\rdpwd.sys
          Kernel Driver       No        Manual
          Stopped   OK        Ignore    No        No

redbook   Digital CD Audio Playback Filter Driver
          c:\windows\system32\drivers\redbook.sys
          Kernel Driver       Yes       System
          Running   OK        Normal    No        Yes

secdrv    Secdrv
          c:\windows\system32\drivers\secdrv.sys
          Kernel Driver       No        Manual
          Stopped   OK        Normal    No        No

serenum   Serenum Filter Driver
          c:\windows\system32\drivers\serenum.sys
          Kernel Driver       Yes       Manual
          Running   OK        Normal    No        Yes

serial    Serial port driver
          c:\windows\system32\drivers\serial.sys
          Kernel Driver       Yes       System
          Running   OK        Ignore    No        Yes

sfloppy   Sfloppy
          c:\windows\system32\drivers\sfloppy.sys
          Kernel Driver       No        System
          Stopped   OK        Ignore    No        No

simbad    Simbad    Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
sparrow   Sparrow   Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
srv       Srv       c:\windows\system32\drivers\srv.sys
          File System Driver  No        Manual
          Stopped   OK        Normal    No        No

swenum    Software Bus Driver
          c:\windows\system32\drivers\swenum.sys
          Kernel Driver       Yes       Manual
          Running   OK        Normal    No        Yes

symc810   symc810   Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
symc8xx   symc8xx   Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No

symmpi    symmpi    Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
sym_hi    sym_hi    Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
sym_u3    sym_u3    Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
tcpip     TCP/IP Protocol Driver
          c:\windows\system32\drivers\tcpip.sys
          Kernel Driver       Yes       System
          Running   OK        Normal    No        Yes

tdpipe    TDPIPE
          c:\windows\system32\drivers\tdpipe.sys
          Kernel Driver       No        Manual
          Stopped   OK        Ignore    No        No

tdtcp     TDTCP
          c:\windows\system32\drivers\tdtcp.sys
          Kernel Driver       No        Manual
          Stopped   OK        Ignore    No        No

termdd    Terminal Device Driver
          c:\windows\system32\drivers\termdd.sys
          Kernel Driver       Yes       System
          Running   OK        Normal    No        Yes

toside    TosIde    Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
udfs      Udfs
          c:\windows\system32\drivers\udfs.sys    File
System Driver       No        Disabled  Stopped   OK
          Normal    No        No
ultra     ultra     Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
update    Microcode Update Driver
          c:\windows\system32\drivers\update.sys
          Kernel Driver       Yes       Manual
          Running   OK        Normal    No        Yes

vgasave   VGA Display Controller.
          c:\windows\system32\drivers\vga.sys
          Kernel Driver       Yes       System
          Running   OK        Ignore    No        Yes

viaide    ViaIde    Not Available       Kernel Driver
          No        Disabled  Stopped   OK
          Normal    No        No
volsnap   Storage volumes
          c:\windows\system32\drivers\volsnap.sys
          Kernel Driver       Yes       Boot
          Running   OK        Normal    No        Yes

wanarp    Remote Access IP ARP Driver
          c:\windows\system32\drivers\wanarp.sys
          Kernel Driver       Yes       Manual
          Running   OK        Normal    No        Yes

wdica     WDICA     Not Available       Kernel Driver
          No        Manual    Stopped   OK
          Ignore    No        No
wlbs      Network Load Balancing
          c:\windows\system32\drivers\wlbs.sys
          Kernel Driver       No        Manual
          Stopped   OK        Normal    No        No


[Signed Drivers]

Device Name         Signed    Device Class
          Driver Version      Driver Date
          Manufacturer        INF Name  Driver Name
          Device ID
Not Available       Not Available       Not Available
          Not Available       Not Available       Not
Available Not Available       Not Available
          HTREE\ROOT\0
ACPI Multiprocessor PC        No        COMPUTER
          5.2.3790.0          10/1/2002 (Standard
computers)          hal.inf   Not Available
          ROOT\ACPI_HAL\0000
Microsoft ACPI-Compliant System         No
          SYSTEM    5.2.3790.0          10/1/2002
          Microsoft acpi.inf  Not Available
          ACPI_HAL\PNP0C08\0
Processor No        PROCESSOR 5.2.3790.0
          10/1/2002 (Standard processor types)
          cpu.inf   Not Available
          ACPI\GENUINEINTEL_-_X86_FAMILY_15_MODEL_2\_0

ACPI Sleep Button   No        SYSTEM    5.2.3790.0
          10/1/2002 (Standard system devices)
          machine.inf         Not Available
          ACPI\PNP0C0E\2&DABA3FF&0
PCI bus   No        SYSTEM    5.2.3790.0
          10/1/2002 (Standard system devices)
          machine.inf         Not Available
          ACPI\PNP0A03\0
Intel(R) E7501 Chipset Host Controller - 254C     No
          SYSTEM    5.2.3790.0          10/1/2002 Intel
          machine.inf         Not Available
          PCI\VEN_8086&DEV_254C&SUBSYS_00000000&REV_01\
3&267A616A&0&00
Intel(R) E7000 Series Host RASUM Controller - 2541
          No        SYSTEM    5.2.3790.0
          10/1/2002 Intel     machine.inf         Not
Available
          PCI\VEN_8086&DEV_2541&SUBSYS_34248086&REV_01\
3&267A616A&0&01
Intel(R) E7000 Series Hub Interface C PCI-to-PCI Bridge
- 2545    No        SYSTEM    5.2.3790.0
          10/1/2002 Intel     machine.inf         Not
Available
          PCI\VEN_8086&DEV_2545&SUBSYS_00000000&REV_01\
3&267A616A&0&18
Intel(r) 82870 I/OxAPIC Interrupt Controller      No
          SYSTEM    5.2.3790.0          10/1/2002 Intel
          machine.inf         Not Available
          PCI\VEN_8086&DEV_1461&SUBSYS_34248086&REV_04\
4&38945BE0&0&E018
Intel(r) 82870 Hub Interface to PCI Bridges       No
          SYSTEM    5.2.3790.0          10/1/2002 Intel
          machine.inf         Not Available
          PCI\VEN_8086&DEV_1460&SUBSYS_00000000&REV_04\
4&38945BE0&0&E818
DEC 21154 PCI to PCI bridge   No        SYSTEM
          5.2.3790.0          10/1/2002 DEC
          machine.inf         Not Available
          PCI\VEN_1011&DEV_0026&SUBSYS_00000000&REV_05\
5&36D4E837&0&10E818
Mylex eXtremeRAID 2000 Disk Array Controller
(Accelerated)       No        SCSIADAPTER
          7.0.14.0  9/9/2002  Mylex     oem2.inf  Not
Available
          PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\
6&175A51B6&0&4010E818
SCSI Enclosure Services Processor       No
          SYSTEM    1.0.0.0   11/8/2001 Eurologic
          oem0.inf  Not Available
```

```
        SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE
V_0017\7&14E2B2C9&0&1F0
SCSI Enclosure Services Processor       No
        SYSTEM    1.0.0.0   11/8/2001 Eurologic
        oem0.inf  Not Available
        SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE
V_0017\7&14E2B2C9&0&2F0
Mylex RAID Disk Device          No        DISKDRIVE
        5.2.3790.0          10/1/2002 Mylex
        disk.inf  Not Available
        SCSI\DISK&VEN_MYLEX&PROD_EXTREMERAID_2000&REV
_0702\7&14E2B2C9&0&400
Mylex GAM Device    No        SYSTEM    5.2.3790.0
        10/1/2002 Mylex     scsidev.inf         Not
Available
        SCSI\PROCESSOR&VEN_MYLEX&PROD_GAM_DEVICE&REV_
\7&14E2B2C9&0&660
Intel(R) PRO/1000 MT Dual Port Network Connection No
        NET       6.3.6.31  10/1/2002 Intel
        nete1000.inf        Not Available
        PCI\VEN_8086&DEV_1010&SUBSYS_34248086&REV_01\
5&36D4E837&0&28E818
Intel(R) PRO/1000 MT Dual Port Network Connection No
        NET       6.3.6.31  10/1/2002 Intel
        nete1000.inf        Not Available
        PCI\VEN_8086&DEV_1010&SUBSYS_34248086&REV_01\
5&36D4E837&0&29E818
Intel(r) 82870 I/OxAPIC Interrupt Controller      No
        SYSTEM    5.2.3790.0          10/1/2002 Intel
        machine.inf         Not Available
        PCI\VEN_8086&DEV_1461&SUBSYS_34248086&REV_04\
4&38945BE0&0&F018
Intel(r) 82870 Hub Interface to PCI Bridges       No
        SYSTEM    5.2.3790.0          10/1/2002 Intel
        machine.inf         Not Available
        PCI\VEN_8086&DEV_1460&SUBSYS_00000000&REV_04\
4&38945BE0&0&F818
DEC 21154 PCI to PCI bridge   No        SYSTEM
        5.2.3790.0          10/1/2002 DEC
        machine.inf         Not Available
        PCI\VEN_1011&DEV_0026&SUBSYS_00000000&REV_05\
5&279870C6&0&08F818
Mylex eXtremeRAID 2000 Disk Array Controller
(Accelerated)       No        SCSIADAPTER
        7.0.14.0  9/9/2002  Mylex     oem2.inf  Not
Available
        PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\
6&3B7FF379&0&4008F818
SCA Hotswap Backplane         No        SYSTEM
        5.2.3790.0          10/1/2002 ESG-SHV
        scsidev.inf         Not Available
        SCSI\PROCESSOR&VEN_ESG-
SHV&PROD_SCA_HSBP_M15&REV_0.10\7&2CA21E49&0&060
Mylex RAID Disk Device          No        DISKDRIVE
        5.2.3790.0          10/1/2002 Mylex
        disk.inf  Not Available
        SCSI\DISK&VEN_MYLEX&PROD_EXTREMERAID_2000&REV
_0702\7&2CA21E49&0&400
Mylex RAID Disk Device          No        DISKDRIVE
        5.2.3790.0          10/1/2002 Mylex
        disk.inf  Not Available
        SCSI\DISK&VEN_MYLEX&PROD_EXTREMERAID_2000&REV
_0702\7&2CA21E49&0&410
Mylex GAM Device    No        SYSTEM    5.2.3790.0
        10/1/2002 Mylex     scsidev.inf         Not
Available
        SCSI\PROCESSOR&VEN_MYLEX&PROD_GAM_DEVICE&REV_
\7&2CA21E49&0&660
DEC 21154 PCI to PCI bridge   No        SYSTEM
        5.2.3790.0          10/1/2002 DEC
        machine.inf         Not Available
        PCI\VEN_1011&DEV_0026&SUBSYS_00000000&REV_05\
5&279870C6&0&10F818
Mylex eXtremeRAID 2000 Disk Array Controller
(Accelerated)       No        SCSIADAPTER
        7.0.14.0  9/9/2002  Mylex     oem2.inf  Not
Available
        PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\
6&17F0941A&0&4010F818
SCSI Enclosure Services Processor       No
        SYSTEM    1.0.0.0   11/8/2001 Eurologic
        oem0.inf  Not Available
        SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE
V_0017\7&3008E2BD&0&1F0
SCSI Enclosure Services Processor       No
        SYSTEM    1.0.0.0   11/8/2001 Eurologic
        oem0.inf  Not Available
        SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE
V_0014\7&3008E2BD&0&2F0
Mylex RAID Disk Device          No        DISKDRIVE
        5.2.3790.0          10/1/2002 Mylex
        disk.inf  Not Available
        SCSI\DISK&VEN_MYLEX&PROD_EXTREMERAID_2000&REV
_0702\7&3008E2BD&0&400
Mylex GAM Device    No        SYSTEM    5.2.3790.0
        10/1/2002 Mylex     scsidev.inf         Not
Available
        SCSI\PROCESSOR&VEN_MYLEX&PROD_GAM_DEVICE&REV_
\7&3008E2BD&0&660
Adaptec AIC-7902-based Ultra320 SCSI    No
        SCSIADAPTER         5.2.3790.0
        10/1/2002 Adaptec   pnpscsi.inf         Not
Available
        PCI\VEN_9005&DEV_801F&SUBSYS_34248086&REV_03\
5&279870C6&0&20F818
Adaptec AIC-7902-based Ultra320 SCSI    No
        SCSIADAPTER         5.2.3790.0
        10/1/2002 Adaptec   pnpscsi.inf         Not
Available
        PCI\VEN_9005&DEV_801F&SUBSYS_34248086&REV_03\
5&279870C6&0&21F818
Intel(R) E7000 Series Hub Interface C RASUM Controller
- 2546    No        SYSTEM    5.2.3790.0
        10/1/2002 Intel     machine.inf         Not
Available
        PCI\VEN_8086&DEV_2546&SUBSYS_34248086&REV_01\
3&267A616A&0&19
Intel(R) 82801DB PCI Bridge - 244E      No
        SYSTEM    5.2.3790.0          10/1/2002 Intel
        machine.inf         Not Available
        PCI\VEN_8086&DEV_244E&SUBSYS_00000000&REV_42\
3&267A616A&0&F0
RAGE XL PCI Family (Microsoft Corporation)        No
        DISPLAY   5.10.2600.6014      8/8/2001  ATI
Technologies Inc.   atiixpad.inf        Not Available
        PCI\VEN_1002&DEV_4752&SUBSYS_34248086&REV_27\
4&27A7C225&0&60F0
Default Monitor     No        MONITOR   5.1.2001.0
        6/6/2001  (Standard monitor types)
        monitor.inf         Not Available
        DISPLAY\DEFAULT_MONITOR\5&1FCD025&0&80000000&
01&0C
PCI standard ISA bridge       No        SYSTEM
        5.2.3790.0          10/1/2002 (Standard
system devices)     machine.inf         Not Available
        PCI\VEN_8086&DEV_2480&SUBSYS_00000000&REV_02\
3&267A616A&0&F8
ISAPNP Read Data Port         No        SYSTEM
        5.2.3790.0          10/1/2002 (Standard
system devices)     machine.inf         Not Available
        ISAPNP\READDATAPORT\0
Motherboard resources         No        SYSTEM
        5.2.3790.0          10/1/2002 (Standard
system devices)     machine.inf         Not Available
        ACPI\PNP0C02\4&1E30281&0
Programmable interrupt controller       No
        SYSTEM    5.2.3790.0          10/1/2002
        (Standard system devices)     machine.inf
        Not Available       ACPI\PNP0000\4&1E30281&0

Direct memory access controller         No
        SYSTEM    5.2.3790.0          10/1/2002
        (Standard system devices)     machine.inf
        Not Available       ACPI\PNP0200\4&1E30281&0

System timer        No        SYSTEM    5.2.3790.0
        10/1/2002 (Standard system devices)
        machine.inf         Not Available
        ACPI\PNP0100\4&1E30281&0
System CMOS/real time clock   No        SYSTEM
        5.2.3790.0          10/1/2002 (Standard
system devices)     machine.inf         Not Available
        ACPI\PNP0B00\4&1E30281&0
System speaker      No        SYSTEM    5.2.3790.0
        10/1/2002 (Standard system devices)
        machine.inf         Not Available
        ACPI\PNP0800\4&1E30281&0
Numeric data processor        No        SYSTEM
        5.2.3790.0          10/1/2002 (Standard
system devices)     machine.inf         Not Available
        ACPI\PNP0C04\4&1E30281&0
Logitech PS/2 Port Mouse      No        MOUSE
        5.2.3790.0          10/1/2002 Logitech
        msmouse.inf         Not Available
        ACPI\PNP0F12\4&1E30281&0
Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
        No        KEYBOARD  5.2.3790.0
        10/1/2002 (Standard keyboards)
        keyboard.inf        Not Available
        ACPI\PNP0303\4&1E30281&0
Microsoft ACPI-Compliant Embedded Controller      No
        SYSTEM    5.2.3790.0          10/1/2002
        Microsoft acpi.inf  Not Available
        ACPI\PNP0C09\4&1E30281&0
Standard floppy disk controller         No        FDC
        5.2.3790.0          10/1/2002 (Standard
floppy disk controllers)      fdc.inf   Not Available
        ACPI\PNP0700\4&1E30281&0
Floppy disk drive   No        FLOPPYDISK
        5.2.3790.0          10/1/2002 (Standard
floppy disk drives) flpydisk.inf        Not Available
        FDC\GENERIC_FLOPPY_DRIVE\5&2CD86306&0&0
Communications Port No        PORTS     5.2.3790.0
        10/1/2002 (Standard port types)
        msports.inf         Not Available
        ACPI\PNP0501\1
Communications Port No        PORTS     5.2.3790.0
        10/1/2002 (Standard port types)
        msports.inf         Not Available
        ACPI\PNP0501\2
ECP Printer Port    No        PORTS     5.2.3790.0
        10/1/2002 (Standard port types)
        msports.inf         Not Available
        ACPI\PNP0401\4&1E30281&0
Printer Port Logical Interface          No
        SYSTEM    5.2.3790.0          10/1/2002
        (Standard system devices)     machine.inf
        Not Available
        LPTENUM\MICROSOFTRAWPORT\5&34CA7A0C&0&LPT1

Intel(r) 82802 Firmware Hub Device      No
        SYSTEM    5.2.3790.0          10/1/2002 Intel
        machine.inf         Not Available
        ACPI\INT0800\4&1E30281&0
```

```
Intel(r) 82801CA Ultra ATA Storage Controller-248B
          No        HDC       5.2.3790.0
          10/1/2002 Intel     mshdc.inf Not Available
          PCI\VEN_8086&DEV_248B&SUBSYS_34248086&REV_02\
3&267A616A&0&F9
Primary IDE Channel No        HDC       5.2.3790.0
          10/1/2002 (Standard IDE ATA/ATAPI
controllers)      mshdc.inf Not Available
          PCIIDE\IDECHANNEL\4&380A97AE&0&0
CD-ROM Drive      No        CDROM     5.2.3790.0
          10/1/2002 (Standard CD-ROM drives)
          cdrom.inf Not Available
          IDE\CDROMLITEON_CD-
ROM_LTN526S_____________________YS0J____\5&13055604&0&0.0
.0
Secondary IDE Channel         No        HDC
          5.2.3790.0          10/1/2002 (Standard IDE
ATA/ATAPI controllers)        mshdc.inf Not Available
          PCIIDE\IDECHANNEL\4&380A97AE&0&1
Intel(R) 82801CA/CAM SMBus Controller - 2483      No
          SYSTEM    5.2.3790.0          10/1/2002 Intel
          machine.inf         Not Available
          PCI\VEN_8086&DEV_2483&SUBSYS_34248086&REV_02\
3&267A616A&0&FB
ACPI Fixed Feature Button     No        SYSTEM
          5.2.3790.0          10/1/2002 (Standard
system devices)     machine.inf         Not Available
          ACPI\FIXEDBUTTON\2&DABA3FF&0
Logical Disk Manager          No        SYSTEM
          5.2.3790.0          10/1/2002 (Standard
system devices)     machine.inf         Not Available
          ROOT\DMIO\0000
Volume Manager      No        SYSTEM    5.2.3790.0
          10/1/2002 (Standard system devices)
          machine.inf         Not Available
          ROOT\FTDISK\0000
Generic volume      No        VOLUME    5.2.3790.0
          10/1/2002 Microsoft volume.inf          Not
Available
          STORAGE\VOLUME\1&30A96598&0&SIGNATURE9C45B0C3
OFFSET7E00LENGTH680728800
Generic volume      No        VOLUME    5.2.3790.0
          10/1/2002 Microsoft volume.inf          Not
Available
          STORAGE\VOLUME\1&30A96598&0&SIGNATURE9C45B0C3
OFFSET680730600LENGTHC80730000
Generic volume      No        VOLUME    5.2.3790.0
          10/1/2002 Microsoft volume.inf          Not
Available
          STORAGE\VOLUME\1&30A96598&0&SIGNATURE9C45B0C3
OFFSET13O0E60600LENGTHDC285C9E00
Generic volume      No        VOLUME    5.2.3790.0
          10/1/2002 Microsoft volume.inf          Not
Available
          STORAGE\VOLUME\1&30A96598&0&SIGNATURE2A3FF97D
OFFSET7E00LENGTH88A16D800
Generic volume      No        VOLUME    5.2.3790.0
          10/1/2002 Microsoft volume.inf          Not
Available
          STORAGE\VOLUME\1&30A96598&0&SIGNATURED8B3E94F
OFFSET7E00LENGTH1115293200
Generic volume      No        VOLUME    5.2.3790.0
          10/1/2002 Microsoft volume.inf          Not
Available
          STORAGE\VOLUME\1&30A96598&0&SIGNATURE9C45B0C0
OFFSET7E00LENGTH680728800
Generic volume      No        VOLUME    5.2.3790.0
          10/1/2002 Microsoft volume.inf          Not
Available
          STORAGE\VOLUME\1&30A96598&0&SIGNATURE9C45B0C0
OFFSET680730600LENGTHC80730000
Generic volume      No        VOLUME    5.2.3790.0
          10/1/2002 Microsoft volume.inf          Not
Available
          STORAGE\VOLUME\1&30A96598&0&SIGNATURE9C45B0C0
OFFSET13O0E60600LENGTHDC285C9E00
AFD Networking Support Environment      Not Available
          LEGACYDRIVER        Not Available       Not
Available Not Available       Not Available       Not
Available ROOT\LEGACY_AFD\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
dmboot    Not Available       LEGACYDRIVER        Not
Available Not Available       Not Available       Not
Available Not Available       ROOT\LEGACY_DMBOOT\0000

dmload    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 Not Available       Not Available       Not
Available ROOT\LEGACY_GPC\0000
IPSEC driver          Not Available       LEGACYDRIVER
          Not Available       Not Available       Not
Available Not Available       Not Available
          ROOT\LEGACY_IPSEC\0000
ksecdd    Not Available       LEGACYDRIVER        Not
Available Not Available       Not Available       Not
Available Not Available       ROOT\LEGACY_KSECDD\0000

mac2w2k   Not Available       LEGACYDRIVER        Not
Available Not Available       Not Available       Not
Available Not Available       ROOT\LEGACY_MAC2W2K\0000

mnmdd     Not Available       LEGACYDRIVER        Not
Available Not Available       Not Available       Not
Available Not Available       ROOT\LEGACY_MNMDD\0000

mountmgr  Not Available       LEGACYDRIVER        Not
Available Not Available       Not Available       Not
Available Not Available       ROOT\LEGACY_MOUNTMGR\0000

NDIS System Driver  Not Available       LEGACYDRIVER
          Not Available       Not Available       Not
Available Not Available       Not Available
          ROOT\LEGACY_NDIS\0000
Remote Access NDIS TAPI Driver          Not Available
          LEGACYDRIVER        Not Available       Not
Available Not Available       Not Available       Not
Available ROOT\LEGACY_NDISTAPI\0000
NDIS Usermode I/O Protocol    Not Available
          LEGACYDRIVER        Not Available       Not
Available Not Available       Not Available       Not
Available ROOT\LEGACY_NDISUIO\0000
NDProxy   Not Available       LEGACYDRIVER        Not
Available Not Available       Not Available       Not
Available Not Available       ROOT\LEGACY_NDPROXY\0000

NetBios over Tcpip  Not Available       LEGACYDRIVER
          Not Available       Not Available       Not
Available Not Available       Not Available
          ROOT\LEGACY_NETBT\0000
Null      Not Available       LEGACYDRIVER        Not
Available Not Available       Not Available       Not
Available Not Available       ROOT\LEGACY_NULL\0000

Partition Manager   Not Available       LEGACYDRIVER
          Not Available       Not Available       Not
Available Not Available       Not Available
          ROOT\LEGACY_PARTMGR\0000
Parvdm    Not Available       LEGACYDRIVER        Not
Available Not Available       Not Available       Not
Available Not Available       ROOT\LEGACY_PARVDM\0000

Remote Access Auto Connection Driver    Not Available
          LEGACYDRIVER        Not Available       Not
Available Not Available       Not Available       Not
Available ROOT\LEGACY_RASACD\0000
RDPCDD    Not Available       LEGACYDRIVER        Not
Available Not Available       Not Available       Not
Available Not Available       ROOT\LEGACY_RDPCDD\0000

TCP/IP Protocol Driver        Not Available
          LEGACYDRIVER        Not Available       Not
Available Not Available       Not Available       Not
Available ROOT\LEGACY_TCPIP\0000
VGA Display Controller.       Not Available
          LEGACYDRIVER        Not Available       Not
Available Not Available       Not Available       Not
Available ROOT\LEGACY_VGASAVE\0000
volsnap   Not Available       LEGACYDRIVER        Not
Available Not Available       Not Available       Not
Available Not Available       ROOT\LEGACY_VOLSNAP\0000

Remote Access IP ARP Driver   Not Available
          LEGACYDRIVER        Not Available       Not
Available Not Available       Not Available       Not
Available ROOT\LEGACY_WANARP\0000
Audio Codecs        No        MEDIA     5.2.3790.0
          10/1/2002 (Standard system devices)
          wave.inf  Not Available
          ROOT\MEDIA\MS_MMACM
Legacy Audio Drivers          No        MEDIA
          5.2.3790.0          10/1/2002 (Standard
system devices)     wave.inf  Not Available
          ROOT\MEDIA\MS_MMDRV
Media Control Devices         No        MEDIA
          5.2.3790.0          10/1/2002 (Standard
system devices)     wave.inf  Not Available
          ROOT\MEDIA\MS_MMMCI
Legacy Video Capture Devices  No        MEDIA
          5.2.3790.0          10/1/2002 (Standard
system devices)     wave.inf  Not Available
          ROOT\MEDIA\MS_MMVCD
Video Codecs        No        MEDIA     5.2.3790.0
          10/1/2002 (Standard system devices)
          wave.inf  Not Available
          ROOT\MEDIA\MS_MMVID
WAN Miniport (L2TP) No        NET       5.2.3790.0
          10/1/2002 Microsoft netrasa.inf         Not
Available ROOT\MS_L2TPMINIPORT\0000
WAN Miniport (IP)   No        NET       5.2.3790.0
          10/1/2002 Microsoft netrasa.inf         Not
Available ROOT\MS_NDISWANIP\0000
WAN Miniport (PPPOE)          No        NET
          5.2.3790.0          10/1/2002 Microsoft
          netrasa.inf         Not Available
          ROOT\MS_PPPOEMINIPORT\0000
WAN Miniport (PPTP) No        NET       5.2.3790.0
          10/1/2002 Microsoft netrasa.inf         Not
Available ROOT\MS_PPTPMINIPORT\0000
```

```
Direct Parallel     No        NET       5.2.3790.0
          10/1/2002 Microsoft netrasa.inf         Not
Available ROOT\MS_PTIMINIPORT\0000
Terminal Server Device Redirector       No
          SYSTEM    5.2.3790.0          10/1/2002
          (Standard system devices)     machine.inf
          Not Available       ROOT\RDPDR\0000
Terminal Server Keyboard Driver         No
          SYSTEM    5.2.3790.0          10/1/2002
          (Standard system devices)     machine.inf
          Not Available       ROOT\RDP_KBD\0000
Terminal Server Mouse Driver  No        SYSTEM
          5.2.3790.0          10/1/2002 (Standard
system devices)     machine.inf         Not Available
          ROOT\RDP_MOU\0000
Plug and Play Software Device Enumerator          No
          SYSTEM    5.2.3790.0          10/1/2002
          (Standard system devices)     machine.inf
          Not Available       ROOT\SYSTEM\0000
Microcode Update Device       No        SYSTEM
          5.2.3790.0          10/1/2002 (Standard
system devices)     machine.inf         Not Available
          ROOT\SYSTEM\0001

[Environment Variables]

Variable  Value     User Name
ComSpec   %SystemRoot%\system32\cmd.exe <SYSTEM>
Path
          %SystemRoot%\system32;%SystemRoot%;%SystemRoo
t%\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
Stepping 7, GenuineIntel      <SYSTEM>
PROCESSOR_REVISION  0207      <SYSTEM>
NUMBER_OF_PROCESSORS          2         <SYSTEM>
ClusterLog          C:\WINDOWS\Cluster\cluster.log
          <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       NT
AUTHORITY\SYSTEM
TMP       %USERPROFILE%\Local Settings\Temp       NT
AUTHORITY\SYSTEM
TEMP      %USERPROFILE%\Local Settings\Temp
          SQL02\Administrator
TMP       %USERPROFILE%\Local Settings\Temp
          SQL02\Administrator

[Print Jobs]

Document  Size      Owner     Notify    Status    Time
Submitted Start Time          Until Time
          Elapsed Time        Pages Printed       Job
ID        Priority  Parameters          Driver    Print
Processor Host Print Queue    Data Type Name

[Network Connections]

Local Name          Remote Name         Type
          Status    User Name

[Running Tasks]

Name      Path      Process ID          Priority  Min
Working Set         Max Working Set     Start Time
          Version   Size      File Date
system idle process Not Available       0         0
          Not Available       Not Available       Not
Available Not Available       Not Available       Not
Available
system    Not Available       4         8         0
          1413120   Not Available       Not Available
          Not Available       Not Available
smss.exe  Not Available       476       11
          204800    1413120   23/4/2004 15:50     Not
Available Not Available       Not Available
csrss.exe Not Available       536       13        Not
Available Not Available       23/4/2004 15:50     Not
Available Not Available       Not Available
winlogon.exe        c:\windows\system32\winlogon.exe
          560       13        204800    1413120
          23/4/2004 15:50     5.2.3790.0
(srv03_rtm.030324-2048)       536,50 KB (549.376 bytes)
          25/3/2003 09:00
services.exe        c:\windows\system32\services.exe
          604       9         204800    1413120
          23/4/2004 15:50     5.2.3790.0
(srv03_rtm.030324-2048)       102,00 KB (104.448 bytes)
          25/3/2003 09:00
lsass.exe c:\windows\system32\lsass.exe 616       9
          204800    1413120   23/4/2004 15:50
          5.2.3790.0 (srv03_rtm.030324-2048)      13,00
KB (13.312 bytes)   25/3/2003 09:00
svchost.exe         c:\windows\system32\svchost.exe
          768       8         204800    1413120
          23/4/2004 15:50     5.2.3790.0
(srv03_rtm.030324-2048)       13,00 KB (13.312 bytes)
          25/3/2003 09:00
svchost.exe         c:\windows\system32\svchost.exe
          820       8         204800    1413120
          23/4/2004 15:50     5.2.3790.0
(srv03_rtm.030324-2048)       13,00 KB (13.312 bytes)
          25/3/2003 09:00
svchost.exe         c:\windows\system32\svchost.exe
          960       8         204800    1413120
          23/4/2004 15:50     5.2.3790.0
(srv03_rtm.030324-2048)       13,00 KB (13.312 bytes)
          25/3/2003 09:00
explorer.exe        c:\windows\explorer.exe       1236
          8         204800    1413120   23/4/2004 16:07
          6.00.3790.0 (srv03_rtm.030324-2048)
          1.008,50 KB (1.032.704 bytes) 25/3/2003 09:00

sqlmangr.exe        c:\program files\microsoft sql
server\80\tools\binn\sqlmangr.exe       1316      8
          204800    1413120   23/4/2004 16:07
          2000.080.0760.00    72,57 KB (74.308 bytes)
          20/4/2004 14:23
wmiprvse.exe        Not Available       1428      8
          Not Available       Not Available
          23/4/2004 16:07     Not Available       Not
Available Not Available
wpabaln.exe         c:\windows\system32\wpabaln.exe
          284       8         204800    1413120
          26/4/2004 16:09     5.2.3790.0
(srv03_rtm.030324-2048)       31,00 KB (31.744 bytes)
          25/3/2003 09:00
cmd.exe   c:\windows\system32\cmd.exe   1792      8
          204800    1413120   27/4/2004 11:49
          5.2.3790.0 (srv03_rtm.030324-2048)
          374,00 KB (382.976 bytes)     25/3/2003 09:00

sqlservr.exe        c:\program files\microsoft sql
server\mssql\binn\sqlservr.exe          1816      13
          204800    1413120   27/4/2004 11:49
          2000.080.0760.00    7,17 MB (7.520.337 bytes)
          20/4/2004 14:21
isqlw.exe c:\program files\microsoft sql
server\80\tools\binn\isqlw.exe          1564      8
          204800    1413120   27/4/2004 11:49
          2000.080.0760.00    344,56 KB (352.828 bytes)
          20/4/2004 14:22
cmd.exe   c:\windows\system32\cmd.exe   1252      8
          204800    1413120   27/4/2004 12:37
          5.2.3790.0 (srv03_rtm.030324-2048)
          374,00 KB (382.976 bytes)     25/3/2003 09:00

helpctr.exe
          c:\windows\pchealth\helpctr\binaries\helpctr.
exe       1720      8         204800    1413120
          27/4/2004 12:50     5.2.3790.0
(srv03_rtm.030324-2048)       764,00 KB (782.336 bytes)
          20/4/2004 13:46
wmiprvse.exe        Not Available       312       8
          Not Available       Not Available
          27/4/2004 12:50     Not Available       Not
Available Not Available
helpsvc.exe
          c:\windows\pchealth\helpctr\binaries\helpsvc.
exe       1844      8         204800    1413120
          27/4/2004 12:50     5.2.3790.0
(srv03_rtm.030324-2048)       720,00 KB (737.280 bytes)
          20/4/2004 13:46

[Loaded Modules]

Name      Version   Size      File Date Manufacturer
          Path
winlogon  5.2.3790.0 (srv03_rtm.030324-2048)
          536,50 KB (549.376 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\winlogon.exe
ntdll     5.2.3790.0 (srv03_rtm.030324-2048)
          722,50 KB (739.840 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\ntdll.dll
kernel32  5.2.3790.0 (srv03_rtm.030324-2048)
          965,00 KB (988.160 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\kernel32.dll
msvcrt    7.0.3790.0 (srv03_rtm.030324-2048)
          319,50 KB (327.168 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\msvcrt.dll
advapi32  5.2.3790.0 (srv03_rtm.030324-2048)
          559,50 KB (572.928 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\advapi32.dll
rpcrt4    5.2.3790.0 (srv03_rtm.030324-2048)
          643,50 KB (658.944 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\rpcrt4.dll
user32    5.2.3790.0 (srv03_rtm.030324-2048)
          562,00 KB (575.488 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\user32.dll
gdi32     5.2.3790.0 (srv03_rtm.030324-2048)
          263,00 KB (269.312 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\gdi32.dll
userenv   5.2.3790.0 (srv03_rtm.030324-2048)
          732,50 KB (750.080 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\userenv.dll
nddeapi   5.2.3790.0 (srv03_rtm.030324-2048)      16,00
KB (16.384 bytes)   25/3/2003 09:00     Microsoft
```

```
Corporation          c:\windows\system32\nddeapi.dll

crypt32    5.131.3790.0 (srv03_rtm.030324-2048)
           598,00 KB (612.352 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\crypt32.dll
msasn1     5.2.3790.0 (srv03_rtm.030324-2048)      58,00
KB (59.392 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\msasn1.dll

secur32    5.2.3790.0 (srv03_rtm.030324-2048)      63,00
KB (64.512 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\secur32.dll

winsta     5.2.3790.0 (srv03_rtm.030324-2048)      51,00
KB (52.224 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\winsta.dll

netapi32   5.2.3790.0 (srv03_rtm.030324-2048)
           317,00 KB (324.608 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\netapi32.dll
profmap    5.2.3790.0 (srv03_rtm.030324-2048)      22,00
KB (22.528 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\profmap.dll

regapi     5.2.3790.0 (srv03_rtm.030324-2048)      48,50
KB (49.664 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\regapi.dll

ws2_32     5.2.3790.0 (srv03_rtm.030324-2048)      87,50
KB (89.600 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\ws2_32.dll

ws2help    5.2.3790.0 (srv03_rtm.030324-2048)      19,50
KB (19.968 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\ws2help.dll

psapi      5.2.3790.0 (srv03_rtm.030324-2048)      21,50
KB (22.016 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\psapi.dll
version    5.2.3790.0 (srv03_rtm.030324-2048)      17,00
KB (17.408 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\version.dll

setupapi   5.2.3790.0 (srv03_rtm.030324-2048)
           1.014,50 KB (1.038.848 bytes) 25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\setupapi.dll
msgina     5.2.3790.0 (srv03_rtm.030324-2048)      1,14
MB (1.191.936 bytes)          25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\msgina.dll
shsvcs     6.00.3790.0 (srv03_rtm.030324-2048)
           121,50 KB (124.416 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\shsvcs.dll
shlwapi    6.00.3790.0 (srv03_rtm.030324-2048)
           281,00 KB (287.744 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\shlwapi.dll
sfc        5.2.3790.0 (srv03_rtm.030324-2048)      4,50
KB (4.608 bytes)     25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\sfc.dll
sfc_os     5.2.3790.0 (srv03_rtm.030324-2048)
           133,00 KB (136.192 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\sfc_os.dll
wintrust   5.131.3790.0 (srv03_rtm.030324-2048)
           161,50 KB (165.376 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\wintrust.dll
ole32      5.2.3790.0 (srv03_rtm.030324-2048)      1,13
MB (1.187.328 bytes)          25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\ole32.dll
imagehlp   5.2.3790.0 (srv03_rtm.030324-2048)
           142,50 KB (145.920 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\imagehlp.dll
comctl32   6.0 (srv03_rtm.030324-2048)   907,00 KB
(928.768 bytes)      20/4/2004 10:19     Microsoft
Corporation
           c:\windows\winsxs\x86_microsoft.windows.commo
n-controls_6595b64144ccf1df_6.0.100.0_x-
ww_8417450b\comctl32.dll
winscard   5.2.3790.0 (srv03_rtm.030324-2048)      98,50
KB (100.864 bytes)   25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\winscard.dll

wtsapi32   5.2.3790.0 (srv03_rtm.030324-2048)      17,50
KB (17.920 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\wtsapi32.dll

sxs        5.2.3790.0 (srv03_rtm.030324-2048)
           733,00 KB (750.592 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\sxs.dll
winmm      5.2.3790.0 (srv03_rtm.030324-2048)
           166,00 KB (169.984 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\winmm.dll
rsaenh     5.2.3790.0 (srv03_rtm.030324-2048)
           176,83 KB (181.072 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\rsaenh.dll
wldap32    5.2.3790.0 (srv03_rtm.030324-2048)
           158,00 KB (161.792 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\wldap32.dll
cscdll     5.2.3790.0 (srv03_rtm.030324-2048)      99,00
KB (101.376 bytes)   25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\cscdll.dll

wlnotify   5.2.3790.0 (srv03_rtm.030324-2048)      87,50
KB (89.600 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\wlnotify.dll

winspool   5.2.3790.0 (srv03_rtm.030324-2048)
           140,00 KB (143.360 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\winspool.drv
mpr        5.2.3790.0 (srv03_rtm.030324-2048)      56,00
KB (57.344 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\mpr.dll
shell32    6.00.3790.0 (srv03_rtm.030324-2048)     7,79
MB (8.166.400 bytes)          25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\shell32.dll
comctl32   5.82 (srv03_rtm.030324-2048)   561,00 KB
(574.464 bytes)      20/4/2004 10:19     Microsoft
Corporation
           c:\windows\winsxs\x86_microsoft.windows.commo
n-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)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\uxtheme.dll
mprapi     5.2.3790.0 (srv03_rtm.030324-2048)      81,00
KB (82.944 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\mprapi.dll
activeds   5.2.3790.0 (srv03_rtm.030324-2048)
           189,00 KB (193.536 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\activeds.dll
adsldpc    5.2.3790.0 (srv03_rtm.030324-2048)
           142,50 KB (145.920 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\adsldpc.dll
credui     5.2.3790.0 (srv03_rtm.030324-2048)
           159,00 KB (162.816 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\credui.dll
atl        3.05.2283 83,00 KB (84.992 bytes)
           25/3/2003 09:00     Microsoft Corporation
           c:\windows\system32\atl.dll
oleaut32   5.2.3790.0          486,00 KB (497.664 bytes)
           25/3/2003 09:00     Microsoft Corporation
           c:\windows\system32\oleaut32.dll
rtutils    5.2.3790.0 (srv03_rtm.030324-2048)      32,00
KB (32.768 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\rtutils.dll

samlib     5.2.3790.0 (srv03_rtm.030324-2048)      49,00
KB (50.176 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\samlib.dll

cscui      5.2.3790.0 (srv03_rtm.030324-2048)
           305,00 KB (312.320 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\cscui.dll
clbcatq    2001.12.4720.0 (srv03_rtm.030324-2048)
           481,00 KB (492.544 bytes)     20/4/2004 13:43
           Microsoft Corporation
           c:\windows\system32\clbcatq.dll
comres     2001.12.4720.0 (srv03_rtm.030324-2048)
           778,00 KB (796.672 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\comres.dll
ntmarta    5.2.3790.0 (srv03_rtm.030324-2048)
           114,00 KB (116.736 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\ntmarta.dll
wbemprox   5.2.3790.0 (srv03_rtm.030324-2048)      17,50
KB (17.920 bytes)    20/4/2004 13:43     Microsoft
Corporation
           c:\windows\system32\wbem\wbemprox.dll
wbemcomn   5.2.3790.0 (srv03_rtm.030324-2048)
           211,50 KB (216.576 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\wbem\wbemcomn.dll
wbemsvc    5.2.3790.0 (srv03_rtm.030324-2048)      42,50
KB (43.520 bytes)    20/4/2004 13:43     Microsoft
Corporation
           c:\windows\system32\wbem\wbemsvc.dll
fastprox   5.2.3790.0 (srv03_rtm.030324-2048)
           443,00 KB (453.632 bytes)     20/4/2004 13:43
           Microsoft Corporation
           c:\windows\system32\wbem\fastprox.dll
msvcp60    6.05.2144.0          388,00 KB (397.312 bytes)
           25/3/2003 09:00     Microsoft Corporation
           c:\windows\system32\msvcp60.dll
ntdsapi    5.2.3790.0 (srv03_rtm.030324-2048)      76,00
KB (77.824 bytes)    25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\ntdsapi.dll

dnsapi     5.2.3790.0 (srv03_rtm.030324-2048)
           147,50 KB (151.040 bytes)     25/3/2003 09:00
           Microsoft Corporation
           c:\windows\system32\dnsapi.dll
services   5.2.3790.0 (srv03_rtm.030324-2048)
           102,00 KB (104.448 bytes)     25/3/2003 09:00
```

```
          Microsoft Corporation
          c:\windows\system32\services.exe
scesrv    5.2.3790.0 (srv03_rtm.030324-2048)
          316,50 KB (324.096 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\scesrv.dll
authz     5.2.3790.0 (srv03_rtm.030324-2048)      67,00
KB (68.608 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\authz.dll
umpnpmgr  5.2.3790.0 (srv03_rtm.030324-2048)
          121,50 KB (124.416 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\umpnpmgr.dll
ncobjapi  5.2.3790.0 (srv03_rtm.030324-2048)      34,50
KB (35.328 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\ncobjapi.dll

eventlog  5.2.3790.0 (srv03_rtm.030324-2048)      60,50
KB (61.952 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\eventlog.dll

lsass     5.2.3790.0 (srv03_rtm.030324-2048)      13,00
KB (13.312 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\lsass.exe
lsasrv    5.2.3790.0 (srv03_rtm.030324-2048)
          780,50 KB (799.232 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\lsasrv.dll
samsrv    5.2.3790.0 (srv03_rtm.030324-2048)
          452,00 KB (462.848 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\samsrv.dll
cryptdll  5.2.3790.0 (srv03_rtm.030324-2048)      34,00
KB (34.816 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\cryptdll.dll

msprivs   5.2.3790.0 (srv03_rtm.030324-2048)      46,50
KB (47.616 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\msprivs.dll

kerberos  5.2.3790.0 (srv03_rtm.030324-2048)
          332,50 KB (340.480 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\kerberos.dll
msv1_0    5.2.3790.0 (srv03_rtm.030324-2048)
          127,00 KB (130.048 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\msv1_0.dll
netlogon  5.2.3790.0 (srv03_rtm.030324-2048)
          409,00 KB (418.816 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\netlogon.dll
w32time   5.2.3790.0 (srv03_rtm.030324-2048)
          216,00 KB (221.184 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\w32time.dll
iphlpapi  5.2.3790.0 (srv03_rtm.030324-2048)      82,50
KB (84.480 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\iphlpapi.dll

schannel  5.2.3790.0 (srv03_rtm.030324-2048)
          149,50 KB (153.088 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\schannel.dll
wdigest   5.2.3790.0 (srv03_rtm.030324-2048)      61,00
KB (62.464 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\wdigest.dll

rassfm    5.2.3790.0 (srv03_rtm.030324-2048)      20,50
KB (20.992 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\rassfm.dll

kdcsvc    5.2.3790.0 (srv03_rtm.030324-2048)
          221,00 KB (226.304 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\kdcsvc.dll
ntdsa     5.2.3790.0 (srv03_rtm.030324-2048)      1,45
MB (1.520.640 bytes)          25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\ntdsa.dll
ntdsatq   5.2.3790.0 (srv03_rtm.030324-2048)      32,00
KB (32.768 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\ntdsatq.dll

mswsock   5.2.3790.0 (srv03_rtm.030324-2048)
          254,00 KB (260.096 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\mswsock.dll
esent     5.2.3790.0 (srv03_rtm.030324-2048)      1,01
MB (1.056.256 bytes)          25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\esent.dll
scecli    5.2.3790.0 (srv03_rtm.030324-2048)
          179,50 KB (183.808 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\scecli.dll
wshtcpip  5.2.3790.0 (srv03_rtm.030324-2048)      18,00
KB (18.432 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\wshtcpip.dll

dssenh    5.2.3790.0 (srv03_rtm.030324-2048)
          131,33 KB (134.480 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\dssenh.dll
svchost   5.2.3790.0 (srv03_rtm.030324-2048)      13,00
KB (13.312 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\svchost.exe

rpcss     5.2.3790.0 (srv03_rtm.030324-2048)
          276,50 KB (283.136 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\rpcss.dll
termsrv   5.2.3790.0 (srv03_rtm.030324-2048)
          216,50 KB (221.696 bytes)     20/4/2004 13:43
          Microsoft Corporation
          c:\windows\system32\termsrv.dll
icaapi    5.2.3790.0 (srv03_rtm.030324-2048)      10,50
KB (10.752 bytes)   20/4/2004 13:43     Microsoft
Corporation         c:\windows\system32\icaapi.dll

mstlsapi  5.2.3790.0 (srv03_rtm.030324-2048)
          104,50 KB (107.008 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\mstlsapi.dll
dmserver  5.2.3790.0 (srv03_rtm.030324-2048)      24,00
KB (24.576 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\dmserver.dll

es        2001.12.4720.0 (srv03_rtm.030324-2048)
          221,50 KB (226.816 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\es.dll
sens      5.2.3790.0 (srv03_rtm.030324-2048)      35,50
KB (36.352 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\sens.dll
netman    5.2.3790.0 (srv03_rtm.030324-2048)
          209,00 KB (214.016 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\netman.dll
rasapi32  5.2.3790.0 (srv03_rtm.030324-2048)
          227,50 KB (232.960 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\rasapi32.dll

rasman    5.2.3790.0 (srv03_rtm.030324-2048)      56,50
KB (57.856 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\rasman.dll

tapi32    5.2.3790.0 (srv03_rtm.030324-2048)
          175,00 KB (179.200 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\tapi32.dll
wzcsvc    5.2.3790.0 (srv03_rtm.030324-2048)
          272,50 KB (279.040 bytes)     25/3/2003 09:15
          Microsoft Corporation
          c:\windows\system32\wzcsvc.dll
wmi       5.2.3790.0 (srv03_rtm.030324-2048)      6,50
KB (6.656 bytes)    25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\wmi.dll
dhcpcsvc  5.2.3790.0 (srv03_rtm.030324-2048)
          101,50 KB (103.936 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\dhcpcsvc.dll
wzcsapi   5.2.3790.0 (srv03_rtm.030324-2048)      24,50
KB (25.088 bytes)   25/3/2003 09:15     Microsoft
Corporation         c:\windows\system32\wzcsapi.dll

netshell  5.2.3790.0 (srv03_rtm.030324-2048)      1,67
MB (1.747.456 bytes)          25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\netshell.dll
clusapi   5.2.3790.0 (srv03_rtm.030324-2048)      56,00
KB (57.344 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\clusapi.dll

hnetcfg   5.2.3790.0 (srv03_rtm.030324-2048)
          243,50 KB (249.344 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\hnetcfg.dll
wininet   6.00.3790.0 (srv03_rtm.030324-2048)
          609,00 KB (623.616 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\wininet.dll
wmisvc    5.2.3790.0 (srv03_rtm.030324-2048)
          131,00 KB (134.144 bytes)     20/4/2004 13:43
          Microsoft Corporation
          c:\windows\system32\wbem\wmisvc.dll
vssapi    5.2.3790.0 (srv03_rtm.030324-2048)
          528,00 KB (540.672 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\vssapi.dll
wbemcore  5.2.3790.0 (srv03_rtm.030324-2048)
          457,00 KB (467.968 bytes)     20/4/2004 13:43
          Microsoft Corporation
          c:\windows\system32\wbem\wbemcore.dll
esscli    5.2.3790.0 (srv03_rtm.030324-2048)
          235,50 KB (241.152 bytes)     20/4/2004 13:43
          Microsoft Corporation
          c:\windows\system32\wbem\esscli.dll
comsvcs   2001.12.4720.0 (srv03_rtm.030324-2048)  1,14
MB (1.199.616 bytes)          20/4/2004 13:43
          Microsoft Corporation
          c:\windows\system32\comsvcs.dll
wmiutils  5.2.3790.0 (srv03_rtm.030324-2048)      90,50
KB (92.672 bytes)   20/4/2004 13:43     Microsoft
Corporation
          c:\windows\system32\wbem\wmiutils.dll
repdrvfs  5.2.3790.0 (srv03_rtm.030324-2048)
          165,00 KB (168.960 bytes)     20/4/2004 13:43
          Microsoft Corporation
          c:\windows\system32\wbem\repdrvfs.dll
wmiprvsd  5.2.3790.0 (srv03_rtm.030324-2048)
          405,50 KB (415.232 bytes)     20/4/2004 13:43
          Microsoft Corporation
          c:\windows\system32\wbem\wmiprvsd.dll
```

```
wbemess   5.2.3790.0 (srv03_rtm.030324-2048)
          256,50 KB (262.656 bytes)     20/4/2004 13:43
          Microsoft Corporation
          c:\windows\system32\wbem\wbemess.dll
ncprov    5.2.3790.0 (srv03_rtm.030324-2048)      43,00
KB (44.032 bytes)   20/4/2004 13:43     Microsoft
Corporation         c:\windows\system32\wbem\ncprov.dll

rasdlg    5.2.3790.0 (srv03_rtm.030324-2048)
          642,00 KB (657.408 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\rasdlg.dll
rasadhlp  5.2.3790.0 (srv03_rtm.030324-2048)      6,50
KB (6.656 bytes)    25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\rasadhlp.dll

pchsvc    5.2.3790.0 (srv03_rtm.030324-2048)      31,50
KB (32.256 bytes)   20/4/2004 13:46     Microsoft
Corporation
          c:\windows\pchealth\helpctr\binaries\pchsvc.d
ll
wbemcons  5.2.3790.0 (srv03_rtm.030324-2048)      69,00
KB (70.656 bytes)   20/4/2004 13:43     Microsoft
Corporation
          c:\windows\system32\wbem\wbemcons.dll
explorer  6.00.3790.0 (srv03_rtm.030324-2048)
          1.008,50 KB (1.032.704 bytes) 25/3/2003 09:00
          Microsoft Corporation
          c:\windows\explorer.exe
browseui  6.00.3790.0 (srv03_rtm.030324-2048)     1,01
MB (1.057.280 bytes)          25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\browseui.dll
shdocvw   6.00.3790.0 (srv03_rtm.030324-2048)     1,33
MB (1.393.664 bytes)          25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\shdocvw.dll
apphelp   5.2.3790.0 (srv03_rtm.030324-2048)
          122,00 KB (124.928 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\apphelp.dll
themeui   6.00.3790.0 (srv03_rtm.030324-2048)
          360,50 KB (369.152 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\themeui.dll
msimg32   5.2.3790.0 (srv03_rtm.030324-2048)      4,50
KB (4.608 bytes)    25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\msimg32.dll

linkinfo  5.2.3790.0 (srv03_rtm.030324-2048)      16,50
KB (16.896 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\linkinfo.dll

ntshrui   6.00.3790.0 (srv03_rtm.030324-2048)
          136,00 KB (139.264 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\ntshrui.dll
urlmon    6.00.3790.0 (srv03_rtm.030324-2048)
          501,50 KB (513.536 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\urlmon.dll
webcheck  6.00.3790.0 (srv03_rtm.030324-2048)
          261,50 KB (267.776 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\webcheck.dll
wsock32   5.2.3790.0 (srv03_rtm.030324-2048)      22,00
KB (22.528 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\wsock32.dll

stobject  5.2.3790.0 (srv03_rtm.030324-2048)
          117,50 KB (120.320 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\stobject.dll
batmeter  6.00.3790.0 (srv03_rtm.030324-2048)     28,50
KB (29.184 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\batmeter.dll

powrprof  6.00.3790.0 (srv03_rtm.030324-2048)     14,50
KB (14.848 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\powrprof.dll

printui   5.2.3790.0 (srv03_rtm.030324-2048)
          536,50 KB (549.376 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\printui.dll
cfgmgr32  5.2.3790.0 (srv03_rtm.030324-2048)      17,50
KB (17.920 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\cfgmgr32.dll

drprov    5.2.3790.0 (srv03_rtm.030324-2048)      12,50
KB (12.800 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\drprov.dll

ntlanman  5.2.3790.0 (srv03_rtm.030324-2048)      41,00
KB (41.984 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\ntlanman.dll

netui0    5.2.3790.0 (srv03_rtm.030324-2048)      75,50
KB (77.312 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\netui0.dll

netui1    5.2.3790.0 (srv03_rtm.030324-2048)
          184,00 KB (188.416 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\netui1.dll
davclnt   5.2.3790.0 (srv03_rtm.030324-2048)      23,50
KB (24.064 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\davclnt.dll

browselc  6.00.3790.0 (srv03_rtm.030324-2048)     62,00
KB (63.488 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\browselc.dll

shdoclc   6.00.3790.0 (srv03_rtm.030324-2048)
          588,50 KB (602.624 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\shdoclc.dll
sensapi   5.2.3790.0 (srv03_rtm.030324-2048)      6,00
KB (6.144 bytes)    25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\sensapi.dll

jscript   5.6.0.8515          436,00 KB (446.464 bytes)
          25/3/2003 09:00     Microsoft Corporation
          c:\windows\system32\jscript.dll
winrnr    5.2.3790.0 (srv03_rtm.030324-2048)      15,00
KB (15.360 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\winrnr.dll

mydocs    6.00.3790.0 (srv03_rtm.030324-2048)     88,00
KB (90.112 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\mydocs.dll

zipfldr   6.00.3790.0 (srv03_rtm.030324-2048)
          316,00 KB (323.584 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\zipfldr.dll
actxprxy  6.00.3790.0 (srv03_rtm.030324-2048)     95,00
KB (97.280 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\actxprxy.dll

sendmail  6.00.3790.0 (srv03_rtm.030324-2048)     52,00
KB (53.248 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\sendmail.dll

sqlmangr  2000.080.0760.00    72,57 KB (74.308 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\sqlmangr.exe
sqlunirl  2000.080.0728.00    176,56 KB (180.800 bytes)
          25/3/2003 09:00     Microsoft Corporation
          c:\windows\system32\sqlunirl.dll
comdlg32  6.00.3790.0 (srv03_rtm.030324-2048)
          261,00 KB (267.264 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\comdlg32.dll
w95scm    2000.080.0760.00    48,56 KB (49.728 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\w95scm.dll
odbc32    3.525.1022.0 (srv03_rtm.030324-2048)
          232,00 KB (237.568 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\odbc32.dll
sqlsvc    2000.080.0760.00    92,56 KB (94.784 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\sqlsvc.dll
odbcbcp   2000.085.1022.00 (srv03_rtm.030324-2048)
          24,00 KB (24.576 bytes)       25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\odbcbcp.dll
sqlresld  2000.080.0382.00    28,56 KB (29.248 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\sqlresld.dll
odbcint   3.525.1022.0 (srv03_rtm.030324-2048)    92,00
KB (94.208 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\odbcint.dll

resutils  5.2.3790.0 (srv03_rtm.030324-2048)      59,00
KB (60.416 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\resutils.dll

mfc42u    6.05.3014.0         960,00 KB (983.040 bytes)
          25/3/2003 09:00     Microsoft Corporation
          c:\windows\system32\mfc42u.dll
sqlsvc    2000.080.0194.00    24,00 KB (24.576 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\resources\1033\sqlsvc.rll
sqlmangr  2000.080.0194.00    96,00 KB (98.304 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\resources\1033\sqlmangr.rll
wpabaln   5.2.3790.0 (srv03_rtm.030324-2048)      31,00
KB (31.744 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\wpabaln.exe

cmd       5.2.3790.0 (srv03_rtm.030324-2048)
          374,00 KB (382.976 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\cmd.exe
sqlservr  2000.080.0760.00    7,17 MB (7.520.337 bytes)
          20/4/2004 14:21     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\sqlservr.exe
opends60  2000.080.0194.00    24,06 KB (24.639 bytes)
          20/4/2004 14:21     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\opends60.dll
ums       2000.080.0760.00    52,55 KB (53.808 bytes)
          20/4/2004 14:21     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\ums.dll
```

```
sqlsort   2000.080.0760.00    576,56 KB (590.396 bytes)
          20/4/2004 14:21     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\sqlsort.dll
msvcirt   7.0.3790.0 (srv03_rtm.030324-2048)      50,00
KB (51.200 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\msvcirt.dll

sqlevn70  2000.080.0760.00    28,00 KB (28.672 bytes)
          20/4/2004 14:21     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\resources\1033\sqlevn70.rll
xolehlp   2001.12.4720.0 (srv03_rtm.030324-2048)  8,50
KB (8.704 bytes)    20/4/2004 13:43     Microsoft
Corporation         c:\windows\system32\xolehlp.dll

msdtcprx  2001.12.4720.0 (srv03_rtm.030324-2048)
          427,50 KB (437.760 bytes)     20/4/2004 13:43
          Microsoft Corporation
          c:\windows\system32\msdtcprx.dll
mtxclu    2001.12.4720.0 (srv03_rtm.030324-2048)  74,50
KB (76.288 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\mtxclu.dll

ssnetlib  2000.080.0760.00    80,56 KB (82.492 bytes)
          20/4/2004 14:21     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\ssnetlib.dll
ssnmpn70  2000.080.0534.00    24,56 KB (25.148 bytes)
          20/4/2004 14:21     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\ssnmpn70.dll
security  5.2.3790.0 (srv03_rtm.030324-2048)      5,50
KB (5.632 bytes)    25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\security.dll

ssmslpcn  2000.080.0760.00    28,56 KB (29.244 bytes)
          20/4/2004 14:21     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\ssmslpcn.dll
sqloledb  2000.085.1022.00 (srv03_rtm.030324-2048)
          536,00 KB (548.864 bytes)     20/4/2004 13:46
          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)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\msdart.dll
msdatl3   2.80.1022.0 (srv03_rtm.030324-2048)     96,00
KB (98.304 bytes)   20/4/2004 13:46     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)     20/4/2004 13:46
          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)   20/4/2004 13:46     Microsoft
Corporation         c:\program files\common
files\system\ole db\oledb32r.dll
xpstar    2000.080.0760.00    280,56 KB (287.296 bytes)
          20/4/2004 14:21     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\xpstar.dll
sqlresld  2000.080.0382.00    28,56 KB (29.248 bytes)
          20/4/2004 14:22     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\sqlresld.dll
sqlsvc    2000.080.0760.00    92,56 KB (94.784 bytes)
          20/4/2004 14:22     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\sqlsvc.dll

w95scm    2000.080.0760.00    48,56 KB (49.728 bytes)
          20/4/2004 14:22     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\w95scm.dll
shfolder  6.00.3790.0 (srv03_rtm.030324-2048)     23,00
KB (23.552 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\shfolder.dll

sqlsvc    2000.080.0194.00    24,00 KB (24.576 bytes)
          20/4/2004 14:22     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\resources\1033\sqlsvc.rll
xpstar    2000.080.0760.00    36,00 KB (36.864 bytes)
          20/4/2004 14:21     Microsoft Corporation
          c:\program files\microsoft sql
server\mssql\binn\resources\1033\xpstar.rll
isqlw     2000.080.0760.00    344,56 KB (352.828 bytes)
          20/4/2004 14:22     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\isqlw.exe
sqlgui    2000.080.0760.00    444,56 KB (455.232 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\sqlgui.dll
imm32     5.2.3790.0 (srv03_rtm.030324-2048)
          105,50 KB (108.032 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\imm32.dll
sqlqry    2000.080.0760.00    392,56 KB (401.984 bytes)
          20/4/2004 14:22     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\sqlqry.dll
pfutil80  2000.080.0760.00    272,56 KB (279.104 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\pfutil80.dll
semsfc    2000.080.0760.00    228,56 KB (234.048 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\semsfc.dll
pfclnt80  2000.080.0760.00    404,56 KB (414.272 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\pfclnt80.dll
sqlgui    2000.080.0194.00    56,00 KB (57.344 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\resources\1033\sqlgui.rll
semsfc    2000.080.0194.00    24,00 KB (24.576 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\resources\1033\semsfc.rll
pfclnt80  2000.080.0194.00    28,00 KB (28.672 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\resources\1033\pfclnt80.rll
pfutil80  2000.080.0382.00    144,00 KB (147.456 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\resources\1033\pfutil80.rll
sqlqry    2000.080.0194.00    180,00 KB (184.320 bytes)
          20/4/2004 14:22     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\resources\1033\sqlqry.rll
isqlw     2000.080.0382.00    240,00 KB (245.760 bytes)
          20/4/2004 14:22     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\resources\1033\isqlw.rll
odbccp32  3.525.1022.0 (srv03_rtm.030324-2048)
          100,00 KB (102.400 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\odbccp32.dll

sqllex    2000.080.0194.00    148,06 KB (151.616 bytes)
          20/4/2004 14:23     Microsoft Corporation
          c:\program files\microsoft sql
server\80\tools\binn\sqllex.dll
sqlsrv32  2000.085.1022.00 (srv03_rtm.030324-2048)
          404,00 KB (413.696 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\sqlsrv32.dll
sqlsrv32  2000.085.1022.00 (srv03_rtm.030324-2048)
          88,00 KB (90.112 bytes)       25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\sqlsrv32.rll
dbnetlib  2000.085.1022 (srv03_rtm.030324-2048)   76,00
KB (77.824 bytes)   25/3/2003 09:00     Microsoft
Corporation         c:\windows\system32\dbnetlib.dll

dbmslpcn  2000.080.0760.00    28,56 KB (29.244 bytes)
          20/4/2004 14:21     Microsoft Corporation
          c:\windows\system32\dbmslpcn.dll
helpctr   5.2.3790.0 (srv03_rtm.030324-2048)
          764,00 KB (782.336 bytes)     20/4/2004 13:46
          Microsoft Corporation
          c:\windows\pchealth\helpctr\binaries\helpctr.
exe
hcappres  5.2.3790.0 (srv03_rtm.030324-2048)      6,50
KB (6.656 bytes)    20/4/2004 13:46     Microsoft
Corporation
          c:\windows\pchealth\helpctr\binaries\hcappres
.dll
itss      5.2.3790.0 (srv03_rtm.030324-2048)
          119,50 KB (122.368 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\itss.dll
msxml3    8.40.9419.0           1,28 MB (1.337.344 bytes)
          25/3/2003 09:00     Microsoft Corporation
          c:\windows\system32\msxml3.dll
pchshell  5.2.3790.0 (srv03_rtm.030324-2048)
          100,50 KB (102.912 bytes)     20/4/2004 13:46
          Microsoft Corporation
          c:\windows\pchealth\helpctr\binaries\pchshell
.dll
mlang     6.00.3790.0 (srv03_rtm.030324-2048)
          570,00 KB (583.680 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\mlang.dll
mshtml    6.00.3790.0 (srv03_rtm.030324-2048)     2,78
MB (2.916.352 bytes)          25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\mshtml.dll
msimtf    5.2.3790.0 (srv03_rtm.030324-2048)
          149,00 KB (152.576 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\msimtf.dll
msctf     5.2.3790.0 (srv03_rtm.030324-2048)
          287,00 KB (293.888 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\msctf.dll
msls31    3.10.349.0            147,00 KB (150.528 bytes)
          25/3/2003 09:00     Microsoft Corporation
          c:\windows\system32\msls31.dll
mshtmled  6.00.3790.0 (srv03_rtm.030324-2048)
          443,50 KB (454.144 bytes)     25/3/2003 09:00
          Microsoft Corporation
          c:\windows\system32\mshtmled.dll
vbscript  5.6.0.8515            404,00 KB (413.696 bytes)
          25/3/2003 09:00     Microsoft Corporation
          c:\windows\system32\vbscript.dll
mfc42     6.05.3014.0           960,00 KB (983.040 bytes)
          25/3/2003 09:00     Microsoft Corporation
          c:\windows\system32\mfc42.dll
msinfo    5.2.3790.0 (srv03_rtm.030324-2048)
          358,50 KB (367.104 bytes)     20/4/2004 13:46
```

```
          Microsoft Corporation
          c:\windows\pchealth\helpctr\binaries\msinfo.d
ll
riched32  5.2.3790.0 (srv03_rtm.030324-2048)      3,50
KB (3.584 bytes)     25/3/2003 09:00     Microsoft
Corporation          c:\windows\system32\riched32.dll

riched20  5.31.23.1218        406,00 KB (415.744 bytes)
          25/3/2003 09:00     Microsoft Corporation
          c:\windows\system32\riched20.dll
helpsvc   5.2.3790.0 (srv03_rtm.030324-2048)
          720,00 KB (737.280 bytes)     20/4/2004 13:46
          Microsoft Corporation
          c:\windows\pchealth\helpctr\binaries\helpsvc.
exe

[Services]

Display Name        Name      State     Start Mode
          Service Type        Path      Error Control
          Start Name          Tag ID
Alerter   Alerter   Stopped   Disabled  Share Process
          c:\windows\system32\svchost.exe -k
localservice        Normal    NT AUTHORITY\LocalService
          0
Application Layer Gateway Service       ALG
          Stopped   Manual    Own Process
          c:\windows\system32\alg.exe   Normal    NT
AUTHORITY\LocalService        0
Application Management        AppMgmt   Stopped
          Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Windows Audio       AudioSrv  Stopped   Disabled  Share
Process   c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Background Intelligent Transfer Service BITS
          Stopped   Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Computer Browser    Browser   Stopped   Manual    Share
Process   c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Indexing Service    CiSvc     Stopped   Disabled  Share
Process   c:\windows\system32\cisvc.exe Normal
          LocalSystem         0
ClipBook  ClipSrv   Stopped   Disabled  Own Process
          c:\windows\system32\clipsrv.exe
          Normal    LocalSystem         0
COM+ System Application       COMSysApp Stopped
          Manual    Own Process
          c:\windows\system32\dllhost.exe
/processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}
          Normal    LocalSystem         0
Cryptographic Services        CryptSvc  Stopped
          Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Distributed File System       Dfs       Stopped
          Manual    Own Process
          c:\windows\system32\dfssvc.exe
          Normal    LocalSystem         0
DHCP Client         Dhcp      Stopped   Manual    Share
Process   c:\windows\system32\svchost.exe -k
networkservice      Normal    NT
AUTHORITY\NetworkService      0
Logical Disk Manager Administrative Service
          dmadmin   Stopped   Manual    Share Process
          c:\windows\system32\dmadmin.exe /com
          Normal    LocalSystem         0
Logical Disk Manager          dmserver  Running   Auto
          Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
DNS Client          Dnscache  Stopped   Manual    Share
Process   c:\windows\system32\svchost.exe -k
networkservice      Normal    NT
AUTHORITY\NetworkService      0
Error Reporting Service       ERSvc     Stopped
          Manual    Share Process
          c:\windows\system32\svchost.exe -k winerr
          Ignore    LocalSystem         0
Event Log Eventlog  Running   Auto      Share Process
          c:\windows\system32\services.exe
          Normal    LocalSystem         0
COM+ Event System   EventSystem         Running
          Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Help and Support    helpsvc   Running   Manual    Share
Process   c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Human Interface Device Access HidServ   Stopped
          Disabled  Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
HTTP SSL  HTTPFilter          Stopped   Manual    Share
Process   c:\windows\system32\lsass.exe Normal
          LocalSystem         0
IMAPI CD-Burning COM Service  ImapiService
          Stopped   Disabled  Own Process
          c:\windows\system32\imapi.exe Normal
          LocalSystem         0
Intersite Messaging IsmServ   Stopped   Disabled  Own
Process   c:\windows\system32\ismserv.exe
          Normal    LocalSystem         0
Kerberos Key Distribution Center        kdc
          Stopped   Disabled  Share Process
          c:\windows\system32\lsass.exe Normal
          LocalSystem         0
Server    lanmanserver        Stopped   Manual    Share
Process   c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Workstation         lanmanworkstation   Stopped
          Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
License Logging     LicenseService      Stopped
          Disabled  Own Process
          c:\windows\system32\llssrv.exe
          Normal    NT AUTHORITY\NetworkService   0

TCP/IP NetBIOS Helper         LmHosts   Stopped
          Manual    Share Process
          c:\windows\system32\svchost.exe -k
localservice        Normal    NT AUTHORITY\LocalService
          0
Messenger Messenger Stopped   Disabled  Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
NetMeeting Remote Desktop Sharing       mnmsrvc
          Stopped   Disabled  Own Process
          c:\windows\system32\mnmsrvc.exe
          Normal    LocalSystem         0
Distributed Transaction Coordinator     MSDTC
          Stopped   Manual    Own Process
          c:\windows\system32\msdtc.exe Normal    NT
AUTHORITY\NetworkService      0
Windows Installer   MSIServer Stopped   Manual    Share
Process   c:\windows\system32\msiexec.exe /v
          Normal    LocalSystem         0
MSSQLSERVER         MSSQLSERVER         Stopped
          Manual    Own Process
          c:\progra~1\micros~1\mssql\binn\sqlservr.exe
          Normal    LocalSystem         0
MSSQLServerADHelper MSSQLServerADHelper Stopped
          Manual    Own Process         c:\program
files\microsoft sql server\80\tools\binn\sqladhlp.exe
          Normal    LocalSystem         0
Network DDE         NetDDE    Stopped   Disabled  Share
Process   c:\windows\system32\netdde.exe
          Normal    LocalSystem         0
Network DDE DSDM    NetDDEdsdm          Stopped
          Disabled  Share Process
          c:\windows\system32\netdde.exe
          Normal    LocalSystem         0
Net Logon Netlogon  Stopped   Manual    Share Process
          c:\windows\system32\lsass.exe Normal
          LocalSystem         0
Network Connections Netman    Running   Manual    Share
Process   c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Network Location Awareness (NLA)        Nla
          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         Stopped
          Manual    Share Process
          c:\windows\system32\lsass.exe Normal
          LocalSystem         0
Protected Storage   ProtectedStorage    Stopped
          Manual    Share Process
          c:\windows\system32\lsass.exe Normal
          LocalSystem         0
Remote Access Auto Connection Manager   RasAuto
          Stopped   Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Remote Access Connection Manager        RasMan
          Stopped   Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Remote Desktop Help Session Manager     RDSessMgr
          Stopped   Manual    Own Process
          c:\windows\system32\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      Stopped
          Manual    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 -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  Stopped   Manual    Share
Process   c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Secondary Logon     seclogon  Stopped   Manual    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
Connection Sharing (ICS)      SharedAccess
          Stopped   Disabled  Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Shell Hardware Detection      ShellHWDetection
          Stopped   Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Ignore    LocalSystem         0
Print Spooler       Spooler   Stopped   Manual    Own
Process   c:\windows\system32\spoolsv.exe
          Normal    LocalSystem         0
SQLSERVERAGENT      SQLSERVERAGENT      Stopped
          Manual    Own Process
          c:\progra~1\micros~1\mssql\binn\sqlagent.exe
          Normal    LocalSystem         0
Windows Image Acquisition (WIA)         stisvc
          Stopped   Disabled  Share Process
          c:\windows\system32\svchost.exe -k imgsvc
          Normal    NT AUTHORITY\LocalService     0

Microsoft Software Shadow Copy Provider swprv
          Stopped   Manual    Own Process
          c:\windows\system32\svchost.exe -k swprv
          Normal    LocalSystem         0
Performance Logs and Alerts   SysmonLog Stopped
          Manual    Own Process
          c:\windows\system32\smlogsvc.exe
          Normal    NT Authority\NetworkService   0

Telephony TapiSrv   Stopped   Manual    Share Process
          c:\windows\system32\svchost.exe -k tapisrv
          Normal    LocalSystem         0
Terminal Services   TermService         Running
          Manual    Share Process
          c:\windows\system32\svchost.exe -k termsvcs
          Normal    LocalSystem         0
Themes    Themes    Stopped   Disabled  Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Telnet    TlntSvr   Stopped   Disabled  Own Process
          c:\windows\system32\tlntsvr.exe
          Normal    NT AUTHORITY\LocalService     0

Distributed Link Tracking Server        TrkSvr
          Stopped   Disabled  Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Distributed Link Tracking Client        TrkWks
          Stopped   Manual    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    Own
Process   c:\windows\system32\vssvc.exe Normal
          LocalSystem         0
Windows Time        W32Time   Stopped   Manual    Share
Process   c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
WebClient WebClient Stopped   Disabled  Share Process
          c:\windows\system32\svchost.exe -k
localservice        Normal    NT AUTHORITY\LocalService
          0
WinHTTP Web Proxy Auto-Discovery Service
          WinHttpAutoProxySvc Stopped   Manual    Share
Process   c:\windows\system32\svchost.exe -k
localservice        Normal    NT AUTHORITY\LocalService
          0
Windows Management Instrumentation      winmgmt
          Running   Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Ignore    LocalSystem         0
Portable Media Serial Number Service    WmdmPmSN
          Stopped   Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Windows Management Instrumentation Driver Extensions
          Wmi       Stopped   Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
WMI Performance Adapter       WmiApSrv  Stopped
          Manual    Own Process
          c:\windows\system32\wbem\wmiapsrv.exe
          Normal    LocalSystem         0
Automatic Updates   wuauserv  Stopped   Manual    Share
Process   c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0
Wireless Configuration        WZCSVC    Stopped
          Manual    Share Process
          c:\windows\system32\svchost.exe -k netsvcs
          Normal    LocalSystem         0

[Program Groups]

Group Name          Name      User Name
Accessories         Default User:Accessories
          Default User
Accessories\Accessibility     Default
User:Accessories\Accessibility          Default User
Accessories\Entertainment     Default
User:Accessories\Entertainment          Default User

Startup   Default User:Startup          Default User

Accessories         All Users:Accessories         All
Users
Accessories\Accessibility     All
Users:Accessories\Accessibility         All Users
Accessories\Communications    All
Users:Accessories\Communications        All Users
Accessories\Entertainment     All
Users:Accessories\Entertainment         All Users
Accessories\System Tools      All
Users:Accessories\System Tools          All Users
Administrative Tools          All Users:Administrative
Tools     All Users
Microsoft SQL Server          All Users:Microsoft SQL
Server    All Users
Microsoft SQL Server - Switch All Users:Microsoft SQL
Server - Switch     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         SQL02\Administrator:Accessories
          SQL02\Administrator
Accessories\Accessibility
          SQL02\Administrator:Accessories\Accessibility
          SQL02\Administrator
Accessories\Entertainment
          SQL02\Administrator:Accessories\Entertainment
          SQL02\Administrator
Administrative Tools
          SQL02\Administrator:Administrative Tools
          SQL02\Administrator
Startup   SQL02\Administrator:Startup
          SQL02\Administrator

[Startup Programs]

Program   Command   User Name Location
desktop   desktop.ini         NT AUTHORITY\SYSTEM
          Startup
desktop   desktop.ini         SQL02\Administrator
          Startup
desktop   desktop.ini         .DEFAULT  Startup
desktop   desktop.ini         All Users Common Startup

Service Manager
          c:\progra~1\micros~1\80\tools\binn\sqlmangr.e
xe  /n    All Users Common Startup

[OLE Registration]

Object    Local Server
Sound (OLE2)        sndrec32.exe
Media Clip          mplay32.exe
Video Clip          mplay32.exe /avi
MIDI Sequence       mplay32.exe /mid
Sound     Not Available
Media Clip          Not Available
WordPad Document    "%programfiles%\windows
nt\accessories\wordpad.exe"
```

```
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
          Company
actxprxy.dll        6.0.3790.0          95 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
advpack.dll         6.0.3790.0          94 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
asctrls.ocx         6.0.3790.0          90 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
browselc.dll        6.0.3790.0          62 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
browseui.dll        6.0.3790.0          1.033 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
cdfview.dll         6.0.3790.0          144 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
comctl32.dll        5.82.3790.0         561 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
dxtrans.dll         6.3.3790.0          198 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
dxtmsft.dll         6.3.3790.0          344 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
iecont.dll          <File Missing>      Not Available
          Not Available       Not Available       Not
Available
iecontlc.dll        <File Missing>      Not Available
          Not Available       Not Available       Not
Available
iedkcs32.dll        16.0.3790.0         300 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
iepeers.dll         6.0.3790.0          230 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
iesetup.dll         6.0.3790.0          59 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
ieuinit.inf         Not Available       20 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32 Not
Available
iexplore.exe        6.0.3790.0          90 KB
          25/3/2003 09:00:00  C:\Program Files\Internet
Explorer  Microsoft Corporation
imgutil.dll         5.2.3790.0          35 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
inetcpl.cpl         6.0.3790.0          303 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
inetcplc.dll        6.0.3790.0          109 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
inseng.dll          6.0.3790.0          72 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
mlang.dll 6.0.3790.0          570 KB    25/3/2003
09:00:00  C:\WINDOWS\system32 Microsoft Corporation

msencode.dll        2002.10.4.0         112 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32 Not
Available
mshta.exe 6.0.3790.0          26 KB     25/3/2003
09:00:00  C:\WINDOWS\system32 Microsoft Corporation

mshtml.dll          6.0.3790.0          2.848 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
mshtml.tlb          6.0.3790.0          1.319 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
mshtmled.dll        6.0.3790.0          444 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
mshtmler.dll        6.0.3790.0          55 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
msident.dll         6.0.3790.0          47 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
msidntld.dll        6.0.3790.0          15 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
msieftp.dll         6.0.3790.0          230 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
msrating.dll        6.0.3790.0          132 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
mstime.dll          6.0.3790.0          491 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
occache.dll         6.0.3790.0          89 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
proctexe.ocx        6.3.3790.0          78 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32 Intel
Corporation
sendmail.dll        6.0.3790.0          52 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
shdoclc.dll         6.0.3790.0          589 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
shdocvw.dll         6.0.3790.0          1.361 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
shfolder.dll        6.0.3790.0          23 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
shlwapi.dll         6.0.3790.0          281 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
tdc.ocx   1.3.0.3130          58 KB     25/3/2003
09:00:00  C:\WINDOWS\system32 Microsoft Corporation

url.dll   6.0.3790.0          36 KB     25/3/2003
09:00:00  C:\WINDOWS\system32 Microsoft Corporation

urlmon.dll          6.0.3790.0          502 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
webcheck.dll        6.0.3790.0          262 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          Microsoft Corporation
wininet.dll         6.0.3790.0          609 KB
          25/3/2003 09:00:00  C:\WINDOWS\system32
          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\LocalService\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       Medium
Internet  High
Restricted sites    High
```

IIS Registry Parameters

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\I
netInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\I
netInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"="LDAPSVC"
"PoolThreadLimit"=dword:000007fe
"ThreadTimeout"=dword:00015180

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\I
netInfo\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:80,e4,12,ab,a3,23,c4,01,
10,25,00,00,00,00,00,00
```

WWW Service Registry Parameters

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,
4e,00,54,00,5c,00,53,00,\

79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6
e,00,65,00,74,00,73,\

00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,00,69,00,6e,0
0,66,00,6f,00,2e,00,\
  65,00,78,00,65,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):49,00,49,00,53,00,41,00,44,00,
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."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC\ASP]
"NOTE"="This is for backward compatibility only."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC\ASP\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\\WINNT\\System32\\inetsrv"
"CertMapList"="C:\\WINNT\\System32\\inetsrv\\iiscrmap.d
ll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\\WINNT\\System32\\LogFiles"
"AcceptExOutstanding"=dword:00000028

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC\Parameters\ADCLaunch]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC\Parameters\ADCLaunch\AdvancedDataFactory]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC\Parameters\ADCLaunch\RDSServer.DataFactory]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC\Parameters\Script Map]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC\Parameters\Virtual Roots]
"/"="c:\\inetpub\\wwwroot,,205"
"/Scripts"="c:\\inetpub\\scripts,,1"
"/IISHelp"="c:\\winnt\\help\\iishelp,,1"
"/IISAdmin"="C:\\WINNT\\System32\\inetsrv\\iisadmin,,1"
"/IISSamples"="c:\\inetpub\\iissamples,,1"
"/MSADC"="c:\\program files\\common
files\\system\\msadc,,1"
"/Printers"="C:\\WINNT\\web\\printers,,201"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC\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:aa,f0,22,7c,0d,67,c3,01,10,3d,00,00,00,00,00,
00
"WbemAdapFileTime"=hex:00,a3,f0,97,ab,d4,c0,01
"WbemAdapFileSize"=dword:00001d10
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,0
0,00,00,30,00,00,00,02,\

00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,0
0,00,00,00,01,00,00,\

00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,0
1,01,00,00,00,00,00,\

05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,0
2,00,00,00,00,00,05,\

20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,0
2,00,01,01,00,00,00,\

00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,0
0,01,02,00,00,00,00,\

00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,0
0,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,05,12,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W
3SVC\Enum]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

COM+ Settings

```
TPCC.AllTxns:
          Activation:
                    Enable Object Pooling selected
                    Minimun Pool Size: 270
                    Maximun Pool Size: 270
                    Creation timeout: 60.000
                    Enable Object Construction
                    Enable Just in Time Activation
          Concurrency:
                    Concorrency required
```

TPCC Application Registry Parameters

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"Path"="C:\\Inetpub\\wwwroot\\"
"NumberOfDeliveryThreads"=dword:00000032
"MaxConnections"=dword:0000445c
"MaxPendingDeliveries"=dword:00000834
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="SQL02"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"
```

Client System Configuration

```
System Information report written at: 04/27/2004
02:24:46 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         CLI04
System Manufacturer Intel
System Model        SBT2
System Type         X86-based PC
Processor x86 Family 6 Model 8 Stepping 6 GenuineIntel
~1000 Mhz
Processor x86 Family 6 Model 8 Stepping 6 GenuineIntel
~1000 Mhz
BIOS Version        PhoenixBIOS 4.0 Release 6.0
Windows Directory   C:\WINNT
System Directory    C:\WINNT\System32
Boot Device         \Device\Harddisk0\Partition1
Locale    United States
User Name CLI04\Administrator
Time Zone Central Daylight Time
Total Physical Memory         1,048,044 KB
Available Physical Memory     806,624 KB
Total Virtual Memory          3,570,060 KB
Available Virtual Memory      3,179,164 KB
Page File Space     2,522,016 KB
Page File C:\pagefile.sys
```

Benchcraft Profile

```
Profile:  1700w
File Path:          C:\Documents and
Settings\Administrator\Desktop\1700w.pro
Version:  4

          Number of Engines: 2

                    Name: DRIVER1
                    Description:
                    Directory: c:\logfiles\rte03-0.log
                    Machine: rte03
                    Parameter Set: PARAM2
                    Index: 0
                    Seed: 25987
                    Configured Users: 8500
                    Pipe Name: DRIVER1557781
                    Connect Rate: 360
                    Start Rate: 360
                    Max. Concurrency: -1
                    Concurrency Rate: 10
                    CLIENT_NURAND: 233
                    CPU: 0
                    Additional Options:

                    Name: DRIVER2
                    Description:
                    Directory: c:\logfiles\rte03-1.log
                    Machine: rte03
                    Parameter Set: PARAM2
                    Index: 100000000
                    Seed: 25987
                    Configured Users: 8500
                    Pipe Name: DRIVER2579890
                    Connect Rate: 360
                    Start Rate: 360
                    Max. Concurrency: -1
                    Concurrency Rate: 10
                    CLIENT_NURAND: 233
                    CPU: 1
                    Additional Options:

          Number of User groups: 2

                    Driver Engine: DRIVER1
                    IIS Server: cli01
                    SQL Server: sql02
                    Database: tpcc
                    User: sa
                    Protocol: HTML
                    w_id Range: 1 - 850
                    w_id Min Warehouse: 1
                    w_id Max Warehouse: 1700
                    Scale: Normal
                    User Count: 8500
                    District id: 1
                    Scale Down: No

                    Driver Engine: DRIVER2
                    IIS Server: cli01
                    SQL Server: sql02
                    Database: tpcc
                    User: sa
                    Protocol: HTML
                    w_id Range: 851 - 1700
                    w_id Min Warehouse: 1
                    w_id Max Warehouse: 1700
                    Scale: Normal
                    User Count: 8500
                    District id: 1
                    Scale Down: No

          Number of Parameter Sets: 2


                    ~Default
                    Default Parameter Set
                                        Txn     Think   Key
RT      RT      Menu
                                        Weight  Time
Time    Delay   Fence   Delay
                    New Order           10.00
12.05       18.01       0.10        5.00        0.10
                    Payment             10.00
12.05        3.01       0.10        5.00        0.10
                    Delivery             1.00
5.05         2.01       0.10        5.00        0.10
                    Stock Level          1.00
5.05         2.01       0.10       20.00        0.10
                    Order Status         1.00
10.05        2.01       0.10        5.00        0.10

                    PARAM2

                                        Txn     Think   Key
RT      RT      Menu
                                        Weight  Time
Time    Delay   Fence   Delay
                    New Order           44.92
12.07       18.01       0.10        5.00        0.10
                    Payment             43.02
12.07        3.01       0.10        5.00        0.10
                    Delivery             4.02
5.07         2.01       0.10        5.00        0.10
                    Stock Level          4.02
5.07         2.01       0.10       20.00        0.10
                    Order Status         4.02
10.07        2.01       0.10        5.00        0.10
```

Appendix D – 60-Day Space

TPC-C 60 Day Space Requirements

Warehouses	1700				TpmC	21.197,00
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	1.700	**184**	**24**	10		218
District	17.000	**1.896**	**24**	96		2016
Customer	51.000.000	**37.090.912**	**2.381.776**	1.973.634		41446322
History	51.000.000	**2.833.344**	**16**		570.980	2833360
NewOrder	15.300.000	**241.904**	**664**	12.128		254696
Orders	51.000.000	**1.563.224**	**863.440**		2.756.132	2426664
OrderLine	510.007.366	**31.875.464**	**79.384**		7.044.565	31954848
Item	100.000	**9.528**	**48**	479		10055
Stock	170.000.000	**54.400.000**	**121.840**	2.726.092		57247932
Total		128.016.456	3.447.216	4.712.440	10.371.678	136.176.112

	Misc_fg	CS_fg
	218	
	2016	
	0	41446322
	3404340	
	254696	
	5182796	
	31954848	
	10055	
	0	57247932
	40.808.970	98.694.254
files=	**2**	**2**
size=	**3.328.000**	**6.400.000**
Total=	6.656.000	12.800.000
8K blocks	53.248.000	102.400.000
	OK	OK

	MB	
Dynamic Space	35.422	Sum of Data for Order, Orderline and History
Static Space	97.563	Sum of Data+Index+5%-Dynamic Space
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)
Daily Growth	7.067	(Dynamic Space/(W*62.5))*tpmc
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed

60 Day Space MB	521.565					
60 Day Space GB	509,34	GB				
Log Size	64.999,99	MB				
KB Per New Order	5,18	KB				
8 hr log MB	51.467	MB				
8 hr log GB	50,2611	GB				

Space Usage	GB Needed	Disks Measured	GB Priced	Disk Size	Formatted Size	
60 Day Space DB	509,34	56	1.959	36GB	34,986	33,92
			0,00	9GB	8,473	
			0,00	4GB	3,999	
Total DB	509,34	56,00	1959,22	OK		
8-hr log + mirror	100,5221	4	139,94	36GB	34,986	
OS, Swap	3	1	8,473	9GB		
Total Storage	1.122,20	GB	2.107,63	GB		

tpmC	**21.197,00**									
	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	**2.833.344**	**16**	3096592	56	263.248	40	263.288	0,0561	570.980,26	557,60
Order	**1.563.224**	**863.440**	1967744	1729816	404.520	866.376	1.270.896	0,2709	2.756.132,19	2.691,54
Order-Line	**31.875.464**	**79.384**	35044320	158888	3.168.856	79.504	3.248.360	0,6924	7.044.565,07	6.879,46
										10.128,59

	sum(*) Before	**sum(*) After**	**Num New-**
d_next_o_id	**51.017.000**	55708650	**4.691.650**

	Before MB	**After MB**	**Grow MB**	**KB/New-Order**	**8-Hr Growth MB**	**8-Hr Growth GB**
Log	**658,76**	24391,16	**23732,40**	**5,1798**	**51.467,34**	**50,26**
				5.304,1523	**bytes**	

64.999,99	**1,0134774**	**37,524868**
Database tpcc log used (%)		