
Itautec Philco S.A.

TPC Benchmark™ C
Full Disclosure Report
for
Servidor Itautec 5450
using
Microsoft SQL Server 2000
Enterprise Edition SP3
and
Microsoft Windows Server 2003
Enterprise Edition

Second Edition
Submitted January 5, 2005

Second Edition – January 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, November 3, 2003, 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 this 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.

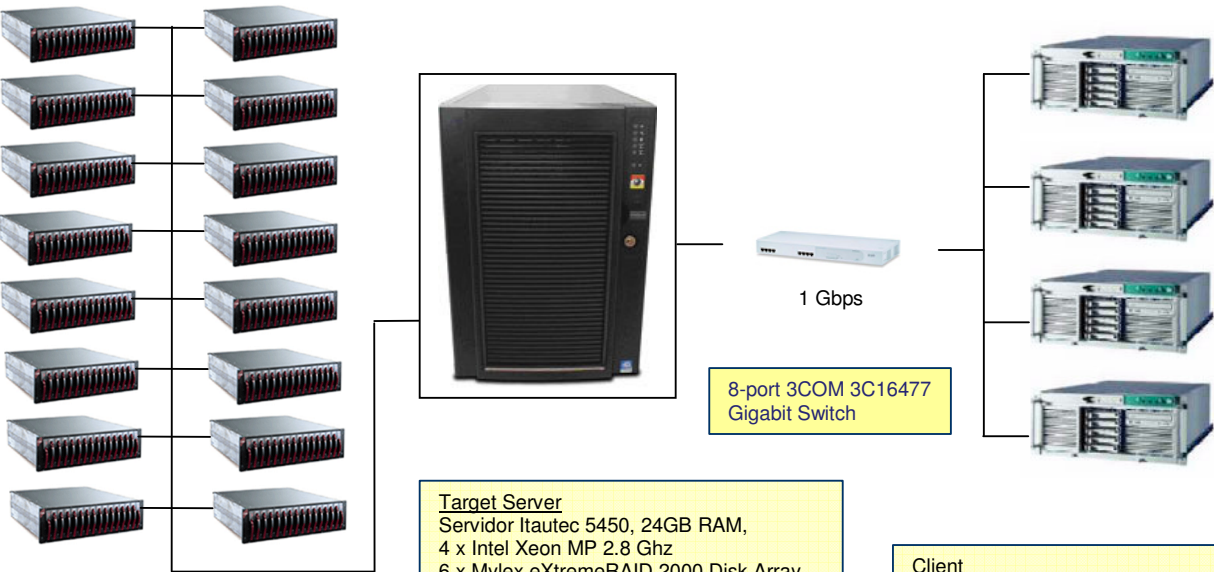
Microsoft, Windows 2003 and SQL Server 2000 are either trademarks or registered trademarks of Microsoft Corporation.

Intel and Xeon are registered trademarks of Intel Corporation.

TPC Benchmark, TPC-C and tpmC are registered trademarks of the Transaction Processing Performance Council.

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

		Servidor Itautec 5450 4P C/S with Servidor Itautec 3040HU		TPC-C Rev. 5.1	
				Report Date: 11/03/2003	
Total System Cost		TPC-C Throughput		Price/Performance	
R\$ 1,554,274.02		58,161.52		R\$ 26.72	
Processors		Database Manager		Operating System	
4 Intel Xeon MP Processor 2.8 GHz – Server 2 Intel Pentium III Xeon Processor 1.0 GHz - Client		Microsoft SQL Server 2000 Enterprise Edition SP3		Windows Server 2003 Enterprise Edition Microsoft Visual C++ Microsoft COM+	
Other Software		Number of Users		47,000	



Storage
16 x Itautec Storage S320
209 36.4GB 15 Krpm drives

Target Server
Servidor Itautec 5450, 24GB RAM,
4 x Intel Xeon MP 2.8 Ghz
6 x Mylex eXtremeRAID 2000 Disk Array
Controllers
1 36.4GB 15 Krpm drive in internal bay

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

System Components	Server		Each Client	
	Quantity	Description	Quantity	Description
Processor	4	2.8 GHz Intel Xeon MP w/ 2 MB cache / 400 MHz frontside BUS	2	1.0 GHz Pentium III Xeon w/ 256 KB cache / 133 MHz frontside BUS
Memory	12	2GB Register DDR ECC 266MHz	2	512 MB Register SDRAM ECC DIMM
Disk Controllers	6	Mylex eXtremeRAID 2000	1	Adaptec AIC-7899 Ultra2
Disk Drives	1	36.4 GB SCSI Drive 15 Krpm	1	18.2 GB 10Krpm SCSI Drive
	209	36.4 GB SCSI Drive 15 Krpm		
Total Storage	7644 GB		18.2 GB	
Tape Drives	1	20/40 GB DAT		

Numerical Quantities Summary

MQTh, Computed Maximum Qualified Throughput

58,161.52 tpmC

Response Times (in seconds)

	Average	90 th	Max
New Order	0.67	1.10	12.12
Payment	0.59	1.00	14.12
Order Status	0.62	1.04	7.79
Delivery (interactive)	0.10	0.11	1.09
Delivery (deferred)	0.26	0.46	1.47
Stock Level	1.05	1.90	8.17
Menu	0.10	0.11	9.51

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%
Order-Status	4.02%
Delivery	4.01%
Stock-Level	4.00%

Keying/Think Times (in seconds)

	Min		Average		Max	
New Order	18.00	0.00	18.02	12.07	18.03	120.71
Payment	3.00	0.00	3.02	12.06	3.03	120.71
Order Status	2.00	0.00	2.02	10.04	2.03	100.70
Delivery	2.00	0.00	2.02	5.07	2.03	50.71
Stock Level	2.00	0.00	2.02	5.08	2.03	50.71

Test Duration

Ramp-up time	58 minutes
Measurement interval	120 minutes
Number of checkpoints	4
Checkpoint interval	30 minutes
Number of transactions (all types) completed in measurement interval	16,149,100

	Itaotec Philco	Servidor Itaotec 5450	TPC-C Rev. 5.1				
			Report Date: 11/03/2003				
Description	Part Number	Third Party Brand	Unit Price Pricing	Quantity	Extended Price	3 yr. Maint. Price	
Server Hardware							
Servidor Itaotec 5450							
Base System with 1 x Xeon MP 2.8Ghz/2M	F5094	Itaotec	1	34.869	1	34.869 -	
Intel Xeon MP 2.8Ghz/2M BTO Option	G0463	Itaotec	1	22.177	3	66.532 -	
2 GB DDR266 PC1600 ECC Reg memory	F7880	Itaotec	1	12.764	12	153.168 -	
36 GB 15K HDD	F4637	Itaotec	1	1.849	1	1.849 -	
On-Board Intel PRO1000XT LAN	Included	Itaotec	1	-	1	- -	
On-Board Intel PRO100+ LAN	Included	Itaotec	1	-	1	- -	
CD-ROM, Internal SCSI Adapter	Included	Itaotec	1	-	1	- -	
Upgrade to 3-year / 4-hour response / 7days - 24hrs	1994/2003_03	Itaotec	1	27.002	1	- 27.002	
Monitor 15" SYNC MASTER 551S	E9801	Itaotec	1	450	1	450 -	
Subtotal					256.869	27.002	
Disk Subsystem							
Extreme RAID 2000 4 Channel Controller (+ 2 spare)	A4805	Itaotec	1	7.759	8	62.072 -	
Itaotec Storage Ultrab SC2100CTR-AC (+ 2 spare)	F1390	Itaotec	1	21.185	18	381.331 -	
36GB Ultra320 15K rpm (+10% spare)	F6853	Itaotec	1	2.527	230	581.320 -	
EXT SCSI Cable Eurou VHDCI68P (+2 spare)	E8158	Itaotec	1	722	12	8.666 -	
EXT SCSI Cable HD68/VHD68 (+ 1 spare)	86610	Itaotec	1	239	7	1.673 -	
Kit Rack for Storage	F1781	Itaotec	1	750	16	11.999 -	
Rack Itaotec F2400	F2400	Itaotec	1	4.195	3	12.584 -	
Subtotal					1.059.644	-	
Server Software							
Microsoft Windows 2003 Enterprise Edition	P72-00264	Microsoft	1	8.611	1	8.611 -	
SQL Server 2000 Ent. Edition (per processor)	810-00848	Microsoft	1	70.557	4	282.228 -	
Subtotal					290.839	-	
Client Hardware							
Servidor Itaotec 3040HU							
Base System Wwith 1 x Pentium III 1Ghz/256KB	B4G6Q	Itaotec	1	6.509	4	26.036 -	
1 x Pentium III 1Ghz/512KB BTO Option	A9347	Itaotec	1	1.369	4	5.476 -	
2 x 512MB Memory	F0251	Itaotec	1	1.300	4	5.200 -	
1 x 18GB 10K rpm	F0555	Itaotec	1	620	4	2.480 -	
Intel Pro/1000 T Server Adapter	F3608	Itaotec	1	1.849	8	14.792 -	
CD-ROM, On-Board LAN	Included	Itaotec	1	-	4	- -	
Upgrade to 3-year / 4-hour response / 7days - 24hrs	1994/2003_03	Itaotec	1	2.166	4	- 8.664	
Monitor 15" SYNC MASTER 551S	E9801	Itaotec	1	450	1	450 -	
Subtotal					54.434	8.664	
Client Software							
Microsoft Windows 2000 Server	C11-00821	Microsoft	1	2.985	4	11.940 -	
Microsoft Visual C++ Professional 6.0	254-00257	Microsoft	1	393	1	393 -	
Subtotal					12.333	-	
Network components							
SWITCH 8P BASELINE 10/100/1000	3C16477-ME	Itaotec	1	2.148	1	2.148 -	
Subtotal					2.148	-	
TOTAL					1.676.267	35.666	
Large volume purchase with cash in advance Discount on Itaotec Hardware (11.5%)					157.659		
TOTAL					1.518.608	35.666	
Pricing: 1 - Itaotec					3-Yr. Cost of Ownership: R\$ 1.554.274,02		
					tpmC Rating: 58.161,52		
					R\$ / tpmC: 26,72		
Audited By Lorna Livingtree							

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

NUMERICAL QUANTITIES SUMMARY	4
TABLE OF CONTENTS	6
ABSTRACT	8
OVERVIEW	8
AUDITOR	8
PREFACE	9
INTRODUCTION	9
GENERAL ITEMS	10
BENCHMARK SPONSOR	10
APPLICATION CODE AND DEFINITION STATEMENTS	10
PARAMETER SETTINGS	10
CONFIGURATION DIAGRAMS	10
MEASURED CONFIGURATION	11
PRICED CONFIGURATION	12
CLAUSE 1 - LOGICAL DATABASE DESIGN RELATED ITEMS	13
TABLE DEFINITIONS	13
PHYSICAL ORGANIZATION OF THE DATABASE	13
INSERT AND DELETE OPERATIONS	13
HORIZONTAL OR VERTICAL PARTITIONING	13
REPLICATION	13
TABLE ATTRIBUTES	13
CLAUSE 2 - TRANSACTION AND TERMINAL PROFILES RELATED ITEMS	14
RANDOM NUMBER GENERATION	14
SCREEN LAYOUT	14
TERMINAL VERIFICATION	14
INTELLIGENT TERMINALS	14
TRANSACTION PROFILES	14
TRANSACTION MIX	15
DEFERRED DELIVERY MECHANISM	15
CLAUSE 3 - TRANSACTION AND SYSTEM PROPERTIES RELATED ITEMS	16
ATOMICITY	16
COMPLETED TRANSACTION	16
ABORTED TRANSACTION	16
CONSISTENCY	16
ISOLATION	16
DURABILITY	16
LOSS OF DATA / LOSS OF LOG	16
INSTANTANEOUS INTERRUPTION AND LOSS OF MEMORY	17
CLAUSE 4: SCALING AND DATABASE POPULATION RELATED ITEMS	18
CARDINALITY OF THE TABLES	18
DISTRIBUTION OF DATABASE TABLES AND LOGS	18

DATABASE MODEL	18
MAPPING PARTITIONS/REPLICATION	19
60-DAY SPACE	19
CLAUSE 5: PERFORMANCE METRICS AND RESPONSE TIME RELATED ITEMS	20
MEASURED TPMC	20
RESPONSE TIMES	20
KEYING AND THINK TIMES	20
RESPONSE TIME DISTRIBUTION CURVES	21
NEW ORDER RESPONSE TIME VS. THROUGHPUT PERFORMANCE	23
NEW ORDER THINK TIME DISTRIBUTION	24
NEW ORDER THROUGHPUT VS. ELAPSED TIME	24
STEADY STATE METHODOLOGY	25
WORK PERFORMED DURING STEADY STATE	25
MEASUREMENT PERIOD DURATION AND CHECKPOINT DURATION	25
REGULATION OF TRANSACTION MIX	25
TRANSACTION STATISTICS	25
CHECKPOINT COUNT AND LOCATION	25
CLAUSE 6: SUT, DRIVER, AND COMMUNICATION DEFINITION RELATED ITEMS	27
RTE PARAMETERS	27
LOST TERMINAL CONNECTIONS	27
EMULATED COMPONENTS	27
BENCHMARKED AND TARGETED SYSTEM CONFIGURATION DIAGRAMS	27
NETWORK CONFIGURATION	27
NETWORK BANDWIDTH	27
OPERATOR INTERVENTION	27
CLAUSE 7 - PRICING RELATED ITEMS	28
HARDWARE AND SOFTWARE LIST	28
AVAILABILITY DATE	28
MEASURED TPMC	28
COUNTRY SPECIFIC PRICING	28
USAGE PRICING	28
SYSTEM PRICING	29
CLAUSE 9 - AUDIT RELATED ITEMS	30
AUDITOR	30
AVAILABILITY OF THE FULL DISCLOSURE REPORT	30
APPENDIX A – SOURCE CODE	33
APPENDIX B: DATABASE DESIGN	91
APPENDIX C: TUNABLE PARAMETERS	125
APPENDIX D – 60-DAY SPACE	165

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted on the Servidor Itautec 5450. The tests were conducted by Centro de Informática / UFPE in Brazil. The tests were run in a client/server configuration using three Servidor Itautec 3040HU as clients. The operating system used for the benchmark was Microsoft Windows Server 2003 Enterprise 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 5.1 of the Transaction Processing Council's TPC Benchmark C Standard Specification. Two standard TPC Benchmark TM 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 5450	Microsoft SQL Server 2000 Enterprise Edition Microsoft Windows Server 2003 Enterprise Edition	R\$ 1,554,274.02	58,161.52	26.72	November 3, 2003

Auditor

The benchmark configuration, methodology, and results were audited by Lorna Livingtree 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 5.1.

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

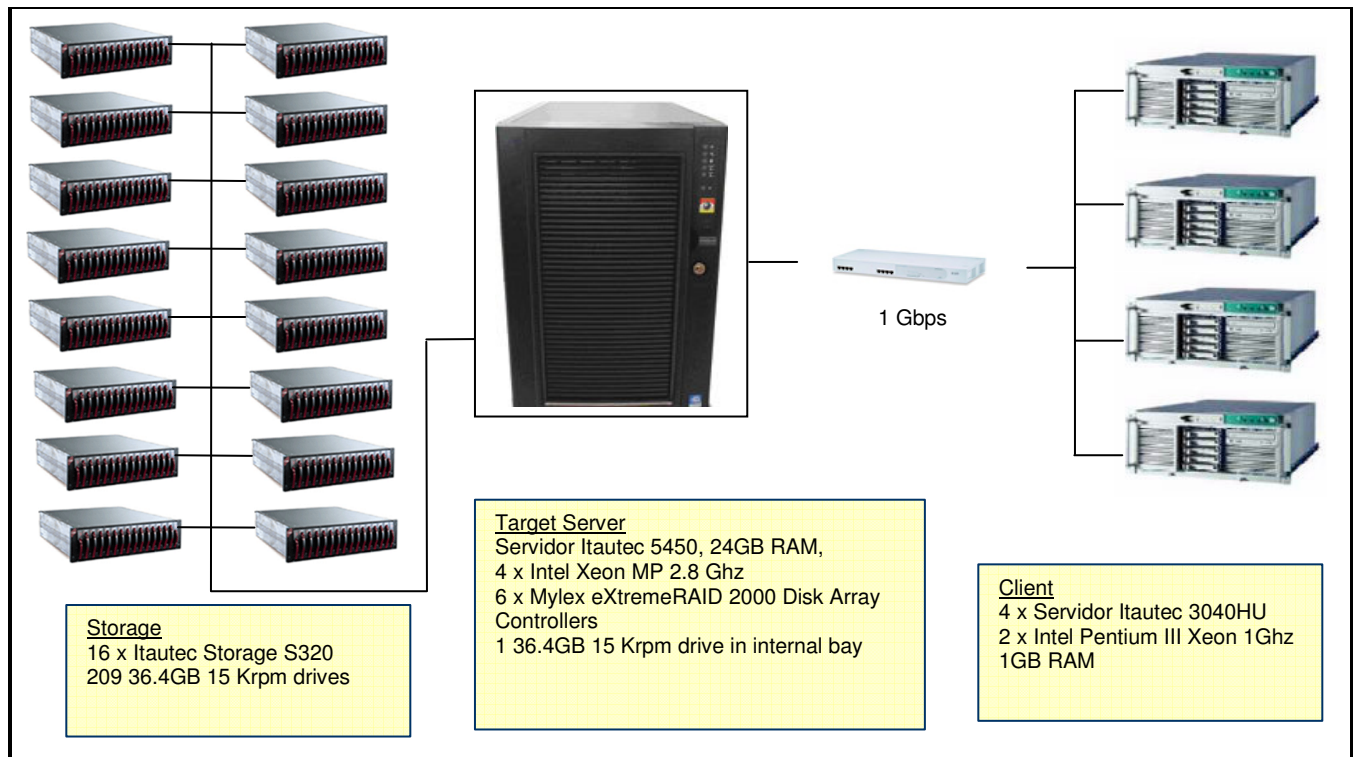


Figure 0.1 – Measured Configuration

Priced Configuration

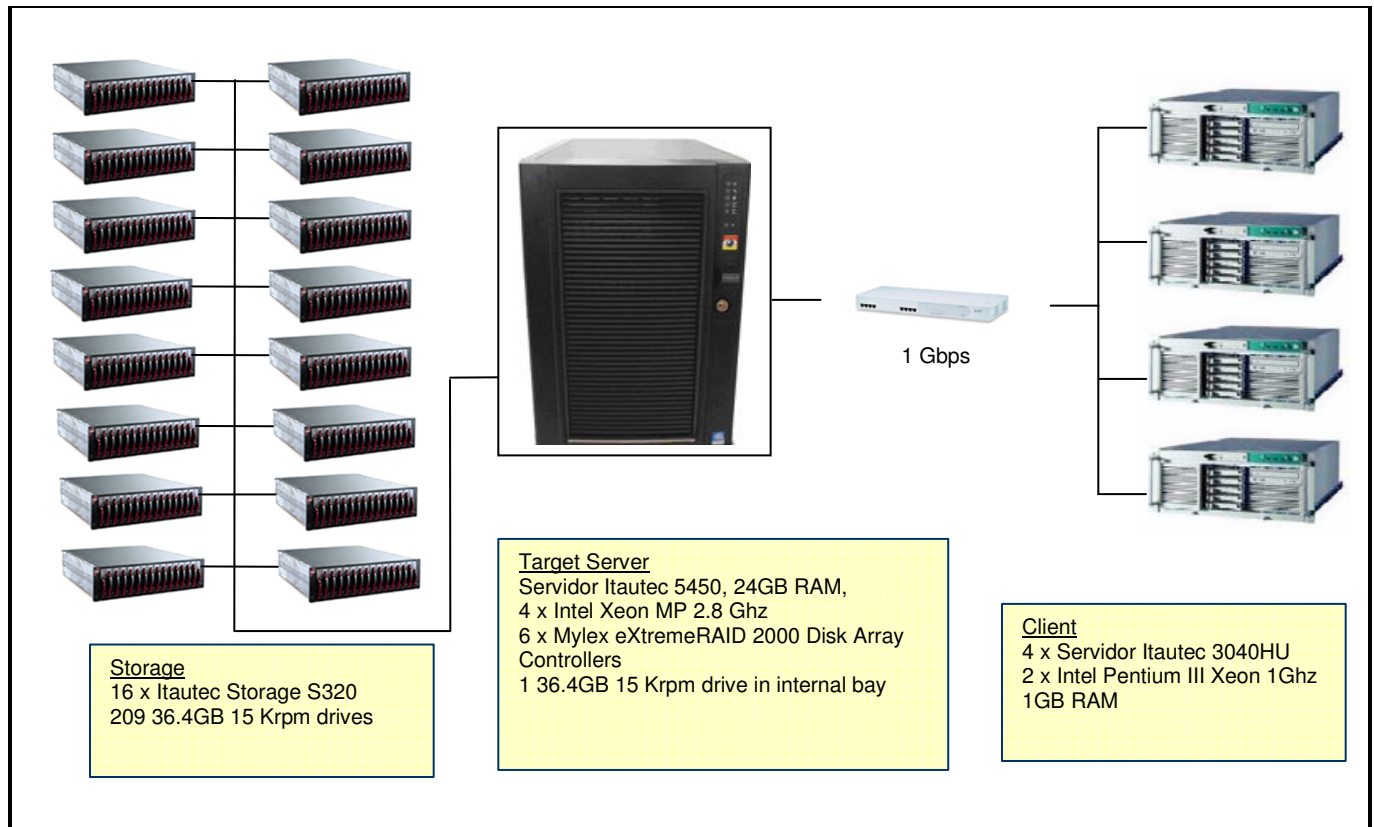


Figure 0.2 – Priced Configuration

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 209 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	0.99%
	Average Lines Per Order	10.00
Payment	Home Warehouse Transactions	85%
	Remote Warehouse Transactions	15%
	Non-Primary Key Access	60.02%
Order Status	Non-Primary Key Access	59.99%
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	4.02
Delivery	4.01
Stock Level	4.00

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 4,700 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) 47,000 users were started via the RTEs.
- 3) The system was run in 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 4700 warehouses, and the audited performance run used 4700 warehouses.

Table 4.1: Table Cardinality

Table	Initial Cardinality
Warehouse	4700
District	47000
Customer	141000000
New Order	42300000
Orders	141000000
History	141000000
Order Line	1409999206
Item	100000
Stock	470000000
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 209 36.4GB 15Krpm SCSI disks for the database. There were 195 disks connected to 5 controllers configured as RAID 0 and 14 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 5 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.

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/I, 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.

Table 4.2: Data Distribution

Controller	DB Components	Partition	Size	Disks
0	Miscellaneous Customer and Stock Backup	C:\dev\misc1 C:\dev\cs1 C:\dev\backup1	75,00GB 100,01GB 1157,45GB	39 – 36 GB 15K
1	Miscellaneous Customer and Stock Backup	C:\dev\misc2 C:\dev\cs2 C:\dev\backup2	75,00GB 100,01GB 1157,45GB	39 – 36 GB 15K
2	Miscellaneous Customer and Stock Backup	C:\dev\misc3 C:\dev\cs3 C:\dev\backup3	75,00GB 100,01GB 1157,45GB	39 – 36 GB 15K
3	Miscellaneous Customer and Stock Backup	C:\dev\misc4 C:\dev\cs4 C:\dev\backup4	75,00GB 100,01GB 1157,45GB	39 – 36 GB 15K
4	Miscellaneous Customer and Stock Backup	C:\dev\misc5 C:\dev\cs5 C:\dev\backup5	75,00GB 100,01GB 1157,45GB	39 – 36 GB 15K
5	Transaction Log	C:\dev\log	239,12 GB	14 – 36 GB 15K

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: **58,161.52** tpmC

Price per tpmC: **R\$ 26.72** 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	0.67	1.10	12.12
Payment	0.59	1.00	14.12
Order Status	0.62	1.04	7.79
Delivery (interactive)	0.10	0.11	1.09
Delivery (deferred)	0.26	0.46	1.47
Stock Level	1.05	1.90	8.17
Menu	0.10	0.11	9.51

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	18.02	18.00	18.03
Payment	3.02	3.00	3.03
Order Status	2.02	2.00	2.03
Delivery	2.02	2.00	2.03
Stock Level	2.02	2.00	2.03

Table 5.3: Think Times

Transaction	Average	Minimum	Maximum
New Order	12.07	0.00	120.71
Payment	12.06	0.00	120.71
Order Status	10.04	0.00	100.70
Delivery	5.07	0.00	50.71
Stock Level	5.08	0.00	50.71

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

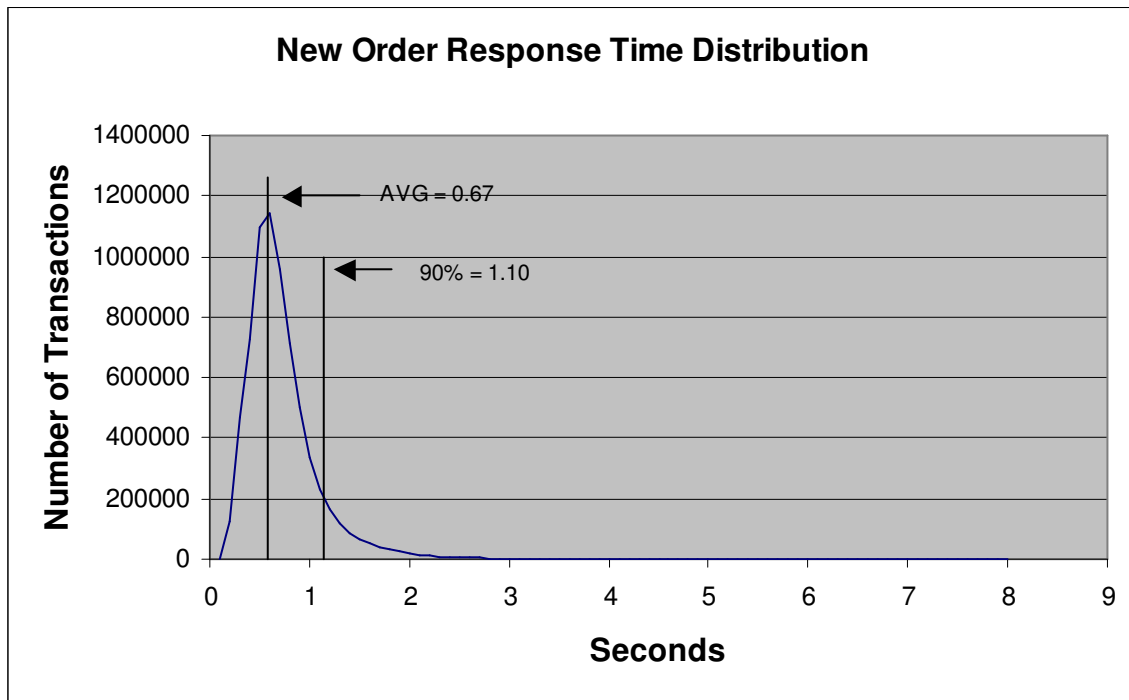


Figure 5.2 - Payment Transaction Response Time Distribution

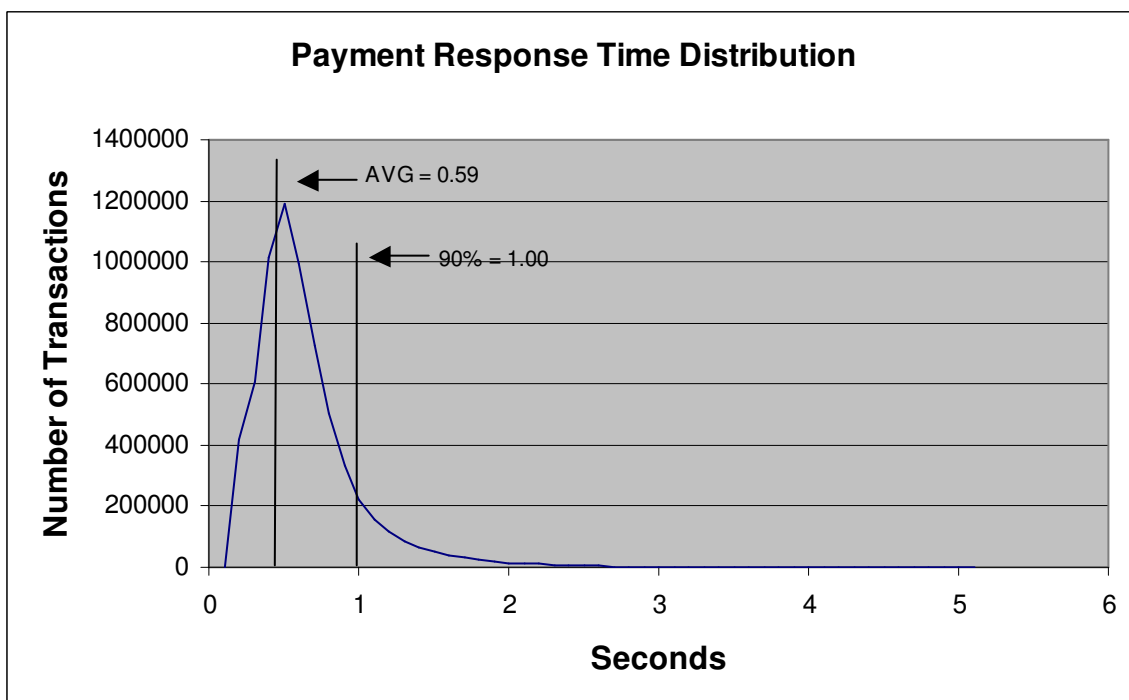


Figure 5.3 – Stock Level Transaction Response Time Distribution

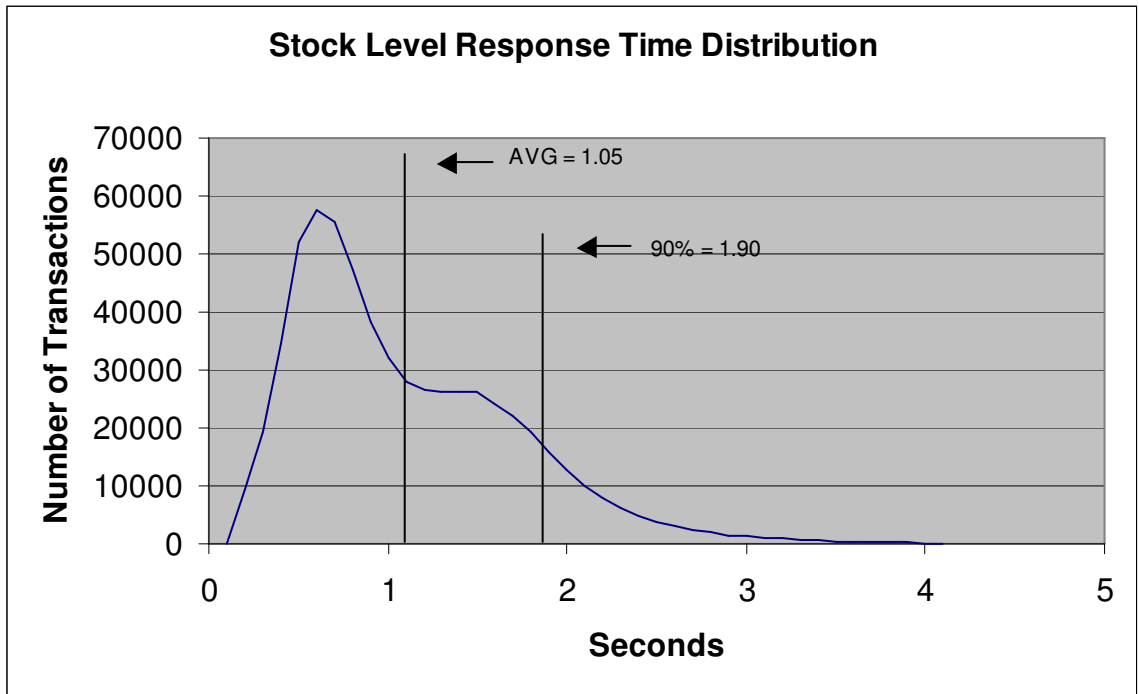


Figure 5.4 – Order Status Transaction Response Time Distribution

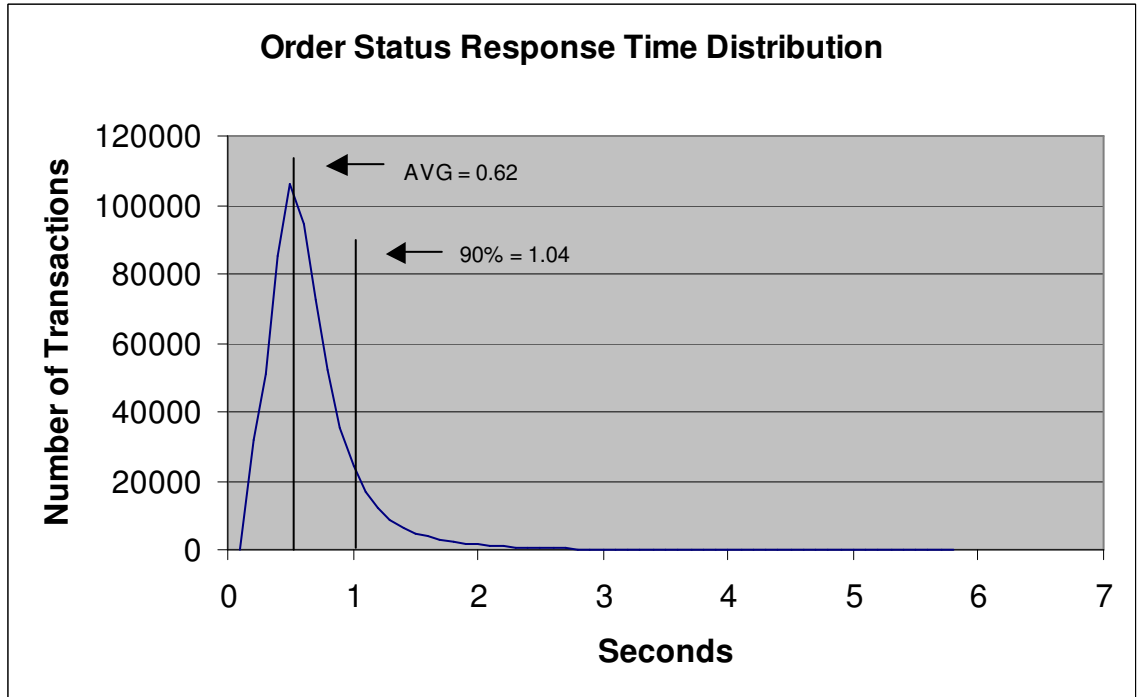


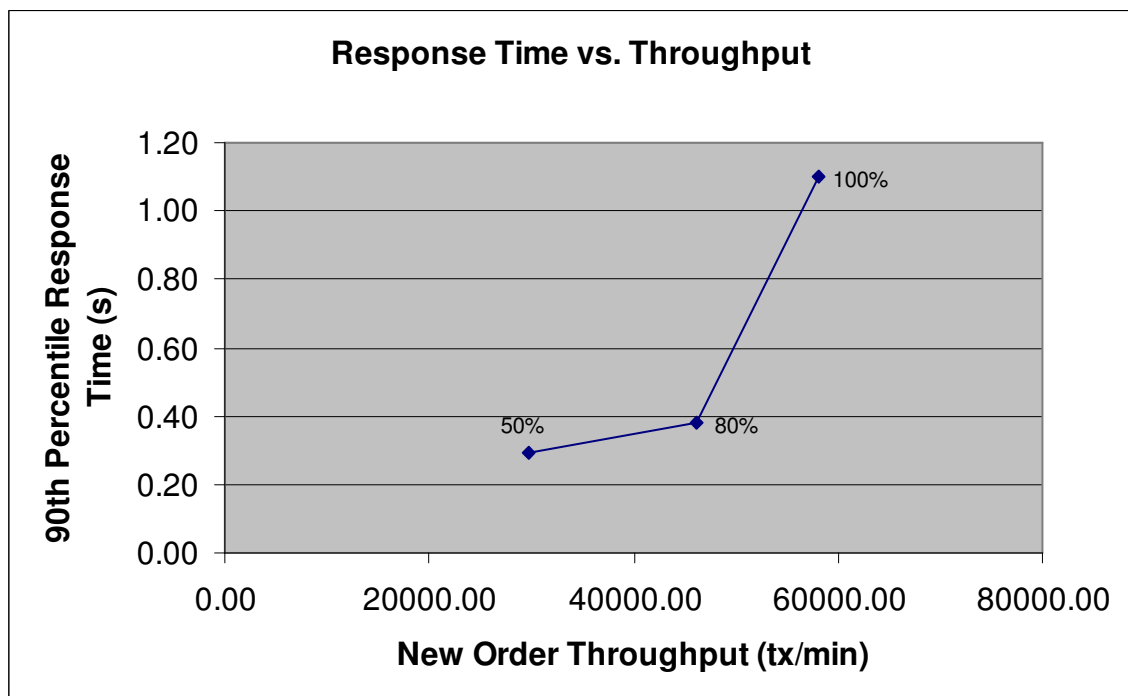
Figure 5.5 – Delivery Transaction Response Time Distribution



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.

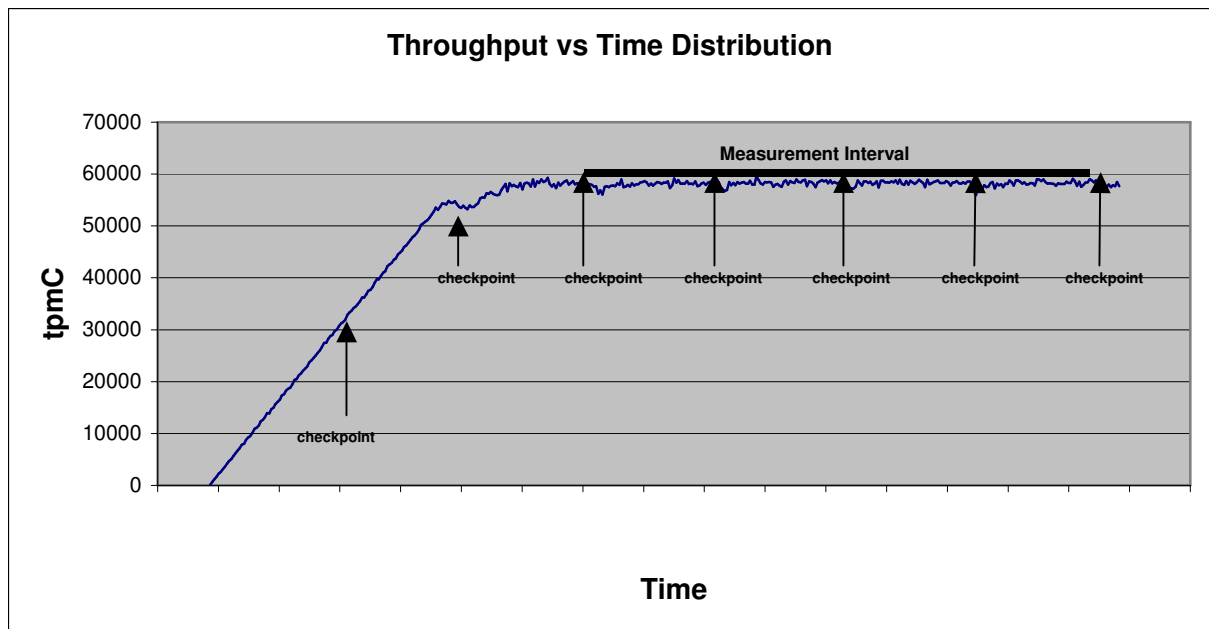
Figure 5.7 – New Order Think Time Distribution



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	00:02:00	02:02:00	7200
1 st Checkpoint	00:02:18	00:27:15	1497
2 nd Checkpoint	00:32:13	00:57:34	1581
3 rd Checkpoint	01:02:08	01:27:30	1522
4 th Checkpoint	01:32:03	01:58:50	1607

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 1497 seconds from the start of the measurement, and the checkpoint interval is 30 minutes.

Clause 6: SUT, Driver, and Communication Definition Related Items

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 identifying 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 5-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 availability 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: 11/03/2003

Software Availability Date: 11/03/2003

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: 58,161.52 tpmC

Price Performance Metric: R\$ 26.72

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:

- 4 Microsoft Windows 2000 Server Licenses
- 1 Microsoft Windows Server 2003 Enterprise Edition License
- 4 Microsoft SQL Server 2000 Enterprise Edition Licenses (per processor)
- 1 Microsoft Visual C++ 32 bit 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 Lorna Livingtree 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 Administrator, 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

or:

Laboratório de Análise de Performance
Centro de Informática / UFPE
Rua Prof. Luis Freire, S/N – Cidade Universitária
50740-540 – Recife – PE
Phone: +55 (81) 3453-9213



PERFORMANCE METRICS INC.
TPC Certified Auditors

November 03, 2003

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 5450
Database Manager: Microsoft SQL Server 2000 Enterprise Edition 32-bit
Operating System: Microsoft Windows 2003 Server, Enterprise Edition
Transaction Monitor: Microsoft COM+

System Under Test: Servidor Itautec 5450 with:				
CPU's	Memory	Disks (total)	90% Response	TpmC
4 Intel Xeon @ 2.8 Ghz	Main: 24 GB	209 @ 36.4GB 1 @ 36.4 GB (OS)	1.10	58,161.52
4 Clients each with:				
2 Intel Xeon @ 1.0 Ghz	Main: 1 GB	1 @ 18 GB	Na	Na

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

- The transactions were correctly implemented.
- The database files were properly sized and populated.
- The database was properly scaled with 4700 warehouses.

PERFORMANCE METRICS INC.
TPC Certified Auditors

- The ACID properties were successfully demonstrated.
- Log loss and data loss durability were demonstrated on a subset of the SUT configured with a database properly populated for 4,400 warehouses.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was present on the tested system.
- Eight hours of growth space for the dynamic tables was present on the tested system.
- The data for the 60 day space calculation was verified.
- The controller cache was disabled on the log disk controller.
- The steady state portion of the test was 120 minutes.
- At least one checkpoint was taken before the measured interval.
- Four complete checkpoints were taken during the measured interval.
- The system pricing was checked for major components and maintenance.

Auditor Notes:
None.

Sincerely,

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

Lorna Livingtree
Auditor

Appendix A – Source Code

isapi_dll/src/tpcc.def

Isapi_dll/src/tpcc.h

```
/* FILE: TPCC.H Microsoft TPC-C
 * Kit Ver. 4.20.000 Copyright
 * Microsoft, 1999 All Rights Reserved
 *
 *
 * 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
    free element or -1 if this entry in use.
    int
    w_id: //warehouse id
    assigned at welcome form
    int
    d_id: //district id
    assigned at welcome form
    int
    isyncid: //synchronization id
    int
    tickcount: //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
    //total allocated terminal array entries
    ifreelist;
    int
    //next available terminal array element or -1
    if none
    int
    imasterSyncId;

    //synchronization id
    CLIENTDATA *pClientData; //pointer to
    allocated client data
    } TERM;
typedef TERM *PTERM;
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_LOC_ID,
    ERR_DELIVERY_THREAD_FAILED,
    ERR_GETPROCADDR_FAILED,
    ERR_HTML_ILLEGAL,
    ERR_INVALID_SYNC_CONNECTION,
    ERR_INVALID_TERMINID,
    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_ID_KEY,
ERR_NEWORDER_MISSING_QTY_KEY,
ERR_NEWORDER_MISSING_SUPPWKEY,
ERR_NEWORDER_NOTITEMID_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_INVALID,
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_CID_INVALID,
ERR_PAYMENT_CID_AND_CLT,
ERR_PAYMENT_CUSTOMER_INVALID,
ERR_PAYMENT_INVALID,
ERR_PAYMENT_DISTRICT_INVALID,
ERR_PAYMENT_HAW_INVALID,
ERR_PAYMENT_HAW_RANGE,
ERR_PAYMENT_LAST_NAME_TOO_LONG,
ERR_PAYMENT_MISSING_CID_KEY,
ERR_PAYMENT_MISSING_CID_CLT,
ERR_PAYMENT_MISSING_CID_KEY,
ERR_PAYMENT_MISSING_CLT,
ERR_PAYMENT_MISSING_CLT_KEY,
ERR_PAYMENT_MISSING_CMT_KEY,
ERR_PAYMENT_MISSING_DID_KEY,
ERR_PAYMENT_MISSING_HAW_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;
    };

    *SzTextDetail, DWORD dwSystemErr)
    {
        m_Error = Err;
        m_SzTextDetail = new
        char[strlen(SzTextDetail)+1]; strcpy( m_SzTextDetail,
        SzTextDetail );
    };
};

November, 2003
```

```

dwSystemErr;

m_SystemErr =
    m_szErrorText = NULL;
};

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

WEBCRLT m_Error;
char
DWORD
    *m_szTextDetail; //
    *m_szErrorText;
    m_SystemErr;

int ErrType() {return
ERR_TYPE_WEBDLL;};

int ErrNum() {return m_Error;};
char *ErrText();
};

```

```

//These constants have already been defined in
engit.h, include 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 APENTRY DllMain(HANDLE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved);
void WriteMessageEventLog(LPCTSTR lpszMsg);
void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB,
int *pCmd, int *pFormId, int *pTermId, int *pSyncId);
void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer); Cmd(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 iErrType, 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 TerminateAll(void);
int TerminateAll(void);
void TerminateAll(int id);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int
iType, int iErrNum, 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);

```

Itaotec Philco TPC-C FULL DISCLOSURE REPORT
© 2003 Itaotec Philco, all rights reserved

```

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 GetNewOrderData(LPCTSTR lpszQueryString,
NEW_ORDER_DATA *pNewOrderData);
void GetPaymentData(LPCTSTR lpszQueryString, PAYMENT_DATA
*pPaymentData);
void GetOrderStatusData(LPCTSTR lpszQueryString,
ORDER_STATUS_DATA *pOrderStatusData);
void 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"
////////////////////////////////////
//
#define APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////
// English (U.S.) resources
//
#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#include _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

#define _MAC
////////////////////////////////////
//
// Version
//
VS_VERSION_INFO VERSIONINFO
FILEVERSION 0.4.0.0
PRODUCTVERSION 0.4.0.0

```

```

FILEFLAGS_MASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            (DIALOG)\0"
            VALUE "Comments", "TPC-C HTML DLL Server
            VALUE "CompanyName", "Microsoft\0"
            VALUE "FileDescription", "TPC-C HTML DLL
            Server (DLL)\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
    #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

```

isapi_dll/src/tpcc.cpp

```
#include "tpcc.h"
//ISAPI DLL information header
```

```
dwDelBuffFreeCount;
// number of buffers free
```

```

*      *      u1_reason_for_call reason for call
*      *      lpReserved          LPVOID
*      *      reserved for future use
*      *      RETURNS:            BOOL FALSE
*      *      errors occurred in
*      *      initialization
*      *      TRUE
*      *      successfully initialized
*      *      DLL
*      *      BOOL APIENTRY DllMain(HANDLE hModule, DWORD
*      *      u1_reason_for_call, LPVOID lpReserved)
*      *      {
*      *      DWORD i;
*      *      char szEvent[LEN_ERR_STRING] = "\0";
*      *      char szLogFile[128];
*      *      char szDllName[128];
*      *      // debugging...
*      *      // DebugBreak();
*      *      try
*      *      {
*      *      switch( u1_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);
*      *      ReadTPCCRegistrySettings( &reg ) )
*      *      if (
*      *      new CWEBCLNT_ERR( ERR_MISSING_REGISTRY_ENTRIES );
*      *      throw
*      *      min( Reg.dwMaxPendingDeliveries, 10000 ); // min with
*      *      10000 as a sanity constraint
*      *      dwNumDeliveryThreads = min(
*      *      Reg.dwNumberOfDeliveryThreads, 100 ); // min with 100
*      *      as a sanity constraint
*      *      Terminate();
*      *      // 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");
if
(pCTPCC_ENCINA_new == NULL)
throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
else if
{
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");
hLibInstanceCdb = LoadLibrary( szDllName );
if (hLibInstanceCdb == 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(hLibInstanceCdb, "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");
hLibInstanceCdb = LoadLibrary( szDllName );
if (hLibInstanceCdb == 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(hLibInstanceCdb, "CTPCC_ODBC_new");
if (pCTPCC_ODBC_new == NULL)
throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
}
if
{
(dwNumDeliveryThreads)
if
{
for deferred delivery txns:
hdoneEvent = CreateEvent( NULL, TRUE /*
manual reset */, FALSE /* initially not signalled */,
NULL );

```

```

tion);
InitializeCriticalSection(&delBuffCriticalSection);
dwDelBuffSemaphore = CreateSemaphore( NULL, 0,
dwDelBuffSize, NULL );
dwDelBuffFreeCount = dwDelBuffSize;

InitJulianTime(NULL);

create unique log file name based on deliLog-yyymmdd-
hhmm.log
SYSTEMTIME Time;
GetLocalTime( &Time );
wprintf( szLogFile, "%sdelivery-
%2.2d%2.2d%2.2d-%2.2d%2.2d.log",
Time.wMonth, Time.wDay, Time.wHour, Time.wMinute );
txndelillog = new CTxnLog(szLogFile,
TXN_LOG_WRITE);

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

allocate structures for delivery buffers and thread
mgmt
pbeliHandles = new
HANDLE[dwNumDeliveryThreads];
pdelBuff = new
DELIVERY_TRANSACTION[dwDelBuffSize];
launch DeliveryWorkerThread to perform actual delivery
txns
for(i=0; i<dwNumDeliveryThreads; i++) {
pbeliHandles[i] = (HANDLE) _beginthread(
DeliveryWorkerThread, 0, NULL );
if (pbeliHandles[i] == INVALID_HANDLE_VALUE)
throw new CWEBCLNT_ERR(
ERR_DELIVERY_THREAD_FAILED );
}
break;
case DLL_PROCESS_DETACH:
if
{
(dwNumDeliveryThreads)
{
(txndelillog != NULL)
{

```

```

//write event into txn log for STOP
txndelillog->WriteCtrlRectLog(TXN_EVENT_STOP,
szMyComputerName, sizeof(szMyComputerName));

// This will do a clean shutdown of the
delivery log file
CTxnLog *txndelillogLocal = txndelillog;
txndelillog= NULL;
delete txndelillogLocal;
}

delete [] pbeliHandles;
delete [] pdelBuff;

CloseHandle( hWorkerSemaphore );
CloseHandle( hBoneEvent );
DeleteCriticalSection(&delBuffCriticalSection
);
}

DeleteCriticalSection(&termCriticalSection);
if
(hLibInstanceTm != NULL)
FreeLibrary( hLibInstanceTm );
hLibInstanceTm
= NULL;
if
(hLibInstanceCbb != NULL)
FreeLibrary( hLibInstanceCbb );
hLibInstanceCbb
= NULL;
Sleep(500);
break;
default: /* nothing */;
}
catch (CBaseErr *e)
{
WriteMessageToEventLog( e-
>ErrorText() );
delete e;
TerminateExtension(0);
return FALSE;
}
catch (...)
{
WriteMessageToEventLog(TEXT("Unhandled
exception. DLL could not load."));
TerminateExtension(0);
return FALSE;
}
}

```

```

}
return TRUE;
}

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

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

return TRUE;
}

/* FUNCTION: TerminateExtension
* PURPOSE: This function is called by the inet
service when the DLL is about to be unloaded.
* ARGUMENTS: None
* RETURNS: TRUE inet service
expected return value.
*/
BOOL WINAPI TerminateExtension( DWORD dwFlags )
{
if (pdeliHandles)
{
SetEvent( hBoneEvent );
for(DWORD i=0,
i<dwNumDeliveryThreads; i++) waitforsingleobject(
pdeliHandles[i], INFINITE );
}
TerminateAll();
return TRUE;
}

/* FUNCTION: HttpExtensionProc
* PURPOSE: This function is the main entry
point for the TPC 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.
*

```



```

void WriteMessageToEventLog(LPCTSTR lpszMsg)
{
    TCHAR szMsg[256];
    HANDLE hEventSource;
    LPCTSTR 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
        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
* 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 //delivery transaction finished
    time

```

```

SYSTEMTIME
//delivery transaction start time
    trans_start;
    assert(txndelilog != NULL);
    try
    {
        if (Reg.cdb.Protocol == ODBC)
            pTxn = pCTPCC_ODBC.New(
            Reg.szbbServer, Reg.szbbuser, Reg.szbbPassword,
            szMyComputerName, Reg.szbbName, Reg.szspPrefix );
        else if (Reg.cdb.Protocol == DBLIB)
            pTxn = pCTPCC_DBLIB.New(
            Reg.szbbServer, Reg.szbbuser, Reg.szbbPassword,
            szMyComputerName, Reg.szbbName);
        pDeliveryData = pTxn->
        >BuffAddr_delivery();
    }
    catch (CBaseErr *e)
    {
        char szTmp[1024];
        wsprintf( szTmp, "Error in delivery
        Txn thread. Could not connect to database. "
        "%s",
        Server->User, Password, database);
        e->ErrorText(
        Reg.szbbServer, Reg.szbbuser, Reg.szbbPassword,
        Reg.szbbName );
        WriteMessageToEventLog( szTmp );
        delete e;
        goto ErrorExit;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
        exception caught in DeliveryWorkerThread."));
        goto ErrorExit;
    }
    while (TRUE)
    {
        try
        {
            //while delivery thread
            //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)
    ;
    *(pDelBuff+dwDelBuffBusyIndex);
    delivery =
    dwDelBuffFreeCount++;
    dwDelBuffBusyIndex++;
    if // wrap-around
    (dwDelBuffBusyIndex == dwDelBuffSize)
    {
        dwDelBuffBusyIndex = 0;
    }
    LeaveCriticalSection(&delBuffCriticalSection)
    ;
    >w_id = delivery.w_id;
    >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=
    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-
        delete e;
    }
    catch (...)
    {
        // unhandled exception;
        shouldnt'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
 *     TRUE error cannot post delivery info
 *     FALSE
 *     delivery information posted successfully
 */
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;
        GetLocalTime(&(pDelBuff+dwDelBuffFreeIndex)-
        >queue);
        dwDelBuffFreeCount--;
        dwDelBuffFreeIndex++;
        if (dwDelBuffFreeIndex ==
            dwDelBuffSize)
            // 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)
    }

```

```

wake up a worker thread
    ReleaseSemaphore( hWorkerSemaphore,
1, NULL );
}
return berror;
}

/* FUNCTION: ProcessQueryString
 * PURPOSE: This function extracts the relevant
 * information out of the http command passed in from
 * the browser.
 * COMMENTS: If this is the initial connection
 * i.e. client is at welcome screen then there will not
 * be a terminal id or current form id. If this is the
 * case
 * then the
 * pTermid and pFormid return values are undefined.
 */
void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB,
int *pCmd, int *pFormid, int *pTermid, int *pSyncid)
{
    char *ptr = pECB->lpszQueryString;
    char szBuffer[25];
    int i;

    //allowable client command strings i.e.
    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;
            // 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] , Reg_dwMaxConnections,
    szDBNames[Reg_edb_Protocol],
    dwNumDeliveryThreads, dwDelBuffSize );
    strcat( szBuffer, szTmp);
    if (Reg_erxnMon == 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.
// set per user. show options fy
options which will be used by the transaction
monitor:<BR>
color=\blue"><PRE>
"<font face="Courier New\
"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);
NAME=\w_id\ SIZE=6<BR>
"warehouse ID = <INPUT
SIZE=20 VALUE=\d_id\
"district ID = <INPUT NAME=\d_id\
SIZE=20 VALUE=\%s\ "><BR>
"</PRE></font><HR>"

```

```

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] = { 0 };
    char szPassword[32] = { 0 };
    char szDatabase[32] = "tpcc";
    "sa";
    // 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_ILLEGAL_FORMAT, 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_ILLEGAL_FORMAT, 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 =
pCTPCCTUXEDO_new();
    else if (Reg.eTxnMon == ENCINA)
Term.pClientData[iNewTerm].ptxn =
pCTPCCENCINA_new();
    else if (Reg.eTxnMon == COM)
Term.pClientData[iNewTerm].ptxn =
Term.pClientData[iNewTerm].ptxn =
pCTPCCCOM_new( Reg.BCOM_SinglePool );
    else if (Reg.eDB_Protocol == ODBC)
Term.pClientData[iNewTerm].ptxn =
pCTPCCODBC_new( szServer, szUser, szPassword,
szMyComputerName, szDatabase, Reg.szspPrefix );
    else if (Reg.eDB_Protocol == DBLIB)
Term.pClientData[iNewTerm].ptxn =
pCTPCCDBLIB_new( szServer, szUser, szPassword,
szMyComputerName, szDatabase );
}
catch (...)
{
    Term.Delete(iNewTerm);
    // pass
    exception upward
}
MakeMainMenuForm(iNewTerm,
Term.pClientData[iNewTerm].SyncId, 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);
    sprintf( 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()

```

<pre> { static SERRMSG errmsgs[] = { { 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. GetProcAddress DLL=" }, { ERR_HTML_ILLEGAL_FORMED, "Required key field is missing from HTML string." }, { ERR_INVALID_SYNC_CONNECTION, "Invalid Terminal Sync ID." }, { ERR_INVALID_TERMINAL_ID, "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\"." }, } } </pre>		<pre> { 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 suppw." }, { 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 Suppw key \"SP\"." }, { ERR_NEWORDER_NOITEMS_ENTERED, "New Order No order lines entered." }, { ERR_NEWORDER_QTY_INVALID, "New Order Qty 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 suppw." }, { ERR_NEWORDER_SUPPW_INVALID, "New Order Suppw 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." }, } </pre>
		<pre> { 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_DISTRICT_KEY, "Order Status missing District key \"DID\"." }, { ERR_PAYMENT_COI_INVALID, "Customer district invalid must be numeric." }, { ERR_PAYMENT_CID_AND_CLT, "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_TOO_LONG, "Payment Customer last name longer than 16 characters." }, { ERR_PAYMENT_MISSING_COI_KEY, "Payment missing Customer district key \"COI\"." }, { 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\"." }, } </pre>

```

    },
    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)

```

```

*pvale = 0; // terminating null
*queryString = ptr;
return;

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

/* FUNCTION: GetIntKeyValue
 * PURPOSE: This function parses a http
 * formatted string for a specific key value.
 * ARGUMENTS: char
 *               *queryString http string from client
 *               browser
 *               *pkey key
 *               value to look for
 *               *pvale key
 *               key not found WEBERROR
 *               value not numeric WEBERROR
 *               RETURNS: integer
 *               *ERROR: if (the pkey value is not found)
 *                       then
 *                       (NoKeyErr != NO_ERR)
 *                       throw CWEBCINTL_ERR(err)
 *               * return 0
 *               else if (non-
 *               numeric char found) then
 *               (NotIntErr != NO_ERR) then
 *               throw CWEBCINTL_ERR(err)
 *               return 0
 *               * COMMENTS: http keys are formatted either
 *               KEY=value& or KEY=value0. This DLL formats
 *               fields in such a manner that the keys can be extracted
 *               in the
 *               above manner.
 *               */
int GetIntKeyValue(char **queryString, char *pkey,
WEBERROR NoKeyErr, WEBERROR NotIntErr)
{
    char *ptr0;
    char *ptr;
    if ( ! (ptr=strstr(*queryString, pkey)) )
        goto ErrorNokey;
    ptr += strlen(pkey);
    if ( *ptr != '=' )
        goto ErrorNokey;
    ptr++;
}

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
 *               *queryString http string from client
 *               browser
 *               *pkey key
 *               value to look for
 *               *pvale key
 *               character array into which to place key's
 *               value
 *               *imax int
 *               maximum length of key value array.
 *               *err WEBERROR
 *               value to throw
 *               RETURNS: nothing.
 *               * ERROR: if (the pkey value is not found)
 *                       then
 *                       if
 *                       (err == 0)
 *                       return (empty string)
 *                       else
 *                       throw CWEBCINTL_ERR(err)
 *               * COMMENTS: http keys are formatted either
 *               KEY=value& or KEY=value0. This DLL formats
 *               fields in such a manner that the keys can be extracted
 *               in the
 *               above manner.
 *               */
void GetKeyValue(char **queryString, char *pkey, char
*pvalue, int *imax, WEBERROR err)
{
    char *ptr;
    if ( ! (ptr=strstr(*queryString, pkey)) )
        goto ErrorExit;
    ptr += strlen(pkey);
    if ( *ptr != '=' )
        goto ErrorExit;
    ptr++;
    *imax--; // one position is for terminating
    while( *ptr && *ptr != '&' && *imax )
    {
        *pvalue++ = *ptr++;
        *imax--;
    }
}

```

```

        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 (NotInErr != 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
 * TPC.BLL
 * is first loaded by the
 * inet service.
 */
void TermInit(void)
{
    EnterCriticalSection(&TermCriticalSection);
    Term.iMastersSyncId = 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 =
        // 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 TPC.BLL
 */
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)
            delete 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: assigned terminal id
 */
int TermAdd(void)
{
    DWORD i;
    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=0xFFFFFFF; i<Reg.dwMaxConnections; i++)
        {

```

```

            Term.pClientData[i].iTickCount >
                if (iTickCount >
                    iTickCount)
                {
                    iNewTerm = i;
                }
            // if oldest term is less than one
            minute old, it probably means that more connections
            specified as "Max Connections" at install. In this
            case,
            // do not bump existing connection;
            instead, return error to requester.
            if ((GetTickCount() - iTickCount) <
                60000)
            {
                LeaveCriticalSection(&TermCriticalSection);
                throw new CWEBCLNT_ERR(
                    ERR_MAX_CONNECTIONS_EXCEEDED );
            }
            Term.pClientData[iNewTerm].iTickCount =
                GetTickCount();
            Term.pClientData[iNewTerm].iSyncId =
                Term.iMastersSyncId++;
            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
 * RETURNS: 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 )

```



```

NAME="\CID*" \ SIZE=4" Name:
Credit: %Disc:<BR>"

"Customer: <INPUT
Number of Lines:
w_tax: D_tax:<BR>
<BR>"

Name Qty Stock B/g Item
Amount<BR>"

SIZE=4" <INPUT NAME="\IID00*" \ SIZE=6"
<INPUT NAME="\Qty00*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID01*" \ SIZE=6"
<INPUT NAME="\Qty01*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID02*" \ SIZE=6"
<INPUT NAME="\Qty02*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID03*" \ SIZE=6"
<INPUT NAME="\Qty03*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID04*" \ SIZE=6"
<INPUT NAME="\Qty04*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID05*" \ SIZE=6"
<INPUT NAME="\Qty05*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID06*" \ SIZE=6"
<INPUT NAME="\Qty06*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID07*" \ SIZE=6"
<INPUT NAME="\Qty07*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID08*" \ SIZE=6"
<INPUT NAME="\Qty08*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID09*" \ SIZE=6"
<INPUT NAME="\Qty09*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID10*" \ SIZE=6"
<INPUT NAME="\Qty10*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID11*" \ SIZE=6"
<INPUT NAME="\Qty11*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID12*" \ SIZE=6"
<INPUT NAME="\Qty12*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID13*" \ SIZE=6"
<INPUT NAME="\Qty13*" \ SIZE=1><BR>
SIZE=4" <INPUT NAME="\IID14*" \ SIZE=6"
<INPUT NAME="\Qty14*" \ SIZE=1><BR>

Total:<BR>"

"</font></PRE><HR>"
"<INPUT TYPE=\submit\"
NAME=\CMD" VALUE=\Menu">"</FORM></HTML>"
);
}
else
{
%6.6d District: %2.2d c += sprintf(szfForm+c, "warehouse:
pNewOrderData->w_id,
pNewOrderData->d_id);
if ( bvalid )

```

```

{
"%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
pNewOrderData-
>o_entry_d.day,
pNewOrderData-
>o_entry_d.month,
pNewOrderData-
>o_entry_d.year,
pNewOrderData-
>o_entry_d.hour,
pNewOrderData-
>o_entry_d.minute,
pNewOrderData-
>o_entry_d.second);
}

c += sprintf(szfForm+c,
"%4.4d Name: %16s Credit: %2s
",
pNewOrderData->c_last, pNewOrderData->c_id,
pNewOrderData->c_credit);

if ( bvalid )
{
c += sprintf(szfForm+c,
"%Disc: %5.2f
",
"%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
pNewOrderData-
>o_entry_d.day,
pNewOrderData-
>o_entry_d.month,
pNewOrderData-
>o_entry_d.year,
pNewOrderData-
>o_entry_d.hour,
pNewOrderData-
>o_entry_d.minute,
pNewOrderData-
>o_entry_d.second);
}

c += sprintf(szfForm+c,
"%6.6d %5.2f %$7.2f %2.2d
%3.3d %1.1s %$6.2f %$7.2f %$7.2f
",
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 += sprintf(szfForm+c,
"%Disc:<BR>"

```

```

%8.8d Number of Lines:
D_tax:<BR> <BR>"

"Order Number:
w_tax:
" Supp_w
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 )
{
"Execution Status: Transaction committed.
Total: %$8.2f",
pNewOrderData-
>total_amount);
else
{
"Execution Status: Item number is not valid.
Total:");
strcpy( szForm+c,
"<BR></font></PRE><HR>"
"<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
* when the client terminal id is no longer needed.
*/

void MakePaymentForm(int itemid, PAYMENT_DATA
*pPaymentData, BOOL binput, char *szForm)
{
int c;

c = sprintf(szfForm,
"<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=\TERMINID" VALUE=\%d">"

```

```

NAME="\SYNCID\" VALUE="%d\"%s"
Payment<BR> "
    "Date: "
    Term.pClientData[iTermId].iSyncId;
    if ( !bInput )
    {
        %2.2d-%4.4d %2.2d:%2.2d:%2.2d C += sprintf(szForm+C, "%2.2d-
        >h_date.month,
        >h_date.year,
        >h_date.hour,
        >h_date.minute,
        >h_date.second);
    }
    if ( bInput )
    {
        c += sprintf(szForm+C,
            "<BR> District: <INPUT NAME=\"%DID*\" SIZE=1><BR> <BR> <BR>
            NAME=\"%CID*\" SIZE=4\"
            NAME=\"%CWT*\" SIZE=4\"
            NAME=\"%CDI*\" SIZE=1><BR>\"
            <INPUT NAME=\"%CLT*\" SIZE=16\"
            Since:<BR> \"
            Credit:<BR> \"
            Disc:<BR> \"
            Phone:<BR> \"
            $<INPUT NAME=\"%HAM*\" SIZE=7>
            Balance:<BR> \"
            <BR>Cust-data: <BR><BR> <BR> <BR>
            <BR></font></PRE><HR>\"
            NAME=\"%CMD\" VALUE=\"%Process\"
            NAME=\"%CMD\" VALUE=\"%Menu\">
            "</BODY></FORM></HTML>"
        Term.pClientData[iTermId].w_idid);
    }
} else
{
    c += sprintf(szForm+C,
        "<BR> District: <BR>warehouse:
        \"%-20s\"
        \"%-20s\"
        \"%-20s <BR>\"
        \"%-20s <BR>\"
        \"%-20s %-2s %5.5s-%4.4s

```

```

Warehouse: %6.6d Cust-District: %2.2d<br>
Since: %2.2d%2.2d-%4.4d<br>
Credit: %-2s<br>"

Term.pClientData[iTermId].w_id, pPaymentData->d_id
>w_street_1, pPaymentData->d_street_1
>w_street_2, pPaymentData->d_street_2
pPaymentData->w_state, pPaymentData->w_zip,
pPaymentData->w_zip+5
pPaymentData->d_state, pPaymentData->d_city,
pPaymentData->d_zip,
pPaymentData->c_w_id,
pPaymentData->c_id,
pPaymentData->c_first,
pPaymentData->c_middle, pPaymentData->c_last
>c_since.day, pPaymentData->c_since.month,
pPaymentData->c_since.year
>c_street_1, pPaymentData->c_credit
);

c += sprintf(szForm+c, " %-20s
%Disc: %5.2f<br> ",
100.0*pPaymentData->c_discount);

c += wsprintf(szForm+c,
"%5.5s-%4.4s Phone: %6.6s-%3.3s-%4.4s<br>
<br> ",
pPaymentData->c_city,
pPaymentData->c_zip,
pPaymentData->c_phone,
pPaymentData->c_phone+9,
pPaymentData->c_phone+12 );

c += sprintf(szForm+c,
"%7.2f New Cust-balance: %14.2f<br>
%13.2f<br> <br> ",
pPaymentData->c_balance
, pPaymentData-
>c_credit_lim
);

' B' && pPaymentData->c_credit[0] ==
if ( pPaymentData->c_credit[1] == 'C' )
c += wsprintf(szForm+c,
"%5.50s<br>
%5.50s<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='\"..De livery..\>"
        "<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
*poOrderStatusData, BOOL bInput, char *szForm)
{
    int
    szTtC char szBr[] = " <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='\"%d\">"
        "<INPUT TYPE='\"hidden\"' "
NAME='\"ERROR\"' VALUE='\"0\">"
        "<INPUT TYPE='\"hidden\"' "
NAME='\"FORMID\"' VALUE='\"%d\">"
        "<INPUT TYPE='\"hidden\"' "
NAME='\"TERMD\"' 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='\"DIP\">" SIZE=15><BR>"
        );
    }
}

```

```

NAME="CID%" SIZE=4" Name:
<INPUT NAME="\CLT%" SIZE=23><BR>
"Order-Number:
Carrier-Number:
"Supply-w Item-Id
Qty Amount Delivery-date<BR> <BR> <BR>
<BR>
" <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<HR></Font></PRE>
TYPE="\submit" NAME="\CMD" VALUE="\..NewOrder..\>
VALUE="\submit"
"VALUES="\Menu"
</BODY></FORM></HTML>
");
}
else
{
c += sprintf(szForm+C,
"District: %2.2d<BR>"
"Customer: %4.4d Name:
porderStatusData->d_id,
porderStatusData-
>c_first, porderStatusData->c_middle, porderStatusData-
>c_last);
}
c += sprintf(szForm+C, "Cust-
Balance: %9.2f<BR> <BR>",
porderStatusData-
>c_balance);
c += sprintf(szForm+C,
"Order-Number: %8.8d
Entry-date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d
Carrier-Number: %2.2d<BR>"
"Supply-w Item-Id
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_o1_cnt; i++)
{
c += sprintf(szForm+C,
"%6.6d %6.6d %2.2d %8.2f %2.2d-%2.2d-%
4.4d<BR>",
porderStatusData->o1[i].o1_supply_w_id,
porderStatusData->o1[i].o1_i_id,
porderStatusData->o1[i].o1_quantity,
porderStatusData->o1[i].o1_amount,
porderStatusData->o1[i].o1_delivery_d.day,

```

```

porderStatusData->o1[i].o1_delivery_d.month,
porderStatusData->o1[i].o1_delivery_d.year);
}
strcpy( szForm+C, szBR, (15-i)*5
);
c += (15-i)*5;
strcpy(szForm+C,
"<Font></PRE><HR><INPUT
TYPE="\submit" NAME="\CMD" VALUE="\..NewOrder..\>
NAME="\CMD" VALUE="\..Payment..">
"VALUES="\submit"
NAME="\CMD" VALUE="\..Delivery..">
"VALUES="\submit"
NAME="\CMD" VALUE="\..Order-Status..">
"VALUES="\submit"
NAME="\CMD" VALUE="\..Stock-Level..">
"VALUES="\submit"
NAME="\CMD" VALUE="\..Exit..">
"VALUES="\submit"
");
}
}
/* 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 itemid, DELIVERY_DATA
*pDeliveryData, BOOL binput, char *szForm)
{
int c;
c = sprintf(szForm,
"<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD><BODY>
<FORM ACTION="\tpcc.dll"
METHOD="\GET" ">
" <INPUT TYPE="\hidden"
NAME="\STATUSID" VALUE="\%d">
" <INPUT TYPE="\hidden"
NAME="\ERROR" VALUE="\0">
" <INPUT TYPE="\hidden"
NAME="\FORMID" VALUE="\%d">
" <INPUT TYPE="\hidden"
NAME="\TERMD" 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, itemid,
Term.pClientData[itemid].syncid,
Term.pClientData[itemid].w_id);
if ( binput )
{
strcpy( szForm+C,
"Carrier Number: <INPUT
NAME="\OCD%" SIZE=1d> <BR>"

```

```

"Execution Status: <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> </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>"
>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 in the required
* input variables. it then calls the pNewOrder
* constructor to construct the transaction, constructs
* the output form and writes it back to client
* browser.
*/
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK
*pECB, int itemid, char *szBuffer)
{
PNEW_ORDER_DATA pNewOrder;
pNewOrder = Term.pClientData[itemid].ptxn-
>BufAddr_NewOrder();
ZeroMemory(pNewOrder,
sizeof(NEW_ORDER_DATA));
pNewOrder->i_id =
Term.pClientData[itemid].w_id;
getNewOrderData(pECB->lpszQueryString,
pNewOrder);
Term.pClientData[itemid].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
 * required input variables. It then calls 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.
 *
 * iTermId Client browser terminal id
 */
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB,
int iTermId, char *szBuffer)
{
    PAYMENT_DATA pPayment;

    payment = Term.pClientData[iTermId].pTxn-
    payment();
    >BuffAddr_Payment();
    pPayment = Term.pClientData[iTermId].pTxn-
    payment->w_id;
    Term.pClientData[iTermId].w_id;
    GetPaymentData(pECB->lpszQueryString,
    pPayment);

    Term.pClientData[iTermId].pTxn->Payment();

    payment = Term.pClientData[iTermId].pTxn-
    payment();
    >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.
 *
 * iTermId client browser terminal id
 */
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer)
{
    ORDER_STATUS_DATA pOrderStatus;

    pOrderStatus =
    Term.pClientData[iTermId].pTxn->BuffAddr_OrderStatus();
    ZeroMemory(pOrderStatus,
    sizeof(ORDER_STATUS_DATA));

```

```

        pOrderStatus->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.
 *
 * 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-
    Delivery();
    ZeroMemory(pDelivery, sizeof(DELIVERY_DATA));
    Term.pClientData[iTermId].w_id;

    pDelivery->o_carrier_id =
    GetIntKeyValue(&ptr, "OCD",
    ERR_DELIVERY_MISSING_OCD_KEY,
    ERR_DELIVERY_CARRIER_INVALID);
    pDelivery->o_carrier_id < 1 || 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;
            }
        }
        // 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.
 *
 * 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 ));

    Term.pClientData[iTermId].w_id;
    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
 * http command string
 * *pNewOrderData
 * order data structure
 */

```

```

void GetNewOrderData(LPSTR lpszQueryString,
NEW_ORDER_DATA *pNewOrderData)
{
    char szTmp[26];
    int i;
    int items;
    int ol_i_id, ol_quantity;
    int *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,
ERR_NEW_ORDER_FORM_MISSING_DID,
ERR_NEW_ORDER_DISTRICT_INVALID);
    pNewOrderData->c_id = GetIntKeyValue(&ptr,
ERR_NEW_ORDER_CUSTOMER_KEY,
ERR_NEW_ORDER_CUSTOMER_INVALID);

    i++;
    for(i=0, items=0; i<MAX_OL_NEW_ORDER_ITEMS;
    i++)
    {
        GetKeyValue(&ptr, szSP[i], szTmp,
        sizeof(szTmp), ERR_NEW_ORDER_MISSING_SUPPW_KEY);
        if ( szTmp[0] )
        {
            if ( !IsNumeric(szTmp) )
            throw new
            pNewOrderData-
            >OL[items].ol_supply_w_id = atoi(szTmp);

            ol_i_id = pNewOrderData-
            >OL[items].ol_i_id =
            GetIntKeyValue(&ptr, szIID[i],
            ERR_NEW_ORDER_MISSING_IID_KEY,
            ERR_NEW_ORDER_ITEMID_INVALID);
            if ( ol_i_id > 999999 ||
            ol_i_id < 1 )
            throw new
            CWEBCLNT_ERR( ERR_NEW_ORDER_ITEMID_RANGE );

            pNewOrderData->OL[items].ol_quantity =
            ol_quantity =
            GetIntKeyValue(&ptr, szQTY[i],
            ERR_NEW_ORDER_MISSING_QTY_KEY,
            ERR_NEW_ORDER_QTY_INVALID);
            if ( ol_quantity > 99 ||
            ol_quantity < 1 )

```

```

CWEBCLNT_ERR( ERR_NEW_ORDER_QTY_RANGE );
        throw new
        items++;
    }
    else
    {
        // nothing entered for
        supply warehouse, so item id and qty must also be blank
        szIID[i], szTmp, sizeof(szTmp),
        ERR_NEW_ORDER_MISSING_IID_KEY);
        if ( szTmp[0] )
        throw new
        CWEBCLNT_ERR( ERR_NEW_ORDER_ITEMID_WITHOUT_SUPPW );
        GetKeyValue(&ptr,
        szQTY[i], szTmp, sizeof(szTmp),
        ERR_NEW_ORDER_MISSING_QTY_KEY);
        if ( szTmp[0] )
        throw new
        CWEBCLNT_ERR( ERR_NEW_ORDER_QTY_WITHOUT_SUPPW );
    }
    if ( items == 0 )
    throw new CWEBCLNT_ERR(
    ERR_NEW_ORDER_NOITEMS_ENTERED );
    pNewOrderData->o_ol_cnt = items;

    /* FUNCTION: GetPaymentData
    * PURPOSE: This function extracts and
    validates the payment form data from an http command
    string.
    * ARGUMENTS: LPSTR Client browser
    lpszQueryString http command string
    * *pPaymentData PAYMENT_DATA
    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,
        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
            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 );
        }
        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 )

```

```

ERR_ORDERSTATUS_MISSING_CID_CLT );
                                throw new CWEBCLNT_ERR(
                                    _strupr( szTmp );
                                    if ( strlen(pOrderStatusdata-
>c_last) > LAST_NAME_LEN )
ERR_ORDERSTATUS_CLT_RANGE );
                                throw new CWEBCLNT_ERR(
szTmp);
                                strcpy(pOrderStatusdata->c_last,
                                }
                                {
                                // parse customer id and verify
                                that last name was NOT entered
                                if ( !IsNumeric(szTmp) )
ERR_ORDERSTATUS_CID_INVALID );
                                throw new CWEBCLNT_ERR(
                                pOrderStatusdata->c_id =
atoi(szTmp);
                                GetKeyValue(&ptr, "CLT*", szTmp,
                                sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CLT_KEY);
                                if ( szTmp[0] != 0 )
ERR_ORDERSTATUS_CID_AND_CLT );
                                throw new CWEBCLNT_ERR(
                                }
                                }
/* 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
* 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
* 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
//
#define IDB_BITMAP1 102
#define IDB_BITMAP2 103
#define IDB_BITMAP3 104
#define IDB_BITMAP4 105
#define IDB_BITMAP5 106
#define IDB_BITMAP6 107
#define IDB_BITMAP7 108
#define IDB_BITMAP8 109
#define IDB_BITMAP9 110
#define IDB_BITMAP10 111
#define IDB_BITMAP11 112
#define IDB_BITMAP12 113
#define IDB_BITMAP13 114
#define IDB_BITMAP14 115
#define IDB_BITMAP15 116
#define IDB_BITMAP16 117
#define IDB_BITMAP17 118
#define IDB_BITMAP18 119
#define IDB_BITMAP19 120
#define IDB_BITMAP20 121
#define IDB_BITMAP21 122
#define IDB_BITMAP22 123
#define IDB_BITMAP23 124
#define IDB_BITMAP24 125
#define IDB_BITMAP25 126
#define IDB_BITMAP26 127
#define IDB_BITMAP27 128
#define IDB_BITMAP28 129
#define IDB_BITMAP29 130
#define IDB_BITMAP30 131
#define IDB_BITMAP31 132
#define IDB_BITMAP32 133
#define IDB_BITMAP33 134
#define IDB_BITMAP34 135
#define IDB_BITMAP35 136
#define IDB_BITMAP36 137
#define IDB_BITMAP37 138
#define IDB_BITMAP38 139
#define IDB_BITMAP39 140
#define IDB_BITMAP40 141
#define IDB_BITMAP41 142
#define IDB_BITMAP42 143
#define IDB_BITMAP43 144
#define IDB_BITMAP44 145
#define IDB_BITMAP45 146
#define IDB_BITMAP46 147
#define IDB_BITMAP47 148
#define IDB_BITMAP48 149
#define IDB_BITMAP49 150
#define IDB_BITMAP50 151
#define IDB_BITMAP51 152
#define IDB_BITMAP52 153
#define IDB_BITMAP53 154
#define IDB_BITMAP54 155
#define IDB_BITMAP55 156
#define IDB_BITMAP56 157
#define IDB_BITMAP57 158
#define IDB_BITMAP58 159
#define IDB_BITMAP59 160
#define IDB_BITMAP60 161
#define IDB_BITMAP61 162
#define IDB_BITMAP62 163
#define IDB_BITMAP63 164
#define IDB_BITMAP64 165
#define IDB_BITMAP65 166
#define IDB_BITMAP66 167
#define IDB_BITMAP67 168
#define IDB_BITMAP68 169
#define IDB_BITMAP69 170
#define IDB_BITMAP70 171
#define IDB_BITMAP71 172
#define IDB_BITMAP72 173
#define IDB_BITMAP73 174
#define IDB_BITMAP74 175
#define IDB_BITMAP75 176
#define IDB_BITMAP76 177
#define IDB_BITMAP77 178
#define IDB_BITMAP78 179
#define IDB_BITMAP79 180
#define IDB_BITMAP80 181
#define IDB_BITMAP81 182
#define IDB_BITMAP82 183
#define IDB_BITMAP83 184
#define IDB_BITMAP84 185
#define IDB_BITMAP85 186
#define IDB_BITMAP86 187
#define IDB_BITMAP87 188
#define IDB_BITMAP88 189
#define IDB_BITMAP89 190
#define IDB_BITMAP90 191
#define IDB_BITMAP91 192
#define IDB_BITMAP92 193
#define IDB_BITMAP93 194
#define IDB_BITMAP94 195
#define IDB_BITMAP96 197
#define IDB_BITMAP98 199
#define IDB_BITMAP99 200
#define IDB_BITMAP100 201
#define IDB_BITMAP101 202
#define IDB_BITMAP102 203
#define IDB_BITMAP103 204
#define IDB_BITMAP104 205
#define IDB_BITMAP105 206
#define IDB_BITMAP106 207
#define IDB_BITMAP107 208
#define IDB_BITMAP108 209
#define IDB_BITMAP109 210
#define IDB_BITMAP110 211
#define IDB_BITMAP111 212
#define IDB_BITMAP112 213
#define IDB_BITMAP113 214
#define IDB_BITMAP114 215
#define IDB_BITMAP115 216
#define IDB_BITMAP116 217
#define IDB_BITMAP117 218
#define IDB_BITMAP118 219
#define IDB_BITMAP119 220
#define IDB_BITMAP120 221
#define IDB_BITMAP121 222
#define IDB_BITMAP122 223
#define IDB_BITMAP123 224
#define IDB_BITMAP124 225
#define IDB_BITMAP125 226
#define IDB_BITMAP126 227
#define IDB_BITMAP127 228
#define IDB_BITMAP128 229
#define IDB_BITMAP129 230
#define IDB_BITMAP130 231
#define IDB_BITMAP131 232
#define IDB_BITMAP132 233
#define IDB_BITMAP133 234
#define IDB_BITMAP134 235
#define IDB_BITMAP135 236
#define IDB_BITMAP136 237
#define IDB_BITMAP137 238
#define IDB_BITMAP138 239
#define IDB_BITMAP139 240
#define IDB_BITMAP140 241
#define IDB_BITMAP141 242
#define IDB_BITMAP142 243
#define IDB_BITMAP143 244
#define IDB_BITMAP144 245
#define IDB_BITMAP145 246
#define IDB_BITMAP146 247
#define IDB_BITMAP147 248
#define IDB_BITMAP148 249
#define IDB_BITMAP149 250
#define IDB_BITMAP150 251
#define IDB_BITMAP151 252
#define IDB_BITMAP152 253
#define IDB_BITMAP153 254
#define IDB_BITMAP154 255
#define IDB_BITMAP155 256
#define IDB_BITMAP156 257
#define IDB_BITMAP157 258
#define IDB_BITMAP158 259
#define IDB_BITMAP159 260
#define IDB_BITMAP160 261
#define IDB_BITMAP161 262
#define IDB_BITMAP162 263
#define IDB_BITMAP163 264
#define IDB_BITMAP164 265
#define IDB_BITMAP165 266
#define IDB_BITMAP166 267
#define IDB_BITMAP167 268
#define IDB_BITMAP168 269
#define IDB_BITMAP169 270
#define IDB_BITMAP170 271
#define IDB_BITMAP171 272
#define IDB_BITMAP172 273
#define IDB_BITMAP173 274
#define IDB_BITMAP174 275
#define IDB_BITMAP175 276
#define IDB_BITMAP176 277
#define IDB_BITMAP177 278
#define IDB_BITMAP178 279
#define IDB_BITMAP179 280
#define IDB_BITMAP180 281
#define IDB_BITMAP181 282
#define IDB_BITMAP182 283
#define IDB_BITMAP183 284
#define IDB_BITMAP184 285
#define IDB_BITMAP185 286
#define IDB_BITMAP186 287
#define IDB_BITMAP187 288
#define IDB_BITMAP188 289
#define IDB_BITMAP189 290
#define IDB_BITMAP190 291
#define IDB_BITMAP191 292
#define IDB_BITMAP192 293
#define IDB_BITMAP193 294
#define IDB_BITMAP194 295
#define IDB_BITMAP195 296
#define IDB_BITMAP196 297
#define IDB_BITMAP197 298
#define IDB_BITMAP198 299
#define IDB_BITMAP199 300
#define IDB_BITMAP200 301
#define IDB_BITMAP201 302
#define IDB_BITMAP202 303
#define IDB_BITMAP203 304
#define IDB_BITMAP204 305
#define IDB_BITMAP205 306
#define IDB_BITMAP206 307
#define IDB_BITMAP207 308
#define IDB_BITMAP208 309
#define IDB_BITMAP209 310
#define IDB_BITMAP210 311
#define IDB_BITMAP211 312
#define IDB_BITMAP212 313
#define IDB_BITMAP213 314
#define IDB_BITMAP214 315
#define IDB_BITMAP215 316
#define IDB_BITMAP216 317
#define IDB_BITMAP217 318
#define IDB_BITMAP218 319
#define IDB_BITMAP219 320
#define IDB_BITMAP220 321
#define IDB_BITMAP221 322
#define IDB_BITMAP222 323
#define IDB_BITMAP223 324
#define IDB_BITMAP224 325
#define IDB_BITMAP225 326
#define IDB_BITMAP226 327
#define IDB_BITMAP227 328
#define IDB_BITMAP228 329
#define IDB_BITMAP229 330
#define IDB_BITMAP230 331
#define IDB_BITMAP231 332
#define IDB_BITMAP232 333
#define IDB_BITMAP233 334
#define IDB_BITMAP234 335
#define IDB_BITMAP235 336
#define IDB_BITMAP236 337
#define IDB_BITMAP237 338
#define IDB_BITMAP238 339
#define IDB_BITMAP239 340
#define IDB_BITMAP240 341
#define IDB_BITMAP241 342
#define IDB_BITMAP242 343
#define IDB_BITMAP243 344
#define IDB_BITMAP244 345
#define IDB_BITMAP245 346
#define IDB_BITMAP246 347
#define IDB_BITMAP247 348
#define IDB_BITMAP248 349
#define IDB_BITMAP249 350
#define IDB_BITMAP250 351
#define IDB_BITMAP251 352
#define IDB_BITMAP252 353
#define IDB_BITMAP253 354
#define IDB_BITMAP254 355
#define IDB_BITMAP255 356
#define IDB_BITMAP256 357
#define IDB_BITMAP257 358
#define IDB_BITMAP258 359
#define IDB_BITMAP259 360
#define IDB_BITMAP260 361
#define IDB_BITMAP261 362
#define IDB_BITMAP262 363
#define IDB_BITMAP263 364
#define IDB_BITMAP264 365
#define IDB_BITMAP265 366
#define IDB_BITMAP266 367
#define IDB_BITMAP267 368
#define IDB_BITMAP268 369
#define IDB_BITMAP269 370
#define IDB_BITMAP270 371
#define IDB_BITMAP271 372
#define IDB_BITMAP272 373
#define IDB_BITMAP273 374
#define IDB_BITMAP274 375
#define IDB_BITMAP275 376
#define IDB_BITMAP276 377
#define IDB_BITMAP277 378
#define IDB_BITMAP278 379
#define IDB_BITMAP279 380
#define IDB_BITMAP280 381
#define IDB_BITMAP281 382
#define IDB_BITMAP282 383
#define IDB_BITMAP283 384
#define IDB_BITMAP284 385
#define IDB_BITMAP285 386
#define IDB_BITMAP286 387
#define IDB_BITMAP287 388
#define IDB_BITMAP288 389
#define IDB_BITMAP289 390
#define IDB_BITMAP290 391
#define IDB_BITMAP291 392
#define IDB_BITMAP292 393
#define IDB_BITMAP293 394
#define IDB_BITMAP294 395
#define IDB_BITMAP295 396
#define IDB_BITMAP296 397
#define IDB_BITMAP297 398
#define IDB_BITMAP298 399
#define IDB_BITMAP299 400
#define IDB_BITMAP300 401
#define IDB_BITMAP301 402
#define IDB_BITMAP302 403
#define IDB_BITMAP303 404
#define IDB_BITMAP304 405
#define IDB_BITMAP305 406
#define IDB_BITMAP306 407
#define IDB_BITMAP307 408
#define IDB_BITMAP308 409
#define IDB_BITMAP309 410
#define IDB_BITMAP310 411
#define IDB_BITMAP311 412
#define IDB_BITMAP312 413
#define IDB_BITMAP313 414
#define IDB_BITMAP314 415
#define IDB_BITMAP315 416
#define IDB_BITMAP316 417
#define IDB_BITMAP317 418
#define IDB_BITMAP318 419
#define IDB_BITMAP319 420
#define IDB_BITMAP320 421
#define IDB_BITMAP321 422
#define IDB_BITMAP322 423
#define IDB_BITMAP323 424
#define IDB_BITMAP324 425
#define IDB_BITMAP325 426
#define IDB_BITMAP326 427
#define IDB_BITMAP327 428
#define IDB_BITMAP328 429
#define IDB_BITMAP329 430
#define IDB_BITMAP330 431
#define IDB_BITMAP331 432
#define IDB_BITMAP332 433
#define IDB_BITMAP333 434
#define IDB_BITMAP334 435
#define IDB_BITMAP335 436
#define IDB_BITMAP336 437
#define IDB_BITMAP337 438
#define IDB_BITMAP338 439
#define IDB_BITMAP339 440
#define IDB_BITMAP340 441
#define IDB_BITMAP341 442
#define IDB_BITMAP342 443
#define IDB_BITMAP343 444
#define IDB_BITMAP344 445
#define IDB_BITMAP345 446
#define IDB_BITMAP346 447
#define IDB_BITMAP347 448
#define IDB_BITMAP348 449
#define IDB_BITMAP349 450
#define IDB_BITMAP350 451
#define IDB_BITMAP351 452
#define IDB_BITMAP352 453
#define IDB_BITMAP353 454
#define IDB_BITMAP354 455
#define IDB_BITMAP355 456
#define IDB_BITMAP356 457
#define IDB_BITMAP357 458
#define IDB_BITMAP358 459
#define IDB_BITMAP359 460
#define IDB_BITMAP360 461
#define IDB_BITMAP361 462
#define IDB_BITMAP362 463
#define IDB_BITMAP363 464
#define IDB_BITMAP364 465
#define IDB_BITMAP365 466
#define IDB_BITMAP366 467
#define IDB_BITMAP367 468
#define IDB_BITMAP368 469
#define IDB_BITMAP369 470
#define IDB_BITMAP370 471
#define IDB_BITMAP371 472
#define IDB_BITMAP372 473
#define IDB_BITMAP373 474
#define IDB_BITMAP374 475
#define IDB_BITMAP375 476
#define IDB_BITMAP376 477
#define IDB_BITMAP377 478
#define IDB_BITMAP378 479
#define IDB_BITMAP379 480
#define IDB_BITMAP380 481
#define IDB_BITMAP381 482
#define IDB_BITMAP382 483
#define IDB_BITMAP383 484
#define IDB_BITMAP384 485
#define IDB_BITMAP385 486
#define IDB_BITMAP386 487
#define IDB_BITMAP387 488
#define IDB_BITMAP388 489
#define IDB_BITMAP389 490
#define IDB_BITMAP390 491
#define IDB_BITMAP391 492
#define IDB_BITMAP392 493
#define IDB_BITMAP393 494
#define IDB_BITMAP394 495
#define IDB_BITMAP395 496
#define IDB_BITMAP396 497
#define IDB_BITMAP397 498
#define IDB_BITMAP398 499
#define IDB_BITMAP399 500
#define IDB_BITMAP400 501
#define IDB_BITMAP401 502
#define IDB_BITMAP402 503
#define IDB_BITMAP403 504
#define IDB_BITMAP404 505
#define IDB_BITMAP405 506
#define IDB_BITMAP406 507
#define IDB_BITMAP407 508
#define IDB_BITMAP408 509
#define IDB_BITMAP409 510
#define IDB_BITMAP410 511
#define IDB_BITMAP411 512
#define IDB_BITMAP412 513
#define IDB_BITMAP413 514
#define IDB_BITMAP414 515
#define IDB_BITMAP415 516
#define IDB_BITMAP416 517
#define IDB_BITMAP417 518
#define IDB_BITMAP418 519
#define IDB_BITMAP419 520
#define IDB_BITMAP420 521
#define IDB_BITMAP421 522
#define IDB_BITMAP422 523
#define IDB_BITMAP423 524
#define IDB_BITMAP424 525
#define IDB_BITMAP425 526
#define IDB_BITMAP426 527
#define IDB_BITMAP427 528
#define IDB_BITMAP428 529
#define IDB_BITMAP429 530
#define IDB_BITMAP430 531
#define IDB_BITMAP431 532
#define IDB_BITMAP432 533
#define IDB_BITMAP433 534
#define IDB_BITMAP434 535
#define IDB_BITMAP435 536
#define IDB_BITMAP436 537
#define IDB_BITMAP437 538
#define IDB_BITMAP438 539
#define IDB_BITMAP439 540
#define IDB_BITMAP440 541
#define IDB_BITMAP441 542
#define IDB_BITMAP442 543
#define IDB_BITMAP443 544
#define IDB_BITMAP444 545
#define IDB_BITMAP445 546
#define IDB_BITMAP446 547
#define IDB_BITMAP447 548
#define IDB_BITMAP448 549
#define IDB_BITMAP449 550
#define IDB_BITMAP450 551
#define IDB_BITMAP451 552
#define IDB_BITMAP452 553
#define IDB_BITMAP453 554
#define IDB_BITMAP454 555
#define IDB_BITMAP455 556
#define IDB_BITMAP456 557
#define IDB_BITMAP457 558
#define IDB_BITMAP458 559
#define IDB_BITMAP459 560
#define IDB_BITMAP460 561
#define IDB_BITMAP461 562
#define IDB_BITMAP462 563
#define IDB_BITMAP463 564
#define IDB_BITMAP464 565
#define IDB_BITMAP465 566
#define IDB_BITMAP466 567
#define IDB_BITMAP467 568
#define IDB_BITMAP468 569
#define IDB_BITMAP469 570
#define IDB_BITMAP470 571
#define IDB_BITMAP471 572
#define IDB_BITMAP472 573
#define IDB_BITMAP473 574
#define IDB_BITMAP474 575
#define IDB_BITMAP475 576
#define IDB_BITMAP476 577
#define IDB_BITMAP477 578
#define IDB_BITMAP478 579
#define IDB_BITMAP479 580
#define IDB_BITMAP480 581
#define IDB_BITMAP481 582
#define IDB_BITMAP482 583
#define IDB_BITMAP483 584
#define IDB_BITMAP484 585
#define IDB_BITMAP485 586
#define IDB_BITMAP486 587
#define IDB_BITMAP487 588
#define IDB_BITMAP488 589
#define IDB_BITMAP489 590
#define IDB_BITMAP490 591
#define IDB_BITMAP491 592
#define IDB_BITMAP492 593
#define IDB_BITMAP493 594
#define IDB_BITMAP494 595
#define IDB_BITMAP495 596
#define IDB_BITMAP496 597
#define IDB_BITMAP497 598
#define IDB_BITMAP498 599
#define IDB_BITMAP499 600
#define IDB_BITMAP500 601
#define IDB_BITMAP501 602
#define IDB_BITMAP502 603
#define IDB_BITMAP503 604
#define IDB_BITMAP504 605
#define IDB_BITMAP505 606
#define IDB_BITMAP506 607
#define IDB_BITMAP507 608
#define IDB_BITMAP508 609
#define IDB_BITMAP509 610
#define IDB_BITMAP510 611
#define IDB_BITMAP511 612
#define IDB_BITMAP512 613
#define IDB_BITMAP513 614
#define IDB_BITMAP514 615
#define IDB_BITMAP515 616
#define IDB_BITMAP516 617
#define IDB_BITMAP517 618
#define IDB_BITMAP518 619
#define IDB_BITMAP519 620
#define IDB_BITMAP520 621
#define IDB_BITMAP521 622
#define IDB_BITMAP522 623
#define IDB_BITMAP523 624
#define IDB_BITMAP524 625
#define IDB_BITMAP525 626
#define IDB_BITMAP526 627
#define IDB_BITMAP527 628
#define IDB_BITMAP528 629
#define IDB_BITMAP529 630
#define IDB_BITMAP530 631
#define IDB_BITMAP531 632
#define IDB_BITMAP532 633
#define IDB_BITMAP533 634
#define IDB_BITMAP534 635
#define IDB_BITMAP535 636
#define IDB_BITMAP536 637
#define IDB_BITMAP537 638
#define IDB_BITMAP538 639
#define IDB_BITMAP539 640
#define IDB_BITMAP540 641
#define IDB_BITMAP541 642
#define IDB_BITMAP542 643
#define IDB_BITMAP543 644
#define IDB_BITMAP544 645
#define IDB_BITMAP545 646
#define IDB_BITMAP546 647
#define IDB_BITMAP547 648
#define IDB_BITMAP548 649
#define IDB_BITMAP549 650
#define IDB_BITMAP550 651
#define IDB_BITMAP551 652
#define IDB_BITMAP552 653
#define IDB_BITMAP553 654
#define IDB_BITMAP554 655
#define IDB_BITMAP555 656
#define IDB_BITMAP556 657
#define IDB_BITMAP557 658
#define IDB_BITMAP558 659
#define IDB_BITMAP559 660
#define IDB_BITMAP560 661
#define IDB_BITMAP561 662
#define IDB_BITMAP562 663
#define IDB_BITMAP563 664
#define IDB_BITMAP564 665
#define IDB_BITMAP565 666
#define IDB_BITMAP566 667
#define IDB_BITMAP567 668
#define IDB_BITMAP568 669
#define IDB_BITMAP569 670
#define IDB_BITMAP570 671
#define IDB_BITMAP571 672
#define IDB_BITMAP572 673
#define IDB_BITMAP573 674
#define IDB_BITMAP574 675
#define IDB_BITMAP575 676
#define IDB_BITMAP576 677
#define IDB_BITMAP577 678
#define IDB_BITMAP578 679
#define IDB_BITMAP579 680
#define IDB_BITMAP580 681
#define IDB_BITMAP581 682
#define IDB_BITMAP582 683
#define IDB_BITMAP583 684
#define IDB_BITMAP584 685
#define IDB_BITMAP585 686
#define IDB_BITMAP586 687
#define IDB_BITMAP587 688
#define IDB_BITMAP588 689
#define IDB_BITMAP589 690
#define IDB_BITMAP590 691
#define IDB_BITMAP591 692
#define IDB_BITMAP592 693
#define IDB_BITMAP593 694
#define IDB_BITMAP594 695
#define IDB_BITMAP595 696
#define IDB_BITMAP596 697
#define IDB_BITMAP597 698
#define IDB_BITMAP598 699
#define IDB_BITMAP599 700
#define IDB_BITMAP600 701
#define IDB_BITMAP601 702
#define IDB_BITMAP602 703
#define IDB_BITMAP603 704
#define IDB_BITMAP604 705
#define IDB_BITMAP605 706
#define IDB_BITMAP606 707
#define IDB_BITMAP607 708
#define IDB_BITMAP608 709
#define IDB_BITMAP609 710
#define IDB_BITMAP610 711
#define IDB_BITMAP611 712
#define IDB_BITMAP612 713
#define IDB_BITMAP613 714
#define IDB_BITMAP614 715
#define IDB_BITMAP615 716
#define IDB_BITMAP616 717
#define IDB_BITMAP617 718
#define IDB_BITMAP618 719
#define IDB_BITMAP619 720
#define IDB_BITMAP620 721
#define IDB_BITMAP621 722
#define IDB_BITMAP622 723
#define IDB_BITMAP623 724
#define IDB_BITMAP624 725
#define IDB_BITMAP625 726
#define IDB_BITMAP626 727
#define IDB_BITMAP627 728
#define IDB_BITMAP628 729
#define IDB_BITMAP629 730
#define IDB_BITMAP630 731
#define IDB_BITMAP631 732
#define IDB_BITMAP632 733
#define IDB_BITMAP633 734
#define IDB_BITMAP634 735
#define IDB_BITMAP635 736
#define IDB_BITMAP636 737
#define IDB_BITMAP637 738
#define IDB_BITMAP638 739
#define IDB_BITMAP639 740
#define IDB_BITMAP640 741
#define IDB_BITMAP641 742
#define IDB_BITMAP642 743
#define IDB_BITMAP643 744
#define IDB_BITMAP644 745
#define IDB_BITMAP645 746
#define IDB_BITMAP646 747
#define IDB_BITMAP647 748
#define IDB_BITMAP648 749
#define IDB_BITMAP649 750
#define IDB_BITMAP650 751
#define IDB_BITMAP651 752
#define IDB_BITMAP652 753
#define IDB_BITMAP653 754
#define IDB_BITMAP654 755
#define IDB_BITMAP655 756
#define IDB_BITMAP656 757
#define IDB_BITMAP657 758
#define IDB_BITMAP658 759
#define IDB_BITMAP659 760
#define IDB_BITMAP660 761
#define IDB_BITMAP661 762
#define IDB_BITMAP662 763
#define IDB_BITMAP663 764
#define IDB_BITMAP664 765
#define IDB_BITMAP665 766
#define IDB_BITMAP666 767
#define IDB_BITMAP667 768
#define IDB_BITMAP668 769
#define IDB_BITMAP669 770
#define IDB_BITMAP670 771
#define IDB_BITMAP671 772
#define IDB_BITMAP672 773
#define IDB_BITMAP673 774
#define IDB_BITMAP674 775
#define IDB_BITMAP675 776
#define IDB_BITMAP676 777
#define IDB_BITMAP677 778
#define IDB_BITMAP678 779
#define IDB_BITMAP679 780
#define IDB_BITMAP680 781
#define IDB_BITMAP681 782
#define IDB_BITMAP682 783
#define IDB_BITMAP683 784
#define IDB_BITMAP684 785
#define IDB_BITMAP685 786
#define IDB_BITMAP686 787
#define IDB_BITMAP687 788
#define IDB_BITMAP688 789
#define IDB_BITMAP689 790
#define IDB_BITMAP690 791
#define IDB_BITMAP691 792
#define IDB_BITMAP692 793
#define IDB_BITMAP693 794
#define IDB_BITMAP694 795
#define IDB_BITMAP695 796
#define IDB_BITMAP696 797
#define IDB_BITMAP697 798
#define IDB_BITMAP698 799
#define IDB_BITMAP699 800
#define IDB_BITMAP700 801
#define IDB_BITMAP701 802
#define IDB_BITMAP702 803
#define IDB_BITMAP703 804
#define IDB_BITMAP704 805
#define IDB_BITMAP705 806
#define IDB_BITMAP706 807
#define IDB_BITMAP707 808
#define IDB_BITMAP708 809
#define IDB_BITMAP709 810
#define IDB_BITMAP710 811
#define IDB_BITMAP711 812
#define IDB_BITMAP712 813
#define IDB_BITMAP713 814
#define IDB_BITMAP714 815
#define IDB_BITMAP715 816
#define IDB_BITMAP716 817
#define IDB_BITMAP717 818
#define IDB_BITMAP718 819
#define IDB_BITMAP719 820
#define IDB_BITMAP720 821
#define IDB_BITMAP721 822
#define IDB_BITMAP722 823
#define IDB_BITMAP723 824
#define IDB_BITMAP724 825
#define IDB_BITMAP725 826
#define IDB_BITMAP726 827
#define IDB_BITMAP727 828
#define IDB_BITMAP728 829
#define IDB_BITMAP729 830
#define IDB_BITMAP730 831
#define IDB_BITMAP731 832
#define IDB_BITMAP732 833
#define IDB_BITMAP733 834
#define IDB_BITMAP734 835
#define IDB_BITMAP735 836
#define IDB_BITMAP736 837
#define IDB_BITMAP737 838
#define IDB_BITMAP738 839
#define IDB_BITMAP739 840
#define IDB_BITMAP740 841
#define IDB_BITMAP741 842
#define IDB_BITMAP742 843
#define IDB_BITMAP743 844
#define IDB_BITMAP744 845
#define IDB_BITMAP745 846
#define IDB_BITMAP746 847
#define IDB_BITMAP747 848
#define IDB_BITMAP748 849
#define IDB_BITMAP749 850
#define IDB_BITMAP750 851
#define IDB_BITMAP751 852
#define IDB_BITMAP752 853
#define IDB_BITMAP753 854
#define IDB_BITMAP754 855
#define IDB_BITMAP755 856
#define IDB_BITMAP756 857
#define IDB_BITMAP757 858
#define IDB_BITMAP758 859
#define IDB_BITMAP759 860
#define IDB_BITMAP760 861
#define IDB_BITMAP761 862
#define IDB_BITMAP762 863
#define IDB_BITMAP763 864
#define IDB_BITMAP764 865
#define IDB_BITMAP765 866
#define IDB_BITMAP766 867
#define IDB_BITMAP767 868
#define IDB_BITMAP768 869
#define IDB_BITMAP769 870
#define IDB_BITMAP770 871
#define IDB_BITMAP771 872
#define IDB_BITMAP772 873
#define IDB_BITMAP773 874
#define IDB_BITMAP774 875
#define IDB_BITMAP775 876
#define IDB_BITMAP776 877
#define IDB_BITMAP777 878
#define IDB_BITMAP778 879
#define IDB_BITMAP779 880
#define IDB_BITMAP780 881
#define IDB_BITMAP781 882
#define IDB_BITMAP782 883
#define IDB_BITMAP783 884
#define IDB_BITMAP784 885
#define IDB_BITMAP785 886
#define IDB_BITMAP786 887
#define IDB_BITMAP787 888
#define IDB_BITMAP788 889
#define IDB_BITMAP789 890
#define IDB_BITMAP790 891
#define IDB_BITMAP791 892
#define IDB_BITMAP792 893
#define IDB_BITMAP793 894
#define IDB_BITMAP794 895
#define IDB_BITMAP795 896
#define IDB_BITMAP796 897
#define IDB_BITMAP797 898
#define IDB_BITMAP798 899
#define IDB_BITMAP799 900
#define IDB_BITMAP800 901
#define IDB_BITMAP801 902
#define IDB_BITMAP802 903
#define IDB_BITMAP803 904
#define IDB_BITMAP804 905
#define IDB_BITMAP805 906
#define IDB_BITMAP806 907
#define IDB_BITMAP807 908
#define IDB_BITMAP808 909
#define IDB_BITMAP809 910
#define IDB_BITMAP810 911
#define IDB_BITMAP811 912
#define IDB_BITMAP812 913
#define IDB_BITMAP813 914
#define IDB_BITMAP814 915
#define IDB_BITMAP815 916
#define IDB_BITMAP816 917
#define IDB_BITMAP817 918
#define IDB_BITMAP818 919
#define IDB_BITMAP819 920
#define IDB_BITMAP820 921
#define IDB_BITMAP821 922
#define IDB_BITMAP822 923
#define IDB_BITMAP823 924
#define IDB_BITMAP824 925
#define IDB_BITMAP825 926
#define IDB_BITMAP826 927
#define IDB_BITMAP827 928
#define IDB_BITMAP828 929
#define IDB_BITMAP829 930
#define IDB_BITMAP830 931
#define IDB_BITMAP831 932
#define IDB_BITMAP832 933
#define IDB_BITMAP833 934
#define IDB_BITMAP834 935
#define IDB_BITMAP835 936
#define IDB_BITMAP836 937
#define IDB_BITMAP837 938
#define IDB_BITMAP838 939
#define IDB_BITMAP839 940
#define IDB_BITMAP840 941
#define IDB_BITMAP841 942
#define IDB_BITMAP842 943
#define IDB_BITMAP843 944
#define IDB_BITMAP844 945
#define IDB_BITMAP845 946
#define IDB_BITMAP846 947
#define IDB_BITMAP847 948
#define IDB_BITMAP848 949
#define IDB_BITMAP849 950
#define IDB_BITMAP850 951
#define IDB_BITMAP851 952
#define IDB_BITMAP852 953
#define IDB_BITMAP853 954
#define IDB_BITMAP854 955
#define IDB_BITMAP855 956
#define IDB_BITMAP856 957
#define IDB_BITMAP857 958
#define IDB_BITMAP858 959
#define IDB_BITMAP859 960
#define IDB_BITMAP860 961
#define IDB_BITMAP861 962
#define IDB_BITMAP862 963
#define IDB_BITMAP863 964
#define IDB_BITMAP864 965
#define IDB_BITMAP865 966
#define IDB_BITMAP866 967
#define IDB_BITMAP867 968
#define IDB_BITMAP868 969
#define IDB_BITMAP869 970
#define IDB_BITMAP870 971
#define IDB_BITMAP871 972
#define IDB_BITMAP872 973
#define IDB_BITMAP873 974
#define IDB_BITMAP874 975
#define IDB_BITMAP875 976
#define IDB_BITMAP876 977
#define IDB_BITMAP877 978
#define IDB_BITMAP878 979
#define IDB_BITMAP879 980
#define IDB_BITMAP880 981
#define IDB_BITMAP881 982
#define IDB_BITMAP882 983
#define IDB_BITMAP883 984
#define IDB_BITMAP884 985
#define IDB_BITMAP885 986
#define IDB_BITMAP886 987
#define IDB_BITMAP887 988
#define IDB_BITMAP888 989
#define IDB_BITMAP889 990
#define IDB_BITMAP890 991
#define IDB_BITMAP891 992
#define IDB_BITMAP892 993
#define IDB_BITMAP893 994
#define IDB_BITMAP894 995
#define IDB_BITMAP895 996
#define IDB_BITMAP896 997
#define IDB_BITMAP897 998
#define IDB_BITMAP898 999
#define IDB_BITMAP899 1000

```

```

* 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 DDLIB
    preg->edb_protocol = unspecified;
    size = sizeof(szTmp);
    if ( RegQueryValueEx(hkey, "DB_Protocol", 0,
&type, (BYTE *)&szTmp, &size) == ERROR_SUCCESS )
    {
        if ( !strcmp(szTmp,
szDBNames[ODBC]) )
            preg->edb_protocol =
ODBC;
        else if ( !strcmp(szTmp,
szDBNames[DLIB]) )
            preg->edb_protocol =
DLIB;
    }
    preg->etxnmon = None;
    // determine txn monitor to use; may be
    either TUXEDO, or blank
    size = sizeof(szTmp);
    if ( RegQueryValueEx(hkey, "TxnMonitor", 0,
&type, (BYTE *)&szTmp, &size) == ERROR_SUCCESS )
    {
        if ( !strcmp(szTmp,
szTxnMonNames[TUXEDO]) )
            preg->etxnmon = TUXEDO;
        else if ( !strcmp(szTmp,
szTxnMonNames[ENCINA]) )
            preg->etxnmon = ENCINA;
        else if ( !strcmp(szTmp,
szTxnMonNames[COM]) )
            preg->etxnmon = COM;
    }
    preg->bcom_singlepool = FALSE;
    size = sizeof(szTmp);
    if ( RegQueryValueEx(hkey, "COM_SinglePool",
0, &type, (BYTE *)&szTmp, &size) == ERROR_SUCCESS )

```

common/src/ReadRegistry.cpp

```

/* FILE: READREGISTRY.CPP
* Kit Ver. 4.20.000
* Microsoft, 1999 All Rights Reserved
* Copyright
* not yet audited
*/

```

```

{
    if ( !strcmp(szTmp, "YES") )
        preg->bConn_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->szSPrefix );
    if ( RegQueryValueEx(hkey, "L'Spprefix", 0,
        &type, (BYTE *)&preg->szSPrefix, &size) !=
        ERROR_SUCCESS )
        preg->szSPrefix[0] = L'\0';

    RegCloseKey(hkey);

    return FALSE;
}

```

common/src/ReadRegistry.h

```

/*
 * FILE: ReadRegistry.h
 * Kit ver. 4.20.000 Microsoft TPC-C
 * Microsoft, 1999 Copyright
 *
 * 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 bConn_SinglePool;
    DWORD dwMaxConnections;
    DWORD dwMaxPendingDeliveries;
    char szPath[128];
    char szDbserver[32];
    char szDbname[32];
    char szDbuser[32];
    wchar_t szSPrefix[32]; // tpc_c.odb.c.d11
    stored procedures prefix
} TPCCREGISTRYDATA, *PTPCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *preg
);

```

common/src/error.h

```

/*
 * FILE: ERROR.H
 * Kit ver. 4.20.000 Microsoft TPC-C
 * Microsoft, 1999 Copyright
 *
 * All Rights Reserved
 *
 * 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
 * needed to be declared virtual
 */
#pragma once
#ifdef _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 _SERRMSG
{
    int iError;
    char szMsg[256];
} SERRMSG;

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

#define ERR_TYPE_LOGIC -1
#define ERR_SUCCESS 0
#define ERR_BAD_ITEM_ID 1
#define ERR_TYPE_DELIVERY_POST 2
#define ERR_TYPE_DELIVERY_POST_FAILED 3
#define ERR_TYPE_WEBDLL 3
#define ERR_TYPE_SQL 4
#define ERR_TYPE_SERVER_GENERATED_ERROR 4
#define ERR_TYPE_DBLIB 5
#define ERR_TYPE_ODBC 6
#define ERR_TYPE_SOCKET 7
#define ERR_TYPE_COMMUNICATION_SOCKET_CLIENT RTE
only
#define ERR_TYPE_DEADLOCK 8
#define ERR_TYPE_DBLIB_AND_ODBC_ONLY_DEADLOCK_CONDITION 8
#define ERR_TYPE_COM 9
#define ERR_TYPE_FROM_COM_CALL 9
#define ERR_TYPE_TUXEDO 10
//tuxedo error

```

```

11 #define ERR_TYPE_OS
12 //operating system error
13 #define ERR_TYPE_MEMORY
14 //memory allocation error
15 #define ERR_TYPE_TPCC_ODBC
16 //error from tpcc odbc txn module
17 #define ERR_TYPE_DELISRV
18 //error from tpcc dblib txn module
19 #define ERR_TYPE_TXNLOG
20 //delivery server error
21 #define ERR_TYPE_TXNLOG
22 //txn log error
23 #define ERR_TYPE_BCONN
24 //Benchmark connection Class
25 #define ERR_TYPE_TPCC_CONN
26 //Benchmark connection Class
27 #define ERR_TYPE_ENCINA
28 //Encina error
29 #define ERR_TYPE_COMPONENT
30 //error from COM component
31 #define ERR_TYPE_RTE
32 //Benchmark rte
33 #define ERR_TYPE_AUTOMATION
34 //Benchmark automation errors
35 #define ERR_TYPE_DRIVER
36 //Driver engine errors
37 #define ERR_TYPE_RTE_BASE
38 //Framework errors
39 #define ERR_BUF_OVERFLOW
40 //Buffer overflow during receive
41 //HTTP/SOAP dll generated error
42 #define ERR_TYPE_TPCW_CONN
43 //Benchmark connection Class
44 #define ERR_TYPE_TPCW_HTML
45 //error from Tpcwhtml dll
46 #define ERR_TYPE_TPCW_USER
47 //error from TPC-W user class
48 #define ERR_TYPE_TPCW_ENG_BASE
49 #define ERR_TYPE_TPCW_ENG_OS
50 #define ERR_TYPE_TPCW_ENG_34
51 #define ERR_TYPE_HTML_RESP
52 #define ERR_TYPE_TPCW_ODBC
53 #define ERR_TYPE_SCHANNEL
54 #define ERR_TYPE_THINK_LIST
55

```

```

#define ERR_INS_MEMORY
//Insufficient memory to continue."
#define ERR_UNKNOWN
//Unknown error."
#define ERR_MSG_BUF_SIZE 512
#define INV_ERROR_CODE
//Insufficient Buffer
//size to receive HTML pages."

class CBaseErr
{
public:
    CBaseErr(LPCTSTR szLoc = NULL)
    {
        m_idmsg = szLoc;
        //take the error code immediately
        //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;
        m_szLoc = new
        char[strlen(szLoc)+1/*m_szLoc_size*/];
        strcpy(m_szLoc, szLoc);
        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
    {
        none = 0,
        send,
        socket,
        bind,
        connect,
        listen,
        host,
        ertcv,
        getHostByName,
        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,
ewrapViewOfFile,
ecreateFileMapping,
ecreateFileSecurityDescriptor,
ecreateSecurityDescriptor,
ecreateNamedPipe,
econnectNamedPipe,
ewaitForSingleObject,
eregOpenKeyEX,
eregQueryValueEX = 20,
ebeginThread,
eregEnumValue,
eregSetValueEX,
eregCreateKeyEX,
ewaitForMultipleObjects,
eregisterClasses,
ecreateWindow,
ereleaseSemaphore,
ereleaseSemaphore,
eread,
ereadFile,
ewrapFile,
esetFilePointer,
enew,
ecloseHandle,
};

eAction, LPCTSTR szLocation);
CSystemErr(Action
eAction, LPCTSTR szLocation);
CSystemErr(int iError,
LPCTSTR szLocation);
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
 *
 * 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 ORDER_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 24
 * #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 16
 * #define PARTS_LEN 12
 * #define PART_NAME_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.
#ifdef SQLTYPES
typedef struct
{
/* SQLSMALLINT */
short year;
/* unsigned short */
SQLSMALLINT * month;
/* unsigned short */
SQLSMALLINT * day;
/* unsigned short */
SQLSMALLINT * hour;
/* unsigned short */
SQLSMALLINT * minute;
/* unsigned short */
SQLSMALLINT * second;
/* unsigned long */
SQLINTEGER * fraction;
} TIMESTAMP_STRUCT;
#endif

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

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

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

typedef struct
{
// input params

```



```

virtual pDeliveryData
BuffAddr_Delivery() = 0;
virtual pStock_LevelData
BuffAddr_StockLevel() = 0;
virtual pOrder_StatusData
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
 * Copyright
 *
 * Microsoft, 2002 All Rights Reserved
 *
 *
 * 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
 * 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>
#define ICECAP
#include <icapexp.h>
#endif
// need to declare functions for export
#define DllDecl __declspec( dllexport )
#include "...\common\src\error.h"

```

Itaitec Philco TPC-C FULL DISCLOSURE REPORT
© 2003 Itaitec Philco, all rights reserved

```

#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 // how many retries on deadlock
static long iMaxRetries = 10;
// number of
current dblib connections
static int iConnectionCount = 0;
const int iErrObjExpProvider = 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);
//
initialize dblib break;
//
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
*/

```

56

```

* message state int
* message severity int
* message description *msgtext char
printable
* RETURNS: int continue if
error is SQLETIME else INT_CANCEL action
* cancel operation INT_CANCEL
* COMMENTS: This function also sets the dead
lock dbproc variable if necessary.
*/
// typedef INT (SQLAPI *DBMSGHANDLE_PROC)(PDBPROCESS,
DBINT, INT, LPCSTR, LPCSTR, LPCSTR, DBUSMALLINT);
int msg_handler(DBPROCESS *dbproc, DBINT msgno, int
msgstate, int severity, LPCSTR procname, DBUSMALLINT line)
{
CTPCC_DBLIB
*pConn;
assert(dbproc != NULL);
pConn = (CTPCC_DBLIB*)dbgetuserdata(dbproc);
if (pConn != NULL)
{
pConn->SetSqlError( msgno,
msgstate, severity, msgtext );
return 0;
}
}

/* FUNCTION: void UtilStrCpy(char * pdest, char * pSrc,
int n)
* PURPOSE: This function copies n characters
from string pSrc to pdest and places a
null character at the end
of the destination string.
* ARGUMENTS: char destination string pointer
*pdest char
* pSrc source string pointer
int n
number of characters to copy
* RETURNS: None
* COMMENTS: Unlike strcpy this function
ensures that the result string is
always null
terminated.
*/
inline static void UtilStrCpy(char * pdest, const BYTE
* pSrc, int n)

```

November, 2003

```

{
    strncpy(pdest, (char *)psrc, n);
    pdest[n] = '\0';
    return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
*/
char* CTPCC_DBLIB_ERR::ErrorText(void)
{
    int i;
    static SERRMSG errorMsgs[] =
    {
        { ERR_WRONG_SP_VERSION,
        "wrong version of stored procs on database
        server"
        },
        { ERR_INVALID_CUST,
        "Invalid Customer id.name."
        },
        { ERR_NO_SUCH_ORDER,
        "No orders found for customer."
        },
        { ERR_RETRIED_TRANS,
        "Retries before transaction succeeded."
        },
        { 0, ""
        }
    };

    number--;
    static char szNotFound[] = "Unknown error
    number";
    for(i=0; errorMsgs[i].szMsg[0]; i++)
    {
        if ( m_erno == errorMsgs[i].iError
        )
        {
            break;
        }
        if ( ierrorMsgs[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 (

```

```

server
LPCSTR szServer, // name of SQL
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; *pdata;
    const BYTE
    // initialization
    m_dbproc = NULL;
    m_DblLibErr = (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::ebbSetMaxProcs);
        }
        // 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::ebbProcHandler);
        }
        if (dbprocsmsghandle(login, msg_handler) ==
        NULL)
        {
            ThrowError(CDBLIBERR::ebbProcHandler);
        }
        DBSETUSER(login, szUser);
        DBSETPWD(login, szPassword);
        DBSETHOST(login, szHost);
        DBSETPACKET(login, (unsigned
        short)DEFCLPACSIZE);
        // use dblib ver 6.0 client behavior
        // set time to wait for login
        if (dbsetlogintime(60) == FAIL)
            ThrowError(CDBLIBERR::ebbSet);
        // set time to wait for statement execution
        if (dbsettime(180) == FAIL)
            ThrowError(CDBLIBERR::ebbSet);
        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
dbsetuserdata(m_dbproc, (LPVOID)this);
// Use 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
dbpcinit(m_dbproc, "tpcc_version", 0);
if (dbrpcexec(m_dbproc) == FAIL)
    ThrowError(CDBLIBERR::edbrpcExec);
if (dbresults(m_dbproc) != SUCCEED)
    ThrowError(CDBLIBERR::edbrResults);
if (dbnextrow(m_dbproc) != REG_ROW)
    ThrowError(CDBLIBERR::edbnxtRow);
char szSrvVersion[16];
pdata=dbdata(m_dbproc, 1);
if (pdata)
    UtilStrCpy(szSrvVersion, pdata,
    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 )
{
    resources
    // close db connection and deallocate
    dbclose(m_dbproc);
    InterlockedDecrement( &iconnectionCount );
    if (m_DblLibErr != NULL)
        delete m_DblLibErr;
    if (m_SqlErr != NULL)
        delete m_SqlErr;
}
void CTPCC_DBLIB::SetDblLibError(int severity, int
dberr, int oserr, LPCSTR dberrstr, LPCSTR oserrstr)
{
    delete m_DblLibErr;
    m_DblLibErr = 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::setError( 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 Delib error.
        {
            {
                CSQLERR *psqlErr;
                psqlErr = m_SqlErr; // Clear our
                m_SqlErr = NULL; // Clear our
                pointer to instance; catch handler will delete
                throw psqlErr;
            }
        }
        CDBLIBERR *pdblLibErr;
        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.
            {
                pdblLibErr = new CDBLIBERR(eAction);
            }
            else
            {
                pdblLibErr = m_DbLibLibErr;
                pdblLibErr->m_eAction = eAction;
                m_DbLibLibErr = NULL; //
                clear our pointer to instance; catch handler will
                delete
            }
    }

```

```

    }
    throw pdblLibErr;
    // 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 RETCODE rc;
        int iRowsRead = 0;
        while (TRUE)
        {
            rc = dbnextrow(m_dbproc);
            if (rc == NO_MORE_ROWS)
                break;
            if (rc == FAIL)
            {
                if (iExpectedCount >= 0)
                    break;
            }
            iRowsRead++;
        }
        ThrowError(CDBLIBERR::eBnNextRow);
        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 RETCODE rc;
        int iResultsRead = 0;
        while (TRUE)
        {
            rc = dbresults(m_dbproc);
            if (rc == NO_MORE_RESULTS)
                break;
            if (rc == FAIL)
            {
                if (iExpectedCount >= 0)
                    ThrowError(CDBLIBERR::eBResults);
                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
            {
                dbrcinit(m_dbproc,
                    "tpcc_stocklevel", 0);
                dbrcparam(m_dbproc,
                    NULL, 0, SQLINT4, -1, -1, (BYTE *)
                    &m_txn.StockLevel.w_id); // @w_id int
                dbrcparam(m_dbproc,
                    NULL, 0, SQLINT1, -1, -1, (BYTE *)
                    &m_txn.StockLevel.d_id); // @d_id
                dbrcparam(m_dbproc,
                    NULL, 0, SQLINT2, -1, -1, (BYTE *)
                    &m_txn.StockLevel.threshold); // @threshold smallint
                if (dbrcexec(m_dbproc)
                    == FAIL)
                    ThrowError(CDBLIBERR::eBrcpExec);
                if (dbresults(m_dbproc)
                    != SUCCEED)
                    ThrowError(CDBLIBERR::eBResults);
                if (dbnextrow(m_dbproc)
                    != REG_ROW)
                    ThrowError(CDBLIBERR::eBnNextRow);
                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 ==
                    >m_msgtext, serrTimeoutExpired) != NULL)) &&
                    (iMaxRetries))
                {
                    // hit
                    deadlock; backoff for increasingly longer period
                    delete e;
                }
            }
        }
    }

```



```

daterec.year;
m_txn.NewOrder.o_entry_d.year =
daterec.month;
m_txn.NewOrder.o_entry_d.month =
daterec.day;
m_txn.NewOrder.o_entry_d.day =
daterec.hour;
m_txn.NewOrder.o_entry_d.hour =
daterec.minute;
m_txn.NewOrder.o_entry_d.minute =
daterec.second;
m_txn.NewOrder.o_entry_d.second =
{
if
commit_flag =
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 ==
strchr(e-
>m_msgtext, 'ERRtimeoutExpired') != NULL)) &&
(++iTryCount <=
iMaxRetries))
{
// hit
deadlock; backoff for increasingly longer period
delete e;
Sleep(10 *
iTryCount);
}
else
throw;
}
// while (TRUE)
{
if (iTryCount)
throw new
CTPC_DBLIB_ERR(CTPC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

```

```

void CTPCC_DBLIB::Payment()
{
DBDATETIME
DBDATEREc daterec;
int
const BYTE
iTryCount = 0;
ResetError();
while (TRUE)
{
try
{
"tpcc_payment", 0);
dbrcpccparam(m_dbproc,
&m_txn.Payment.w_id);
dbrcpccparam(m_dbproc,
&m_txn.Payment.w_id);
dbrcpccparam(m_dbproc,
&m_txn.Payment.c_w_id);
dbrcpccparam(m_dbproc,
&m_txn.Payment.h_amount);
dbrcpccparam(m_dbproc,
&m_txn.Payment.d_id);
dbrcpccparam(m_dbproc,
&m_txn.Payment.c_d_id);
dbrcpccparam(m_dbproc,
&m_txn.Payment.c_id);
// if customer id is
zero, then payment is by name
if (m_txn.Payment.c_id ==
0)
dbrcpccparam(m_dbproc, NULL, 0, SQLCHAR, -1,
strlen(m_txn.Payment.c_last), (unsigned char
*)m_txn.Payment.c_last);
== FAIL)
ThrowError(CDBLIBERR::eDbrcpccExec);
if (dbresults(m_dbproc)
!= SUCCEEDED)
ThrowError(CDBLIBERR::eDbResults);
if (dbnextrow(m_dbproc)
!= REG_ROW)
ThrowError(CDBLIBERR::eDbNextrow);
if (dbnumcols(m_dbproc)
!= 27)
ThrowError(CDBLIBERR::eWrongNumCols);
if
(pdata=dbdata(m_dbproc, 1))
m_txn.Payment.c_id = *((DBINT *) pdata);
if
(pdata=dbdata(m_dbproc, 2))

```

```

utilStrCpy(m_txn.Payment.c_last, pdata,
dbdatalen(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,
dbdatalen(m_dbproc, 4));
if
(pdata=dbdata(m_dbproc, 5))
utilStrCpy(m_txn.Payment.w_street_2, pdata,
dbdatalen(m_dbproc, 5));
if
(pdata=dbdata(m_dbproc, 6))
utilStrCpy(m_txn.Payment.w_city, pdata,
dbdatalen(m_dbproc, 6));
if
(pdata=dbdata(m_dbproc, 7))
utilStrCpy(m_txn.Payment.w_state, pdata,
dbdatalen(m_dbproc, 7));
if
(pdata=dbdata(m_dbproc, 8))
utilStrCpy(m_txn.Payment.w_zip, pdata,
dbdatalen(m_dbproc, 8));
if
(pdata=dbdata(m_dbproc, 9))
utilStrCpy(m_txn.Payment.d_street_1, pdata,
dbdatalen(m_dbproc, 9));
if
(pdata=dbdata(m_dbproc, 10))
utilStrCpy(m_txn.Payment.d_street_2, pdata,
dbdatalen(m_dbproc, 10));
if
(pdata=dbdata(m_dbproc, 11))
utilStrCpy(m_txn.Payment.d_city, pdata,
dbdatalen(m_dbproc, 11));
if
(pdata=dbdata(m_dbproc, 12))
utilStrCpy(m_txn.Payment.d_state, pdata,
dbdatalen(m_dbproc, 12));
if
(pdata=dbdata(m_dbproc, 13))

```

```

        UtilIstrCpy(m_txn.Payment.d_zip, pdata,
dbdatlen(m_dbproc, 13));
        if
(pdata=dbdata(m_dbproc, 14))
        UtilIstrCpy(m_txn.Payment.c_first, pdata,
dbdatlen(m_dbproc, 14));
        if
(pdata=dbdata(m_dbproc, 15))
        UtilIstrCpy(m_txn.Payment.c_middle, pdata,
dbdatlen(m_dbproc, 15));
        if
(pdata=dbdata(m_dbproc, 16))
        UtilIstrCpy(m_txn.Payment.c_street_1, pdata,
dbdatlen(m_dbproc, 16));
        if
(pdata=dbdata(m_dbproc, 17))
        UtilIstrCpy(m_txn.Payment.c_street_2, pdata,
dbdatlen(m_dbproc, 17));
        if
(pdata=dbdata(m_dbproc, 18))
        UtilIstrCpy(m_txn.Payment.c_city, pdata,
dbdatlen(m_dbproc, 18));
        if
(pdata=dbdata(m_dbproc, 19))
        UtilIstrCpy(m_txn.Payment.c_state, pdata,
dbdatlen(m_dbproc, 19));
        if
(pdata=dbdata(m_dbproc, 20))
        UtilIstrCpy(m_txn.Payment.c_zip, pdata,
dbdatlen(m_dbproc, 20));
        if
(pdata=dbdata(m_dbproc, 21))
        UtilIstrCpy(m_txn.Payment.c_phone, pdata,
dbdatlen(m_dbproc, 21));
        if
(pdata=dbdata(m_dbproc, 22))
        {
            datetime =
*((DBDATEIME *) pdata);
            datecrack(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))
        UtilIstrCpy(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))
            UtilIstrCpy(m_txn.Payment.c_data, pdata,
dbdatlen(m_dbproc, 27));
            DiscardNextRows(0);
            if (m_txn.Payment.c_id ==
0)
            {
                throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
            }
            m_txn.Payment.exec_status_code = eOk;
            return;
        }
        catch (CSQLERR *e)
        {
            if ((e->m_msgno == 1205
            (e->m_msgno ==
iErrorlebbpProvider &&
>m_msgtext, serrTimeoutExpired) != NULL) &&
            (m_maxRetries))
            {
                // hit
                deadlock; backoff for increasingly longer period
                delete e;
                Sleep(10 *
iTryCount);
            }
            else
            {
                throw;
            }
        }
        if (iTryCount)
        {
            if (iTryCount)
            {
                throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
            }
        }
        void CTPCC_DBLIB::orderStatus()
        {
            int
            DBDATEIME          datetime;
            DBDATEREC daterec;

```

```

        int
        RETCODE          rc;
        const BYTE
        *pdata;
        ResetError();
        while (TRUE)
        {
            try
            {
                dbrcpinit(m_dbproc,
                "tpcc_orderstatus", 0);
                dbrcpparam(m_dbproc,
                &m_txn.orderstatus.w_id, (BYTE *)
                NULL, 0, SQLINT4, -1, -1, (BYTE *)
                NULL, 0, SQLINT1, -1, -1, (BYTE *)
                NULL, 0, SQLINT4, -1, -1, (BYTE *)
                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)
                {
                    dbrcpparam(m_dbproc, NULL, 0, SQLCHAR, -1,
                    strlen(m_txn.orderstatus.c_last), (unsigned char
                    *)&m_txn.orderstatus.c_last);
                }
                if (dbrcpexec(m_dbproc)
                == FAIL)
                {
                    ThrowError(CDBLIBERR::eDbrcpExec);
                }
                // get order lines
                if (dbresults(m_dbproc)
                != SUCCEED)
                {
                    if (m_DblLibErr
                    == 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::eBbNextRow);
                    }
                }
            }
            catch (Exception *)
            {
                iTryCount = 0;
            }
        }
    }
}

```

```

        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),
            *)&m_txn.OrderStatus.oL[i].oL_amount, 8);
        if(pdata=dbdata(m_dbproc, 5)) {
            datetime = *((DBDATEIME *) 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;
            }
            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 =
                dbdatecrack(m_dbproc, &daterec, &datetime);
                m_txn.OrderStatus.o.o_entry_d.year =
                daterec.year;
                m_txn.OrderStatus.o.o_entry_d.month =
                daterec.month;
                m_txn.OrderStatus.o.o_entry_d.day =
                daterec.day;
                m_txn.OrderStatus.o.o_entry_d.hour =
                daterec.hour;
                m_txn.OrderStatus.o.o_entry_d.minute =
                daterec.minute;
                m_txn.OrderStatus.o.o_entry_d.second =
                daterec.second;
            }
            m_txn.OrderStatus.o.o_carrier_id =
            (* (DBSMALLINT *) pdata);
            if(pdata=dbdata(m_dbproc,
            7))
                dbconvert(m_dbproc, SQLNUMERIC,
                (LPCBYTE)pdata, dbdatlen(m_dbproc, 7),
                *)&m_txn.OrderStatus.c_balance, 8);
            if(pdata=dbdata(m_dbproc,
            8))
                m_txn.OrderStatus.o.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 ==
            >m_msgtext, serrTimeoutExpired) != NULL)) &&
            !m_maxRetries))
            {
                // 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
            int
            const BYTE
            *pdata;
            i;
            iTryCount = 0;
            ResetError();
            while (TRUE)
            {
                try
                {
                    dbrcpinit(m_dbproc,
                    "tpcc_delivery", 0);
                    NULL, 0, SQLINT4, -1, -1, (BYTE *)
                    &m_txn.Delivery_w_id);
                    dbrcpccparam(m_dbproc,
                    NULL, 0, SQLINT1, -1, -1, (BYTE *)
                    &m_txn.Delivery.o_carrier_id);
                    if (dbrcpexec(m_dbproc)
                    == FAIL)
                        ThrowError(CDBLIBERR::eDbRcpExec);
                    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
||
iErrorDbbProvider &&
(e->m_msgno ==
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_DblLibErr != NULL)
            {
                delete m_DblLibErr;
                m_DblLibErr = (CDBLIBERR*)NULL;
            }
            if (m_SqlErr != NULL)
            {
                delete m_SqlErr;
                m_SqlErr = (CSQLERR*)NULL;
            }
            return;
        }
    }

```

db_dblib_dll/src/tpcc_dblib.h

Itaotec Philco TPC-C FULL DISCLOSURE REPORT
© 2003 Itaotec Philco, all rights reserved

```

/* FILE: TPCCC_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
#ifdef 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.
#ifdef D11DECL
#define D11DECL __declspec( dllexport )
#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 ErrType() {return
ERR_TYPE_SQL;};
    int ErrNum() {return m_msgno;};
    char *ErrMsg() {return
m_msgtext;};
};
class CDBLIBERR : public CBaseErr
{
public:
    enum ACTION
    {
        enone,
        eunknown,
        elogin,
        // error from dblogin
    };

```

```

        // error from dbopen,
        eDbuse,
        // error from dbuse
        eBbsqlExec,
        // error from dbsqlExec
        eDbset,
        // error from one of the dbset*
        routines
        eBbnextRow,
        // error from dbnextrow
        eWrongRowCount,
        // more or less rows returned than expected
        eWrongNumCols,
        // more or less columns returned than
        expected
        eBbresults,
        // error from dbresults
        eBbpcExec,
        // error from dbpcExec
        eBbsetMaxProcs,
        // error from dbsetMaxprocs
        eBbProcHandler,
        // error from either dbprocerrhandle or
        dbprocmsgshandle
    };
    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 ErrType() {return
ERR_TYPE_DBLIB;};
    int ErrNum() {return m_dberror;};
    char *ErrMsg() {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_erno = iErr; m_iTryCount = 0; };
CTPCC_DBLIB_ERR( int iErr, int
iTryCount ) { m_erno = iErr; m_iTryCount = iTryCount;
};
int m_erno;
int m_iTryCount;
int ErrType() {return
ERR_TYPE_TPC_DBLIB;};
int ErrNum() {return m_erno;};
char *ErrText();
private: // declare variables and private
functions here...
PDBPROCESS m_dbproc;
CDBLIBERR *m_DblLibErr;
// not allocated until needed (maybe never)
CSQLERR // 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
PAYMENT_DATA
DELIVERY_DATA
STOCK_LEVEL_DATA
ORDER_STATUS_DATA
};
m_txn;
public:
CTPCC_DBLIB(LPCSTR szServer, LPCSTR
szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR
szDatabase );
~CTPCC_DBLIB(void);
inline PNEW_ORDER_DATA
BuffAddr_NewOrder() { return
&m_txn.NewOrder;
};
inline PPAYMENT_DATA
BuffAddr_Payment() { return
&m_txn.Payment;
};

```

```

inline PDELIVERY_DATA
BuffAddr_Delivery() { return
&m_txn.Delivery;
};
inline PSTOCK_LEVEL_DATA
BuffAddr_StockLevel() { return
&m_txn.StockLevel;
};
inline PORDER_STATUS_DATA
BuffAddr_OrderStatus() { return
&m_txn.OrderStatus;
};
void NewOrder
();
void Payment
();
void Delivery
();
void StockLevel
();
void OrderStatus
();
// these are public because they
must be called from the dlib err_handler and
msg_handler
// outside of the class
void SetDbLibError(int severity,
LPCSTR dberstr, 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: TPC_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"
//tpccit transaction header contains
definitions of structures specific to TPC-C
#include "...\common\src\error.h"
#include "...\common\src\txn_base.h"
#include "tpcc_com.h"

```

```

#include "...\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "...\tpcc_com_all\src\tpcc_com_all_i.c"
// wrapper routine for class constructor
__declspec(dllexport) CTPCC_COM* CTPCC_COM_new(BOOL
bSinglePool)
{
return new CTPCC_COM(bSinglePool);
}
CTPCC_COM::CTPCC_COM(BOOL bSinglePool)
{
HRESULT hr = NULL;
long lRet = 0;
ULONG ulTmpSize = 0;
m_pTxn
= NULL;
m_pNewOrder
= NULL;
m_pPayment
= NULL;
m_pStockLevel
= NULL;
m_pOrderStatus
= NULL;
m_bSinglePool
= bSinglePool;
ulTmpSize = (ULONG) sizeof(COM_DATA);
VariantInit(&m_VTxn);
m_VTxn.Vt = VT_SAFEARRAY;
m_VTxn.parray = SafeArrayCreateVector(VT_UI1,
ulTmpSize, ulTmpSize);
if (m_VTxn.parray
throw new CCOMERR( E_FAIL );
memset((void*)m_VTxn.parray-
>pvdata, 0, ulTmpSize);
m_pTxn = (COM_DATA*)m_VTxn.parray->pvdata;
hr = CoInitializeEx(NULL,
COINIT_MULTITHREADED);
if (FAILED(hr))
{
throw new CCOMERR( hr );
}
// create components
if (m_bSinglePool)
{
hr = CoCreateInstance(CLSID_TPC_C,
IID_ITPC_C, IID_TPC_C, 0, &m_pNewOrder);
throw new CCOMERR(hr);
NULL, CLSCTX_SERVER, IID_ITPC_C, IID_ITPC_C, 0, &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_ITPC_C, (void **) &m_pNewOrder);
if (FAILED(hr))
throw new CCOMERR(hr);
hr =
CoCreateInstance(CLSID_Payment, NULL, CLSCTX_SERVER,
IID_ITPC_C, (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_bsinglPool)
    {
        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_pPayment);
        if (!m_bsinglPool)
        {
            ReleaseInterface(m_pStockLevel);
            ReleaseInterface(m_pOrderStatus);
        }
        CountInitialize();
    }

    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
 * Kit Ver. 4.20.000
 * 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 DllExport
#define DllExport __declspec( dllimport )
#endif

class CCOMERR : public CBaseErr
{
private:
    char m_szErrorText[64];

public:
    // use this interface for genuine
    COM errors
    {
        CCOMERR( HRESULT hr )
        {
            m_hr = hr;
            m_iErrorType = 0;
            m_iError = 0;
        }

        // use this interface to
        impersonate a non-COM error type
        CCOMERR( int iErrorType, int iError
        )
        {
            m_iErrorType =
            m_iError = iError;
            m_hr = S_OK;
        }

        int
        m_hr;
        int
        m_iErrorType;
        int
        m_iError;

        // A CCOMERR class can impersonate
        another class, which happens if the error
        // was not actually a COM Services
        error, but was simply transmitted back via COM.
        {
            int iErrorType;
            if (m_iErrorType == 0)
                return
            else
                return
        }

        int iErrorNum() {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;
                Delivery;
            };
            PAYMENT_DATA
            {
                Payment;
            };
            DELIVERY_DATA
            {
                Delivery;
            };
        };
    };

    STOCK_LEVEL_DATA StockLevel;
    ORDER_STATUS_DATA OrderStatus;
    VARIANT m_vTxn;
    CTPCC_COM(BOOL bSinglePool);
    ~CTPCC_COM(void);

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

    void NewOrder
    ();
    void Payment
    ();
    void StockLevel
    ();
    void OrderStatus
    ();
    void Delivery
    ();
    void Delivery
    (); // not supported

    throw new CCOMERR(E_NOTIMPL); }

    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_atl/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_UNSUPPORTED_FEATURE,
    ERR_UNKNOWN_DB_PROTOCOL
};

class COMPONENT_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,
    DWORD dwSystemErr)
    {
        m_Error = Err;
        m_szTextDetail = new
        char[strlen(szTextDetail)+1]; strcpy( m_szTextDetail,
        szTextDetail );
        m_SystemErr =
        m_szErrorText = NULL;
    };
    ~COMPONENT_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);

private:
    // helper methods
    BOOL m_bCanBePooled;
    CTPCC_BASE *m_pTxn;
    struct COM_DATA
    {
}

```

```
////////////////////////////////////////////////////
//      OrderStatus :
//      class COrderStatus :
//          public CTPCC_Common,
//          public CComCocClass<CorderStatus,
//              &CLSID_OrderStatus>
//      {
//      public:
//      DECLARE_REGISTRY_RESOURCEID(IDR_ORDERSTATUS)
//
//      BEGIN_COM_MAP(CorderStatus)
//          // COM_INTERFACE_ENTRY2(IUnknown,
//          ComObjectRootEX)
//          COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
//          COM_INTERFACE_ENTRY_CHAIN(ITPCC_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_out){ return E_NOTIMPL;}
//          HRESULT __stdcall OrdersStatus(
//              VARIANT txn_in, VARIANT* txn_out) {return
//                  E_NOTIMPL;}
//      };
////////////////////////////////////////////////////
//      CPayment
//      class CPayment :
//          public CTPCC_Common,
//          public CComCocClass<CPayment, &CLSID_Payment>
//      {
//      public:
//      DECLARE_REGISTRY_RESOURCEID(IDR_PAYMENT)
//
//      BEGIN_COM_MAP(CPayment)
//          // COM_INTERFACE_ENTRY2(IUnknown,
//          ComObjectRootEX)
//          COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
//          COM_INTERFACE_ENTRY_CHAIN(ITPCC_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_out){ return E_NOTIMPL;}
//          HRESULT __stdcall OrdersStatus(
//              VARIANT txn_in, VARIANT* txn_out) {return
//                  E_NOTIMPL;}
//      };
////////////////////////////////////////////////////
//      CStockLevel
//      class CStockLevel :
```

```

    public CTPCCommon,
    public ComCoclass<StockLevel,
    &CLSTD_StockLevel>
    {
        public:
            DECLARE_REGISTRY_RESOURCEID(IDR_STOCKLEVEL)

    BEGIN_COM_MAP(CStockLevel)
        COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(CTPCCommon)
    END_COM_MAP()

    // ITPC
    public:
        HRESULT STDMETHODCALLTYPE NewOrder(
            VARIANT txn_in, VARIANT* txn_out) {return
            E_NOTIMPL;}

        HRESULT STDMETHODCALLTYPE Payment(
            VARIANT txn_in, VARIANT* txn_out) {return
            E_NOTIMPL;}

        HRESULT STDMETHODCALLTYPE StockLevel(
            VARIANT* txn_out) {return E_NOTIMPL;}

        HRESULT STDMETHODCALLTYPE 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
    * Kit Ver. 4.20.000
    * Microsoft, 1999 Copyright
    * All Rights Reserved
    *
    *
    * 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 <atbase.h>
//You may derive a class from CComModule and use it if
//you want to override
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"

header contains definitions of //tpckit transaction
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_1.c"
#include "Methods.h"
#include ".\..\..\tpcc_com_ps\src\tpcc_com_ps_1.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);
            MAX_COMPUTERNAME_LENGTH+1;
            DWORD dwSize =
            GetComputerName(szMyComputerName, &dwSize);
            szMyComputerName[dwSize]
            = 0;
            ReadTPCCRegistrySettings( &Reg )
            if (
            COMPONENT_ERROR( ERR_MISSING_REGISTRY_ENTRIES );
            throw new
            DBLIB)
            if (Reg.edb_Protocol ==
            {
                strcpy(
                szDllName, Reg.szPath );
                strcat(
                szDllName, "tpcc_dblib.dll");
                hLibInstanceDb
                = LoadLibrary( szDllName );
                if
                (hLibInstanceDb == NULL)
                new COMPONENT_ERROR( ERR_LOADDLL_FAILED, szDllName,
                GetLastError() );
                throw
                pointer to wrapper for class constructor

                PCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
                GetProcAddress(hLibInstanceDb, "CTPCC_DBLIB_new");
                if
                (pCTPCC_DBLIB_new == NULL)
                throw
                new COMPONENT_ERROR( ERR_GETPROCADDR_FAILED, szDllName,
                GetLastError() );
                == ODBC)
                {
                    else if (Reg.edb_Protocol
                    {
                        strcpy(
                        szDllName, Reg.szPath );
                        strcat(
                        szDllName, "tpcc_odbc.dll");
                        hLibInstanceDb
                        = LoadLibrary( szDllName );
                        if
                        (hLibInstanceDb == NULL)
                        new COMPONENT_ERROR( ERR_LOADDLL_FAILED, szDllName,
                        GetLastError() );
                        throw
                        pointer to wrapper for class constructor
                    }
                }
            }
        }
    }
}

```

```

= (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb, "CTPCC_ODBC_new");
if
(pCTPCC_ODBC_new == NULL)
throw
new COMPONENT_ERROR( ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
}
else
throw new
COMPONENT_ERROR( 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);
}
////////////////////////////////////
// DllRegistersServer - Adds entries to the system
registry
STDAPI DllRegistersServer(void)
{
    // registers object, typelib and all
    interfaces in typelib
    return _Module.RegisterServer(TRUE);
}
}

```



```

pNewOrder, sizeof(NEW_ORDER_DATA));
    memcpy( &pdata->u.NewOrder,
    pdata->retval = ERR_SUCCESS;
    pdata->error = 0;
    return S_OK;
}
catch (CBaseErr *e)
{
    // check for lost database
    connection; if yes, component is toast
    ERR_TYPE_DBLIB) && ((e->ErrorType() ==
    ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
    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;
    }
    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)
{
    PAYMENT_DATA    pPayment;
    COM_DATA
    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
    ERR_TYPE_DBLIB) && ((e->ErrorType() ==
    ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
    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;
    }
    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
    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
        ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||

```

```

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)
{
    ORDER_STATUS_DATA    pOrderStatus;
    COM_DATA
    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
        ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
        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(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADER( )

```

/* 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

```

Itautec Philco TPC-C FULL DISCLOSURE REPORT
© 2003 Itautec Philco, all rights reserved

```

/* 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 "oaid.h"
#include "ocidl.h"
#include "tpcc_com_ps.h"

#ifndef __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;

#define __TPCCLIB_LIBRARY_DEFINED__
#ifdef __cplusplus
extern "C" const IID LIBID_TPCCLib;
extern "C" const CLSID CLSID_TPCC;
#endif /* __cplusplus */
class DECLSPEC_UUID("122A3128-2520-11D3-BA71-00C04FBFE088")
TPCC;
#ifdef EXTERN_C
extern "C" const CLSID CLSID_NewOrder;
#endif /* __cplusplus */
class DECLSPEC_UUID("975BAABF-84A7-11D2-BA47-00C04FBFE088")
NewOrder;
#ifdef EXTERN_C
extern "C" const CLSID CLSID_OrderStatus;
#endif /* __cplusplus */
class DECLSPEC_UUID("266836AD-A50D-11D2-BA4E-00C04FBFE088")
OrderStatus;
#ifdef EXTERN_C
extern "C" const CLSID CLSID_Payment;
#endif /* __cplusplus */
class DECLSPEC_UUID("CD02F7EF-A4FA-11D2-BA4E-00C04FBFE088")
Payment;
#ifdef EXTERN_C
extern "C" const CLSID CLSID_StockLevel;
#endif /* __cplusplus */
class DECLSPEC_UUID("2668369E-A50D-11D2-BA4E-00C04FBFE088")
StockLevel;
#endif

```

```
#endif /* __TPCLIB_LIBRARY_DEFINED__ */
/* Additional prototypes for ALL interfaces */
/* end of Additional prototypes */
#ifdef __cplusplus
}
#endif
#endif
```

tpcc_com_all/src/tpcc_com_all.idl

```

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

```

```
interface TPCP:
  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-BA71-00C04FBFE088),
    version(1,0),
    helpstring("TPC-C 1.0 Type Library")
]
library TPCCLib
{
    importlib("stdole32.tlb");
    importlib("stdole2.tlb");

    [
        uuid(122A3128-2520-11D3-BA71-00C04FBFE088),
        helpstring("All Txns Class")
    ]
    coclass TPCC
    {
        [default] interface ITPCC;
    };
}

[
    uuid(975BA8BF-84A7-11D2-BA47-00C04FBFE088),
```

Itautec Philco TPC-C FULL DISCLOSURE REPORT
© 2003 Itautec Philco, all rights reserved

```

    helpstring("NewOrder Class")
}
class NewOrder
{
    [default] interface ITPCC;
};

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

[
    uuid(CD027EF-A4FA-11D2-BA4E-
        helpstring("Payment Class"))
]
class Payment
{
    [default] interface ITPCC;
};

[
    uuid(2668369E-A50D-11D2-BA4E-
        helpstring("StockLevel Class"))
]
class 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
#ifdef _AFX_RESOURCE_DLL || defined(_AFX_TARGET_ENU)
#include "afxres.h"
#pragma code_page(1252)
#endif

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

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

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

#ifdef APSTUDIO_INVOKED
//
// #ifdef _MAC
//
// Version
//
VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGS 0x3fL
#ifdef _DEBUG
    FILEFLAGS 0x1L
#else
    FILEFLAGS 0x0L
#endif
#ifdef _DEBUG
    FILEOS 0x4L
    FILETYPE 0x2L
    FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "tpcc_com_all"
            VALUE "FileVersion", "1.0.0.1\0"
            VALUE "InternalName", "TPCCMORDER\0"
            VALUE "LegalCopyright", "Copyright 1997\0"
            VALUE "OriginalFilename",
                "tpcc_com_all.dll\0"
            VALUE "ProductVersion", "tpcc_com_all"
            VALUE "ProductName", "1.0.0.1\0"
            VALUE "OLES32Register", "\0"
        END
    END
END
#endif
#endif

```

```
END
BLOCK "VarFileInfo"
BEGIN
    VALUE "Translation", 0x409, 1200
END

#endif // !_MAC

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

IDR_TPCC                REGISTRY DISCARDABLE
IDR_NEWORDER            REGISTRY DISCARDABLE
IDR_COM_LNO_rgs         "tpcc_com_lno_rgs"
IDR_ORDERSTATUS         "tpcc_com_lno_rgs"
IDR_COM_OS_rgs          "tpcc_com_os_rgs"
IDR_PAYMENT             REGISTRY DISCARDABLE
IDR_STOCKLEVEL          "tpcc_com_pay_rgs"
IDR_COM_S1_rgs          "tpcc_com_s1_rgs"

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

STRINGTABLE DISCARDABLE
BEGIN
    IDS_PROJNAME         "tpcc_com_all"
END

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

////////////////////////////////////
//
//  APSTUDIO_INVOKED
//
////////////////////////////////////

#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'
    {
        BA71-00C04FBFE088, CLSID = s '{122A3128-2520-11D3-
    }
    TPCC.AllTxns = s 'TPCC Class'
}
```

Itaotec Philco TPC-C FULL DISCLOSURE REPORT
© 2003 Itaotec Philco, all rights reserved

```
CurVer = s 'TPCC.AllTxns.1'
}
NORemove CLSID
{
    Forceremove {122A3128-2520-11D3-
BA71-00C04FBFE088} = s 'TPCC Class'
    {
        Progid = s
        VersionIndependentProgid
        InprocServer32 = s
        {
            va1
        }
    }
    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(novtable)
        __declspec(uuid()), __declspec(selectany),
        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
            #endif
            #include <guiddef.h>
            #ifdef _cplusplus
                extern "C" {
            #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 x1;
    unsigned short x2;
    unsigned short x3;
    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) \
MIDL_DEFINE_GUID(IID, \
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0 \
XC0,0x4F,0xBF,0xE0,0x88);

MIDL_DEFINE_GUID(CLSID, \
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0XC0 \
,0x4F,0xBF,0xE0,0x88);

MIDL_DEFINE_GUID(CLSID, \
CLSID_NewOrder,0x9738A8BF,0x84A7,0x11D2,0xBA,0x47,0x00, \
0XC0,0x4F,0xBF,0xE0,0x88);

MIDL_DEFINE_GUID(CLSID, \
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x \
00,0XC0,0x4F,0xBF,0xE0,0x88);

MIDL_DEFINE_GUID(CLSID, \
CLSID_Payment,0xC0D2F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00, \
XC0,0x4F,0xBF,0xE0,0x88);

MIDL_DEFINE_GUID(CLSID, \
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x \
0,0XC0,0x4F,0xBF,0xE0,0x88);

#ifndef MIDL_DEFINE_GUID
#define MIDL_DEFINE_GUID
#endif
#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'
    }
    'TPCC.NewOrder.1'
    ProgID = s
    VersionIndependentProgID
    = s 'TPCC.NewOrder'
    'MODULE%'
    InprocServer32 = s
    {
        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'
    }
    'TPCC.OrderStatus.1'
    ProgID = s
    VersionIndependentProgID
    = s 'TPCC.OrderStatus'
    'MODULE%'
    InprocServer32 = s
    {
        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'
    }
    'TPCC.Payment.1'
    ProgID = s
    VersionIndependentProgID
    = s 'TPCC.Payment'
    'MODULE%'
    InprocServer32 = s
    {
        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, Wl, Zp8, env=win32 (32b run)
  protocol : dce ms_ext, C_ext
  error checks: allocation ref bounds_check enum
  stub_data
  VC_declspec() decoration level:
  _declspec(novtable)
  _declspec(uuid()), _declspec(selectany),
  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"
```

```

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;
};

#ifdef /* C style interface */
typedef struct ITPCCvtbl
{
    BEGIN_INTERFACE
        HRESULT ( STDMETHODCALLTYPE *QueryInterface )(
            ITPCC * This,
            /* [in] */ REFIID riid,
            /* [iid_is] [out] */ void **ppvobject);
        ULONG ( STDMETHODCALLTYPE *AddRef )(
            ITPCC * This);
        ULONG ( STDMETHODCALLTYPE *Release )(
            ITPCC * This);
        HRESULT ( STDMETHODCALLTYPE *NewOrder )(
            ITPCC * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out);
        HRESULT ( STDMETHODCALLTYPE *Payment )(
            ITPCC * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out);
        HRESULT ( STDMETHODCALLTYPE *Delivery )(
            ITPCC * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out);
        HRESULT ( STDMETHODCALLTYPE *StockLevel )(
            ITPCC * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out);
        HRESULT ( STDMETHODCALLTYPE *OrderStatus )(
            ITPCC * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out);
        HRESULT ( STDMETHODCALLTYPE *CallSetComplete )(
            ITPCC * This);
    END_INTERFACE
} ITPCCvtbl;

interface ITPCC
{
    CONST_VTBL struct ITPCCvtbl *lpVtbl;
};

```

```

#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 */

#ifdef /* C style interface */
#define __stdcall ITPCC_NewOrder_Proxy( \
    ITPCC * This, \
    /* [in] */ VARIANT txn_in, \
    /* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_NewOrder_Stub( \
    ITPCCvtbl *This, \
    RPCChannelBuffer *prpcChannelBuffer, \
    PRPC_MESSAGE _prpcMessage, \
    DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_Payment_Stub( \
    ITPCCvtbl *This, \
    RPCChannelBuffer *prpcChannelBuffer, \
    PRPC_MESSAGE _prpcMessage, \
    DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_Delivery_Stub( \
    ITPCCvtbl *This, \
    RPCChannelBuffer *prpcChannelBuffer, \
    PRPC_MESSAGE _prpcMessage, \
    DWORD *_pdwStubPhase);

void __RPC_STUB ITPCC_StockLevel_Stub( \
    ITPCCvtbl *This, \
    RPCChannelBuffer *prpcChannelBuffer, \
    PRPC_MESSAGE _prpcMessage, \
    DWORD *_pdwStubPhase);

void __RPC_STUB ITPCC_OrderStatus_Proxy( \
    ITPCC * This, \
    /* [in] */ VARIANT txn_in, \
    /* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_CallSetComplete_Proxy( \
    ITPCC * This);

```

```

IRpcStubBuffer *This,
RPCChannelBuffer *_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(
    ITPCCvtbl *This,
    RPCChannelBuffer *_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(
    ITPCCvtbl *This,
    RPCChannelBuffer *_prpcChannelBuffer,
    PRPC_MESSAGE _prpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC * This);

void __RPC_STUB ITPCC_CallSetComplete_Stub(
    ITPCCvtbl *This,
    RPCChannelBuffer *_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-00C04F8FE08B}'
    }
    TPCC.StockLevel = s 'StockLevel Class'
    {
        CurVer = s 'TPCC.StockLevel.1'
    }
    NoRemove CLSID
    {
        ForceRemove {2668369E-A50D-11D2-BA4E-00C04F8FE08B} = s 'StockLevel Class'
    }
    ProgID = s
    'TPCC.StockLevel.1'
    VersionIndependentProgID
    = s 'TPCC.StockLevel'
    '%MODULE%'
    {
        ThreadingModel = s 'Both'
    }
}
```

tpcc.com ps/src/dlldata.c

```

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

DO NOT ALTER THIS FILE

This file is generated 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 );

```

Itautec Philco TPC-C FULL DISCLOSURE REPORT
© 2003 Itautec Philco, all rights reserved

```

/* End of list */
PROXYFILE_LIST_END

DLLDATA_ROUTINES( approxFileList, GET_DLL_CLSID )

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

/* end of generated dll\data file */

tpcc_com_ps/src/tpcc_com_ps.def

LIBRARY
    "tpcc_com_ps"

DESCRIPTION 'Proxy/Stub DLL'

EXPORTS
    DllGetObject @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(novtable)
    _declspec(uuid()), _declspec(selectany),
    _declspec(uuid()), MIDL_INTERFACE()
    */
@@@MIDL_FILE_HEADER( )

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

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

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

#ifdef COM_NO_WINDOWS_H
#include "windows.h"
#endif

```

```
#include "ole2.h"
#ifdef _COM_NO_WINDOWS_H_
#define __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

#pragma once

/* Forward Declarations */

#ifndef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__
typedef interface ITPCCTPCC;
#endif /* __TPCC_FWD_DEFINED__ */

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

#ifdef __cplusplus
extern "C" {
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;

#define __ITPCCTPCC_INTERFACE_DEFINED__
#define __ITPCCTPCC_INTERFACE_DEFINED__

/* Interface ITPCCTPCC */
/* [unique][helpstring][uuid][oleautomation][object] */

EXTERN_C const IID IID_ITPCCTPCC;

#if defined(__cplusplus) && !defined(CINTERFACE)
MIDL_INTERFACE("FEE6A2-84B1-11d2-B47F-00C04FAFF08B")
ITPCCTPCC : public IUnknown
{
public:
    virtual HRESULT STDMETHODCALLTYPE NewOrder(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

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

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

    virtual HRESULT STDMETHODCALLTYPE StockLevel(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;
};
#endif

```

```
(This)->lpvbt1 -> AddrRef(This)

#define ITPCC_Release(This) \
(This)->lpvbt1 -> Release(This)

#define ITPCC_NewOrder(This,txn_in,txn_out) \
(This)->lpvbt1 -> NewOrder(This,txn_in,txn_out)

#define ITPCC_Payment(This,txn_in,txn_out) \
(This)->lpvbt1 -> Payment(This,txn_in,txn_out)

#define ITPCC_Delivery(This,txn_in,txn_out) \
(This)->lpvbt1 -> Delivery(This,txn_in,txn_out)

#define ITPCC_StockLevel(This,txn_in,txn_out) \
(This)->lpvbt1 -> StockLevel(This,txn_in,txn_out)

#define ITPCC_OrderStatus(This,txn_in,txn_out) \
(This)->lpvbt1 -> OrderStatus(This,txn_in,txn_out)

#define ITPCC_CallSetComplete(This) \
(This)->lpvbt1 -> CallSetComplete(This)

#define /* COBJMACROS */

#define /* C style interface */

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

void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcChannelBuffer *This,
    IRpcChannelBuffer *prpccChannelBuffer,
    PRPC_MESSAGE _prpccMessage,
    DWORD _pdwstsubPhase);

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

void __RPC_STUB ITPCC_Payment_Stub(
    IRpcChannelBuffer *This,
    IRpcChannelBuffer *prpccChannelBuffer,
    PRPC_MESSAGE _prpccMessage,
    DWORD _pdwstsubPhase);

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

void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcChannelBuffer *This,
    IRpcChannelBuffer *prpccChannelBuffer,
    PRPC_MESSAGE _prpccMessage,
    DWORD _pdwstsubPhase);

HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC *This,
```

```

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

void __RPC_STUB_IPTCC_StockLevel_Stub(
    ITPCCStubBuffer *This,
    IPRPCChannelBuffer *prpcChannelBuffer,
    IPRPC_MESSAGE prpcMessage,
    DWORD *_pdwStubPhase);

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

void __RPC_STUB_IPTCC_OrderStatus_Stub(
    ITPCCStubBuffer *This,
    IPRPCChannelBuffer *prpcChannelBuffer,
    IPRPC_MESSAGE prpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall IPTCC_CallSetComplete_Proxy(
    ITPCC * This);

void __RPC_STUB_IPTCC_CallSetComplete_Stub(
    ITPCCStubBuffer *This,
    IPRPCChannelBuffer *prpcChannelBuffer,
    IPRPC_MESSAGE prpcMessage,
    DWORD *_pdwStubPhase);

#endif /* __IPTCC_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_UserMarshal(unsigned long *, unsigned char *, 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: Microsoft TPC-C
 * Kit Ver. 4.20.000
 * Microsoft, 1999 Copyright
 * All Rights Reserved
 * *
 * not yet audited

```

```

*
* PURPOSE: Defines the interface used by TPCC.
* This interface can be implemented by C++ components.
*
* Change history:
*
*
// Forward declare all types defined
// interface ITPCC:
import "oidl.idl";
import "ocidl.idl";

[
    object
    automation,
    uuid(FEEE6AA2-8481-11d2-8A47-
00C04FBFE08B),
    helpstring("ITPCC Interface"),
    pointer_default(unique)
]
interface ITPCC : IUnknown
{
    HRESULT STDMETHODCALLTYPE NewOrder
    (
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );
    HRESULT STDMETHODCALLTYPE Payment
    (
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );
    HRESULT STDMETHODCALLTYPE Delivery
    (
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );
    HRESULT STDMETHODCALLTYPE StockLevel
    (
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );
    HRESULT STDMETHODCALLTYPE OrderStatus
    (
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );
}

```

```

);
HRESULT STDMETHODCALLTYPE 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(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AMD64)
#define _cplplusplus
extern "C" {
#endif

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

#if !defined(_MIDL_USE_GUIDDEF_)
#define _MIDL_USE_GUIDDEF_
#endif
#define INITGUID
#define GUIDDEF
#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)

#define _MIDL_USE_GUIDDEF_
#define 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) \
{ l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8} }

#define _MIDL_USE_GUIDDEF_
_MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEE6AA2,0x8481,0x11d2,0x8A,0x47,0x00,0xC0,
0x4F,0xBF,0xE0,0x88);

#undef _MIDL_DEFINE_GUID

#define _cplplusplus
#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(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#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)

#define _MIDL_USE_GUIDDEF_
#define IID_DEFINED
#define IID_DEFINED_
typedef struct _IID

```

```

#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 = {0x8a885004, 0x1ceb, 0x11c9, {0x9f, 0xe8, 0x08, 0x00, 0x2b, 0x10, 0x48, 0x60}}, {2, 0}};

extern const MIDL_TYPE_FORMAT_STRING
_MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
_MIDL_ProcFormatString;

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo;

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

#if !defined(_RPC_WIN32_)
#error Invalid build platform for this stub.
#endif

#if ! (TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this
stub because it uses these features:
#error -Oif or -Oicf, [wire_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 */

```

```

FC_AUTO_HANDLE */
0x33,
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, /* 012 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
= 24 */
/* 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, /* 012 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, /* 012 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 */

```

```

/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset
= 24 */
/* 168 */ 0x8,
/*
*/
/* Procedure CallSetComplete */
/* 170 */ 0x33,
/* FC_AUTO_HANDLE */
/* 096c,
/* 0
/* 8
/* x86 Stack size/offset
= 8 */
/* 180 */ NdrFcShort( 0x0 ),
/* 182 */ NdrFcShort( 0x8 ),
/* 184 */ 0x4,
/* 012 Flags: has
return, */
/* 1
*/
/* Return value */
/* 186 */ NdrFcShort( 0x70 ), /* Flags: out, return,
baseType, */
/* 188 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset
= 4 */
/* 190 */ 0x8,
/* FC_LONG */
/* 0
*/
/*
0x0
*/
static const MIDL_TYPE_FORMAT_STRING
_MIDL_TypeFormatString =
{
0,
{
/* 2 */
FC_UP /*
/* 4 */ NdrFcShort( 0x3ca ),
/* 6 */
/*
FC_NONENCAPSULATED_UNION */
0x2b,
/*
FC_ILONG */
/* 8 */ 0x7,
/* Corr desc: FC_USHORT
0x0,
/*
/* 10 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 12 */ NdrFcShort( 0x2 ), /* offset= 2 (14) */
/* 14 */ NdrFcShort( 0x10 ), /* 16 */
/* 16 */ NdrFcShort( 0x2f ), /* 47 */
/* 18 */ NdrFcLong( 0x14 ), /* 20 */
/* Simple arm
type: FC_HYPER */
/* 24 */ NdrFcLong( 0x3 ), /* 3 */
/* Simple arm
type: FC_LONG */
/* 28 */ NdrFcShort( 0x8008 ),
/* 30 */ NdrFcLong( 0x11 ), /* 17 */
/* Simple arm
type: FC_BYTE */
/* 34 */ NdrFcShort( 0x8001 ),
/* 36 */ NdrFcLong( 0x2 ), /* 2 */

```

```

/* simple arm
/* simple arm
/* simple arm
/* simple arm
/* simple arm
/* simple arm
232 (302) */
/* simple arm
226 (308) */
244 (332) */
/* offset= 256
/* 8192 */
/* offset= 268
/* offset= 794
/* 16420 */
/* offset= 788
/* 16401 */
/* offset= 786
/* 16386 */
/* offset= 784
/* 16387 */
/* offset= 782
/* 16404 */
/* offset= 780
/* 16388 */
/* offset= 778
/* 16389 */
/* offset= 776
/* 16395 */
/* offset= 754
/* 16394 */
/* offset= 752
/* 16390 */
/* offset= 762
/* 16391 */
/* offset= 752
/* 16392 */
/* offset= 754
/* 16397 */
/* offset= 752
/* 16393 */

```

/* 314 */ NdrFcShort(0x2),	/* 2 */	/* 1
/* 316 */ 0x9,	/* Corr desc: FC_ULONG */	
/* 318 */ NdrFcShort(0xffffc),	/* -4 */	
/* 320 */ 0x6,	/* FC_SHORT */	
FC_END /*	0x5b,	/*
/* 322 */		
FC_CSTRUCT /*	0x17,	/*
/* 324 */ NdrFcShort(0x8),	0x3,	/* 3
/* 326 */ NdrFcShort(0xfffffff2),	/* 8 */	/* offset= -14
(312) /*	/* FC_LONG */	
/* 328 */ 0x8,	0x8,	/*
FC_LONG /*	/* FC_PAD */	
/* 330 */ 0x5c,	0x5b,	/*
FC_END /*		
/* 332 */	0x2f,	/*
FC_IP /*	0x5a,	/*
FC_CONSTANT_IID /*	/* 0 */	/* 0
/* 334 */ NdrFcLong(0x0),	0x0,	
/* 336 */ NdrFcShort(0x0),	/* 0 */	/* 0
/* 338 */ NdrFcShort(0x0),	0x0,	
/* 340 */ NdrFcShort(0x0),	/* 0 */	/* 0
/* 342 */ 0xc0,	/* 192 */	/* 70
/* 344 */ 0x0,	/* 0 */	
/* 346 */ 0x0,	0x0,	/* 0
/* 348 */ 0x0,	/* 0 */	/* 0
/* 350 */	0x46,	/* 70
FC_IP /*	0x2f,	/*
FC_CONSTANT_IID /*	0x5a,	/*
/* 352 */ NdrFcLong(0x20400),	/* 132096 */	
/* 354 */ NdrFcShort(0x0),	/* 0 */	/* 0
/* 356 */ NdrFcShort(0x0),	/* 0 */	/* 0
/* 358 */ NdrFcShort(0x0),	/* 192 */	/* 0
/* 360 */ 0xc0,	0x0,	
/* 362 */ 0x0,	/* 0 */	/* 0
/* 364 */ 0x0,	0x0,	/* 0
/* 366 */ 0x0,	/* 0 */	/* 0
/* 368 */	0x46,	/* 70
FC_UP [pointer deref] /*	0x12, 0x10,	/*
/* 370 */ NdrFcShort(0x2),	/* offset= 2 (372) */	
/* 372 */	0x12, 0x0,	/*
FC_UP /*		

```

/* 374 */ NdrFcShort( 0x1fc ), /* offset= 508 */
/* 376 */
FC_ENCAPSULATED_UNION /*
/*
/* 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 */
/* 396 */ NdrFcShort( 0x94 ), /* Offset= 148 (546) */
/* 398 */ NdrFcLong( 0xc ), /* 12 */
/* 400 */ NdrFcShort( 0x3 ), /* Offset= 188 (592) */
/* 402 */ NdrFcLong( 0x24 ), /* 36 */
/* 406 */ NdrFcShort( 0x114 ), /* Offset= 276 (686) */
/* 410 */ NdrFcLong( 0x800d ), /* 32781 */
/* 412 */ NdrFcShort( 0x130 ), /* Offset= 304 (720) */
/* 416 */
/* 418 */ NdrFcLong( 0x10 ), /* 16 */
/* 422 */ NdrFcShort( 0x148 ), /* Offset= 328 (750) */
/* 424 */
/* 428 */ NdrFcLong( 0x2 ), /* 2 */
/* 430 */ NdrFcShort( 0x160 ), /* Offset= 352 (780) */
/* 434 */
/* 436 */ NdrFcLong( 0x3 ), /* 3 */
/* 438 */ NdrFcShort( 0x178 ), /* Offset= 376 (814) */
/* 442 */ NdrFcLong( 0x14 ), /* 20 */
/* 444 */ NdrFcShort( 0x190 ), /* Offset= 400 (840) */
/* 446 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (441) */
/* 448 */
FC_CARRY /*
/*
/* 446 */ NdrFcShort( 0x4 ), /* 4 */
/* 448 */ 0x19, /* Correlation field pointer, FC_LONG */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */
FC_PP /*
/*
FC_PAD /*
/* 454 */
FC_VARIABLE_REPEAT /*
/*
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( 0xffffffff ), /* Offset= -146 (322) */
/* 470 */
FC_END /*
/*
FC_LONG /*

```

```

/* 472 */ /* 0x5c, */ /* FC_PAD */ /*
FC_END */ /* 0x5b, */ /*
/* 474 */ /* 0x16, */ /*
FC_PSTRUCT */ /* 0x3, */ /* 3
*/ /* 476 */ /* NdrFcShort( 0x8 ), */ /* 8 */
/* 478 */ /* 0x4b, */ /*
FC_PP */ /* 0x5c, */ /*
FC_PAD */ /* 0x46, */ /*
/* 480 */ /* 0x5c, */ /*
FC_NO_REPEAT */ /*
FC_PAD */ /* 4 */ /*
/* 482 */ /* NdrFcShort( 0x4 ), */ /* 4 */
/* 484 */ /* NdrFcShort( 0x4 ), */ /* 4 */
/* 486 */ /* 0x11, 0x0, */ /* FC_RP */ /*
/* 488 */ /* NdrFcShort( 0xfffffd4 ), */ /* Offset= -44
(444) */ /* 490 */
FC_END */
/* 0x5b, */ /*
/* 0x8, */ /*
/* 492 */ /* 0x8, */ /* FC_LONG */ /*
/* 494 */ /* 0x5b, */ /*
FC_END */ /*
/* 0x21, */ /*
/* 0x3, */ /* 3
*/ /* 496 */ /* NdrFcShort( 0x0 ), */ /* 0 */
/* 498 */ /* 0x19, */ /* Corr desc: field
pointer, FC_ULONG */ /* 0x0, */ /*
/* 500 */ /* NdrFcShort( 0x0 ), */ /* 0 */
/* 502 */ /* NdrFcLong( 0xfffffff ), */ /* -1 */
/* 506 */ /* 0x4c, */ /* FC_EMBEDDED_COMPLEX */
/* 0x0, */ /*
/* 508 */ /* NdrFcShort( 0xfffff50 ), */ /* Offset= -176
(332) */ /* 510 */ /* 0x5c, */ /* FC_PAD */ /*
/* 512 */ /* 0x5b, */ /*
/* 0x1a, */ /*
/* 0x3, */ /* 3
*/ /* 514 */ /* NdrFcShort( 0x8 ), */ /* 8 */
/* 516 */ /* NdrFcShort( 0x0 ), */ /* 0 */
/* 518 */ /* NdrFcShort( 0x6 ), */ /* Offset= 6 (524) */
/* 520 */ /* 0x8, */ /* FC_LONG */ /*
/* 0x36, */ /*
FC_POINTER */ /* 4 */ /*
/* 522 */ /* 0x5c, */ /* FC_PAD */ /*
/* 524 */ /* 0x5b, */ /*
FC_END */ /*
/* 0x11, 0x0, */ /*
FC_RP */ /* 526 */ /* NdrFcShort( 0xffffffe0 ), */ /* 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( 0xfffffff ), */ /* -1 */
/* 540 */ /* 0x4c, */ /* FC_EMBEDDED_COMPLEX */
/* 0x0, */ /*
*/ /* 542 */ /* NdrFcShort( 0xffffff40 ), */ /* Offset= -192
(350) */ /* 544 */ /* 0x5c, */ /* FC_PAD */ /*
/* 546 */ /* 0x5b, */ /*
FC_END */ /*
/* 548 */ /* 0x1a, */ /*
FC_BOGUS_STRUCT */ /* 0x3, */ /* 3
*/ /* 548 */ /* NdrFcShort( 0x8 ), */ /* 8 */
/* 550 */ /* NdrFcShort( 0x0 ), */ /* 0 */
/* 552 */ /* NdrFcShort( 0x6 ), */ /* Offset= 6 (558) */
/* 554 */ /* 0x8, */ /* FC_LONG */ /*
FC_POINTER */ /* 0x36, */ /*
/* 556 */ /* 0x5c, */ /* FC_PAD */ /*
/* 558 */ /* 0x5b, */ /*
FC_END */ /*
/* 0x11, 0x0, */ /*
FC_RP */ /* 560 */ /* NdrFcShort( 0xffffffe0 ), */ /* Offset= -32
(528) */ /* 562 */ /* 0x1b, */ /*
FC_CARRY */ /* 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 */ /* 0x48, */ /*
/* 572 */ /* 0x49, */ /*
FC_VARIABLE_REPEAT */ /*
FC_FIXED_OFFSET */ /* 4 */ /*
/* 574 */ /* NdrFcShort( 0x4 ), */ /* 4 */
/* 576 */ /* NdrFcShort( 0x0 ), */ /* 0 */
/* 578 */ /* NdrFcShort( 0x1 ), */ /* 1 */
/* 580 */ /* NdrFcShort( 0x0 ), */ /* 0 */
/* 582 */ /* NdrFcShort( 0x0 ), */ /* 0 */
/* 584 */ /* 0x12, 0x0, */ /* FC_LP */ /*
/* 586 */ /* NdrFcShort( 0x184 ), */ /* Offset= 388
(974) */ /* 588 */
FC_END */
/* 0x5b, */ /*

```

```

FC_LONG */ /* 0x8, */ /*
/* 590 */ /* 0x5c, */ /* FC_PAD */ /*
/* 592 */ /* 0x5b, */ /*
/* 0x1a, */ /*
/* 0x3, */ /* 3
*/ /* 594 */ /* NdrFcShort( 0x8 ), */ /* 8 */
/* 596 */ /* NdrFcShort( 0x0 ), */ /* 0 */
/* 598 */ /* NdrFcShort( 0x6 ), */ /* Offset= 6 (604) */
/* 600 */ /* 0x8, */ /* FC_LONG */ /*
/* 0x36, */ /*
FC_POINTER */ /* 4 */ /*
/* 602 */ /* 0x5c, */ /* FC_PAD */ /*
/* 604 */ /* 0x5b, */ /*
FC_END */ /*
/* 606 */ /* 0x11, 0x0, */ /*
FC_RP */ /* 606 */ /* NdrFcShort( 0xfffffd4 ), */ /* Offset= -44
(562) */ /* 608 */ /* 0x2f, */ /*
FC_IP */ /* 0x5a, */ /*
FC_CONSTANT_IID */ /* 47 */ /*
/* 610 */ /* NdrFcLong( 0x2f ), */ /* 47 */
/* 612 */ /* NdrFcShort( 0x0 ), */ /* 0 */
/* 614 */ /* NdrFcShort( 0x0 ), */ /* 0 */
/* 616 */ /* NdrFcShort( 0x0 ), */ /* 192 */
/* 618 */ /* 0xc0, */ /* 0x0, */ /* 0
*/ /* 620 */ /* 0x0, */ /* 0
*/ /* 622 */ /* 0x0, */ /* 0
*/ /* 624 */ /* 0x0, */ /* 0
*/ /* 626 */ /* 0x1b, */ /*
FC_CARRY */ /* 0x0, */ /* 0
*/ /* 628 */ /* NdrFcShort( 0x1 ), */ /* 1 */
/* 630 */ /* 0x19, */ /* Corr desc: field
pointer, FC_ULONG */ /* 0x0, */ /*
*/ /* 632 */ /* NdrFcShort( 0x4 ), */ /* 4 */
/* 634 */ /* 0x1, */ /* FC_BYTE */ /*
/* 636 */ /* 0x5b, */ /*
FC_END */ /*
/* 0x1a, */ /*
FC_BOGUS_STRUCT */ /* 0x3, */ /* 3
*/ /* 638 */ /* NdrFcShort( 0x10 ), */ /* 16 */
/* 640 */ /* NdrFcShort( 0x0 ), */ /* 0 */
/* 642 */ /* NdrFcShort( 0xa ), */ /* Offset= 10 (652) */
/* 644 */ /* 0x8, */ /* FC_LONG */ /*
/* 646 */ /* 0x4c, */ /* FC_EMBEDDED_COMPLEX */
/* 0x0, */ /*

```

```

/* 648 */ NdrFcShort( 0xffffffff8 ), /* Offset= -40
/* 650 */ 0x36, /* FC_POINTER */ /*
FC_END */
/* 652 */ 0x12, 0x0, /*
FC_UP */ NdrFcShort( 0xffffffff4 ), /* Offset= -28
(626) */
/* 656 */ 0x1b, /*
FC_CARRY */
/*
/* 658 */ NdrFcShort( 0x4 ), /* 4 */
/* 660 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
/*
/* 662 */ NdrFcShort( 0x0 ), /* 0 */
/* 664 */ 0x4b, /*
FC_PP */
/*
FC_PAD */
/* 666 */ 0x48, /*
FC_VARIABLE_REPEAT */
/*
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( 0xffffffff4 ), /* Offset= -44
(636) */
/* 682 */ 0x5b, /*
FC_END */
/*
FC_LONG */
/* 684 */ 0x5c, /* FC_PAD */
/* 686 */ 0x1a, /*
FC_BOGUS_STRUCT */
/*
/* 688 */ NdrFcShort( 0x8 ), /* 8 */
/* 690 */ NdrFcShort( 0x0 ), /* 0 */
/* 692 */ NdrFcShort( 0x6 ), /* Offset= 6 (698) */
/* 694 */ 0x8, /* FC_LONG */
/*
FC_POINTER */
/* 696 */ 0x5c, /* FC_PAD */
/* 698 */ 0x11, 0x0, /*
FC_PP */
/* 700 */ NdrFcShort( 0xffffffff4 ), /* Offset= -44
(656) */
/* 702 */ 0x1d, /*
FC_SMFARRAY */
/*
/*

```

```

/* 704 */ NdrFcShort( 0x8 ), /* 8 */
/* 706 */ 0x1, /* FC_BYTE */
/*
FC_END */
/* 708 */ 0x15, /*
FC_STRUCT */
/*
/* 710 */ NdrFcShort( 0x10 ), /* 16 */
/* 712 */ 0x8, /* FC_LONG */
/*
FC_SHORT */
/* 714 */ 0x6, /* FC_SHORT */
/*
FC_EMBEDDED_COMPLEX */
/* 716 */ 0x0, /* 0 */
/* 718 */ NdrFcShort( 0xffffffff1 ), /* Offset= -15 (702) */
/*
FC_END */
/* 720 */ 0x1a, /*
FC_BOGUS_STRUCT */
/*
/* 722 */ NdrFcShort( 0x18 ), /* 24 */
/* 724 */ NdrFcShort( 0x0 ), /* 0 */
/* 726 */ NdrFcShort( 0x0 ), /* Offset= 10 (736) */
/* 728 */ 0x8, /* FC_LONG */
/*
FC_POINTER */
/* 730 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/*
/* 732 */ NdrFcShort( 0xffffffff8 ), /* Offset= -24
(708) */
/* 734 */ 0x5c, /* FC_PAD */
/*
FC_END */
/* 736 */ 0x11, 0x0, /*
FC_PP */
/* 738 */ NdrFcShort( 0xffffffff0c ), /* Offset= -244
(494) */
/* 740 */ 0x1b, /*
FC_CARRY */
/*
/* 742 */ NdrFcShort( 0x1 ), /* 1 */
/* 744 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
/*
/* 746 */ NdrFcShort( 0x0 ), /* 0 */
/* 748 */ 0x1, /* FC_BYTE */
/*
FC_END */
/* 750 */ 0x16, /*
FC_PSTRUCT */
/*
/* 752 */ NdrFcShort( 0x8 ), /* 8 */
/* 754 */ 0x4b, /*
FC_PP */
/*
FC_PAD */
/* 756 */ 0x5c, /*

```

```

FC_NO_REPEAT */
/*
FC_PAD */
/* 758 */ NdrFcShort( 0x4 ), /* 4 */
/* 760 */ NdrFcShort( 0x4 ), /* 4 */
/* 762 */ 0x12, 0x0, /* FC_UP */
/* 764 */ NdrFcShort( 0xffffffff8 ), /* Offset= -24
(740) */
/* 766 */ 0x5b, /*
FC_END */
/*
FC_LONG */
/* 768 */ 0x8, /* FC_LONG */
/* 770 */ 0x5b, /*
FC_END */
/*
FC_CARRY */
/*
/* 772 */ NdrFcShort( 0x2 ), /* 2 */
/* 774 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
/*
/* 776 */ NdrFcShort( 0x0 ), /* 0 */
/* 778 */ 0x6, /* FC_SHORT */
/*
FC_END */
/* 780 */ 0x16, /*
FC_PSTRUCT */
/*
/* 782 */ NdrFcShort( 0x8 ), /* 8 */
/* 784 */ 0x4b, /*
FC_PP */
/*
FC_PAD */
/* 786 */ 0x46, /*
FC_NO_REPEAT */
/*
FC_PAD */
/* 788 */ NdrFcShort( 0x4 ), /* 4 */
/* 790 */ NdrFcShort( 0x4 ), /* 4 */
/* 792 */ 0x12, 0x0, /* FC_UP */
/* 794 */ NdrFcShort( 0xffffffff8 ), /* Offset= -24
(770) */
/* 796 */ 0x5b, /*
FC_END */
/*
FC_LONG */
/* 798 */ 0x8, /* FC_LONG */
/* 800 */ 0x5b, /*
FC_END */
/*
FC_CARRY */
/*
/* 802 */ NdrFcShort( 0x4 ), /* 4 */
/* 804 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
/*

```

```

/* 806 */ NdrFcShort( 0x0 ), /* 0 */
/* 808 */ 0x8, /* FC_LONG */
FC_END /*
/* 810 */
/* 816 */
FC_PSTRUCT /*
/* 812 */ NdrFcShort( 0x8 ), /* 8 */
/* 814 */
FC_PP /*
/* 818 */ NdrFcShort( 0x4 ), /* 4 */
/* 820 */ NdrFcShort( 0x4 ), /* 4 */
/* 822 */ 0x12, 0x0, /* FC_UP */
/* 824 */ NdrFcShort( 0xfffffe8 ), /* Offset= -24
(800) */
/* 826 */
FC_END /*
/* 832 */ NdrFcShort( 0x8 ), /* 8 */
/* 834 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
/* 836 */ NdrFcShort( 0x0 ), /* 0 */
/* 838 */ 0x0, /* 0 */
/* 840 */
FC_END /*
/* 842 */ NdrFcShort( 0x8 ), /* 8 */
/* 844 */
FC_PP /*
/* 846 */
FC_PAD /*
/* 848 */ NdrFcShort( 0x4 ), /* 4 */
/* 850 */ NdrFcShort( 0x4 ), /* 4 */
/* 852 */ 0x12, 0x0, /* FC_UP */
/* 854 */ NdrFcShort( 0xfffffe8 ), /* Offset= -24
(830) */
/* 856 */
FC_END /*

```

```

FC_LONG /*
/* 858 */ 0x8,
FC_END /*
/* 860 */
FC_STRUCT /*
/* 862 */ NdrFcShort( 0x8 ), /* 8 */
/* 864 */ 0x8, /* FC_LONG */
/* 866 */ 0x5c, /* FC_PAD */
FC_END /*
/* 868 */
FC_CARRY /*
/* 870 */ NdrFcShort( 0x8 ), /* 8 */
/* 872 */ 0x7, /* Corr desc: FC_USHORT
*/
/* 874 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 876 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 878 */ NdrFcShort( 0xfffffee ), /* Offset= -18
(860) */
/* 880 */ 0x5c, /* FC_PAD */
FC_END /*
/* 882 */
FC_BOGUS_STRUCT /*
/* 884 */ NdrFcShort( 0x28 ), /* 40 */
/* 886 */ NdrFcShort( 0xfffffee ), /* Offset= -18
(868) */
/* 888 */ NdrFcShort( 0x0 ), /* Offset= 0 (888) */
/* 890 */ 0x6, /* FC_SHORT */
FC_SHORT /*
/* 892 */ 0x8, /* FC_LONG */
FC_LONG /*
/* 894 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 896 */ NdrFcShort( 0xfffffd8 ), /* Offset= -520
(376) */
/* 898 */ 0x5c, /* FC_PAD */
FC_END /*
/* 900 */
FC_UP /*
/* 902 */ NdrFcShort( 0xfffffef6 ), /* Offset= -266
(636) */
/* 904 */
FC_UP [simple_pointer] /*
/* 906 */ 0x1, /* FC_BYTE */
FC_PAD /*
/* 908 */

```

```

FC_UP [simple_pointer] /*
/* 910 */ 0x6,
FC_PAD /*
/* 912 */
FC_UP [simple_pointer] /*
/* 914 */ 0x8,
FC_PAD /*
/* 916 */
FC_UP [simple_pointer] /*
/* 918 */ 0xb, /* FC_HYPER */
FC_PAD /*
/* 920 */
FC_UP [simple_pointer] /*
/* 922 */ 0xa, /* FC_FLOAT */
FC_PAD /*
/* 924 */
FC_UP [simple_pointer] /*
/* 926 */ 0xc, /* FC_DOUBLE */
FC_PAD /*
/* 928 */
FC_UP /*
/* 930 */ NdrFcShort( 0xfffffd8 ), /* Offset= -628
(302) */
/* 932 */
FC_UP [pointer_deref] /*
/* 934 */ NdrFcShort( 0xfffffd8 ), /* Offset= -626
(308) */
/* 936 */
FC_UP [pointer_deref] /*
/* 938 */ NdrFcShort( 0xffffda2 ), /* Offset= -606
(332) */
/* 940 */
FC_UP [pointer_deref] /*
/* 942 */ NdrFcShort( 0xffffdb0 ), /* Offset= -592
(350) */
/* 944 */
FC_UP [pointer_deref] /*
/* 946 */ NdrFcShort( 0xffffdb8 ), /* Offset= -578
(368) */
/* 948 */
FC_UP [pointer_deref] /*
/* 950 */ NdrFcShort( 0x2 ), /* Offset= 2 (952) */
/* 952 */
FC_UP /*
/* 954 */ NdrFcShort( 0x14 ), /* Offset= 20 (974) */
/* 956 */
FC_STRUCT /*
/* 958 */ NdrFcShort( 0x10 ), /* 16 */
/* 960 */ 0x6, /* FC_SHORT */
FC_BYTE /*
/* 962 */ 0x1, /* FC_BYTE */

```

```

FC_LONG /* 0x8, /*
/* 964 */ 0xb, /* FC_HYPER */
0x5b, /*
/* 966 */
0x12, 0x0, /*
/* 968 */ NdrFcShort( 0xffffffff ), /* Offset= -12
(956) */
/* 970 */
0x12, 0x8, /*
/* 972 */ [simple_pointer] */
/* FC_CHAR */
0x5c, /*
/* 974 */
0x1a, /*
/* 976 */
0x7, /* 7
*/
/* 976 */ NdrFcShort( 0x20 ), /* 32 */
/* 978 */ NdrFcShort( 0x0 ), /* 0 */
/* 980 */ NdrFcShort( 0x0 ), /* Offset= 0 (980) */
/* 982 */ 0x8, /* FC_LONG */
/*
/* FC_LONG */
/* 984 */ 0x6, /* FC_SHORT */
/*
/* FC_SHORT */
/* 986 */ 0x6, /* FC_SHORT */
/*
/* FC_SHORT */
/* 988 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /*
/* 990 */ NdrFcShort( 0xffffffff28 ), /* Offset= -984
(6) */
/* 992 */ 0x5c, /* FC_PAD */
/*
/* 994 */ 0xb4, /* FC_USER_MARSHAL */
0x83, /*
131 */ /* NdrFcShort( 0x0 ), /* 0 */
/* 996 */ NdrFcShort( 0x10 ), /* 16 */
/* 998 */ NdrFcShort( 0x0 ), /* 0 */
/* 1000 */ NdrFcShort( 0x0 ), /* 0 */
Offset= -1000 (2) */
/* 1004 */
0x11, 0x4, /*
/* 1006 */ NdrFcShort( 0x6 ), /* Offset= 6
(1012) */
/* 1008 */
0x13, 0x0, /*
/* 1010 */ NdrFcShort( 0xffffffffdc ), /*
Offset= -36 (974) */
0xb4, /*
/* FC_USER_MARSHAL */
0x83, /*
131 */ /* NdrFcShort( 0x0 ), /* 0 */
/* 1014 */ NdrFcShort( 0x10 ), /* 16 */
/* 1016 */ NdrFcShort( 0x0 ), /* 0 */
/* 1018 */ NdrFcShort( 0x0 ), /* 0 */
/* 1020 */ NdrFcShort( 0xffffffff4 ), /*
Offset= -12 (1008) */
0x0
}
};

```

```

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

/* Standard interface: __MIDL_itf_tpc_com_ps_0000,
ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

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

/* Object interface: ITPCC, ver. 0.0,
GUID={0xEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x83}} */
#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,
    0
};

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

INTERFACE_PROXY_VTABLE(9) _ITPCC_ProxyVtbl =
{
    &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,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format, /*
    1, /* error bounds-check flag */
    0x20000, /* Ndr Library Version */
    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 *
_tpc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &ITPCCProxyVtbl,
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[]
=
{
    ( CInterfaceStubVtbl *) &ITPCCStubVtbl,
};

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_ProxyStubVtblList,
    (ConstPCInterfaceName *) &
    _tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    & _tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, // table of [asynch_uuid] interfaces */
    0, // Filler1 */
    0, // Filler2 */
    0, // Filler3 */
};

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

common/txnlog/include/rtime.h

```
/* FILE: rtime.h : header file
 * Copyright 1997 Microsoft Corp., All rights reserved.
 * Source code licensed to Tandem Computers for
 * 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: RTIME.H
#define MAX_JULIAN_TIME 0x7FFFFFFF
#define JULIAN_TIME __int64
extern "C"
{
    BOOL InitJulianTime(LPSYSTEMTIME
        lpInitTime);
    JULIAN_TIME MyTickCount(void);
    void MyTickCount(JULIAN_TIME
        *pJulian, DWORD *pTC);
    JULIAN_TIME iday, int hour, int minute, int fsecond;
    JULIAN_TIME ConvertTo64bitTime(int iYear, int
        iMonth, int iDay, int iHour, int iMinute, int iSecond);
    lpInitTime);
}
```

```
int mm, int dd );
void JulianDay( int yr, int
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
 * 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 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 m_Semaphore;
    volatile LONG m_SpinLock;
}
```

```
Waiting;
volatile LONG
#ifdef _DEBUG
// Counters for debugging
volatile LONG
TotalLocks;
volatile LONG
TotalSleeps;
volatile LONG
TotalSpins;
volatile LONG
TotalWaits;
#endif
public:
// Public functions.
SpinLock( void );
inline BOOL ClaimLock(
void );
inline void ReleaseLock(
~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(
(LPULONG) & TotalLocks );
#endif
return ( (*SpinLock) == LockOpen )
&& (InterlockedExchange( (LPULONG)SpinLock, LockClosed )
== LockOpen );
}

/*****
 * Claim the spinlock.
 *
 *****/
```



```

// 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
    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
    BYTE TxnType;
    // = TXN_REC_TYPE_TPCCW
    BYTE TxnSubType;
    // depends on TxnType
    // end of common header

    int ThinkTime;
    think time (ms)
    int WIRT;
    // response time (ms)
    int TxnError;
    TxnStatus
    // error code providing more detail for
    errors
    BYTE TxnStatus;
    // completion status for txn to indicate
    txn sub type:
    // This field below depends on the

```

```

indicates whether the user was a new customer (or
returning)
// - for Home interaction: it
// - for Buy Confirm:
it indicates whether the shipping address was
updated
// - for Search Request:
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); } *PTXN_RECORD_TPCW;
} 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;
    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
    BYTE TxnType;
    // start of txn
    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;
JULIAN_TIME
// user number
//
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 // lower 22 bits
mask for user field in TPCW record 0x003fffff
#define USER_SESSION_MASK 0xffff0000
// 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
TXN_DATA_START
// offset in log file where log records start
4096
#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; // file size in
    bytes
    iFileSize;
    log file
    // driver engine that created this
    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
    {
        //
        TS; JULIAN_TIME // timestamp of
        record
    }

```

```

// position in file      int      // byte
// RecMap[RecMapSize]:
// RecMapSize
// #define 200
} TXN_LOG_HEADER, *PTXN_LOG_HEADER;

/* Header of the sorted pointers blocks in
Temp file (in merging). */
typedef struct BLOCK_HEADER {
    long BlockPos;
    DWORD CmpPos;
    BYTE Read;
    int nRecords;
    BYTE *offset; /* offset of
} BLOCK_HEADER, *PBLOCK_HEADER;

pointers to records in the log file */

#define READ_BUFFER_SIZE 64*1024
#define WRITE_BUFFER_SIZE 8*1024
#define WRITE_BUFFER_SIZE 128*1024
#define NUM_READ_BUFFERS 1
#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;
    DWORD iBytesFreeInBuffer;
    int iNumBuffers;
    int iActiveBuffer;
    int iToBuffer;
    int iFilePointer;
    LARGE_INTEGER iPosition;
    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
    SavePTime;
    int iSavePtFilePointer;
    LARGE_INTEGER iSavePtFilePosition;
    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
    log file sorted; // applies to both input and output
    // is
    BeginTxnts; JULIAN_TIME
    //
    timestamp of first (lowest) txn start
    JULIAN_TIME // 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.
    last records that begin in the buffer.
    // So remember the timestamps for 2
    record written to disk.
    // 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
    {

```

```

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 // sorted
    pointers block header
    BYTE
    **recPointers;
    // record pointer buffers
    for each sorted block
    PTXN_RECORD_HEADER *recBuffers;
    // record buffers for each sorted block
    int // # of pointers
    processed in each block
    BOOL *blockAvailable;
    // whether to check a particular
    block for jmin
    int
    int nBlocks;
    jmin;
    // index (block-wise) of the lowest timestamp
    record
    int
    // average
    record length

```

```

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

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

        int Write(BYTE *ptr, DWORD size);
        static void LogFileTo(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
            pTxnRcd);
            int
            WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF pTxnRcd);
            int WriteToLog(PTXN_RECORD_CONTROL
            pCtrlRec);
            int WriteToLog(PTXN_RECORD_HEADER
            pCtrlRec);
            int WriteToLog(PTXN_RECORD_TPCW
            pTxnRcd);
            //support for TPC-W

            LPTSTR lpStr, DWORD dwLen);
            int WriteCtrlRecToLog(BYTE subtype,
            void CloseTransactionLogFile(void);

            PTXN_RECORD_HEADER
            GetNextRecord(BOOL bSkipCtrlRecs = FALSE);
            PTXN_RECORD_HEADER
            GetNextRecord(JULIAN_TIME SeekTime0, BOOL
            bSkipCtrlRecs = FALSE);

            int Sort(void);
            PTXN_RECORD_HEADER

            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."
            };

```

```

        // "Log file is not sorted",
        ERR_LOG_NOT_SORTED,
        // "Invalid Time Seq.
        // "Internal Error: Record Time
        Sequence invalid."
    };

    CtxNLog_ERR(int iErr) :
    CBaseErr(iErr) {}

    int ErrType() {return
    ERR_TYPE_TXNLOG;}

    char *ErrMsg() { return "TXN
    LOG"; }

    char *ErrMsgText()
    {
        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;
        {
            if ( m_idmsg ==
            i )
                break;
        }

        szMsgs[i] : ERR_UNKNOWN);
    };
}

return(szMsgs[i][0] ?
};

```

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:
: 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.
:-----
: @ccscript 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, 2003
--
-- Creates 5700 warehouse database
--

```

```

--
-----
--
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(), 21))
go
-- create main database files
-----
CREATE DATABASE tpcc
ON PRIMARY
(
NAME = MSSQL_tpcc_root,
FILENAME = 'c:\mssql\tpcc_root.mdf',
SIZE = 8MB,
FILEGROWTH = 0),
FILEGROUP MSSQL_misc_fg
(
NAME = MSSQL_misc1,
FILENAME = 'c:\dev\misc1',
SIZE = 50661MB,
FILEGROWTH = 0),
(
NAME = MSSQL_misc2,
FILENAME = 'c:\dev\misc2',
SIZE = 50661MB,
FILEGROWTH = 0),
(
NAME = MSSQL_misc3,
FILENAME = 'c:\dev\misc3',
SIZE = 50661MB,
FILEGROWTH = 0),
(
NAME = MSSQL_misc4,
FILENAME = 'c:\dev\misc4',
SIZE = 50661MB,
FILEGROWTH = 0),
(
NAME = MSSQL_misc5,
FILENAME = 'c:\dev\misc5',
SIZE = 50661MB,
FILEGROWTH = 0),
FILEGROUP MSSQL_cs_fg
(
NAME = MSSQL_cs1,
FILENAME = 'c:\dev\cs1',
SIZE = 67548MB,
FILEGROWTH = 0),
(
NAME = MSSQL_cs2,
FILENAME = 'c:\dev\cs2',
SIZE = 67548MB,
FILEGROWTH = 0),
(
NAME = MSSQL_cs3,
FILENAME = 'c:\dev\cs3',
SIZE = 67548MB,
FILEGROWTH = 0)

```

```

(
NAME = MSSQL_tpcc_log,
FILENAME = 'c:\dev\log',
SIZE = 24480MB,
FILEGROWTH = 0)
COLLATE Latin1_General_BIN
go
-- Store ending time
-----
update tpcc_timer
set end_date = (select convert(char(30),
getdate(), 21))
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
--
-- Creates TPC-C tables
-----
SET ANSI_NULL_DFLT_OFF ON
go
use tpcc
go

```

[illegible]

```
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
s.ytd
int,
s.order_cnt
smallint,
s.remote_cnt
smallint,
s.data
char(30)
) on MSSQL_CS_fg
go
```

idxcuscsl.sql

```
-----
--
-- File:      IDXCUSCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver.
--           Copyright Microsoft, 2001
--
-- table      -- Creates clustered index on customer
--
-- use tpcc
go
declare      @startdate datetime,
             @enddate datetime
select      @startdate = getdate()
select      'Start date:',
             convert(varchar(30),@startdate,21)
if exists ( select name from sysindexes where name =
             'customer_cl' )
drop index customer.customer_cl
create unique clustered index customer_cl on
customer(c_w_id,c_d_id,c_id)
on MSSQL_CS_fg
select      @enddate = getdate()
select      'End date:',
             convert(varchar(30),@enddate,21)
select      'Elapsed time (in seconds):',
             datediff(second,@startdate,
             @enddate)
go
```

idxcuscnc.sql

Itauec Philco TPC-C FULL DISCLOSURE REPORT
© 2003 Itauec Philco, all rights reserved

```
-----
--
-- File:      IDXCUSNC.SQL
--           Microsoft TPC-C Benchmark Kit Ver.
--           Copyright Microsoft, 2001
--
-- Customer table      -- Creates non-clustered index on
--
-- use tpcc
go
declare      @startdate datetime,
             @enddate datetime
select      @startdate = getdate()
select      'Start date:',
             convert(varchar(30),@startdate,21)
if exists ( select name from sysindexes where name =
             'customer_nc1' )
drop index customer.customer_nc1
create unique nonclustered index customer_nc1 on
customer(c_w_id,c_d_id,c_last,c_first,c_id)
on MSSQL_CS_fg
select      @enddate = getdate()
select      'End date:',
             convert(varchar(30),@enddate,21)
select      'Elapsed time (in seconds):',
             datediff(second,@startdate,
             @enddate)
go
```

idxdiscsl.sql

```
-----
--
-- File:      IDXDISCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver.
--           Copyright Microsoft, 2001
--
-- table      -- Creates clustered index on district
--
-- use tpcc
go
declare      @startdate datetime,
             @enddate datetime
```

```
select      @startdate = getdate()
select      'Start date:',
             convert(varchar(30),@startdate,21)
if exists ( select name from sysindexes where name =
             'district_cl' )
drop index district.district_cl
create unique clustered index district_cl on
district(d_w_id,d_id)
with fillfactor=100 on MSSQL_misc_fg
select      @enddate = getdate()
select      'End date:',
             convert(varchar(30),@enddate,21)
select      'Elapsed time (in seconds):',
             datediff(second,@startdate,
             @enddate)
go
```

idxitmcl.sql

```
-----
--
-- File:      IDXITMCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver.
--           Copyright Microsoft, 2001
--
-- table      -- Creates clustered index on item
--
-- use tpcc
go
declare      @startdate datetime,
             @enddate datetime
select      @startdate = getdate()
select      'Start date:',
             convert(varchar(30),@startdate,21)
if exists ( select name from sysindexes where name =
             'item_cl' )
drop index item.item_cl
create unique clustered index item_cl on item(i_id)
on MSSQL_misc_fg
select      @enddate = getdate()
select      'End date:',
             convert(varchar(30),@enddate,21)
select      'Elapsed time (in seconds):',
             datediff(second,@startdate,
             @enddate)
go
```

idxnodcl.sql

```
-----
--
-- File: IDXNODCL.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver.
-- Copyright Microsoft, 2001
--
--
-- new_order table
-- Creates clustered index on
--
-----
use tpcc
go
declare @startdate datetime, @enddate datetime,
select @startdate = getdate(),
select 'Start date:', convert(varchar(30),@startdate, 21)
if exists ( select name from sysindexes where name =
'new_order_ci' )
drop index new_order.new_order_ci
create unique clustered index new_order_ci on
new_order(o_w_id, o_d_id, o_i_id)
on MSSQL_misc_fg
select @enddate = getdate()
select 'End date:', convert(varchar(30),@enddate, 21)
select 'Elapsed time (in seconds):',
datediff(second, @startdate,
@enddate)
go
idxnodcl.sql
-----
--
-- File: IDXODLCL.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver.
-- Copyright Microsoft, 2001
--
--
-- order_line table
-- Creates clustered index on
--
-----
use tpcc
go
declare @startdate datetime, @enddate datetime,
select @startdate = getdate(),
select 'Start date:',
```

```
convert(varchar(30),@startdate, 21)
if exists ( select name from sysindexes where name =
'order_line_ci' )
drop index order_line.order_line_ci
create unique clustered index order_line_ci on
order_line(o_l_w_id, o_l_d_id, o_l_o_id, o_l_number)
on MSSQL_misc_fg
select @enddate = getdate()
select 'End date:', convert(varchar(30),@enddate, 21)
select 'Elapsed time (in seconds):',
datediff(second, @startdate,
@enddate)
go
```

idxordcl.sql

```
-----
--
-- File: IDXORDCL.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver.
-- Copyright Microsoft, 2001
--
--
-- Creates clustered index on orders
--
-----
table
--
-----
use tpcc
go
declare @startdate datetime, @enddate datetime,
select @startdate = getdate(),
select 'Start date:',
convert(varchar(30),@startdate, 21)
if exists ( select name from sysindexes where name =
'orders_ci' )
drop index orders.orders_ci
create unique clustered index orders_ci on
orders(o_w_id, o_d_id, o_i_id)
on MSSQL_misc_fg
select @enddate = getdate()
select 'End date:', convert(varchar(30),@enddate, 21)
select 'Elapsed time (in seconds):',
datediff(second, @startdate,
@enddate)
go
```

idxstkcl.sql

```
-----
--
-- File: IDXSTKCL.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver.
-- Copyright Microsoft, 2001
--
--
-- table
-- Creates clustered index on stock
--
-----
use tpcc
go
declare @startdate datetime, @enddate datetime,
select @startdate = getdate(),
select 'Start date:', convert(varchar(30),@startdate, 21)
if exists ( select name from sysindexes where name =
'stock_ci' )
drop index stock.stock_ci
create unique clustered index stock_ci on stock(s_i_id,
s_w_id)
on MSSQL_cs_fg
select @enddate = getdate()
select 'End date:', convert(varchar(30),@enddate, 21)
select 'Elapsed time (in seconds):',
datediff(second, @startdate,
@enddate)
go
```

idxwarcl.sql

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

```
go
declare @startdate datetime,
        @enddate datetime,
        @startdate datetime,
        @enddate datetime
select @startdate = getdate()
select 'Start date:',
       convert(varchar(30),@startdate, 21)
if exists ( select name from sysindexes where name =
            'warehouse_cl' )
drop index warehouse.warehouse_cl on
warehouse(warehouse_id)
with fillfactor=100 on MSSQL_misc_fg
select @enddate = getdate()
select 'End date: ',
       convert(varchar(30),@enddate, 21)
select 'Elapsed time (in seconds): ',
       datediff(second, @startdate,
@enddate)
go
```

idxordnc.sql

```
-----
--
-- File:      IDXORDNC.SQL
--
-- 4.41      Microsoft TPC-C Benchmark Kit Ver.
--          Copyright Microsoft, 2001
--
--
--          Creates non-clustered index on
--          orders table
--
-----
use tpcc
go
declare @startdate datetime,
        @enddate datetime
select @startdate = getdate()
select 'Start date:',
       convert(varchar(30),@startdate, 21)
if exists ( select name from sysindexes where name =
            'orders_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, 21)
select 'Elapsed time (in seconds): ',
       datediff(second, @startdate, @enddate)
```

```
@enddate)
go
```

idxhisc.sql

```
-----
--
-- File:      IDXHISCL.SQL
--
-- 4.50      Microsoft TPC-C Benchmark Kit Ver.
--          Copyright Microsoft, 2003
--
--          Creates clustered index on history
--          table
--
-- CAUTION: This index is only beneficial for
-- systems with 8 or more processors.
--
-- CAUTION: It may negatively impact performance
-- on processors.
--
-----
use tpcc
go
declare @startdate datetime,
        @enddate datetime,
        @startdate datetime,
        @enddate datetime
select @startdate = getdate()
select 'Start date:',
       convert(varchar(30),@startdate, 21)
if exists ( select name from sysindexes where name =
            'history_cl' )
drop index history.history_cl
create unique clustered index history_cl on
history(h_c_w_id, h_date, h_c_d_id, h_c_id, h_amount)
on MSSQL_misc_fg
select @enddate = getdate()
select 'End date: ',
       convert(varchar(30),@enddate, 21)
select 'Elapsed time (in seconds): ',
       datediff(second, @startdate,
@enddate)
go
```

```
-----
--
-- File:      IDXODLCL.SQL
--
-- 4.50      Microsoft TPC-C Benchmark Kit Ver.
--          Copyright Microsoft, 2003
--
--          Creates clustered index on
--          order_line table
--
-----
use tpcc
go
declare @startdate datetime,
        @enddate datetime,
        @startdate datetime,
        @enddate datetime
select @startdate = getdate()
select 'Start date:',
       convert(varchar(30),@startdate, 21)
if exists ( select name from sysindexes where name =
            'order_line_cl' )
drop index order_line.order_line_cl
create unique clustered index order_line_cl on
order_line(o_w_id, o_d_id, o_l_o_id, o_l_number)
on MSSQL_misc_fg
select @enddate = getdate()
select 'End date: ',
       convert(varchar(30),@enddate, 21)
select 'Elapsed time (in seconds): ',
       datediff(second, @startdate,
@enddate)
go
```

dbopt1.sql

```
--
-- File:      DBOPT1.SQL
--
-- 4.41      Microsoft TPC-C Benchmark
--          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
```

dbopt2.sql

```
USE tpcc
```

CHECKPOINT

```
sp_configure 'allow updates', 1
```

RECONFIGURE WITH OVERRIDE

```
DECLARE @msn varchar(50)
```

```
--
--      OPTIONS FOR SQL SERVER 2000
--
--      set option values for user-defined indexes
--
```

```
SET @msg
PRINT @msg
SET @msg
PRINT @msg
SET @msg
PRINT @msg
```

```
EXEC sp_indexoption 'customers', '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 'order_id', 'disallowRowLocks', TRUE
EXEC sp_indexoption 'item', 'disallowRowLocks', TRUE
EXEC sp_indexoption 'item', 'disallowPageLocks', TRUE
```

```
print ', '
print '*****'
print 'Pre-specified Locking Hierarchy: '
print 'Lockflag = 0 ==> No pre-specified'
print 'hierarchy'
print 'Lockflag = 1 ==> Lock at Page-level then'
print 'Table-level'
```

Itautec Philco TPC-C FULL DISCLOSURE REPORT
© 2003 Itautec Philco, all rights reserved

```
SELECT name, lockflags
```

ORDER BY LockFlags asc

```
sp_configure 'allow updates', 0
```

RECONFIGURE WITH OVERRIDE

```
EXEC sp_dboption tpcc,
    'auto update statistics',
    'auto create statistics',
    FALSE
EXEC sp_dboption tpcc,
    'auto update statistics',
    'auto create statistics',
    FALSE
```

GO

FALSE

```
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
```

backup.sql

```
-- -- File: BACKUP.SQL -- --
-- -- Microsoft TPC-C Benchmark Kit Ver. 4.41 -- --
-- -- Copyright Microsoft, 2003 -- --
```

```
declare @startdate datetime,
```

```
select @startdate = getdate()
select 'Start date: ',
       convert(varchar(30),@startdate, 21)
```

```
dump database tpcc to tpcc4700w_1_5, tpcc4700w_2_5,  
tpcc4700w_3_5, tpcc4700w_4_5, tpcc4700w_5_5 with init,  
stats = 1
```

```
select @enddate = getdate()
```

November, 2003

restore.sql

100

-- 4.41
--
-- Microsoft TPC-C Benchmark kit ver.
--
-- Copyright Microsoft, 2003

```
declare @startdate datetime,
         @enddate datetime
```

```
select @startdate = getdate()
```

```
select start_date, convert(varchar(30),@startdate, 21)
Load database tpcc4700w_1_5, tpcc4700w_2_5,
tpcc4700w_3_5, tpcc4700w_4_5, tpcc4700w_5_5 with
replace, stats = 1
```

```
select @enddate = getdate()
```

```

select      end date:      convert(varchar(30),@enddate, 21)
select      'Elapsed time (in seconds):',
            datediff(second, @startdate,
                        @enddate)

```

qo

backupdev.sql

```
-- File: BACKUPDEV.S
```

```
-- 4.41
--
-- Microsoft IPC-C Benchmark Kit ver.
--
-- Copyright Microsoft, 2003
--
```

use master

1

```
-----
-- create backup devices
exec sp_addumpdevice
p'disk', 'tpcc4700w_1_5', 'c:\dev\backup1\tpcc4700w_1_5.dm
exec sp_addumpdevice
p'disk', 'tpcc4700w_2_5', 'c:\dev\backup2\tpcc4700w_2_5.dm
exec sp_addumpdevice
p'disk', 'tpcc4700w_3_5', 'c:\dev\backup3\tpcc4700w_3_5.dm
exec sp_addumpdevice
p'disk', 'tpcc4700w_4_5', 'c:\dev\backup4\tpcc4700w_4_5.dm
exec sp_addumpdevice
p'disk', 'tpcc4700w_5_5', 'c:\dev\backup5\tpcc4700w_5_5.dm
go
```

removedb.sql

```
-----
--
-- File: REMOVEDB.SQL
--
4.50 Microsoft TPC-C Benchmark Kit Ver.
Copyright Microsoft, 2003
--
--
--
use master
go
-- remove any existing database and backup files
if exists(select name from master..sysdatabases where
name = 'tpcc')
exec sp_dbbremove tpcc, dropdev
go
if exists(select name from master..sysdevices where
name like 'tpcc4700w%')
exec sp_dropdevice 'tpcc4700w_1_5', 'tpcc4700w_2_5',
'tpcc4700w_3_5', 'tpcc4700w_4_5', 'tpcc4700w_5_5',
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
go
use tpcc
go
if exists ( select name from sysobjects where name =
'tpcc_neworder' )
drop procedure tpcc_neworder
go
create proc tpcc_neworder
smallint, @o_o1_cnt tinyint, @o_w_id
tinyint, @d_id
int, @c_id
@o_o1_cnt tinyint,
@o_all_local tinyint,
@o1_qty1 int = 0, @s_w_id1 smallint = 0,
@o1_qty2 int = 0, @s_w_id2 smallint = 0,
@o1_qty3 int = 0, @s_w_id3 smallint = 0,
@o1_qty4 int = 0, @s_w_id4 smallint = 0,
@o1_qty5 int = 0, @s_w_id5 smallint = 0,
@o1_qty6 int = 0, @s_w_id6 smallint = 0,
@o1_qty7 int = 0, @s_w_id7 smallint = 0,
@o1_qty8 int = 0, @s_w_id8 smallint = 0,
@o1_qty9 int = 0, @s_w_id9 smallint = 0,
@o1_qty10 int = 0, @s_w_id10 smallint = 0,
@o1_qty11 int = 0, @s_w_id11 smallint = 0,
@o1_qty12 int = 0, @s_w_id12 smallint = 0,
@o1_qty13 int = 0, @s_w_id13 smallint = 0,
@o1_qty14 int = 0, @s_w_id14 smallint = 0,
@o1_qty15 int = 0, @s_w_id15 smallint = 0,
```

```
as
declare @w_tax numeric(4,4),
@o_tax numeric(4,4),
@c_last char(16),
@c_credit char(2),
@o_discount numeric(4,4),
@i_price numeric(5,2),
@i_name char(24),
@i_data char(50),
@o_entry_d datetime,
@o_entry_id int,
@o_remote_flag tinyint,
@o_quantity smallint,
@o_dist char(50),
@o_dist char(24),
@i_no int,
@o_id int,
@commit_flag tinyint,
@o1_s_w_id int,
@o1_qty smallint,
@o1_number int,
@c_id_local int
begin
begin transaction n
-- get district tax and next available order id and
update
-- plus initialize local variables
update set district
@o_id = d_tax,
d_next_o_id = d_next_o_id,
@o_entry_d = d_next_o_id + 1,
@i_no = getdate(),
@commit_flag = 1,
d_w_id = @w_id and
d_id = @d_id
where
-- process orderlines
while (@i_no < @o_o1_cnt)
begin
select @i_no = @i_no + 1
-- set i_id, s_w_id, and qty for this lineitem
select @i_id = case @i_no
when 1 then
when 2 then
when 3 then
when 4 then
when 5 then
when 6 then
when 7 then
when 8 then
```

```

        @i_id9
        @i_id10
        @i_id11
        @i_id12
        @i_id13
        @i_id14
        @i_id15

        when 9 then
        when 10 then
        when 11 then
        when 12 then
        when 13 then
        when 14 then
        when 15 then
        end,

        @li_s_w_id = case @li_no
        when 1
            when 2
            when 3
            when 4
            when 5
            when 6
            when 7
            when 8
            when 9
            when 10
            when 11
            when 12
            when 13
            when 14
            when 15
            end,

        @li_qty = case @li_no
        when 1 then
        when 2 then
        when 3 then
        when 4 then
        when 5 then
        when 6 then
        when 7 then
        when 8 then
        when 9 then
        when 10 then
        when 11 then

```

```

        @o1_qty12
        @o1_qty13
        @o1_qty14
        @o1_qty15

        -- get item data (no one updates item)
        select
            @i_price = i_price,
            @i_name = i_name,
            @i_data = i_data,
        from
            item (tablock)
        where
            i_id = @i_id

        -- update stock values
        update
            stock
        set
            s_ytd + @li_qty,
            s_quantity = s_quantity - @li_qty +
                case when (s_quantity - @li_qty < 10)
                then 91 else 0 end,
            s_order_cnt + 1,
            s_remote_cnt = s_remote_cnt
        else 1 end,
        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
    where
        s_i_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)

```

```

        values(@o_id,
            @d_id,
            @w_id,
            @i_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 'g'
            else 'g' end,
            @li_qty
        end
        else
        begin
        -- no item (or stock) found - triggers rollback
        condition
        select '',0,'',0,0
        select @commit_flag = 0

        end

        -- get customer last name, discount, and credit rating
        select
            @c_last = c_last,
            @c_discount = c_discount,
            @c_credit = c_credit,
            @c_id_local = c_id,
            customer (repeat(read)
                = @c_id and
                c_id
                = @w_id and
                c_d_id
                = @d_id)

        -- insert fresh row into orders table
        insert into orders values (
            @o_id,
            @w_id,
            @c_id_local,
            @o_entry_d,
            0,

```



```

-- 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,
  @o_balance,
  @o_id

go

-- 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
as
smallint

declare @d_id      tinyint,
        @c_id      int,
        @total     numeric(12,2),
        @oid1      int,
        @oid2      int,
        @oid3      int,
        @oid4      int,
        @oid5      int,
        @oid6      int,
        @oid7      int,

```

delivery.sql

```

        @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
      from
        new_order (serializable)
      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
      where
        new_order
          no_w_id =
          no_d_id =
          no_o_id =

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

      update
      orders
      set
        o_carrier_id,
        @c_id
      where
        o_w_id =
        o_d_id =
        o_id = @o_id

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

      update
      order_line
      set
        ol_delivery_d
          = getdate(),
          @total
      where
        ol_w_id =
        ol_d_id =
        ol_o_id = @o_id

      -- accumulate lineitem amounts for this order into
      customer

      update
      set
        customer
          c_balance + @total,

```

```

        c_delivery_cnt
      = c_delivery_cnt + 1
    where
      = @w_id and
      = @d_id and
      = @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

```

stocklev.sql

```

go
create proc tpc_c_stocklevel @w_id
    smallint,
    tinyint,
    smallint
as
declare @o_id_low int,
    @o_id_high int
select @o_id_low = (d_next_o_id - 20), @d_id
    = (d_next_o_id - 1)
from distri
where d_w_id
    = @w_id and
    d_id
    = @d_id
select count(distinct(s_i_id))
    stock, order_line
from o_l_w_id
where o_l_w_id
    = @w_id and
    o_l_d_id
    between @o_id_low and
    @o_id_high and
    o_l_o_id
    = o_l_w_id and
    s_l_id
    = o_l_d_id and
    s_quantity
    < @threshold

```

Loader Source Code

tpcc.h

```

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

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

// General headers
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.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 ""
#define USER ""
#define PASSWORD ""
// Default loader arguments
#define BATCH 10000
#define DEFLOADPACKSIZE 32768
#define LOADER_RES_FILE "c:\\MSTPCC.440\\SETUP\\logs\\load.out"
#define LOG_PATH "c:\\MSTPCC.440\\SETUP\\LOGS\\";
#define LOADER_NURAND_C 123
#define DEF_STARTING_WAREHOUSE 1
#define BUILD_INDEX // build both data and indexes
#define INDEX_ORDER // 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;
    WAREHOUSE, DISTRICT, and STOCK
    BOOL table_customer;
    Loading CUSTOMER and HISTORY
    BOOL table_orders;
    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;
char *TPCC_LD_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 16
#define ADDRESS_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 4
#define S_DATA_LEN 24
#define S_NAME_LEN 50
#define D_NAME_LEN 16
#define FIRST_NAME_LEN 16
#define MIDDLE_NAME_LEN 16
#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 25
#define C_SINCE_LEN 23
#define H_DATE_LEN 23
#define OL_DELIVERY_D_LEN 23
#define Q_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 PadString();

tpccldr.c File: TPCCCLR.C Microsoft TPC-C
//
// Kit Ver. 4.41 Copyright
// Microsoft, 1996, 1997, 1998, 2000, 2001
// Purpose: Source file for TPC-C database
// loader

// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS 100000
#define CUSTOMERS_SCALE_DOWN 3000
#define CUSTOMERS_SCALE_DOWN 30
#define DISTRICT_PER_WAREHOUSE 10
#define ORDERS_PER_DISTRICT 3000
#define ORDERS_SCALE_DOWN 30
#define MAX_CUSTOMERS_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 Distinct();

void LoadCustomer();
void LoadCustomerInit();
void LoadCustomerBuf();
void LoadCustomerTable();
void LoadHistoryTable();

void LoadOrders();
void LoadOrdersInit();
void LoadOrdersBuf();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void LoadOrderLineTable();
void LoadPermutation();
void LoadCheckForCommit();
void LoadOpenConnections();
void LoadButIdIndex();
void LoadFormatDate();

// Shared memory structures
typedef struct
{
    long ol_i_id;
    long ol_supply_w_id;
    short ol_quantity;
}

```

```

double ol_amount;
char ol_dist_info[DIST_INFO_LEN+1];
char ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
    long o_id;
    short o_d_id;
    short o_w_id;
    short o_carrier_id;
    short o_ol_cnt;
    short o_attr_local;
} 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
    char 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 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;
HSTMT o_hstmt1;
HSTMT o_hstmt2;
HSTMT o_hstmt3;
ORDERS_STRUCT orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];
long new_order_rows_loaded;
long history_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;
max_items;
customers_per_district;
orders_per_district;
first_new_order;
last_new_order;

TPCCCLR_ARGS *aptr, args;

//=====
// Function name: main
//=====

```

```

int main(int argc, char **argv)
{
    DWORD dwThreadId[MAX_MAIN_THREADS];
    HANDLE hThread[MAX_MAIN_THREADS];
    FILE *floader;
    char buffer[255];
    int i;

    for (i=0; i<MAX_MAIN_THREADS; i++)
        hThread[i] = NULL;

    printf("\n*****\n");
    printf("\n* Microsoft SQL Server\n");
    printf("\n*");
    printf("\n* TPC-C BENCHMARK KIT: Database loader\n");
    printf("\n* Version %s\n", TPCKIT_VER);
    printf("\n*");
    printf("\n*****\n");

    // process command line arguments
    argc = &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_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;
    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   bcpint[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("idxitmc1");

    Initstring(i_name, I_NAME_LEN+1);
    Initstring(i_data, I_DATA_LEN+1);

    sprintf(name, "%s..%s", aptr->database,
"item");

    "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(bcpint, "tablock, order
(i_id), ROWS_PER_BATCH = 10000");
        rc = bcp_control(i_hdbc1, BCPHINTS,
(void*) bcpint);
        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);
        RandomNumber(100L, 10000L)/100.0;
        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);
    }
}

```

```

if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

>index_order == 1))
{
    sprintf(bcp_hint, "tablock, order
(w_id), ROWS_PER_BATCH = %d", aptr->num_warehouses);
    rc = bcp_control(w_hdbc1, BCPHINTS,
(void*) bcp_hint);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
}

rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 1);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0,
SQL_NAME_LEN, NULL, 0, 0, 2);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1,
0, ADDRESS_LEN, NULL, 0, 0, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2,
0, ADDRESS_LEN, NULL, 0, 0, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0,
ADDRESS_LEN, NULL, 0, 0, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0,
STATE_LEN, NULL, 0, 0, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0,
ZIP_LEN, NULL, 0, 0, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
if (rc != SUCCEEDED)
    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++)
{
    MakeA1phastring(6, 10, w_NAME_LEN,
w_name);
    MakeAddress(w_street_1, w_street_2,
w_city, w_state, w_zip);

```

```

RandomNumber(0L, 20000L))/10000.00;

w_ytd = 300000.00;

rc = bcp_sendrow(w_hdbc1);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

warehouse_rows_loaded++;
CheckForCommit(w_hdbc1, 1, hstmt1,
warehouse_rows_loaded, "warehouse", &time_start);

rcint = bcp_done(w_hdbc1);
if (rcint < 0)
    HandleErrorDBC(w_hdbc1);

printf("Finished loading warehouse
table.\n");

// if build index after load...
if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
    buildindex("idxwarc1");

stock_rows_loaded = 0;
district_rows_loaded = 0;

District();
Stock();

}

//=====
// Function : District
//=====
void District()
{
    short d_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 rc;
    RETCODE rcint;
    DBINT rcint;
    char bcp_hint[128];
    char err_log_path[256];

    // Seed with unique number
    seed(4);

    printf("Loading district table...\n");

    // build index before load

```

```

>index_order == 1))
    BuildIndex("idxdisc1");

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 != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
{
    sprintf(bcp_hint, "tablock, order
(d_w_id, d_id), ROWS_PER_BATCH = %u", (aptr-
>num_warehouses * 10);
    rc = bcp_control(w_hdbc1, BCPHINTS,
(void*) bcp_hint);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
}

rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 1);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0,
D_NAME_LEN, NULL, 0, 0, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1,
0, ADDRESS_LEN, NULL, 0, 0, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2,
0, ADDRESS_LEN, NULL, 0, 0, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0,
ADDRESS_LEN, NULL, 0, 0, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0,
STATE_LEN, NULL, 0, 0, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0,
ZIP_LEN, NULL, 0, 0, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

```

```

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

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

rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 11);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

d_ytd = 30000.0;
d_next_o_id = orders_per_district+1;

time_start = (TimeNow() / MILLI);
for (w_id = aptr->starting_warehouse; w_id <=
aptr->num_warehouses; w_id++)
{
    d_w_id = w_id;
    for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
    {
        Makea1phastring(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 =
            if (rc != SUCCEEDED)
                HandleErrorDBC(w_hdbc1);

        district_rows_loaded++;
        checkforcommit(w_hdbc1,
&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("idxdsc1");

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_vid;
    short s_order_cnt;
    short s_remote_cnt;
    char s_data[s_DATA_LEN+1];
    short len;
    long name[20];
    long time_start;
    RETCODE rc;
    DBINT rcint;
    char bcpint[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("idxstk1");

    printf(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 != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
    {
        printf(bcpint, "tablock, order
(s_i_id, s_w_id), ROWS_PER_BATCH = %u", (aptr-
>num_warehouses * 100000));
        rc = bcp_control(w_hdbc1, BCPHINTS,
(void*) bcpint);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
    }

    rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
}

```

```

rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0,
S_DIST_LEN, NULL, 0, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0,
S_DIST_LEN, NULL, 0, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0,
S_DIST_LEN, NULL, 0, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0,
S_DIST_LEN, NULL, 0, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0,
S_DIST_LEN, NULL, 0, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0,
S_DIST_LEN, NULL, 0, 9);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0,
S_DIST_LEN, NULL, 0, 10);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0,
S_DIST_LEN, NULL, 0, 11);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0,
S_DIST_LEN, NULL, 0, 12);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0,
S_DIST_LEN, NULL, 0, 13);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 14);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 15);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *)
&s_remote_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
16);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

```

```

s_data_len, NULL, 0, 0, 17);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

s_ytd = bcp_bind(w_hdbc1, (BYTE *) s_data, 0,
    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++)
    {
        (short)RandomNumber(10L, 100L);
        len = s_quantity =
        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);

        MakeOriginalAlphastring(26, 50, s_data_len, s_data, 10);

        bcp_sendrow(w_hdbc1);
        if (rc != SUCCEEDED)
            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))

```

```

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 *cp;
    bcp_hist[128];
    cond[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("idxcustcl");
        // 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 // if less than 8 processors, do
        "NUMBER_OF_PROCESSORS" );
        if ( num_procs = atoi(getenv(
            "NUMBER_OF_PROCESSORS" ))
            if ( num_procs >= 8 )
                BuildIndex("idxhisc1");
    }
}

```

```

// 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 != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcp_hint, "tablock, order
    (c_w_id, c_d_id, c_id), ROWS_PER_BATCH = %d", (aptr->num_warehouses * 3000));
    rc = bcp_control(c_hdbc1, BCPHINTS,
        (void*) bcp_hint);
    if (rc != SUCCEEDED)
        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 != SUCCEEDED)
    HandleErrorDBC(c_hdbc2);

sprintf(bcp_hint, "tablock");
rc = bcp_control(c_hdbc2, BCPHINTS, (void*)
    bcp_hint);
if (rc != SUCCEEDED)
    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);
    }
}

```

<pre> CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE) LoadCustomerTable, &customer_time_start, 0, &dwThreadId[0]); if (hThread[0] == NULL) { printf("Error, thread is creating creating thread = 0.\n"); exit(-1); } // Start History table thread history table for: d_id = %d, w_id = %d\n", d_id, w_id); CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE) LoadHistoryTable, &history_time_start, 0, &dwThreadId[1]); if (hThread[1] == NULL) { printf("Error, thread is creating creating thread = 1.\n"); exit(-1); } waitForSingleObject(hThread[0], INFINITE); hThread[1], INFINITE); (CloseHandle(hThread[0]) == FALSE) { printf("Error, failed in closing customer thread handle with errno: %d\n", GetLastError()); } if (CloseHandle(hThread[1]) == FALSE) { </pre>	<pre> failed in closing history thread handle with errno: %d\n", GetLastError()); } } // Flush the bulk connection rcint = bcp_done(c_hdbc2); if (rcint < 0) HandleErrorDBC(c_hdbc1); rcint = bcp_done(c_hdbc2); if (rcint < 0) HandleErrorDBC(c_hdbc2); printf("Finished loading customer table.\n"); if ((aptr->build_index == 1) && (aptr- >index_order == 0)) { BuildIndex("idxcusc1"); 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("idxhisc1"); } // build non-clustered index if (aptr->build_index == 1) BuildIndex("idxcuscnc"); // Output the NURAND used for the loader into C_FIRST for C_ID = 1, and C_D_ID = 1 // C_W_ID = 1, and C_D_ID = 1 Q"update customer set c_first = %s -d%\$ -e - c_id = 1 and w_id = %s -s%\$ -c_load = %d' where logs\nurand_load.log" -d%\$ -e - Q"update customer set c_first = %s -d%\$ -e - c_id = 1 and 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); </pre>	<pre> 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,""); } } //===== </pre>
--	---	---

```

// Function : CustomerBufLoad
// Fills shared buffer for HISTORY and CUSTOMER
//=====
void CustomerBufLoad(int d_id, int w_id)
{
    long CUSTOMER_SORT_STRUCT c[CUSTOMERS_PER_DISTRICT];
    for (i=0;i<customers_per_distrcit;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("\n...Loading customer buffer for: d_id
        = %d, w_id = %d\n", d_id, w_id);
    for (i=0;i<customers_per_distrcit;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] = '0';
        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);
    }
}

```

```

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
{
    int i;
    long c_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_discount;
    // fix to avoid ODBC float to numeric
    conversion problem.
    double c_balance;
    char c_ytd_payment;
    double c_payment_cnt;
    short c_delivery_cnt;
    char c_data[C_DATA_LEN+1];
}

```

char	RETCODE	c_since[C_SINCE_LEN+1];
rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLT4, 1);	if (rc != SUCCEED) HandleErrorDBC(c_hdbc1);	
rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLT2, 2);	if (rc != SUCCEED) HandleErrorDBC(c_hdbc1);	
rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLT2, 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 != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 16);
        if (rc != SUCCEEDED)
            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 != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, 5,
NULL, 0, SQLCHARACTER, 17);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 18);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 19);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 20);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, 500,
NULL, 0, 0, 21);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        for (i = 0; i < customers_per_district; i++)
        {
            c_id = customer_buf[i].c_id;
            c_d_id = customer_buf[i].c_d_id;
            c_w_id = customer_buf[i].c_w_id;

            strcpy(c_first,
customer_buf[i].c_first);
            strcpy(c_middle,
customer_buf[i].c_middle);
            strcpy(c_last,
customer_buf[i].c_last);
            strcpy(c_street_1,
customer_buf[i].c_street_1);
            strcpy(c_street_2,
customer_buf[i].c_street_2);
            strcpy(c_city,
customer_buf[i].c_city);
            strcpy(c_state,
customer_buf[i].c_state);
            strcpy(c_zip,
customer_buf[i].c_zip);
            strcpy(c_phone,
customer_buf[i].c_phone);
            strcpy(c_credit,
customer_buf[i].c_credit);

```

```

        Formatdate(&c_since);

        customer_buf[i].c_credit_lim =
            c_credit_lim;
        customer_buf[i].c_discount =
            c_discount;

        // fix to avoid ODBC float to
        numeric conversion problem.
        customer_buf[i].c_balance =
            strcpy(c_balance,
customer_buf[i].c_balance);

        customer_buf[i].c_ytd_payment =
            c_ytd_payment;
        customer_buf[i].c_payment_cnt =
            c_payment_cnt;
        customer_buf[i].c_delivery_cnt =
            c_delivery_cnt;
        customer_buf[i].c_data =
            strcpy(c_data,
customer_buf[i].c_data);

        // Send data to server
        rc = bcp_sendrow(c_hdbc1);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        customer_rows_loaded++;
        CheckForCommit(c_hdbc1, c_hstmt1,
customer_rows_loaded, "customer", &customer_time_start-
>time_start);
    }

    //=====
    // Function : LoadHistoryTable
    //=====
    void LoadHistoryTable(LOADER_TIME_STRUCT
*history_time_start)
    {
        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_DATA_LEN+1];
        RETCODE rc;

        rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

```

```

        rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 4);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 5);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0,
H_DATE_LEN, NULL, 0, SQLCHARACTER, 6);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 7);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0,
H_DATA_LEN, NULL, 0, 8);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

        for (i = 0; i < customers_per_district; i++)
        {
            c_id = customer_buf[i].c_id;
            c_d_id = customer_buf[i].c_d_id;
            c_w_id = customer_buf[i].c_w_id;
            h_amount =
customer_buf[i].h_amount;
            strcpy(h_data,
customer_buf[i].h_data);

            Formatdate(&h_date);

            // send to server
            rc = bcp_sendrow(c_hdbc2);
            if (rc != SUCCEEDED)
                HandleErrorDBC(c_hdbc2);

            history_rows_loaded++;
            CheckForCommit(c_hdbc2, c_hstmt2,
history_rows_loaded, "history", &history_time_start-
>time_start);
        }
    }

    //=====
    // Function : LoadOrders
    //=====
    void LoadOrders()
    {
        LOADER_TIME_STRUCT
        loader_time_struct;
        orders_time_start;
        new_order_time_start;

```



```

failed in creating creating thread = 2.\n");
    }
    waitForsingleobject(
hthread[0], INFINITE );
hthread[1], INFINITE );
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;
    }
}

```

```

= 0;
    for (j=0;j<=14;j++)
    {
        orders_buf[i].o_ol[j].o_l_id = 0;
        orders_buf[i].o_ol[j].o_supply_w_id = 0;
        orders_buf[i].o_ol[j].o_quantity = 0;
        orders_buf[i].o_ol[j].o_amount = 0;
        strcpy(orders_buf[i].o_ol[j].o_l_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_c_id =
o_id+1;
        orders_buf[o_id].o_carrier_id =
o_id+1;
        cust[o_id+1];
        (short)RandomNumber(SL, 1SL);
        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].o_l_id = ol+1;
        orders_buf[o_id].o_ol[ol].o_l_id =
RandomNumber(1L, max_ol_items);
        orders_buf[o_id].o_ol[ol].o_supply_w_id =
w_id;
        orders_buf[o_id].o_ol[ol].o_quantity = 5;
        MakeAlphastring(24, 24,
&orders_buf[o_id].o_ol[ol].o_l_dist_info);
        data
        // Generate ORDER-LINE
        first_new_order
        {
            orders_buf[o_id].o_ol[ol].o_l_amount = 0;
            insure o_l_delivery_d set properly during load
            FormatDate(&orders_buf[o_id].o_ol[ol].o_l_deliv
ery_d);
        }
        else
        {
            orders_buf[o_id].o_ol[ol].o_l_amount =
RandomNumber(1.999999)/100.0;
            insure o_l_delivery_d set properly during load
            // Added to
            // odbc
            datetime format
            strcpy(orders_buf[o_id].o_ol[ol].o_l_delivery_
d,"1899-12-31 00:00:00.000");
        }
    }
}
//=====
// Function : LoadOrdersTable
//=====
void LoadOrdersTable(LOADER_TIME_STRUCT
%orders_time_start)
{

```

```

long int o_id; i;
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_VLEN_DATA, NULL, 0, SQLINT4, 1);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0,
SQL_VLEN_DATA, NULL, 0, SQLINT2, 2);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0,
SQL_VLEN_DATA, NULL, 0, SQLINT2, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0,
SQL_VLEN_DATA, NULL, 0, SQLINT4, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d,
0, o_ENTRY_D_LEN, NULL, 0, SQLCHARACTER, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0,
SQL_VLEN_DATA, NULL, 0, SQLINT2, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0,
SQL_VLEN_DATA, NULL, 0, SQLINT2, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0,
SQL_VLEN_DATA, NULL, 0, SQLINT2, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

for (i = 0; i < orders_per_district; i++)
{
    o_id = orders_buf[i].o_id;
    orders_buf[i].o_d_id = o_d_id;
    orders_buf[i].o_w_id = o_w_id;
    orders_buf[i].o_c_id = o_c_id;
    orders_buf[i].o_carrier_id = o_carrier_id;
    orders_buf[i].o_ol_cnt = o_ol_cnt;
    orders_buf[i].o_all_local = o_all_local;

    FormatDate(&o_entry_d);

```

```

// send data to server
rc = bcp_sendrow(o_hdbc1);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

orders_rows_loaded++;

CheckForCommit(o_hdbc1, o_hstmt1,
>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_VLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0,
SQL_VLEN_DATA, NULL, 0, SQLINT2, 2);

```

```

if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc2);

rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0,
SQL_VLEN_DATA, NULL, 0, SQLINT2, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc2);

for (i = first_new_order; i < last_new_order;
i++)
{
    o_id = orders_buf[i].o_id;
    o_d_id = orders_buf[i].o_d_id;
    o_w_id = orders_buf[i].o_w_id;
    rc = bcp_sendrow(o_hdbc2);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

    new_order_rows_loaded++;

    CheckForCommit(o_hdbc2, o_hstmt2,
    &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_VARCHAR_DATA, NULL, 0, SQLINT4, 1);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0,
SQL_VARCHAR_DATA, NULL, 0, SQLINT2, 2);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0,
SQL_VARCHAR_DATA, NULL, 0, SQLINT2, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &o_l, 0,
SQL_VARCHAR_DATA, NULL, 0, SQLINT4, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &o_l_i_id, 0,
SQL_VARCHAR_DATA, NULL, 0, SQLINT4, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &o_l_supply_w_id, 0,
SQL_VARCHAR_DATA, NULL, 0, SQLINT2, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *)
&o_l_delivery_d, 0, OL_DELIVERY_D_LEN, NULL, 0,
SQL_CHARACTER, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &o_l_quantity, 0,
SQL_VARCHAR_DATA, NULL, 0, SQLINT2, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &o_l_amount, 0,
SQL_VARCHAR_DATA, NULL, 0, SQLFLT8, 9);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0,
DIST_INFO_LEN, NULL, 0, 10);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

for (i = 0; i < orders_per_distinct; 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_l_cnt; j++)

```

```

orders_buf[i].o_o_l[j].o_l =
orders_buf[i].o_o_l[j].o_l_i_id;
orders_buf[i].o_o_l[j].o_l_i_id;
orders_buf[i].o_o_l[j].o_l_supply_w_id =
orders_buf[i].o_o_l[j].o_l_supply_w_id;
orders_buf[i].o_o_l[j].o_l_quantity =
orders_buf[i].o_o_l[j].o_l_quantity;
orders_buf[i].o_o_l[j].o_l_amount =
orders_buf[i].o_o_l[j].o_l_amount;

strcpy(ol_delivery_d, orders_buf[i].o_o_l[j].o_l_
_delivery_d);

strcpy(ol_dist_info, orders_buf[i].o_o_l[j].o_l_
_dist_info);

bcp_sendrow(o_hdbc3);
rc =
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

order_line_rows_loaded++;
CheckForCommit(o_hdbc3,
&order_line_time_start->time_start);
}
// rcint = bcp_batch(o_hdbc3);
// if (rcint < 0)
//     HandleErrorDBC(o_hdbc3);

if ((o_w_id == aprt->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 ((aprt->build_index == 1) &&
(aprt->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,
rows_loaded,
*table_name,
*time_start)
{
    long time_end, time_diff;
    // DBINT rcint;
    if ( !(rows_loaded % aprt->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\n",
in %ld sec - Total = %d (%.2f rps)\n",
table_name,
time_diff,
rows_loaded,
(float) aprt-
>batch / (time_diff ? time_diff : 1L));
        *time_start = time_end;
    }
}

return;

//=====
// Function : OpenConnections
//=====

```



```

    }
    }
    i++;
}

void FormatDate ( char* szTimeOutput )
{
    struct tm when;
    time_t now;
    time( &now );
    when = *localtime( &now );
    mktime( &when );

    // odbc datetime format
    strftime( szTimeOutput , 30 , "%Y-%m-%d
%H:%M:%S.000" , &when );
    return;
}

//=====
// Function : CheckDatabase
//=====
void CheckDatabase()
{
    RETCODE rc;

    char szDriverString[300];
    char szDriverStringOut[1024];
    char* i;
    int i;
    ExitFlag;

    SQL_SMALLINT
    cdbDriverStringOut;
    SQL_CHAR
    TabName[10];
    SQL_INTEGER
    TabNameInd;
    TabCount, TabCountInd;

    ExitFlag = 0;
    SQLAllocHandle(SQL_HANDLE_ENV,
    SQL_NULL_HANDLE, &env );
    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
    (void*)SQL_OV_ODBC3, 0 );
    &v_hdbc);
    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
    SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP,
    (void *)SQL_BCP_ON, SQL_IS_INTEGER );
    // Open connection to SQL Server

```

```

    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
    aptr->server,
    aptr->user,
    aptr->password,
    aptr->database );
    rc = SQLSetConnectAttr( v_hdbc,
    SQL_ATTR_PACKET_SIZE, (SQLPOINTER)aptr->pack_size,
    SQL_IS_INTEGER );
    if (rc != SQL_SUCCESS)
        HandleErrorDBC(v_hdbc);
    rc = SQLDriverConnect ( v_hdbc,
    NULL,
    (SQLCHAR*)&szDriverString[0] ,
    SQL_NTS,
    (SQLCHAR*)&szDriverStringOut[0],
    sizeof(szDriverStringOut),
    &cbDriverStringOut,
    SQL_DRIVER_NOPROMPT );
    // if the rc is SQL_ERROR, the the TPCC
    database probably does not exist
    if (rc == SQL_ERROR)
        printf("The database TPCC does not
appear to exist!\n");
    printf("\nCheck LOGS\\ directory
for database creation errors.\n");
    // cleanup database connections and
handles
    SQLFreeHandle(SQL_HANDLE_STMT,
    v_hstmt);
    SQLDisconnect(v_hdbc);
    SQLFreeHandle(SQL_HANDLE_DBC,
    v_hdbc);
    // since there is not a database,
    exit back to SETUP.CMD
    exit(1);
    if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc ,
    &v_hstmt) != SQL_SUCCESS )
        HandleErrorDBC(v_hdbc);
    if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG,
    &TabCount, 0, &TabCountInd) != SQL_SUCCESS )
        HandleErrorStmt(v_hstmt);
    // count the number of user tables from
sysobjects
    rc = SQLExecDirect(v_hstmt, "select count(*)
from sysobjects where xtype = 'U'", SQL_NTS);
    if ((rc != SQL_SUCCESS) && (rc !=
    SQL_SUCCESS_WITH_INFO))
        HandleErrorStmt(v_hstmt);

```

```

    if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
        HandleErrorStmt(v_hstmt);
    // if the number of tables is less than 9,
    select all the user tables in TPCC
    if (TabCount != 9)
    {
        SQLFreeHandle(SQL_HANDLE_STMT,
    v_hstmt);
        SQLAllocHandle(SQL_HANDLE_STMT,
    v_hdbc , &v_hstmt);
        if ( SQLBindCol(v_hstmt, 1,
    SQL_C_CHAR, &TabName, sizeof(TabName), &TabNameInd) !=
    SQL_SUCCESS )
            HandleErrorStmt(v_hstmt);
        // select the list of user tables
        into a result set
        rc = SQLExecDirect(v_hstmt, "select
* from sysobjects where xtype = 'U'", SQL_NTS);
        if ((rc != SQL_SUCCESS) && (rc !=
    SQL_SUCCESS_WITH_INFO))
            HandleErrorStmt(v_hstmt);
        // go through the result set and
        set the bitmap for each found table
        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';
                    case 'i':
                        TablesBitmap[7]
                        = '1';
                        break;
                    case 's':
                        break;
            }
        }
    }

```

<pre> = '1'; TablesBitmap[8] break; } } // a '0' ExitFlag means do NOT exit the loader early, a '1' means exit the loader early ExitFlag = 0; // iterate through the bitmap to display which table(s) is actually missing for (i = 0; i <= 8; i++) { switch(i) { case 0: if { (TablesBitmap[i] == '0') { printf("The Warehouse table is missing or damaged.\n"); ExitFlag = 1; } break; } case 1: if { (TablesBitmap[i] == '0') { printf("The District table is missing or damaged.\n"); ExitFlag = 1; } break; } case 2: if { (TablesBitmap[i] == '0') { printf("The Customer table is missing or damaged.\n"); ExitFlag = 1; } break; } case 3: if { (TablesBitmap[i] == '0') { 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); } } } } } } } } } } </pre>	<pre> if { 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); } } } </pre>	<pre> SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc); return; } getargs.c File: GETARGS.C Microsoft TPC-C // kit ver. 4.41 // Microsoft, 1996, 1997, 1998, 1999, 2000, 2001. Copyright // Purpose: Source file for command line processing // Includes // #include "tpcc.h" // ===== // Function name: GetArgsLoader // ===== void GetArgsLoader(int argc, char **argv, TPCC_LD_ARGS "pargs) { int i; char *ptr; #ifdef DEBUG printf("[%d]DBG: Entering GetArgsLoader()\n", (int) GetCurrentThreadId()); #endif /* init args struct with some useful values */ pargs->server = SERVER; pargs->user = USER; pargs->password = PASSWORD; pargs->database = DATABASE; pargs->batch = BATCH; pargs->num_warehouses = UNDEF; pargs->tables_all = TRUE; pargs->table_item = FALSE; pargs->table_warehouse = FALSE; pargs->table_customer = FALSE; pargs->table_orders = FALSE; pargs->loader_res_file = LOADER_RES_FILE; pargs->log_path = LOG_PATH; pargs->pack_size = DEF_LD_PACKET_SIZE; 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 */ } </pre>
--	---	---

```

if ( argc == 1 ) GetArgsLoaderUsage();
for ( i = 1; i < argc; ++i)
{
    if ( argv[i][0] != '-' && argv[i][0]
    != '/' )
    {
        printf("\nunrecognized command");
        GetArgsLoaderUsage();
        exit(1);
    }
    ptr = argv[i];
    switch (ptr[1])
    {
        case '?': /* Fall through */
            break;
        case 'd':
            pargs->database
            break;
        case 'p':
            pargs->password
            break;
        case 's':
            pargs->server =
            break;
        case 'u':
            pargs->user =
            break;
        case 'b':
            pargs->batch =
            break;
        case 'w':
            pargs->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;
            }
            if ( strcmp(ptr+2, "warehouse") == 0)
                pargs->table_warehouse = TRUE;
            if ( strcmp(ptr+2, "customer") == 0)

```

```

            pargs->table_customer = TRUE;
            if ( strcmp(ptr+2, "orders") == 0)
            {
                pargs->table_orders = TRUE;
            }
            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 )
    {
        required(ptr+2);
        exit(-2);
    }
    return;
}

```

```

//=====
// Function name: GetArgsLoaderUsage
//=====
void GetArgsLoaderUsage()
{
    #ifdef DEBUG
        printf("\n[%d]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
    %d\n", (long) BATCH);
    printf("-p TDS packet size
    %d\n", (long) DEFDPACKSIZE);
    printf("-f Loader Results Output Filename
    %s\n", LOADER_RES_FILE);
    printf("-s Starting Warehouse
    %d\n", (long) DEF_STARTING_WAREHOUSE);
    printf("-i Build Option (data = 0, data and
    index = 1) %d\n", (long) BUILD_INDEX);
    printf("-o Cluster Index Build Order (before
    = 1, after = 0) %d\n", (long) INDEX_ORDER);
    printf("-c Build Scaled Database (normal = 0,
    tiny = 1) %d\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:      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("[%d]DBG: Entering seed(...)\n", (int)
        GetCurrentThreadId());
    #endif
}

```

```

val);
printf("old Seed %ld New Seed %ld\n", Seed,
#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("[%d]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("[%d]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("[%d]DBG: Entering RandomNumber(...)\n",
        (int) GetCurrentThreadId());
    #endif

    perf enhancement */
    if ( upper == lower )      /* pgd 08-13-96
        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("[%d]DBG: RandomNumber between %ld & %ld
        ==> %ld\n",
        (int)
        GetCurrentThreadId(), lower, upper, rand_num);
    #endif

    return rand_num;
}

/* if 0
//Original code pgd 08/13/96
long RandomNumber(long lower,
    long upper)
{
    long rand_num;

    #ifdef DEBUG
        printf("[%d]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("[%d]DBG: RandomNumber between %ld & %ld
    ==> %ld\n",
           (int)
    GetCurrentThreadId(), lower, upper, rand_num);
#endif

    return rand_num;
}
#endif

// Function : NURand
// Description:
//=====
Tong NURand(int iconst,
            long x,
            long y,
            long C)
{
    Tong rand_num;

#ifdef DEBUG
    printf("[%d]DBG: Entering NURand()\n", (int)
    GetCurrentThreadId());
#endif
    rand_num = (((RandomNumber(0, iconst) |
    RandomNumber(x, y)) + C) % (y-x+1))+x;

#ifdef DEBUG
    printf("[%d]DBG: NURand: num = %ld\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, 2000, 2001
// Purpose: Source file for database loader
// string functions
// Includes "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("[%d]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("[%d]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("[%d]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("[%d]DBG: LastName: num = [%d] ==>
    [%d][%d]\n",
           (int)
    GetCurrentThreadId(), num, num/100, (num/10)%10,
    num%10);
    printf("[%d]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]
//represents a string
//of random alphanumeric
//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
//ASCII so this is a non-issue.
//This is completed, and is suitable to stuff non-printing
//chaos 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[] =
    "0123456789ABCDEFHJKLMNOPQRSTUVWXYZabcdefghijklmnopqr
    stuvwxyz";
    static int charrayMax = 61;

#ifdef DEBUG
    printf("[%d]DBG: Entering MakeAlphaString()\n",
    (int) GetCurrentThreadId());
#endif
    len= RandomNumber(x, y);
    for (i=0; i<len; i++)
    {

```

```

chararraymax[]];
    cc = chararray[RandomNumber(0,
    }
    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("[%d]DBG: Entering
    MakeOriginalAlphaString()\n", (int)
    GetCurrentThreadId());
    #endif

    // verify percentage is valid
    if ((percent < 0) || (percent > 100))
    {
        printf("MakeOriginalAlphaString:
        Invalid percentage: %d\n", percent);
        exit(-1);
    }

    // verify string is at least 8 chars in length
    if ((x + y) <= 8)
    {
        printf("MakeOriginalAlphaString:
        string length must be >= 8\n");
        exit(-1);
    }

    // Make Alpha String
    len = MakeAlphaString(x,y, z, str);
    val = RandomNumber(1,100);
    if (val <= percent)
    {
        start = RandomNumber(0, len - 8);
        strncpy(str + start, "ORIGINAL",
        8);
    }

    #ifdef DEBUG
    printf("[%d]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, "0000111111");

    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("[%d]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
// kit ver. 4.41
// 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("[%d]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, Enterprise" /fastdetect /PAE
```

Microsoft Windows Server 2003 EE Configuration (Enabled Services)

These Windows services are started:

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

Microsoft SQL Server 2000 Configuration Parameters

name	config_value	run_value	minimum	maximum
------	--------------	-----------	---------	---------

Itaotec Philco TPC-C FULL DISCLOSURE REPORT
© 2003 Itaotec Philco, all rights reserved

-----	-----	-----	-----	-----
affinity mask	2147483647	255	-2147483648	
allow updates	0		0	1
awe enabled	0		0	1
c2 audit mode	1		0	1
cost threshold for parallelism	5		0	32767
Cross DB Ownership Chaining	0		0	1
Cursor threshold	0		-1	
default full-text language	-1		0	
default language	1033		0	9999
fill factor (%)	0		0	100
index create memory (KB)	0		704	
Lightweight pooling	0		0	1
locks	1		5000	
max degree of parallelism	0		0	32
max server memory (MB)	1		4	
max text repl size (B)	2147483647		0	
max worker threads	65536		32	32767
media retention	345		0	365
min memory per query (KB)	1024		512	
min server memory (MB)	0		0	
nested triggers	0		0	1
network packet size (B)	4096		512	65536
objects	0		0	
priority boost	1		0	1
query governor cost limit	0		0	
query wait (s)	0		-1	
recovery interval (min)	-1		0	32767
remote access	112		0	1
remote login timeout (s)	0		0	
remote proc trans	0		0	1
remote query timeout (s)	0		0	
scan for startup proc	0		0	1
set working set size	0		0	1
show advanced options	1		0	1
two digit year cutoff	2049		1753	9999

user connections	0	0	32767
user options	0	0	32767

Disk Controller Driver Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ac2w2k]
"Group"="SCSI miniport"
"Start"=dword:00000000
"Tag"=dword:00000020
"Type"=dword:00000001
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):73,00,79,00,73,00,6d,00,6d,00,33,00,32,00,5c,00,44,00,52,00,49,00,56,00,45,00,52,00,53,00,5c,00,64,00,61,00,63,00,32,00,77,00,32,00,6b,00,2e,00,73,00,79,00,73,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ac2w2k\Parameters]
"BusType"=dword:00000008
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ac2w2k\Parameters\PortInterface]
"Z"=dword:00000001
"5"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ac2w2k\Port]
"0"="PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&254dad54&0&4040"
"Count"=dword:00000006
"NextInstance"=dword:00000006
"1"="PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&94a037d8&0&4048"
"2"="PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&2c59aba9&0&4040"
"3"="PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&23605288&0&4048"
"4"="PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&12e15626&0&4040"
"5"="PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&1b065aab&0&4048"
```

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\m ac2w2k]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:0000000f
"ImagePath"=hex(2):73,00,79,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,44,00,52,00,49,00,56,00,45,00,52,00,53,00,5c,00,6d,00,61,00,63,00,32,00,77,00,32,00,6b,00,2e,00,73,00,79,00,73,00,00,00
"Group"="System Bus Extender"
```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sm
ac2wk\Security]
"Security"=hex:01,00,14,80,90,00,00,00,9c,00,00,00,14,0
0,00,00,30,00,00,00,02,\
00,1c,00,01,00,00,02,80,14,00,ff,01,0f,00,01,01,00,0
0,00,00,00,01,00,00,\
00,00,02,00,60,00,00,00,00,00,00,14,00,fd,01,02,00,0
1,01,00,00,00,00,00,\
05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,0
0,00,05,20,00,00,00,\
20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,0
0,05,0b,00,00,00,00,\
00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00
0,23,02,00,00,01,01,\
00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,05,1
2,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sm
ac2wk\Enum]
"0"="root\\LEGACY_MAC2M2K\0000"
"Count"=dword:00000008
"NextInstance"=dword:00000008
"1"="SCSI\DiskVen_MYLEX&Prod_extremeRAID_2000&Rev_070
2\\58d708b0c080400"
"2"="SCSI\DiskVen_MYLEX&Prod_extremeRAID_2000&Rev_070
2\\58d708b0c080400"
"3"="SCSI\DiskVen_MYLEX&Prod_extremeRAID_2000&Rev_070
2\\582c06d47d080400"
"4"="SCSI\DiskVen_MYLEX&Prod_extremeRAID_2000&Rev_070
2\\582cb960ec080400"
"5"="SCSI\DiskVen_MYLEX&Prod_extremeRAID_2000&Rev_070
2\\58e99db856080400"
"6"="SCSI\DiskVen_MYLEX&Prod_extremeRAID_2000&Rev_070
2\\58dd1a6608080400"
"7"="SCSI\DiskVen_MYLEX&Prod_extremeRAID_2000&Rev_070
2\\585ddf08080400"

System Summary

System Information report written at: 11/03/03 16:06:13
System Name: SQL
[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003,
Enterprise Edition	
Version	5.2.3790
OS Manufacturer	Microsoft Corporation
Activation Status	Activation Pending (52 days remaining)
System Name	SQL
System Manufacturer	Intel
System Model	SPSH4 SSHCPBK
System Type	X86-based PC
Processor x86 Family	15 Model 2 Stepping 5 GenuineIntel
~2783 Mhz	
Processor x86 Family	15 Model 2 Stepping 5 GenuineIntel
~2783 Mhz	
Processor x86 Family	15 Model 2 Stepping 5 GenuineIntel
~2783 Mhz	

Itaitec Philco TPC-C FULL DISCLOSURE REPORT

© 2003 Itaitec Philco, all rights reserved

Processor x86 Family 15 Model 2 Stepping 5 GenuineIntel
~2783 Mhz
Processor x86 Family 15 Model 2 Stepping 5 GenuineIntel
~2783 Mhz
Processor x86 Family 15 Model 2 Stepping 5 GenuineIntel
~2783 Mhz
Processor x86 Family 15 Model 2 Stepping 5 GenuineIntel
~2783 Mhz
Processor x86 Family 15 Model 2 Stepping 5 GenuineIntel
~2783 Mhz
BIOS Version/Date Intel Corporation
SH40.868.0089.B.0309251052, 25/9/2003
SMBIOS Version 2.3
Windows Directory C:\WINDOWS
System Directory C:\WINDOWS\System32
Boot Device \Device\HarddiskVolume1
Locale United States
Hardware Abstraction Layer Version = "5.2.3790.0
(Svc03_rtm.030324-2048)"
User Name SQLAdministrator
Time Zone E. South America Standard Time
Total Physical Memory 24,577.00 MB
Available Physical Memory 51,25 MB
Total Virtual Memory 48,12 GB
Available Virtual Memory 2,07 GB
Page File Space 24.93 GB
Page File C:\pagefile.sys
[Hardware Resources]

[Conflicts/Sharing]

Resource Device PCI bus
I/O Port 0x00000000-0x0000003AF Direct memory
access controller
Memory Address 0xf4000000-0xf8bfffff PCI bus
Memory Address 0xf4000000-0xf8bfffff DEC 21154 PCI
to PCI bridge
Memory Address 0xf4000000-0xf8bfffff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)

I/O Port 0x000003c0-0x000003df PCI bus
I/O Port 0x000003c0-0x000003df RAGE XL PCI
Family (Microsoft Corporation)
Memory Address 0xe8000000-0xe8bfffff DEC 21154 PCI
to PCI bridge
Memory Address 0xe8000000-0xe8bfffff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)

Memory Address 0xb8000000-0xb8bfffff DEC 21154 PCI
to PCI bridge
Memory Address 0xb8000000-0xb8bfffff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)

Memory Address 0xec000000-0xf01fffff PCI bus
Memory Address 0xec000000-0xf01fffff DEC 21154 PCI
to PCI bridge
Memory Address 0xec000000-0xf01fffff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)

I/O Port 0x00006000-0x00006fff DEC 21154 PCI
to PCI bridge

I/O Port 0x00006000-0x00006fff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)
I/O Port 0x00003000-0x00004fff PCI bus
I/O Port 0x00003000-0x00004fff DEC 21154 PCI
to PCI bridge
I/O Port 0x00003000-0x00004fff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)
Memory Address 0xd2000000-0xd3fffff DEC 21154 PCI
to PCI bridge
Memory Address 0xd2000000-0xd3fffff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)
Memory Address 0xce000000-0xce3fffff PCI bus
Memory Address 0xce000000-0xce3fffff Intel(R)
PRO/1000 XT Network Connection
Memory Address 0xcc000000-0xcdfffff PCI bus
Memory Address 0xcc000000-0xcdfffff Intel 8255x-
based PCI Ethernet Adapter (10/100)
Memory Address 0xde000000-0xdffffff DEC 21154 PCI
to PCI bridge
Memory Address 0xde000000-0xdffffff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)
Memory Address 0xe4000000-0xebfffff PCI bus
Memory Address 0xe4000000-0xebfffff DEC 21154 PCI
to PCI bridge
Memory Address 0xe4000000-0xebfffff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)
I/O Port 0x00005000-0x00006fff PCI bus
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 0xd0000000-0xd1fffff DEC 21154 PCI
to PCI bridge
Memory Address 0xd0000000-0xd1fffff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)
I/O Port 0x00002460-0x0000247f PCI bus
I/O Port 0x00002460-0x0000247f Intel(R)
PRO/1000 XT Network Connection
Memory Address 0xa0000-0xbffff PCI bus
Memory Address 0xa0000-0xbffff RAGE XL PCI
Family (Microsoft Corporation)
Memory Address 0xf8000000-0xf8bfffff DEC 21154 PCI
to PCI bridge
Memory Address 0xf8000000-0xf8bfffff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)
I/O Port 0x00007000-0x00008fff PCI bus
I/O Port 0x00007000-0x00008fff DEC 21154 PCI
to PCI bridge
I/O Port 0x00007000-0x00008fff Mylex
extremeRAID 2000 Disk Array Controller (Accelerated)

I/O Port 0x00000380-0x00000388 Family (Microsoft Corporation)	PCI bus RAGE XL PCI	0x00000279-0x00000279 OK	ISAPNP Read Data Port	0x00000F50-0x00000F58 OK	Motherboard resources
I/O Port 0x00000380-0x00000388 Family (Microsoft Corporation)	PCI bus RAGE XL PCI	0x00000274-0x00000277 OK	ISAPNP Read Data Port	0x00000374-0x00000375 OK	Motherboard resources
I/O Port 0x00000400-0x000004FF to PCI bridge	DEC 21154 PCI	0x00000060-0x00000060 Microsoft Natural PS/2	Standard 101/102-Key or keyboard	0x0000FE00-0x0000FE20 OK	Motherboard resources
I/O Port 0x00000400-0x000004FF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000064-0x00000064 Microsoft Natural PS/2	Standard 101/102-Key or keyboard	0x00000220-0x00000220 OK	Motherboard resources
I/O Port 0x00000800-0x000008FF to PCI bridge	DEC 21154 PCI	0x00000070-0x00000073 clock	System CMOS/real time	0x00000225-0x00000225 OK	Motherboard resources
I/O Port 0x00000800-0x000008FF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000010-0x0000001F controller	Direct memory access	0x00000228-0x00000228 OK	Motherboard resources
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000080-0x0000008F controller	Direct memory access	0x0000022A-0x0000022E OK	Motherboard resources
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000000-0x0000000F controller	Direct memory access	0x00000102-0x00000105 OK	Motherboard resources
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000020-0x00000021 controller	Programmable interrupt	0x00000107-0x00000107 OK	Motherboard resources
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x000000A0-0x000000A1 controller	Programmable interrupt	0x00000408-0x00000408 OK	Motherboard resources
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000040-0x00000043 controller	System timer	0x00000500-0x0000051F OK	Motherboard resources
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x000000F0-0x000000FF controller	Numeric data processor	0x0000058E-0x00000593 OK	Motherboard resources
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000061-0x00000061 controller	System speaker	0x00000CA2-0x00000CA5 OK	Motherboard resources
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x0000002E-0x0000002F controller	Motherboard resources	0x000003F8-0x000003FF (COM1) OK	Communications Port
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000540-0x0000055F controller	Motherboard resources	0x000002F8-0x000002FF (COM2) OK	Communications Port
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000560-0x00000563 controller	Motherboard resources	0x00000378-0x0000037F OK	ECP Printer Port (LPT1)
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000564-0x00000567 controller	Motherboard resources	0x00000778-0x0000077F OK	ECP Printer Port (LPT1)
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000568-0x0000056F controller	Motherboard resources	0x000003F0-0x000003F5 controller	Standard floppy disk
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x000000E0-0x000000FF controller	Motherboard resources	0x000003F7-0x000003F7 controller	Standard floppy disk
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000600-0x0000061F controller	Motherboard resources	0x00000CA6-0x00000CA6 Embedded Controller OK	Microsoft ACPI-Compliant
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000580-0x0000058D controller	Motherboard resources	0x00000CA7-0x00000CA7 Embedded Controller OK	Microsoft ACPI-Compliant
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000092-0x00000092 controller	Motherboard resources	0x00000900-0x0000093F OK	Motherboard resources
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000804-0x00000804 controller	Motherboard resources	0x00000940-0x0000097F OK	Motherboard resources
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000419-0x0000041B controller	Motherboard resources	0x00002440-0x0000244F OK	CSB5 IDE Controller OK
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x0000041D-0x0000041F controller	Motherboard resources	0x000001F0-0x000001F7 OK	Primary IDE Channel OK
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000400-0x000004D1 controller	Motherboard resources	0x000003F6-0x000003F6 OK	Primary IDE Channel OK
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x000004D6-0x000004D6 controller	Motherboard resources	0x00000170-0x00000177 OK	Secondary IDE Channel
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000C00-0x00000C01 controller	Motherboard resources	0x00000376-0x00000376 OK	Secondary IDE Channel
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000C06-0x00000C08 controller	Motherboard resources	0x00002460-0x0000247F OK	PCI bus OK
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000C14-0x00000C14 controller	Motherboard resources	0x00002460-0x0000247F Intel(R) PRO/1000 XT	PCI bus OK
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000C49-0x00000C4A controller	Motherboard resources	0x00003000-0x00004FFF OK	PCI bus OK
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000C50-0x00000C51 controller	Motherboard resources	0x00003000-0x00004FFF bridge	DEC 21154 PCI to PCI
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000C52-0x00000C52 controller	Motherboard resources	0x00003000-0x00004FFF bridge	Mylex extremeRAID 2000 (Accelerated) OK
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000C6C-0x00000C6C controller	Motherboard resources	0x00004000-0x00004FFF bridge	DEC 21154 PCI to PCI
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000C6F-0x00000C6F controller	Motherboard resources	0x00004000-0x00004FFF bridge	Mylex extremeRAID 2000 (Accelerated) OK
Memory Address 0x0C000000-0xE01FFFFF extremeRAID 2000 Disk Array Controller	Mylex (Accelerated)	0x00000C6F-0x00000C6F controller	Motherboard resources	0x00005000-0x00006FFF PCI bus OK	PCI bus OK
Memory Address 0x0C000000-0xE01FFFFF to PCI bridge	PCI bus DEC 21154 PCI	0x00000CDB6-0x00000CDB7 controller	Motherboard resources	0x00005000-0x00006FFF bridge	DEC 21154 PCI to PCI

c:\windows\system32\msrle32.dll Microsoft Corporation
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\msh261.drv Microsoft Corporation
C:\WINDOWS\system32\msh261.drv
4.4.4000.180.00 KB (184,320 bytes) 25/10/2003 22:29
c:\windows\system32\msv1dc32.dll Microsoft Corporation
C:\WINDOWS\system32\MSV1DC32.DLL
5.2.3790.0 (srv03_rtm.030324-2048) 26, 50 KB (27,136 bytes) 25/3/2003 09:00
c:\windows\system32\iyuv_32.dll Microsoft Corporation
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\msyuv.dll Microsoft Corporation
C:\WINDOWS\system32\MSYUV.DLL
5.2.3790.0 (srv03_rtm.030324-2048) 16, 50 KB (16,896 bytes) 24/3/2003 22:49

[CD-ROM]

Item	Value
Drive T:	CD-ROM Drive
Description	Yes
Media Type	CD-ROM
Name	LITEON CD-ROM LTNS26
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	1289.00 kbytes/sec
SCSI Target ID	0
PNP Device ID	IDE\CDROM\LTNTEON_CD-ROM_LTNS26_
PNP Device ID	YH0X_&F80C83D&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_34128086&REV_27\3&267A616A&0&10

Item	Value
Adapter Type	ATI RAGE XL PCI (B41), ATI Technologies Inc. compatible
Adapter Description	RAGE XL PCI Family (Microsoft Corporation)
Adapter RAM	4,00 MB (4,194.304 bytes)
Installed Drivers	ati2dreg.dll
Driver Version	10.10.5863.6013
Color Planes	1
Color Table Entries	4294967296
Resolution	1024 x 768 x 70 hertz
Bits/Pixel	32
Memory Address	0xC0000000-0xC0FFFFFF
I/O Port	0x00002000-0x000020FF
Memory Address	0xC0020000-0xC020FFFF

Itautec Philco TPC-C FULL DISCLOSURE REPORT

© 2003 Itautec Philco, all rights reserved

IRQ Channel
I/O Port 0x00003B0-0x00000388
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), 25/10/2003 15:18)

[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&35118DFF&0
Number of Function keys	12
I/O Port	0x00000060-0x00000060
I/O Port	0x00000064-0x00000064
IRQ Channel	IRQ 1
Driver	c:\windows\system32\drivers\i8042prt.sys (5.2.3790.0 (srv03_rtm.030324-2048), 68,50 KB (70,144 bytes), 25/3/2003 09:00)

[Pointing device]

Item	Value
Hardware Type	PS/2 Compatible Mouse
Number of Buttons	5
Status	OK
PNP Device ID	ACPI\PNP0F13\4&35118DFF&0
Power Management	Supported
Double Click Threshold	No
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	[000000001] Intel 8255x-based PCI Ethernet Adapter (10/100)
Adapter Type	Ethernet 802.3
Product Type	Intel 8255x-based PCI Ethernet Adapter (10/100)
Installed Yes	Yes
PNP Device	PCI\VEN_8086&DEV_1229&SUBSYS_34128086&REV_0D\3&267A616A&0&18
Last Reset	1/11/2003 16:52
Index	1
Service Name	E1008
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 00:02:B3:55:47:7A
Memory Address 0xC021000-0xC021FFF
I/O Port 0x00002400-0x0000243F
Memory Address 0xC0000000-0xC0FFFFFF
IRQ Channel IRQ 18
Driver c:\windows\system32\drivers\el100b325.sys (6.6.8.1 built by: winDDK, 138,50 KB (141,824 bytes), 25/10/2003 15:18)

Name	[000000002] Intel (R) PRO/1000 XT Network Connection
Adapter Type	Ethernet 802.3
Product Type	Intel (R) PRO/1000 XT Network
Installed Yes	Yes
PNP Device ID	PCI\VEN_8086&DEV_100D&SUBSYS_34128086&REV_02\3&13C080C5&0&18
Last Reset	1/11/2003 16:52
Index	2
Service Name	E1000
IP Address	192.168.10.100
IP Subnet	255.255.255.0
Default IP Gateway	Not Available
DHCP Enabled	No
DHCP Server Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	00:02:B3:55:47:78

Memory Address	0xCE020000-0xCE03FFFF
Memory Address	0xCE000000-0xCE3FFFFF
I/O Port	0x00002460-0x0000247F
IRQ Channel	IRQ 19
Driver	c:\windows\system32\drivers\el100b325.sys (6.3.6.31 built by: winDDK, 99,00 KB (101,376 bytes), 25/10/2003 15:18)

Name	[000000003] RAS Async Adapter
Adapter Type	Not Available
Product Type	RAS Async Adapter
Installed Yes	Yes
PNP Device ID	Not Available
Last Reset	1/11/2003 16:52
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	[000000004] WAN Miniport (L2TP)
Adapter Type	Not Available
Product Type	WAN Miniport (L2TP)
Installed Yes	Yes
PNP Device ID	ROOT\MS_L2TPMINIPORT\0000
Last Reset	1/11/2003 16:52
Index	4
Service Name	Ras12tp
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\ras12tp.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 77,00 KB (78.848
bytes), 25/3/2003 09:00)

Name [000000005] 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 1/11/2003 16:52
Index 5
Service Name PptmMiniport
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\rasppoe.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 70,50 KB (72.192
bytes), 25/3/2003 09:00)

Name [000000006] 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 1/11/2003 16:52
Index 6
Service Name Rasppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver c:\windows\system32\drivers\rasppoe.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 38,00 KB (38.912
bytes), 25/3/2003 09:00)

Name [000000007] Direct Parallel
Adapter Type Direct Parallel
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTMINIPORT\0000
Last Reset 1/11/2003 16:52
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 [000000008] WAN Miniport (IP)
Adapter Type WAN Miniport (IP)
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000

Last Reset 8 1/11/2003 16:52

Index
Service Name Ndiswan
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 MSADF 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 MSADF 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 MSADF NetBIOS [\Device\NetBT_Tcpip_{5D31AC17-CE5F-4FC1-8A1E-CCEEFC9276FC}] SEQUENTIAL
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 MSADF NetBIOS [\Device\NetBT_Tcpip_{5D31AC17-CE5F-4FC1-8A1E-CCEEFC9276FC}] DATAGRAM
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 MSADF NetBIOS [\Device\NetBT_Tcpip_{ED9F1F09-5A4D-45AE-9136-55978896F10C}] SEQUENTIAL
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	No	Supports Encryption No	No
Supports Expedited Data	No	Supports Expedited Data	No
Supports Graceful Closing	No	Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No	Supports Guaranteed Bandwidth	No
Supports Multicasting	No	Supports Multicasting	No
Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{ED9F1F09-544D-45AE-9136-559789B8F10C}] DATAGRAM 1			
Connectionless Service	Yes	Connectionless Service	Yes
Guarantees Delivery No	No	Guarantees Delivery No	No
Guarantees Sequencing	20 bytes	Guarantees Sequencing	20 bytes
Maximum Address Size	62,50 KB (64,000 bytes)	Maximum Address Size	62,50 KB (64,000 bytes)
Maximum Message Size		Maximum Message Size	
Message Oriented	Yes	Message Oriented	Yes
Minimum Address Size	20 bytes	Minimum Address Size	20 bytes
Pseudo Stream Oriented	No	Pseudo Stream Oriented	No
Supports Broadcasting	Yes	Supports Broadcasting	Yes
Supports Connect Data	No	Supports Connect Data	No
Supports Disconnect Data	No	Supports Disconnect Data	No
Supports Encryption No	No	Supports Encryption No	No
Supports Expedited Data	No	Supports Expedited Data	No
Supports Graceful Closing	No	Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No	Supports Guaranteed Bandwidth	No
Supports Multicasting	No	Supports Multicasting	No
Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{61B843F0-386D-4A3F-A92C-486538A579E4}] SEQUENTIAL 2			
Connectionless Service	No	Connectionless Service	No
Guarantees Delivery Yes	Yes	Guarantees Delivery Yes	Yes
Guarantees Sequencing	20 bytes	Guarantees Sequencing	20 bytes
Maximum Address Size	62,50 KB (64,000 bytes)	Maximum Address Size	62,50 KB (64,000 bytes)
Maximum Message Size		Maximum Message Size	
Message Oriented	Yes	Message Oriented	Yes
Minimum Address Size	20 bytes	Minimum Address Size	20 bytes
Pseudo Stream Oriented	No	Pseudo Stream Oriented	No
Supports Broadcasting	No	Supports Broadcasting	No
Supports Connect Data	No	Supports Connect Data	No
Supports Disconnect Data	No	Supports Disconnect Data	No
Supports Encryption No	No	Supports Encryption No	No
Supports Expedited Data	No	Supports Expedited Data	No
Supports Graceful Closing	No	Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No	Supports Guaranteed Bandwidth	No
Supports Multicasting	No	Supports Multicasting	No
Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{61B843F0-386D-4A3F-A92C-486538A579E4}] DATAGRAM 2			
Connectionless Service	Yes	Connectionless Service	Yes
Guarantees Delivery No	No	Guarantees Delivery No	No
Guarantees Sequencing	20 bytes	Guarantees Sequencing	20 bytes
Maximum Address Size	62,50 KB (64,000 bytes)	Maximum Address Size	62,50 KB (64,000 bytes)
Maximum Message Size		Maximum Message Size	
Message Oriented	Yes	Message Oriented	Yes
Minimum Address Size	20 bytes	Minimum Address Size	20 bytes
Pseudo Stream Oriented	No	Pseudo Stream Oriented	No
Supports Broadcasting	Yes	Supports Broadcasting	Yes
Supports Connect Data	No	Supports Connect Data	No
Supports Disconnect Data	No	Supports Disconnect Data	No
Supports Encryption No	No	Supports Encryption No	No
Supports Expedited Data	No	Supports Expedited Data	No
Supports Graceful Closing	No	Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No	Supports Guaranteed Bandwidth	No
Supports Multicasting	No	Supports Multicasting	No
Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{535E17FB-93F2-4008-9284-ADF2FFAF078}] SEQUENTIAL 3			
Connectionless Service	No	Connectionless Service	No
Guarantees Delivery Yes	Yes	Guarantees Delivery Yes	Yes
Guarantees Sequencing	20 bytes	Guarantees Sequencing	20 bytes
Maximum Address Size	62,50 KB (64,000 bytes)	Maximum Address Size	62,50 KB (64,000 bytes)
Maximum Message Size		Maximum Message Size	
Message Oriented	Yes	Message Oriented	Yes
Minimum Address Size	20 bytes	Minimum Address Size	20 bytes
Pseudo Stream Oriented	No	Pseudo Stream Oriented	No
Supports Broadcasting	Yes	Supports Broadcasting	Yes
Supports Connect Data	No	Supports Connect Data	No
Supports Disconnect Data	No	Supports Disconnect Data	No
Supports Encryption No	No	Supports Encryption No	No
Supports Expedited Data	No	Supports Expedited Data	No
Supports Graceful Closing	No	Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No	Supports Guaranteed Bandwidth	No
Supports Multicasting	No	Supports Multicasting	No

Guarantees Sequencing	Yes	Guarantees Sequencing	Yes
Maximum Address Size	62,50 KB (64,000 bytes)	Maximum Address Size	62,50 KB (64,000 bytes)
Maximum Message Size		Maximum Message Size	
Message Oriented	Yes	Message Oriented	Yes
Pseudo Stream Oriented	No	Pseudo Stream Oriented	No
Supports Broadcasting	No	Supports Broadcasting	No
Supports Connect Data	No	Supports Connect Data	No
Supports Disconnect Data	No	Supports Disconnect Data	No
Supports Encryption No	No	Supports Encryption No	No
Supports Expedited Data	No	Supports Expedited Data	No
Supports Graceful Closing	No	Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No	Supports Guaranteed Bandwidth	No
Supports Multicasting	No	Supports Multicasting	No
Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{535E17FB-93F2-4008-9284-ADF2FFAF078}] DATAGRAM 3			
Connectionless Service	Yes	Connectionless Service	Yes
Guarantees Delivery No	No	Guarantees Delivery No	No
Guarantees Sequencing	20 bytes	Guarantees Sequencing	20 bytes
Maximum Address Size	62,50 KB (64,000 bytes)	Maximum Address Size	62,50 KB (64,000 bytes)
Maximum Message Size		Maximum Message Size	
Message Oriented	Yes	Message Oriented	Yes
Minimum Address Size	20 bytes	Minimum Address Size	20 bytes
Pseudo Stream Oriented	No	Pseudo Stream Oriented	No
Supports Broadcasting	Yes	Supports Broadcasting	Yes
Supports Connect Data	No	Supports Connect Data	No
Supports Disconnect Data	No	Supports Disconnect Data	No
Supports Encryption No	No	Supports Encryption No	No
Supports Expedited Data	No	Supports Expedited Data	No
Supports Graceful Closing	No	Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No	Supports Guaranteed Bandwidth	No
Supports Multicasting	No	Supports Multicasting	No
[Winsock]			
Item	Value	Item	Value
File	c:\windows\system32\winsock.dll	File	c:\windows\system32\winsock.dll
Size	2,80 KB (2,864 bytes)	Size	2,80 KB (2,864 bytes)
Version	3.10	Version	3.10
File	c:\windows\system32\wsck32.dll	File	c:\windows\system32\wsck32.dll
Size	22,00 KB (22,528 bytes)	Size	22,00 KB (22,528 bytes)
Version	5.2.3790.0 (srv03_rtm.030324-2048)	Version	5.2.3790.0 (srv03_rtm.030324-2048)
[Ports]			
[Serial]			
Item	Value	Item	Value
Name	OK	Name	OK
Status	OK	Status	OK
PNP Device ID	ACPI\PNP0501\1	PNP Device ID	ACPI\PNP0501\1
Maximum Input Buffer Size	0	Maximum Input Buffer Size	0
Maximum Output Buffer Size	No	Maximum Output Buffer Size	No
Settable Baud Rate	Yes	Settable Baud Rate	Yes
Settable Data Bits	Yes	Settable Data Bits	Yes
Settable Flow Control	Yes	Settable Flow Control	Yes
Settable Parity	Yes	Settable Parity	Yes
Settable Parity Check	Yes	Settable Parity Check	Yes
Settable Stop Bits	Yes	Settable Stop Bits	Yes
Settable RLS	Yes	Settable RLS	Yes
Settable RLSD	Yes	Settable RLSD	Yes
Supports 16 Bit Mode	No	Supports 16 Bit Mode	No
Supports Special Characters	No	Supports Special Characters	No
Baud Rate	9600	Baud Rate	9600
Bits/Byte	8	Bits/Byte	8
Stop Bits	1	Stop Bits	1

Parity	None	Parity	None
Busy	No	Busy	No
Abort Read/Write on Error	No	Abort Read/Write on Error	No
Binary Mode Enabled	Yes	Binary Mode Enabled	Yes
Continue Xmit on Xoff	No	Continue Xmit on Xoff	No
CTS Outflow Control	No	CTS Outflow Control	No
Discard NULL Bytes	No	Discard NULL Bytes	No
DSR Outflow Control	0	DSR Outflow Control	0
DSR Sensitivity	0	DSR Sensitivity	0
DTR Flow Control	Type	DTR Flow Control	Type
EOF Character	0	EOF Character	0
Error Replace Character	0	Error Replace Character	0
Error Replacement Enabled	Yes	Error Replacement Enabled	Yes
Event Character	0	Event Character	0
RTS Flow Control	Enabled	RTS Flow Control	Enabled
RTS Flow Control Type	19	RTS Flow Control Type	19
Xoff Character	19	Xoff Character	19
XoffXmit Threshold	512	XoffXmit Threshold	512
Xon Character	17	Xon Character	17
XonXmit Threshold	2048	XonXmit Threshold	2048
XonXoff InFlow Control	0	XonXoff InFlow Control	0
XonXoff OutFlow Control	0	XonXoff OutFlow Control	0
I/O Port	0x00003F8-0x000003FF	I/O Port	0x00003F8-0x000003FF
IRQ Channel	IRQ 4	IRQ Channel	IRQ 4
Driver	c:\windows\system32\drivers\serial.sys	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)		(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	Status	OK
PNP Device ID	ACPI\PNP0501\2	PNP Device ID	ACPI\PNP0501\2
Maximum Input Buffer Size	0	Maximum Input Buffer Size	0
Maximum Output Buffer Size	No	Maximum Output Buffer Size	No
Settable Baud Rate	Yes	Settable Baud Rate	Yes
Settable Data Bits	Yes	Settable Data Bits	Yes
Settable Flow Control	Yes	Settable Flow Control	Yes
Settable Parity	Yes	Settable Parity	Yes
Settable Parity Check	Yes	Settable Parity Check	Yes
Settable Stop Bits	Yes	Settable Stop Bits	Yes
Settable RLS	Yes	Settable RLS	Yes
Settable RLSD	Yes	Settable RLSD	Yes
Supports 16 Bit Mode	No	Supports 16 Bit Mode	No
Supports Special Characters	No	Supports Special Characters	No
Baud Rate	9600	Baud Rate	9600
Bits/Byte	8	Bits/Byte	8
Stop Bits	1	Stop Bits	1
Parity	None	Parity	None
Busy	No	Busy	No
Abort Read/Write on Error	No	Abort Read/Write on Error	No
Binary Mode Enabled	Yes	Binary Mode Enabled	Yes
Continue Xmit on Xoff	No	Continue Xmit on Xoff	No
CTS Outflow Control	No	CTS Outflow Control	No
Discard NULL Bytes	No	Discard NULL Bytes	No
DSR Outflow Control	0	DSR Outflow Control	0
DSR Sensitivity	0	DSR Sensitivity	0
DTR Flow Control	Type	DTR Flow Control	Type
EOF Character	0	EOF Character	0
Error Replace Character	0	Error Replace Character	0
Error Replacement Enabled	Yes	Error Replacement Enabled	Yes
Event Character	0	Event Character	0
Parity Check	Enabled	Parity Check	Enabled
RTS Flow Control	Type	RTS Flow Control	Type
Xoff Character	19	Xoff Character	19
XoffXmit Threshold	512	XoffXmit Threshold	512
Xon Character	17	Xon Character	17
XonXmit Threshold	2048	XonXmit Threshold	2048
XonXoff InFlow Control	0	XonXoff InFlow Control	0
XonXoff OutFlow Control	0	XonXoff OutFlow Control	0
I/O Port	0x00002F8-0x000002FF	I/O Port	0x00002F8-0x000002FF
IRQ Channel	IRQ 3	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\4d35118dff&0
I/O Port 0x00000378-0x0000037f
DMA 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: 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 30,20 GB (32.425.287.680 bytes)

Volume Name
Volume Serial Number F4EF140E

Drive T: CD-ROM Disc

[Disks]

Item Value
Description \\.\PHYSICALDRIVE2
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 1,30 TB (1.430.729.879.040 bytes)
Total Cylinders 173.943
Total Sectors 2.794.394.295
Total Tracks 44.355.465
Tracks/Cylinder 255
Partition Disk #2, Partition #0
Partition Size 75,00 GB (80.533.684.224 bytes)

Partition Starting Offset 8.257.536 bytes
Partition Disk #2, Partition #1
Partition Size 100,01 GB (107.380.998.144 bytes)
Partition Starting Offset 80.541.974.016 bytes
Partition Disk #2, Partition #2
Partition Size 1,13 TB (1.242.806.874.624 bytes)

Partition Starting Offset 187.923.004.416 bytes

\\.\PHYSICALDRIVE3

Description Not Available
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 4
SCSI Target ID 0
Sectors/Track 63
Size 1,30 TB (1.430.729.879.040 bytes)
Total Cylinders 173.943
Total Sectors 2.794.394.295
Total Tracks 44.355.465
Tracks/Cylinder 255
Partition Disk #3, Partition #0
Partition Size 75,00 GB (80.533.684.224 bytes)
Partition Starting Offset 8.257.536 bytes
Partition Disk #3, Partition #1
Partition Size 100,01 GB (107.380.998.144 bytes)
Partition Starting Offset 80.541.974.016 bytes
Partition Disk #3, Partition #2
Partition Size 1,13 TB (1.242.806.874.624 bytes)
Partition Starting Offset 187.923.004.416 bytes

\\.\PHYSICALDRIVE0

Description Not Available
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 2
SCSI Target ID 0
Sectors/Track 63
Size 34,16 GB (36.676.491.264 bytes)
Total Cylinders 173.943
Total Sectors 2.794.394.295
Total Tracks 44.355.465
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 34,16 GB (36.676.491.264 bytes)

Partition Starting Offset 32.256 bytes

\\.\PHYSICALDRIVE1

Description Not Available
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 2
SCSI Target ID 1
Sectors/Track 63
Size 239,16 GB (256.793.241.600 bytes)
Total Cylinders 31.220
Total Sectors 501.549.300

Total Tracks 7.961.100
Tracks/Cylinder 255
Partition Disk #1, Partition #0
Partition Size 239,16 GB (256.793.209.344 bytes)
Partition Starting Offset 32.256 bytes

\\.\PHYSICALDRIVE6

Description Not Available
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 7
SCSI Target ID 0
Sectors/Track 63
Size 1,30 TB (1.430.729.879.040 bytes)
Total Cylinders 173.943
Total Sectors 2.794.394.295
Total Tracks 44.355.465
Tracks/Cylinder 255
Partition Disk #6, Partition #0
Partition Size 75,00 GB (80.533.684.224 bytes)

Partition Starting Offset 8.257.536 bytes
Partition Disk #6, Partition #1
Partition Size 100,01 GB (107.380.998.144 bytes)
Partition Starting Offset 80.541.974.016 bytes
Partition Disk #6, Partition #2
Partition Size 1,13 TB (1.242.806.874.624 bytes)
Partition Starting Offset 187.923.004.416 bytes

\\.\PHYSICALDRIVE5

Description Not Available
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 6
SCSI Target ID 0
Sectors/Track 63
Size 1,30 TB (1.430.729.879.040 bytes)
Total Cylinders 173.943
Total Sectors 2.794.394.295
Total Tracks 44.355.465
Tracks/Cylinder 255
Partition Disk #5, Partition #0
Partition Size 75,00 GB (80.533.684.224 bytes)
Partition Starting Offset 8.257.536 bytes
Partition Disk #5, Partition #1
Partition Size 100,01 GB (107.380.998.144 bytes)
Partition Starting Offset 80.541.974.016 bytes
Partition Disk #5, Partition #2
Partition Size 1,13 TB (1.242.806.874.624 bytes)
Partition Starting Offset 187.923.004.416 bytes

\\.\PHYSICALDRIVE4

Manufacturer 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	5
SCSI Target ID	0
Sectors/Track	63
Size	1.30 TB (1,430,729,879,040 bytes)
Total Cylinders	17,394
Total Sectors	3,594,295
Tracks/Cylinder	44,355,465
Tracks/Head	255
Partition Disk #0	Partition #0
Partition Size	75,00 GB (80,533,684,224 bytes)
Partition Starting Offset	8,257,536 bytes
Partition Disk #4	Partition #1
Partition Size	100.01 GB (107,380,998,144 bytes)
Partition Starting Offset	80,541,974,016 bytes
Partition Disk #4	Partition #2
Partition Size	1,113 TB (1,242,806,874,624 bytes)
Partition Starting Offset	187,923,004,416 bytes

[SCSI]

Item Name	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\4&254DAD54&0&4040
Memory Address	0xD0000000-0xD1FFFFFF
I/O Port	0x00003000-0x00004FFF
Memory Address	0xD4000000-0xD8FFFFFF
IRQ	24
Driver	C:\windows\system32\drivers\dac2w2k.sys
(7.00-14, 172,75 KB (176,896 bytes), 23/9/2003 15:46)	

Name: MyLex extremeRAID 2000 Disk Array Controller
(Accelerated)
Manufacturer: MyLex
Status: OK
PNP Device ID: PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&254DAD54&0&4040
Memory Address: 0xD0000000-0xD1FFFFFF
I/O Port: 0x00003000-0x00004FFF
Memory Address: 0xD4000000-0xD8FFFFFF
IRQ: 24
Driver: C:\windows\system32\drivers\dac2w2k.sys
(7.00-14, 172,75 KB (176,896 bytes), 23/9/2003 15:46)

Name	MyLex extremeRAID 2000 Disk Array Controller
(Accelerated)	
Manufacturer	MyLex
Status	OK
PNP Device ID	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&94A037D&0&4048
Memory Address	0xD2000000-0xD3FFFFFF
I/O Port	0x00004000-0x00004FFF
Memory Address	0xD8000000-0xDBFFFFFF
IRQ	25
Driver	C:\windows\system32\drivers\dac2w2k.sys
(7.00-14, 172,75 KB (176,896 bytes), 23/9/2003 15:46)	

Name: MyLex extremeRAID 2000 Disk Array Controller
(Accelerated)
Manufacturer: MyLex
Status: OK
PNP Device ID: PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&2C59ABA3&0&4040
Memory Address: 0xDC000000-0xE01FFFFFF

I/O Port	0x00005000-0x00006FFF
Memory Address	0xE4000000-0xEBFFFFFF
IRQ Channel	IRQ 26
Driver	C:\windows\system32\drivers\dac2w2k.sys
(7.00-14, 172,75 KB (176,896 bytes), 23/9/2003 15:46)	

Name: MyLex extremeRAID 2000 Disk Array Controller
(Accelerated)
Manufacturer: MyLex
Status: OK
PNP Device ID: PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&23C0528&0&4048
Memory Address: 0xDE000000-0xDFFFFFFF
I/O Port: 0x00006000-0x00006FFF
Memory Address: 0xE8000000-0xEBFFFFFF
IRQ Channel: IRQ 27
Driver: C:\windows\system32\drivers\dac2w2k.sys
(7.00-14, 172,75 KB (176,896 bytes), 23/9/2003 15:46)

Name	MyLex extremeRAID 2000 Disk Array Controller
(Accelerated)	
Manufacturer	MyLex
Status	OK
PNP Device ID	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&12E1562&0&4040
Memory Address	0xEC000000-0xF01FFFFFF
I/O Port	0x00007000-0x00008FFF
Memory Address	0xF4000000-0xFBFFFFFF
IRQ	28
Driver	C:\windows\system32\drivers\dac2w2k.sys
(7.00-14, 172,75 KB (176,896 bytes), 23/9/2003 15:46)	

Name: MyLex extremeRAID 2000 Disk Array Controller
(Accelerated)
Manufacturer: MyLex
Status: OK
PNP Device ID: PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&1BB65AAB&0&4048
Memory Address: 0xEE000000-0xEEFFFFFF
I/O Port: 0x00008000-0x00008FFF
Memory Address: 0xF8000000-0xFBFFFFFF
IRQ Channel: IRQ 29
Driver: C:\windows\system32\drivers\dac2w2k.sys
(7.00-14, 172,75 KB (176,896 bytes), 23/9/2003 15:46)

[IDE]

Item Name	Value
CSBS IDE Controller	
Manufacturer	ServerWorks
Status	OK
PNP Device ID	PCI\VEN_1166&DEV_0212&SUBSYS_02121166&REV_93\3&267A616A&0&879
I/O Port	0x00002440-0x0000244F
Driver	C:\windows\system32\drivers\pciide.sys
(5.2-3790.0 (SRV03_rtm.030324-2048), 5,50 KB (5,632 bytes), 25/3/2003 09:00)	

Name: Primary IDE Channel
(Standard IDE ATA/ATAPI controllers)
Manufacturer: (Standard IDE ATA/ATAPI controllers)
Status: OK
PNP Device ID: PCIIDE\IDECHANNEL\4&1024D5C6&0&0

I/O Port	0x00001F0-0x000001F7
IRQ Channel	IRQ 14
Driver	C:\windows\system32\drivers\atapic.sys
(5.2-3790.0 (SRV03_rtm.030324-2048), 89,00 KB (91,136 bytes), 25/3/2003 09:00)	

Name: Secondary IDE Channel
(Standard IDE ATA/ATAPI controllers)
Manufacturer: (Standard IDE ATA/ATAPI controllers)
Status: OK
PNP Device ID: PCIIDE\IDECHANNEL\4&1024D5C6&0&01
I/O Port	0x0000170-0x00000177
I/O Port	0x0000376-0x00000376
Driver	C:\windows\system32\drivers\atapic.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	
Not Available	ACPI\IBM37D4\2&0ABA3FF&0	The drivers for this device are not installed.	

[USB]

Device	PNP Device ID
ServerWorks (RCC) PCI to USB Open Host Controller	PCI\VEN_1166&DEV_0220&SUBSYS_02201166&REV_05\3&267A616A&0&87A
USB Root Hub	USB\ROOT_HUB\4&AF5358C&0

[Software Environment]

[System Drivers]				
Name	Description	Start Mode	File	Type
abiosdisk	Accept Status	Error Control	Accept Pause	Kernel Driver
	Accept Stop	Not Available	Kernel Driver	OK
	Abiosdisk	Disabled	Stopped	OK
acpi	Ignore	No	Stopped	OK
	Microsoft ACPI Driver			
	C:\windows\system32\drivers\acpi.sys			
	Kernel driver	Yes	Boot	Yes
	Running	OK	Normal	No
acpiec	Microsoft Embedded Controller Driver			
	C:\windows\system32\drivers\acpiec.sys			
	Kernel driver	Yes	Boot	Yes
	Running	OK	Normal	No
adpu160m	adpu160m	Not Available	Stopped	Kernel Driver
	Normal	No	Stopped	OK
adpu320	adpu320	Not Available	Stopped	Kernel Driver
	Normal	No	Stopped	OK
afcnt	afcnt	Not Available	Stopped	Kernel Driver
	afcnt	Not Available	Stopped	OK
afd	Normal	No	No	No
	AFD Networking Support Environment			
	C:\windows\system32\drivers\afd.sys			
	Kernel driver	Yes	Auto	

ipsec	Stopped	OK	Normal	No	No	
	IPSEC driver					
	c:\windows\system32\drivers\ipsec.sys	Yes	System			
	Kernel Driver	Running	OK	Normal	No	Yes
ipsraidn	ipsraidn	Not Available	Stopped	OK	Kernel Driver	
	No	No	No	No	No	No
isapnp	PNP ISA/EISA Bus Driver					
	c:\windows\system32\drivers\isapnp.sys	Yes	Boot			
	Kernel Driver	Running	OK	Critical	No	Yes
kbdclass	Keyboard Class Driver					
	c:\windows\system32\drivers\kbdclass.sys	Yes	System			
	Kernel Driver	Running	OK	Normal	No	Yes
ksecdd	KSecDD					
	c:\windows\system32\drivers\ksecdd.sys	Yes	Boot			
	Kernel Driver	Running	OK	Normal	No	Yes
lp6nds35	lp6nds35	Not Available	Stopped	OK	Kernel Driver	
	No	No	No	No	No	No
mac2w2k	mac2w2k					
	c:\windows\system32\drivers\mac2w2k.sys	Yes	Boot			
	Kernel Driver	Running	OK	Normal	No	Yes
mmdd	mmdd					
	c:\windows\system32\drivers\mmdd.sys	Yes	System			
	Kernel Driver	Running	OK	Ignore	No	Yes
modem	Modem					
	c:\windows\system32\drivers\modem.sys	No	Manual			
	Kernel Driver	Stopped	OK	Ignore	No	No
mouclass	Mouse Class Driver					
	c:\windows\system32\drivers\mouclass.sys	Yes	System			
	Kernel Driver	Running	OK	Normal	No	Yes
mountmgr	Mount Point Manager					
	c:\windows\system32\drivers\mountmgr.sys	Yes	Boot			
	Kernel Driver	Running	OK	Normal	No	Yes
mraid35x	mraid35x	Not Available	Stopped	OK	Kernel Driver	
	No	No	No	No	No	No
mrxdav	Webdav Client Redirector					
	c:\windows\system32\drivers\mrxdav.sys	File				
	System Driver	No	Manual	Stopped	OK	
mrxsmmb	MRXSMB					
	c:\windows\system32\drivers\mrxsmmb.sys	File				
	System Driver	Yes	System	Running	OK	
msfs	MSFS					
	c:\windows\system32\drivers\msfs.sys	File				
	System Driver	Yes	System	Running	OK	
mup	Mup					
	c:\windows\system32\drivers\mup.sys	File	Boot			
	File System Driver	Yes	Normal	No	Yes	
	Running	OK				

ndis	NDIS System Driver					
	c:\windows\system32\drivers\ndis.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
ndistapi	Remote Access NDIS TAPI Driver					
	c:\windows\system32\drivers\ndistapi.sys	Kernel Driver	Yes	Manual	No	Yes
	Running	OK				
ndisuio	NDIS Usermode I/O Protocol					
	c:\windows\system32\drivers\ndisuio.sys	Kernel Driver	No	Manual	No	No
	Stopped	OK				
ndiswan	Remote Access NDIS WAN Driver					
	c:\windows\system32\drivers\ndiswan.sys	Kernel Driver	Yes	Manual	No	Yes
	Running	OK				
ndproxy	NDIS Proxy					
	c:\windows\system32\drivers\ndproxy.sys	Kernel Driver	Yes	Manual	No	Yes
	Running	OK				
netbios	NetBIOS Interface					
	c:\windows\system32\drivers\netbios.sys	File				
	System Driver	Yes	System	Running	OK	
netbt	Netbios over Tcpip					
	c:\windows\system32\drivers\netbt.sys	System	Yes	Normal	No	Yes
	Running	OK				
nfrd960	nfrd960	Not Available	Stopped	OK	Kernel Driver	
	No	No	No	No	No	No
npfs	NPFS					
	c:\windows\system32\drivers\npfs.sys	File				
	System Driver	Yes	System	Running	OK	
ntfs	NTFS					
	c:\windows\system32\drivers\ntfs.sys	File				
	System Driver	Yes	System	Running	OK	
null	Null					
	c:\windows\system32\drivers\null.sys	System	Yes	Normal	No	Yes
	Running	OK				
parport	Parallel port driver					
	c:\windows\system32\drivers\parport.sys	Kernel Driver	Yes	Manual	No	Yes
	Running	OK				
partmgr	Partition Manager					
	c:\windows\system32\drivers\partmgr.sys	Kernel Driver	Yes	Boot	No	Yes
	Running	OK				
parvdm	Parvdm					
	c:\windows\system32\drivers\parvdm.sys	Kernel Driver	Yes	Auto	No	Yes
	Running	OK				
pci	PCI Bus Driver					
	c:\windows\system32\drivers\pci.sys	Kernel Driver	Yes	Critical	No	Yes
	Running	OK				
pciide	PCIIDE					
	c:\windows\system32\drivers\pciide.sys					

pcmcia	pcmcia					
	c:\windows\system32\drivers\pcmcia.sys	Kernel Driver	Yes	Normal	No	Yes
	Stopped	OK				
pdcomp	PDCOMP					
	c:\windows\system32\drivers\pdcomp.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
pdframe	PDFFRAME					
	c:\windows\system32\drivers\pdframe.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
pdrelli	Pdrelli					
	c:\windows\system32\drivers\pdrelli.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
pdrframe	PDRFRAME					
	c:\windows\system32\drivers\pdrframe.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
perc2	perc2					
	c:\windows\system32\drivers\perc2.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
perc2hib	perc2hib					
	c:\windows\system32\drivers\perc2hib.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
pptpminiport	PPTP Miniport					
	c:\windows\system32\drivers\rasppptp.sys	Manual				
	Running	OK				
processor	Processor Driver					
	c:\windows\system32\drivers\processr.sys	Manual				
	Running	OK				
ptilink	Direct Parallel Link Driver					
	c:\windows\system32\drivers\ptilink.sys	Manual				
	Running	OK				
ql1080	ql1080					
	c:\windows\system32\drivers\ql1080.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
ql10wnt	ql10wnt					
	c:\windows\system32\drivers\ql10wnt.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
ql12160	ql12160					
	c:\windows\system32\drivers\ql12160.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
ql1240	ql1240					
	c:\windows\system32\drivers\ql1240.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
ql1280	ql1280					
	c:\windows\system32\drivers\ql1280.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
ql2100	ql2100					
	c:\windows\system32\drivers\ql2100.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
ql2200	ql2200					
	c:\windows\system32\drivers\ql2200.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
ql2300	ql2300					
	c:\windows\system32\drivers\ql2300.sys	Kernel Driver	Yes	Normal	No	Yes
	Running	OK				
rasacd	Remote Access Auto Connection Driver					
	c:\windows\system32\drivers\rasacd.sys	Kernel Driver	Yes	System		
	Running	OK				
rasl2tp	WAN Miniport (L2TP)					
	c:\windows\system32\drivers\rasl2tp.sys					

rasppoe	Running	OK	Yes	Manual	No	Yes
	Remote Access PPPOE Driver	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\rasppoe.sys	Running	OK	Yes	Manual	Yes
raspti	Direct Parallel	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\raspti.sys	Running	OK	Yes	Manual	Yes
rdbs	Rdbss	Running	OK	Yes	Manual	Yes
System Driver	C:\windows\system32\drivers\rdbs.sys	Running	OK	Yes	Manual	Yes
rdpcdd	RDPCDD	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\rdpcdd.sys	Running	OK	Yes	Manual	Yes
rdpdr	Terminal Server Device Redirector Driver	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\rdpdr.sys	Running	OK	Yes	Manual	Yes
rdpwd	RPWD	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\rdpwd.sys	Running	OK	Yes	Manual	Yes
redbook	Digital CD Audio Playback Filter Driver	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\redbook.sys	Running	OK	Yes	Manual	Yes
secdrv	Secdrv	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\secdrv.sys	Running	OK	Yes	Manual	Yes
serenum	Serenum Filter Driver	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\serenum.sys	Running	OK	Yes	Manual	Yes
serial	Serial port driver	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\serial.sys	Running	OK	Yes	Manual	Yes
sfloppy	SFloppy	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\sfloppy.sys	Running	OK	Yes	Manual	Yes
simbad	Simbad	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\simbad.sys	Running	OK	Yes	Manual	Yes
sparrow	Sparrow	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\sparrow.sys	Running	OK	Yes	Manual	Yes
srv	File System	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\svr.sys	Running	OK	Yes	Manual	Yes
swenum	Software Bus Driver	Running	OK	Yes	Manual	Yes
	C:\windows\system32\drivers\swenum.sys	Running	OK	Yes	Manual	Yes

sync810	Not Available	Stopped	OK	Kernel Driver	Yes
	C:\windows\system32\drivers\sync810.sys	Stopped	OK	Kernel Driver	Yes
sync8xx	Sync8xx	Stopped	OK	Kernel Driver	Yes
	C:\windows\system32\drivers\sync8xx.sys	Stopped	OK	Kernel Driver	Yes
symmpi	Symmpi	Stopped	OK	Kernel Driver	Yes
	C:\windows\system32\drivers\symmpi.sys	Stopped	OK	Kernel Driver	Yes
sym_hi	Sym_hi	Stopped	OK	Kernel Driver	Yes
	C:\windows\system32\drivers\sym_hi.sys	Stopped	OK	Kernel Driver	Yes
sym_u3	Sym_u3	Stopped	OK	Kernel Driver	Yes
	C:\windows\system32\drivers\sym_u3.sys	Stopped	OK	Kernel Driver	Yes
tcpip	TCP/IP Protocol Driver	Running	OK	System	Yes
	C:\windows\system32\drivers\tcpip.sys	Running	OK	System	Yes
tdpipe	TDPIPE	Running	OK	System	Yes
	C:\windows\system32\drivers\tdpipe.sys	Running	OK	System	Yes
tdtcp	TDTCP	Running	OK	System	Yes
	C:\windows\system32\drivers\tdtcp.sys	Running	OK	System	Yes
termdd	Termdd	Running	OK	System	Yes
	C:\windows\system32\drivers\termdd.sys	Running	OK	System	Yes
toside	Toside	Running	OK	System	Yes
	C:\windows\system32\drivers\toside.sys	Running	OK	System	Yes
udfs	UDFS	Running	OK	System	Yes
	C:\windows\system32\drivers\udfs.sys	Running	OK	System	Yes
ultra	Ultra	Running	OK	System	Yes
	C:\windows\system32\drivers\ultra.sys	Running	OK	System	Yes
update	Update	Running	OK	System	Yes
	C:\windows\system32\drivers\update.sys	Running	OK	System	Yes
usbhub	USB Hub	Running	OK	System	Yes
	C:\windows\system32\drivers\usbhub.sys	Running	OK	System	Yes
usbhcd	USB Host Controller	Running	OK	System	Yes
	C:\windows\system32\drivers\usbhcd.sys	Running	OK	System	Yes
vga	VGA	Running	OK	System	Yes
	C:\windows\system32\drivers\vga.sys	Running	OK	System	Yes
viaide	Viaide	Running	OK	System	Yes
	C:\windows\system32\drivers\viaide.sys	Running	OK	System	Yes
volsnap	Volume Manager	Running	OK	System	Yes
	C:\windows\system32\drivers\volmgr.sys	Running	OK	System	Yes

wanarp	Remote Access IP ARP Driver	Running	OK	Normal	No	Yes
	C:\windows\system32\drivers\wanarp.sys	Running	OK	Normal	No	Yes
wdica	WDICA	Running	OK	Normal	No	Yes
	C:\windows\system32\drivers\wdica.sys	Running	OK	Normal	No	Yes
wlbs	Network Load Balancing	Running	OK	Normal	No	Yes
	C:\windows\system32\drivers\wlbs.sys	Running	OK	Normal	No	Yes
[Signed Drivers]						
Device Name	Driver Version	Device Class	Driver Date	INF Name	Driver Name	
Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not
ACPI Multi-processor PC	HTREE\ROOT\0	COMPUTER	10/1/2002	Standard	COMPUTER	
computers)	hal.inf	10/1/2002	Standard	COMPUTER	COMPUTER	
Microsoft	ACPI\HAL\0000	10/1/2002	Standard	COMPUTER	COMPUTER	
Microsoft	ACPI\HAL\0000	10/1/2002	Standard	COMPUTER	COMPUTER	
Processor	ACPI\HAL\PNPC08\0	PROCESSOR 5.2.3790.0	10/1/2002	Standard	PROCESSOR 5.2.3790.0	
Processor	ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_2_0	PROCESSOR 5.2.3790.0	10/1/2002	Standard	PROCESSOR 5.2.3790.0	
Processor	ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_2_1	PROCESSOR 5.2.3790.0	10/1/2002	Standard	PROCESSOR 5.2.3790.0	
Processor	ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_2_2	PROCESSOR 5.2.3790.0	10/1/2002	Standard	PROCESSOR 5.2.3790.0	
Processor	ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_2_3	PROCESSOR 5.2.3790.0	10/1/2002	Standard	PROCESSOR 5.2.3790.0	
Processor	ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_2_4	PROCESSOR 5.2.3790.0	10/1/2002	Standard	PROCESSOR 5.2.3790.0	
Processor	ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_2_5	PROCESSOR 5.2.3790.0	10/1/2002	Standard	PROCESSOR 5.2.3790.0	
Processor	ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_2_6	PROCESSOR 5.2.3790.0	10/1/2002	Standard	PROCESSOR 5.2.3790.0	
Processor	ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_2_7	PROCESSOR 5.2.3790.0	10/1/2002	Standard	PROCESSOR 5.2.3790.0	

Available	7.0.14.0	9/9/2002	Mylex	oem1.inf	Not
4&254DAD5AD0&04040	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\				
SCA Hotswap Backplane	NO	10/1/2002	ESG-SHV	SYSTEM	
scsidrv.inf	NO	Not Available			
SHV&PROD_SCA-HSBP_M15&REV_0.10\5&2D708B0C0&0060	SCSI\PROCESSOR&VEN_ESG-				
SCSI Enclosure Services Processor	SCSI\PROCESSOR&VEN_ESG-				
SYSTEM 1.0.0.0	11/8/2001	Eurologic			
oem0.inf	Not Available				
SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE					
Mylex RAID disk device	NO	DISKDRIVE			
5.2.3790.0	10/1/2002	Mylex			
-0702\5&2D708B0C0&0400	SCSI\DISK&VEN_MYLEX&PROD_EXTREMERATD_2000&REV				
Mylex RAID disk device	NO	DISKDRIVE			
5.2.3790.0	10/1/2002	Mylex			
-0702\5&2D708B0C0&0410	SCSI\DISK&VEN_MYLEX&PROD_EXTREMERATD_2000&REV				
Mylex GAM Device	NO	SYSTEM	5.2.3790.0		
Available	10/1/2002	Mylex	scsidrv.inf	Not	
\5&2D708B0C0&0660	SCSI\PROCESSOR&VEN_MYLEX&PROD_GAM_DEVICE&REV_				
DEC 21154	PCI to PCI bridge	NO	SYSTEM		
PCI 5.2.3790.0	10/1/2002	DEC			
PCI 5.2.3790.0	Not Available				
3&107002D0&0448	PCI\VEN_1011&DEV_0026&SUBSYS_00000000&REV_05\				
Mylex extremeraid 2000 Disk Array Controller					
(Accelerated)	NO	SCSIADAPTER			
Available	7.0.14.0	9/9/2002	Mylex	oem1.inf	Not
4&94A037D0&04048	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\				
SCSI Enclosure Services Processor	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
oem0.inf	Not Available				
SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE					
V_0017\5&2C8B960EC&0&0F0	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
Mylex GAM Device	NO	SYSTEM	5.2.3790.0		
SCSI Enclosure Services Processor	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
oem0.inf	Not Available				
SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE					
V_0017\5&2C8B960EC&0&2F0	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
Mylex RAID disk device	NO	DISKDRIVE			
5.2.3790.0	10/1/2002	Mylex			
-0702\5&2C8B960EC&0&400	SCSI\DISK&VEN_MYLEX&PROD_EXTREMERATD_2000&REV				
Mylex GAM Device	NO	SYSTEM	5.2.3790.0		
Available	10/1/2002	Mylex	scsidrv.inf	Not	
\5&2C8B960EC&0&660	SCSI\PROCESSOR&VEN_MYLEX&PROD_GAM_DEVICE&REV_				
PCI bus	NO	SYSTEM	5.2.3790.0		
machine.inf	Not Available				
ACPI\PNP0A03\3	10/1/2002	(Standard system devices)			
DEC 21154	PCI to PCI bridge	NO	SYSTEM		
5.2.3790.0	10/1/2002	DEC			
machine.inf	Not Available				

3&29E81982&0&040	PCI\VEN_1011&DEV_0026&SUBSYS_00000000&REV_05\				
Mylex extremeraid 2000 Disk Array Controller					
(Accelerated)	NO	SCSIADAPTER			
Available	7.0.14.0	9/9/2002	Mylex	oem1.inf	Not
4&2C59AB9A&0&04040	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\				
SCSI Enclosure Services Processor	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
oem0.inf	Not Available				
SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE					
V_0017\5&0DD1A660&0&0F0	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
Mylex RAID disk device	NO	DISKDRIVE			
5.2.3790.0	10/1/2002	Mylex			
-0702\5&0DD1A660&0&400	SCSI\DISK&VEN_MYLEX&PROD_EXTREMERATD_2000&REV				
Mylex GAM Device	NO	SYSTEM	5.2.3790.0		
Available	10/1/2002	Mylex	scsidrv.inf	Not	
\5&0DD1A660&0&660	SCSI\PROCESSOR&VEN_MYLEX&PROD_GAM_DEVICE&REV_				
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\					
3&29E81982&0&048	PCI\VEN_1011&DEV_0026&SUBSYS_00000000&REV_05\				
Mylex extremeraid 2000 Disk Array Controller					
(Accelerated)	NO	SCSIADAPTER			
Available	7.0.14.0	9/9/2002	Mylex	oem1.inf	Not
4&23E052&0&04048	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\				
SCSI Enclosure Services Processor	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
oem0.inf	Not Available				
SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE					
V_0017\5&5DDFD0&0&0F0	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
Mylex GAM Device	NO	SYSTEM	5.2.3790.0		
SCSI Enclosure Services Processor	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
oem0.inf	Not Available				
SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE					
V_0017\5&5DDFD0&0&2F0	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
Mylex RAID disk device	NO	DISKDRIVE			
5.2.3790.0	10/1/2002	Mylex			
-0702\5&5DDFD0&0&400	SCSI\DISK&VEN_MYLEX&PROD_EXTREMERATD_2000&REV				
Mylex GAM Device	NO	SYSTEM	5.2.3790.0		
Available	10/1/2002	Mylex	scsidrv.inf	Not	
\5&5DDFD0&0&660	SCSI\PROCESSOR&VEN_MYLEX&PROD_GAM_DEVICE&REV_				
PCI bus	NO	SYSTEM	5.2.3790.0		
10/1/2002	(Standard system devices)				

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\					
3&6172E68DD&0&040	PCI\VEN_1011&DEV_0026&SUBSYS_00000000&REV_05\				
Mylex extremeraid 2000 Disk Array Controller					
(Accelerated)	NO	SCSIADAPTER			
Available	7.0.14.0	9/9/2002	Mylex	oem1.inf	Not
4&12E1562D&0&04040	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\				
SCSI Enclosure Services Processor	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
oem0.inf	Not Available				
SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE					
V_0017\5&E99D885&0&0F0	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
Mylex RAID disk device	NO	DISKDRIVE			
5.2.3790.0	10/1/2002	Mylex			
-0702\5&E99D885&0&400	SCSI\DISK&VEN_MYLEX&PROD_EXTREMERATD_2000&REV				
Mylex GAM Device	NO	SYSTEM	5.2.3790.0		
Available	10/1/2002	Mylex	scsidrv.inf	Not	
\5&E99D885&0&660	SCSI\PROCESSOR&VEN_MYLEX&PROD_GAM_DEVICE&REV_				
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\					
3&6172E68DD&0&048	PCI\VEN_1011&DEV_0026&SUBSYS_00000000&REV_05\				
Mylex extremeraid 2000 Disk Array Controller					
(Accelerated)	NO	SCSIADAPTER			
Available	7.0.14.0	9/9/2002	Mylex	oem1.inf	Not
4&18B65AA&0&04048	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\				
SCSI Enclosure Services Processor	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
oem0.inf	Not Available				
SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE					
V_0017\5&2C06D47D&0&0F0	SCSI\PROCESSOR&VEN_EUROLOGC&PROD_ULTRABLOC&RE				
Mylex RAID disk device	NO	DISKDRIVE			
5.2.3790.0	10/1/2002	Mylex			
-0702\5&2C06D47D&0&400	SCSI\DISK&VEN_MYLEX&PROD_EXTREMERATD_2000&REV				
Mylex GAM Device	NO	SYSTEM	5.2.3790.0		
Available	10/1/2002	Mylex	scsidrv.inf	Not	

[illegible]

mnmdd	Not Available	LEGACYDRIVER ROOT\LEGACY_MNDD\0000	Not Available
mountmgr	Not Available	LEGACYDRIVER ROOT\LEGACY_MOUNTMGR\0000	Not Available
NDIS System Driver	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	LEGACYDRIVER	Not Available
Remote Access NDIS TAPI Driver	Root\Legacy_Ndis\0000	Not Available	Not Available
Available	LEGACYDRIVER	Not Available	Not Available
Available	LEGACYDRIVER	Not Available	Not Available
NDIS Usermode TCP/IP Protocol	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available	Not Available
NDProxy	Root\Legacy_NDISUIO\0000	LEGACYDRIVER	Not Available
Available	Not Available	Root\LEGACY_NDPROXY\0000	Not Available
NetBios over Tcpip	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Null	Root\Legacy_NETBT\0000	LEGACYDRIVER	Not Available
Available	Not Available	Root\LEGACY_NULL\0000	Not Available
Partition Manager	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Parvdm	Root\Legacy_PARTMGR\0000	LEGACYDRIVER	Not Available
Available	Not Available	Root\LEGACY_PARVDM\0000	Not Available
Remote Access Auto Connection Driver	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available	Not Available
RDPcdd	Root\Legacy_RASCD\0000	LEGACYDRIVER	Not Available
Available	Not Available	Root\LEGACY_RDPcdd\0000	Not Available
TCP/IP Protocol driver	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available	Not Available
VGA Display Controller	Root\Legacy_TCPIP\0000	Not Available	Not Available
Available	LEGACYDRIVER	Not Available	Not Available
volsnap	Root\Legacy_VGASAVE\0000	LEGACYDRIVER	Not Available
Available	Not Available	Root\LEGACY_VOLSNAP\0000	Not Available
Remote Access IP ARP driver	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available	Not Available
Audio Codecs	Root\Legacy_WMAMP\0000	MEDIA 5.2.3790.0	Standard system devices)
Legacy Audio Wave In	Root\Media\WMACM	No MEDIA	10/1/2002 (Standard system devices)
system devices)	wave.inf	Not Available	Not Available
Root\Media\WS_MMIO	5.2.3790.0	No MEDIA	10/1/2002 (Standard system devices)
Root\Media\WS_MMIO	wave.inf	Not Available	Not Available

Itautec Philco TPC-C FULL DISCLOSURE REPORT
© 2003 Itautec Philco, all rights reserved

141November, 2003

wbemprox KB (17,920 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	25/10/2003 22:26	Microsoft Corporation	17,50
wbemcom	c:\windows\system32\wbem\wbemprox.dll	25/3/2003 09:00	Microsoft Corporation	
wbemcom	c:\windows\system32\wbem\wbemcom.dll	25/3/2003 09:00	Microsoft Corporation	
wbemsvcs	c:\windows\system32\wbem\wbemsvcs.dll	25/3/2003 09:00	Microsoft Corporation	
fastprox	c:\windows\system32\wbem\wbemfastprox.dll	25/10/2003 22:26	Microsoft Corporation	
msvc60	c:\windows\system32\wbem\wbemfastprox.dll	25/3/2003 09:00	Microsoft Corporation	
ntdsapi	c:\windows\system32\wbem\wbemntdsapi.dll	25/3/2003 09:00	Microsoft Corporation	
dnsapi	c:\windows\system32\wbem\wbemdnsapi.dll	25/3/2003 09:00	Microsoft Corporation	
services	c:\windows\system32\wbem\wbemservices.dll	25/3/2003 09:00	Microsoft Corporation	
scscsv	c:\windows\system32\wbem\wbemscscsv.dll	25/3/2003 09:00	Microsoft Corporation	
authz	c:\windows\system32\wbem\wbemauthz.dll	25/3/2003 09:00	Microsoft Corporation	
umppmgr	c:\windows\system32\wbem\wbemumppmgr.dll	25/3/2003 09:00	Microsoft Corporation	
ncobjapi	c:\windows\system32\wbem\wbemncobjapi.dll	25/3/2003 09:00	Microsoft Corporation	
eventlog	c:\windows\system32\wbem\wbemeventlog.dll	25/3/2003 09:00	Microsoft Corporation	
lsass	c:\windows\system32\wbem\wbemlsass.dll	25/3/2003 09:00	Microsoft Corporation	
samsrv	c:\windows\system32\wbem\wbemsamsrv.dll	25/3/2003 09:00	Microsoft Corporation	
cryptdll	c:\windows\system32\wbem\wbemcryptdll.dll	25/3/2003 09:00	Microsoft Corporation	
msprivs	c:\windows\system32\wbem\wbemmsprivs.dll	25/3/2003 09:00	Microsoft Corporation	
kerberos	c:\windows\system32\wbem\wbemkerberos.dll	25/3/2003 09:00	Microsoft Corporation	

msv1_0	c:\windows\system32\kerberos.dll	25/3/2003 09:00	Microsoft Corporation	
netlogon	c:\windows\system32\netlogon.dll	25/3/2003 09:00	Microsoft Corporation	
w32time	c:\windows\system32\w32time.dll	25/3/2003 09:00	Microsoft Corporation	
iphlpapi	c:\windows\system32\iphlpapi.dll	25/3/2003 09:00	Microsoft Corporation	
channel	c:\windows\system32\channel.dll	25/3/2003 09:00	Microsoft Corporation	
wdigest	c:\windows\system32\wdigest.dll	25/3/2003 09:00	Microsoft Corporation	
rasfm	c:\windows\system32\rasfm.dll	25/3/2003 09:00	Microsoft Corporation	
kdcsvc	c:\windows\system32\kdcsvc.dll	25/3/2003 09:00	Microsoft Corporation	
ntdsapi	c:\windows\system32\ntdsapi.dll	25/3/2003 09:00	Microsoft Corporation	
mswsock	c:\windows\system32\mswsock.dll	25/3/2003 09:00	Microsoft Corporation	
esent	c:\windows\system32\esent.dll	25/3/2003 09:00	Microsoft Corporation	
scecli	c:\windows\system32\scecli.dll	25/3/2003 09:00	Microsoft Corporation	
wshtcapi	c:\windows\system32\wshtcapi.dll	25/3/2003 09:00	Microsoft Corporation	
dsenh	c:\windows\system32\dsenh.dll	25/3/2003 09:00	Microsoft Corporation	
svchost	c:\windows\system32\svchost.exe	25/3/2003 09:00	Microsoft Corporation	
rpcss	c:\windows\system32\rpcss.dll	25/3/2003 09:00	Microsoft Corporation	
termerv	c:\windows\system32\termerv.dll	25/10/2003	Microsoft Corporation	

icaapi	c:\windows\system32\icaapi.dll	25/10/2003 22:26	Microsoft Corporation	10,50
mstlsapi	c:\windows\system32\mstlsapi.dll	25/3/2003 09:00	Microsoft Corporation	
dnserver	c:\windows\system32\dnserver.dll	25/3/2003 09:00	Microsoft Corporation	
es	c:\windows\system32\es.dll	25/3/2003 09:00	Microsoft Corporation	
wmisvc	c:\windows\system32\wmisvc.dll	25/10/2003	Microsoft Corporation	
vssapi	c:\windows\system32\vssapi.dll	25/3/2003 09:00	Microsoft Corporation	
sens	c:\windows\system32\sens.dll	25/3/2003 09:00	Microsoft Corporation	
comsvcs	c:\windows\system32\comsvcs.dll	25/10/2003	Microsoft Corporation	
wbemcore	c:\windows\system32\wbemcore.dll	25/10/2003	Microsoft Corporation	
esscli	c:\windows\system32\esscli.dll	25/10/2003	Microsoft Corporation	
wniutlts	c:\windows\system32\wniutlts.dll	25/10/2003	Microsoft Corporation	
repdrvs	c:\windows\system32\repdrvs.dll	25/10/2003	Microsoft Corporation	
wniprsvd	c:\windows\system32\wniprsvd.dll	25/10/2003	Microsoft Corporation	
wbemess	c:\windows\system32\wbemess.dll	25/10/2003	Microsoft Corporation	
ncprov	c:\windows\system32\ncprov.dll	25/10/2003	Microsoft Corporation	
netman	c:\windows\system32\netman.dll	25/3/2003 09:00	Microsoft Corporation	
rasapi32	c:\windows\system32\rasapi32.dll	25/3/2003 09:00	Microsoft Corporation	
rasman	c:\windows\system32\rasman.dll	25/3/2003 09:00	Microsoft Corporation	

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	
wzcsvc	5.2.3790.0 (srv03_rtm.030324-2048) 272,50 KB (279,040 bytes) 25/3/2003 09:15 Microsoft Corporation	
wmi	5.2.3790.0 (srv03_rtm.030324-2048) KB (6.656 bytes) 25/3/2003 09:00 Microsoft Corporation	6,50
dhcpcsvc	2.3790.0 (c:\windows\system32\wininit.dll) 101,50 KB (103,936 bytes) 25/3/2003 09:00 Microsoft Corporation	
wzcsapi	5.2.3790.0 (srv03_rtm.030324-2048) KB (25,088 bytes) 25/3/2003 09:15 Microsoft Corporation	24,50
rasdlog	5.2.3790.0 (srv03_rtm.030324-2048) 642,00 KB (657,408 bytes) 25/3/2003 09:00 Microsoft Corporation	
rasadhlp	5.2.3790.0 (srv03_rtm.030324-2048) KB (6.656 bytes) 25/3/2003 09:00 Microsoft Corporation	6,50
wkssvc	5.2.3790.0 (srv03_rtm.030324-2048) 125,00 KB (128,000 bytes) 25/3/2003 09:00 Microsoft Corporation	
srsvsc	5.2.3790.0 (srv03_rtm.030324-2048) KB (91,136 bytes) 25/3/2003 09:00 Microsoft Corporation	89,00
browser	5.2.3790.0 (srv03_rtm.030324-2048) KB (72,192 bytes) 25/3/2003 09:00 Microsoft Corporation	70,50
netrap	5.2.3790.0 (srv03_rtm.030324-2048) KB (11,776 bytes) 25/3/2003 09:00 Microsoft Corporation	11,50
ntlsapi	5.2.3790.0 (srv03_rtm.030324-2048) KB (8,192 bytes) 25/3/2003 09:00 Microsoft Corporation	8,00
pchsvc	5.2.3790.0 (srv03_rtm.030324-2048) KB (32,256 bytes) 25/10/2003 22:29 Microsoft Corporation	31,50
11	c:\windows\pchealth\helpctr\binaries\pchsvc.dll	
wbemicons	5.2.3790.0 (srv03_rtm.030324-2048) KB (70,656 bytes) 25/10/2003 22:26 Microsoft Corporation	69,00
explorer	5.2.3790.0 (srv03_rtm.030324-2048) KB (1,008,50 KB (1,032,704 bytes) 25/3/2003 09:00 Microsoft Corporation	
browseui	5.2.3790.0 (srv03_rtm.030324-2048) KB (1,057,280 bytes) 25/3/2003 09:00 Microsoft Corporation	1,01
shdocvw	6.00.3790.0 (srv03_rtm.030324-2048) KB (1,393,664 bytes) 25/3/2003 09:00 Microsoft Corporation	1,33
appleip	5.2.3790.0 (srv03_rtm.030324-2048) 122,00 KB (124,928 bytes) 25/3/2003 09:00 Microsoft Corporation	

Microsoft Corporation	c:\windows\system32\appleip.dll	
themeui	6.00.3790.0 (srv03_rtm.030324-2048) 360,50 KB (369,152 bytes) 25/3/2003 09:00 Microsoft Corporation	
msimg32	5.2.3790.0 (srv03_rtm.030324-2048) KB (4,608 bytes) 25/3/2003 09:00 Microsoft Corporation	4,50
LinkInfo	5.2.3790.0 (srv03_rtm.030324-2048) KB (16,896 bytes) 25/3/2003 09:00 Microsoft Corporation	16,50
ntshrui	6.00.3790.0 (srv03_rtm.030324-2048) 136,00 KB (139,264 bytes) 25/3/2003 09:00 Microsoft Corporation	
netshell	5.2.3790.0 (srv03_rtm.030324-2048) MB (1,747,456 bytes) 25/3/2003 09:00 Microsoft Corporation	1,67
cluserapi	5.2.3790.0 (srv03_rtm.030324-2048) KB (57,344 bytes) 25/3/2003 09:00 Microsoft Corporation	56,00
urlmon	6.00.3790.0 (srv03_rtm.030324-2048) 501,50 KB (513,536 bytes) 25/3/2003 09:00 Microsoft Corporation	
webcheck	6.00.3790.0 (srv03_rtm.030324-2048) 361,50 KB (367,776 bytes) 25/3/2003 09:00 Microsoft Corporation	
wsock32	5.2.3790.0 (srv03_rtm.030324-2048) KB (22,528 bytes) 25/3/2003 09:00 Microsoft Corporation	22,00
stobject	5.2.3790.0 (srv03_rtm.030324-2048) 117,50 KB (120,320 bytes) 25/3/2003 09:00 Microsoft Corporation	
batmeter	6.00.3790.0 (srv03_rtm.030324-2048) KB (29,184 bytes) 25/3/2003 09:00 Microsoft Corporation	28,50
powrprof	6.00.3790.0 (srv03_rtm.030324-2048) KB (34,848 bytes) 25/3/2003 09:00 Microsoft Corporation	14,50
browseui	6.00.3790.0 (srv03_rtm.030324-2048) KB (63,488 bytes) 25/3/2003 09:00 Microsoft Corporation	62,00
shdoclc	6.00.3790.0 (srv03_rtm.030324-2048) 588,50 KB (602,624 bytes) 25/3/2003 09:00 Microsoft Corporation	
wininet	6.00.3790.0 (srv03_rtm.030324-2048) 609,00 KB (623,616 bytes) 25/3/2003 09:00 Microsoft Corporation	
drprov	5.2.3790.0 (srv03_rtm.030324-2048) KB (12,800 bytes) 25/3/2003 09:00 Microsoft Corporation	12,50
ntlanman	5.2.3790.0 (srv03_rtm.030324-2048) KB (41,984 bytes) 25/3/2003 09:00 Microsoft Corporation	41,00
netui0	5.2.3790.0 (srv03_rtm.030324-2048) KB (77,312 bytes) 25/3/2003 09:00 Microsoft Corporation	75,50

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	
davclnt	5.2.3790.0 (srv03_rtm.030324-2048) KB (24,064 bytes) 25/3/2003 09:00 Microsoft Corporation	23,50
printui	5.2.3790.0 (srv03_rtm.030324-2048) 536,50 KB (549,376 bytes) 25/3/2003 09:00 Microsoft Corporation	
cfmgpr32	5.2.3790.0 (srv03_rtm.030324-2048) KB (17,920 bytes) 25/3/2003 09:00 Microsoft Corporation	17,50
mydocs	6.00.3790.0 (srv03_rtm.030324-2048) KB (90,112 bytes) 25/3/2003 09:00 Microsoft Corporation	88,00
mscms	5.2.3790.0 (srv03_rtm.030324-2048) KB (70,656 bytes) 25/3/2003 09:00 Microsoft Corporation	69,00
zipfldr	6.00.3790.0 (srv03_rtm.030324-2048) 316,00 KB (323,584 bytes) 25/3/2003 09:00 Microsoft Corporation	
actxprvw	6.00.3790.0 (srv03_rtm.030324-2048) KB (37,280 bytes) 25/3/2003 09:00 Microsoft Corporation	95,00
srchui	1.00 25/10/2003 22:29 Microsoft Corporation	
msacm32	5.2.3790.0 (srv03_rtm.030324-2048) KB (69,120 bytes) 25/3/2003 09:00 Microsoft Corporation	67,50
oleacc	4.2.5406.0 (srv03_rtm.030324-2048) 171,00 KB (175,104 bytes) 25/3/2003 09:00 Microsoft Corporation	
msi	2.0.3790.0 25/3/2003 09:00 Microsoft Corporation	
srchctls	1.00 25/10/2003 22:29 Microsoft Corporation	
msxml3	8.40.9419.0 (srv03_rtm.030324-2048) 25/3/2003 09:00 Microsoft Corporation	
sensapi	5.2.3790.0 (srv03_rtm.030324-2048) KB (6,144 bytes) 25/3/2003 09:00 Microsoft Corporation	6,00
jscrip	5.6.0.8515 25/3/2003 09:00 Microsoft Corporation	
sendmail	6.00.3790.0 (srv03_rtm.030324-2048) KB (53,248 bytes) 25/3/2003 09:00 Microsoft Corporation	52,00
wpaballn	5.2.3790.0 (srv03_rtm.030324-2048) KB (31,744 bytes) 25/3/2003 09:00 Microsoft Corporation	31,00
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)
25/10/2003 22:45 Microsoft Corporation
c:\program files\microsoft sql
server\mssql\bin\sqlservr.exe
opends60 2000.080.0194.00 24.06 KB (24.639 bytes)
25/10/2003 22:46 Microsoft Corporation
c:\program files\microsoft sql
server\mssql\bin\opends60.dll
ums 2000.080.0760.00 52.55 KB (53.808 bytes)
25/10/2003 22:46 Microsoft Corporation
c:\program files\microsoft sql
server\msmsg\bin\ums.dll
sqlsort 2000.080.0760.00 576.56 KB (590.396 bytes)
25/10/2003 22:46 Microsoft Corporation
c:\program files\microsoft sql
server\msmsg\bin\sqlsort.dll
msvcirt 2.0.3790.0 (srv03_rtm.030324-2048) 50.00
KB (51.200 bytes) 25/3/2003 09:00 Microsoft
Corporation c:\windows\system32\msvcirt.dll
sqllevn70 2000.080.0760.00 28.00 KB (28.672 bytes)
25/10/2003 22:46 Microsoft Corporation
c:\program files\microsoft sql
server\msmsg\bin\resources\1033\sqllevn70.rll
xolehlp 2001.12.4720.0 (srv03_rtm.030324-2048) 8.50
KB (8.704 bytes) 25/10/2003 22:26 Microsoft
Corporation c:\windows\system32\xolehlp.dll
msdtcprx 2001.12.4720.0 (srv03_rtm.030324-2048)
447.50 KB (437.760 bytes) 25/10/2003
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
resultls 5.2.3790.0 (srv03_rtm.030324-2048) 59.00
KB (60.416 bytes) 25/3/2003 09:00 Microsoft
Corporation c:\windows\system32\resultls.dll
mfc42u 6.05.3014.0 960.00 KB (983.040 bytes)
25/3/2003 09:00 Microsoft Corporation
c:\windows\system32\mfc42u.dll
winmrn 5.2.3790.0 (srv03_rtm.030324-2048) 15.00
KB (15.360 bytes) 25/3/2003 09:00 Microsoft
Corporation c:\windows\system32\winmrn.dll
ssnetlib 2000.080.0760.00 80.56 KB (82.492 bytes)
25/10/2003 22:46 Microsoft Corporation
c:\program files\microsoft sql
server\msmsg\bin\ssnetlib.dll
ssmnpn70 2000.080.0534.00 24.56 KB (25.148 bytes)
25/10/2003 22:46 Microsoft Corporation
c:\program files\microsoft sql
server\msmsg\bin\ssmnpn70.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)
25/10/2003 22:46 Microsoft Corporation
c:\program files\microsoft sql
server\msmsg\bin\ssmslpcn.dll
sqldfqr 2000.080.0760.00 192.67 KB (197.196 bytes)
25/10/2003 22:46 Microsoft Corporation
c:\program files\microsoft sql
server\msmsg\bin\sqldfqr.dll
sqltoledb 2000.085.1022.00 (srv03_rtm.030324-2048)
536.00 KB (548.864 bytes) 25/10/2003
Microsoft Corporation c:\program
files\common files\system\ole db\sqltoledb.dll

```

```

msdmt 2.80.1022.0 (srv03_rtm.030324-2048)
164.00 KB (167.936 bytes) 25/3/2003 09:00
Microsoft Corporation
c:\windows\system32\msdmt.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
msdatl3 2.80.1022.0 (srv03_rtm.030324-2048)
96.00
KB (98.304 bytes) 25/10/2003 22:29 Microsoft
Corporation c:\program files\common
files\system\ole db\msdatl3.dll
oledb32 2.80.1022.0 (srv03_rtm.030324-2048)
280.00 KB (284.800 bytes) 25/10/2003
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) 25/10/2003 22:29 Microsoft
Corporation c:\program files\common
files\system\ole db\oledb32r.dll
mmc 5.2.3790.0 (srv03_rtm.030324-2048)
762.50 KB (780.800 bytes) 25/3/2003 09:00
Microsoft Corporation
c:\windows\system32\mmc.exe
mmcbase 5.2.3790.0 (srv03_rtm.030324-2048)
70.50
KB (72.192 bytes) 25/3/2003 09:00 Microsoft
Corporation c:\windows\system32\mmcbase.dll
mmcmdmgr 5.2.3790.0 (srv03_rtm.030324-2048)
1.13
MB (1.182.720 bytes) 25/3/2003 09:00
Microsoft Corporation
c:\windows\system32\mmcmdmgr.dll
filemgmt 5.2.3790.0 (srv03_rtm.030324-2048)
327.50 KB (335.360 bytes) 25/3/2003 09:00
Microsoft Corporation
c:\windows\system32\filemgmt.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
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
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
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
mshtmlnd 6.00.3790.0 (srv03_rtm.030324-2048)
443.50 KB (454.144 bytes) 25/3/2003 09:00
Microsoft Corporation
c:\windows\system32\mshtmlnd.dll
ingutil 5.2.3790.0 (srv03_rtm.030324-2048)
35.00
KB (35.840 bytes) 25/3/2003 09:00
Microsoft Corporation c:\windows\system32\ingutil.dll
regedit 5.2.3790.0 (srv03_rtm.030324-2048)
135.50 KB (138.752 bytes) 25/3/2003 09:00
Microsoft Corporation
c:\windows\regedit.exe

```

```

aclui 5.2.3790.0 (srv03_rtm.030324-2048)
111.00 KB (113.664 bytes) 25/3/2003 09:00
Microsoft Corporation
c:\windows\system32\aclui.dll
utilib 5.2.3790.0 (srv03_rtm.030324-2048)
265.50 KB (271.872 bytes) 25/3/2003 09:00
Microsoft Corporation
c:\windows\system32\utilib.dll
clb 5.2.3790.0 (srv03_rtm.030324-2048)
11.50
KB (11.776 bytes) 25/3/2003 09:00 Microsoft
Corporation c:\windows\system32\clb.dll
helptr 5.2.3790.0 (srv03_rtm.030324-2048)
764.00 KB (782.336 bytes) 25/10/2003
Microsoft Corporation
c:\windows\pchealth\helptr\binaries\helptr.
exe
hcapres 5.2.3790.0 (srv03_rtm.030324-2048)
6.50
KB (6.656 bytes) 25/10/2003 22:29 Microsoft
Corporation c:\windows\pchealth\helptr\binaries\hcapres
.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
pchshel1 5.2.3790.0 (srv03_rtm.030324-2048)
100.50 KB (102.912 bytes) 25/10/2003
Microsoft Corporation
c:\windows\pchealth\helptr\binaries\pchshel1
.dll
vbscript 5.6.0.8515 404.00 KB (413.696 bytes)
25/3/2003 09:00 Microsoft Corporation
c:\windows\system32\vbscript.dll
mfcd42 6.05.3014.0 960.00 KB (983.040 bytes)
25/3/2003 09:00 Microsoft Corporation
c:\windows\system32\mfcd42.dll
msinfo 5.2.3790.0 (srv03_rtm.030324-2048)
358.50 KB (367.104 bytes) 25/10/2003
Microsoft Corporation
c:\windows\pchealth\helptr\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) 25/10/2003
Microsoft Corporation
c:\windows\pchealth\helptr\binaries\helpsvc.
exe
[Services]
Display Name Name State Start Mode
Service Type Path ID Error Control
Alerter Alerter Stopped Share Process
c:\windows\system32\svchost.exe -k
LocalService Normal NT AUTHORITY\LocalService
Application Layer Gateway Service ALG
c:\windows\system32\alg.exe Normal NT
AUTHORITY\LocalService
Application Management AppMgmt Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem

```

windows Audio c:\windows\system32\svchost.exe -k netsvcs Share
Process Normal LocalSystem
Background Intelligent Transfer Service BITS
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Computer Browser Running Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Indexing service C1Svc Stopped Disabled Share
Process c:\windows\system32\csvc.exe Normal
LocalSystem
Clipbook C1Psrvc Stopped Disabled Own Process
c:\windows\system32\clipsrv.exe
Normal LocalSystem
COM+ System Application COMSysApp Stopped
Manual Own Process
c:\windows\system32\dllhost.exe
/processid:{024b3bf1-fd88-11d1-960d-00805fc79235}
Normal LocalSystem
Cryptographic Services CryptSvc Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Distributed File System Dfs Stopped
Manual Own Process
c:\windows\system32\dfssvc.exe
Normal LocalSystem
DHCP Client Dhcp Stopped Manual Share
Process c:\windows\system32\svchost.exe -k
netsvcs
Normal LocalSystem
AUTHORITY Network Service NT
Logical Disk Manager Administrative Service
dnadmm Stopped Manual Share Process
c:\windows\system32\dmadmin.exe /com
Normal LocalSystem
Logical Disk Manager dmsvr Running Auto
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
DNS Client Dnscache Running Manual Share
Process c:\windows\system32\svchost.exe -k
netsvcs
Normal NT
AUTHORITY Network Service
Error Reporting Service ERSvc Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k winerr
Normal LocalSystem
Event Log Eventlog Running Auto Share Process
c:\windows\system32\services.exe
Normal LocalSystem
COM+ Event System Eventsvc Running
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Mylex Global Array Manager Server gamscm
Stopped Manual Own Process
system32\gamsrv\gamscm.exe Normal
LocalSystem
Help and Support helpsvc Running Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Human Interface Device Access Hidserv Stopped
c:\windows\system32\hidssc
Normal LocalSystem
HTTP SSL HTTPFilter Stopped Manual Share
Process c:\windows\system32\lsass.exe Normal
LocalSystem
IMAPI CD-Burning COM Service ImapiService
Stopped Disabled Own Process

c:\windows\system32\inapi.exe Normal
LocalSystem
InterSite Messaging Ismserv Stopped Disabled Own
Process c:\windows\system32\ismsserv.exe
Normal LocalSystem
Kerberos Key Distribution Center kdc
Stopped Disabled Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem
Server Tamanserver Running Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Workstation Lamanworkstation Running
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
License Logging Licenservice Stopped
Disabled Own Process
c:\windows\system32\lssrv.exe
Normal NT AUTHORITY\NetworkService
TCP/IP NetBIOS Helper Lmhosts Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k
LocalService Normal NT AUTHORITY\LocalService
Messenger Messenger Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
NetMeeting Remote Desktop Sharing nmmsrvc
Stopped Disabled Own Process
c:\windows\system32\nmmsrvc.exe
Normal LocalSystem
Distributed Transaction Coordinator MSDTC
Stopped Manual Own Process
c:\windows\system32\msdtc.exe Normal NT
AUTHORITY Network Service
Windows Installer MSIServer Stopped Manual Share
Process c:\windows\system32\msiexec.exe /v
Normal LocalSystem
Microsoft Search MSSEARCH Stopped Manual
Files\system\mssearch\bin\mssearch.exe" Normal
LocalSystem
MSSQLSERVER MSSQLSERVER Stopped
Manual Own Process
c:\program-1\micos-1\mssql\bin\sqlservr.exe
MSSQLSERVERADHDPper MSSQLServer Stopped
Files\microsoft sql server\80\tools\bin\sqladhp.exe
Normal LocalSystem
Network DDE NetDDE Stopped Disabled Share
Process c:\windows\system32\netdde.exe
Normal LocalSystem
Network DDE DSM NetDDEsdm Stopped
Disabled Share Process
c:\windows\system32\netdde.exe
Normal LocalSystem
Net Logon Netlogon Stopped Manual Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem
Network Connections Netman Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs
Network Location Awareness (NLA) Nla
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
File Replication Ntfrs Stopped Manual Own
Process c:\windows\system32\ntfrs.exe Ignore
LocalSystem

NT LM Security Support Provider NtLmSsp
Stopped Manual Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem
Removable Storage NtmsSvc Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Plug and Play PlayPlay Running Auto Share
Process c:\windows\system32\services.exe
Normal LocalSystem
IPSEC Services PolicyAgent Stopped
Manual Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem
Protected Storage ProtectedStorage Stopped
Manual Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem
Remote Access Auto Connection Manager RasAuto
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Remote Access Connection Manager Rasman
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Remote Desktop Help Session Manager RDSessMgr
Stopped Manual Own Process
c:\windows\system32\ssmgr.exe
Normal LocalSystem
Routing and Remote Access RemoteAccess
Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Remote Registry RemoteRegistry Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k regsvc
Normal NT AUTHORITY\LocalService
Remote Procedure Call (RPC) Locator RpcLocator
Stopped Manual Own Process
c:\windows\system32\locator.exe
Normal NT AUTHORITY\NetworkService
Remote Procedure Call (RPC) RpcSs Running Auto
Share Process
c:\windows\system32\svchost -k rpcss
Normal LocalSystem
Resultant Policy Object RsoProv
Stopped Manual Share Process
c:\windows\system32\rsoprov.exe
Normal LocalSystem
Special Administration Console Helper sacsvr
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Security Accounts Manager SamSs Running Auto
Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem
Smart Card Scardsvr Stopped Manual Share
Process c:\windows\system32\scardsrvr.exe
Ignore NT AUTHORITY\LocalService
Task Scheduler Schedule Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Secondary Logon Seclogon Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs
Ignore LocalSystem
System Event Notification SENS Running Auto
Share Process

```

c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Internet Connection Firewall (ICF) / Internet
Connection Sharing (ICS) SharedAccess
Stopped disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Shell Hardware Detection ShellHwDetection
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Ignore LocalSystem
Print Spooler Spooler Stopped Manual Own
Process c:\windows\system32\spoolsv.exe
Normal LocalSystem
SQLSERVERAGENT SQLSERVERAGENT Stopped
Manual LocalSystem
c:\program\microsoft\mssql\bin\sqlagent.exe
Normal LocalSystem
Windows Image Acquisition (WIA) tsisvc
Stopped disabled Share Process
c:\windows\system32\svchost.exe -k imgsvc
Normal NT AUTHORITY\LocalService
Microsoft Software Shadow Copy Provider swprv
Stopped Manual Own Process
c:\windows\system32\svchost.exe -k swprv
Normal LocalSystem
Performance Logs and Alerts SystemLog Stopped
Manual Own Process
c:\windows\system32\smlogsvc.exe
Normal NT AUTHORITY\NetworkService
Telephony Tapisrv Stopped Manual Share Process
c:\windows\system32\svchost.exe -k tapisrv
Normal LocalSystem
Terminal Services TermService Running
Manual Share Process
c:\windows\system32\svchost.exe -k termvcs
Normal LocalSystem
Themes Themes Stopped disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Telnet Tlntsvr Stopped disabled Own Process
c:\windows\system32\tlntsvr.exe
Normal NT AUTHORITY\LocalService
Distributed Link Tracking Service TrkSvr
Stopped disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Distributed Link Tracking Client TrkLks
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Terminal Services Session Directory Tssdis
Stopped disabled Own Process
c:\windows\system32\tssdis.exe
Normal LocalSystem
Upload Manager uploadmgr Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Uninterruptible Power Supply UPS Stopped
Manual Own Process
c:\windows\system32\ups.exe
Normal LocalSystem
Virtual Disk Service vds Stopped
Manual Own Process
c:\windows\system32\vdss.exe
Normal LocalSystem
Volume Shadow Copy VSS Stopped Manual Own
Process c:\windows\system32\svsdc.exe Normal
LocalSystem

```

```

Windows Time w32Time Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
WebClient webClient Stopped disabled Share Process
c:\windows\system32\svchost.exe -k
LocalService Normal NT AUTHORITY\LocalService
WinHTTP Web Proxy Auto-Discovery Service
WinHttpAutoProxySvc Stopped Manual Share
Process c:\windows\system32\svchost.exe -k
LocalService Normal NT AUTHORITY\LocalService
Windows Management Instrumentation winmgmt
Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Portable Media Serial Number Service WmdmPmSN
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Windows Management Instrumentation Driver Extensions
Wmi Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
WMI Performance Adapter WmiApSrv Stopped
Manual Own Process
c:\windows\system32\wbem\wmiapshr.exe
Normal LocalSystem
Automatic Updates wuauclt Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
Wireless Configuration Wizard WzCsvc Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem
[Program Groups]
Group Name Name User Name
Accessories Default User User:Accessories
Accessories\Accessibility Default Default User
Accessories\Entertainment Default Default User
Startup Default User:Startup Default User
Accessories All Users:Accessories All
Users
Accessories\Accessibility All
Users
Accessories\Communications All
Users
Accessories\Entertainment All
Users
Accessories\Entertainment 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 NT
Accessories\Accessibility NT
AUTHORITY\SYSTEM:Accessories\Accessibility
AUTHORITY\SYSTEM

```

```

Accessories\Entertainment NT
AUTHORITY\SYSTEM:Accessories\Entertainment
AUTHORITY\SYSTEM NT
Startup NT AUTHORITY\SYSTEM:Startup NT
Accessories SQL\Administrator:Accessories
SQL\Administrator
Accessories\Accessibility
SQL\Administrator:Accessories\Accessibility
Accessories\Entertainment
SQL\Administrator
Administrative Tools
SQL\Administrator:Administrative Tools
Startup
SQL\Administrator:Startup
SQL\Administrator
[Startup Programs]
Command User Name Location
Program desktop.ini NT AUTHORITY\SYSTEM
desktop Startup
desktop desktop.ini SQL\Administrator
desktop desktop.ini .DEFAULT Startup
desktop desktop.ini All Users Common Startup
[OLE Registration]
Object Local Server
Sound (OLE2) sndrec32.exe
Media Clip mplay32.exe /avi
Video Clip mplay32.exe /mid
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
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
actcxprxy.dll	Company	6.0.3790.0		C:\WINDOWS\system32
advpack.dll	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
asctrls.ocx	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
browseui.dll	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
cdffview.dll	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
comctl32.dll	Microsoft Corporation	5.82.3790.0		C:\WINDOWS\system32
dxtmsft.dll	Microsoft Corporation	6.3.3790.0		C:\WINDOWS\system32
iecont.dll	Microsoft Corporation	6.3.3790.0		C:\WINDOWS\system32
iecontlc.dll	Microsoft Corporation	6.3.3790.0		C:\WINDOWS\system32
iedkcs32.dll	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
iepeers.dll	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
iesetup.dll	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
ieuinitt.inf	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
ieupdate.exe	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
Explorer	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
imgutil.dll	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
inetcp1.cpl	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
inetcp1c.dll	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
inseng.dll	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
mlang.dll	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
msencode.dll	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32
mshta.exe	Microsoft Corporation	6.0.3790.0		C:\WINDOWS\system32

mshtml.dll	25/3/2003 09:00:00	2,848 KB
mshtml.tlb	25/3/2003 09:00:00	1,319 KB
mshtml.dll	25/3/2003 09:00:00	444 KB
mshtml.dll	25/3/2003 09:00:00	55 KB
msident.dll	25/3/2003 09:00:00	47 KB
msidntld.dll	25/3/2003 09:00:00	15 KB
msieftp.dll	25/3/2003 09:00:00	230 KB
msrating.dll	25/3/2003 09:00:00	132 KB
msstime.dll	25/3/2003 09:00:00	491 KB
occache.dll	25/3/2003 09:00:00	89 KB
prctexte.ocx	25/3/2003 09:00:00	78 KB
sendmail.dll	25/3/2003 09:00:00	52 KB
shdoclc.dll	25/3/2003 09:00:00	589 KB
shdocvw.dll	25/3/2003 09:00:00	1,361 KB
shfolder.dll	25/3/2003 09:00:00	23 KB
shlwapi.dll	25/3/2003 09:00:00	281 KB
tdc.ocx	25/3/2003 09:00:00	58 KB
url.dll	25/3/2003 09:00:00	36 KB
urlmon.dll	25/3/2003 09:00:00	502 KB
webcheck.dll	25/3/2003 09:00:00	262 KB
wininet.dll	25/3/2003 09:00:00	609 KB
[Connectivity]		
Item Connection Preference	Value	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
Internet 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]

```
[HEX]LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
netinfo\parameters"
"listenbacklog"=dword:00000032
"bitspactimeout"=hex(7)24c,00,44,00,41,00,53,00,
56,00,43,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,4e,00,5
4,00,50,00,53,00,56,
00,43,00,00,00,00,
"poolthreadlimit"=dword:000007fe
"threadtimeout"=dword:00015180

[HEX]LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\I
netinfo\performance"
"Open"="openInFoPerformanceData"
"Close"="closeInFoPerformanceData"
"Collect"="collectInFoPerformanceData"
"Last Counter"=dword:00000084
"Last Help"=dword:0000008e
"First Counter"=dword:000008a4
"First Help"=dword:00000083
"Priority Validation"
Code="hex:76,8e,03,36,ae,a0,c3,01,10,25,00,00,00,00,00,
00,
"wbemadap1Time"=hex:0,3,f0,97,ab,d4,c0,01
"wbemadap1Size"=dword:00002510
"wbemadapStatus"=dword:00000000
```

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
"DisplayName"="World Wide Web Publishing Service"
"DependenceSource"=hex(7):49;00;49;00;53;00;41;00;44;00;
4d;00;49;00;4e;00;00;\\
00;00;00;00
"DependenceGroup"=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\35VCV\Parameters]
    "MajorVersion"=dword:00000005
    "MinorVersion"=dword:00000000
    "InstallPath"="c:\WINNT\System32\inetshr\"
    "CertMapList"="c:\WINNT\System32\inetshr\iisrmap.d11"
    "AccessDeniedMessage"="Error: Access is Denied."
    "Filter DLLs"=""
    "LogFileDirectory"="c:\WINNT\System32\LogFiles"
    "AcceptExOutstanding"=dword:00000028

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDS\Server.DataFactory]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots]
    "C"="c:\inetpub\wwwroot\",205"
    "Scripts"="c:\inetpub\scripts\",1"
    "IISHelp"="c:\winnt\help\iishelp\",1"
    "IISAdmin"="c:\WINNT\System32\inetshr\iisadmin\",1"
    "IISASPM"="c:\inetpub\issamples\",1"
    "IISCommonFiles"="c:\inetpub\files\common files\system\usad\",1"
    "vri_bin"="c:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40\isapi\",1"
    "Printers"="c:\\WINNT\\web\\printers\",201"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Performance]
    "Library"="w3ctrs.dll"
    "Open"="OpenW3perfomanceData"
    "Close"="CloseW3performanceData"
    "Collect"="CollectW3performanceData"
    "Last Counter"=dword:00000988
    "Last Help"=dword:00000989
    "First Counter"=dword:00000986
    "First Help"=dword:00000987
    "Code"=hex:92,36,44,39,a0,c3,01,10,3d,00,00,00,00,00,00,00,00
    "WbemAdapFileTime"=hex:00,a3,f0,97,ab,d4,c0,01
    "WbemAdapFileSize"=dword:00001d10
    "WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Security]
    "Security"=hex:01,00,14,80,a0,00,00,ac,00,00,00,14,0,0,0,30,00,00,02,\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,0,0,0,0,00,01,00,00,\
00,02,00,70,00,04,00,00,00,00,18,00,fd,01,02,00,0,1,01,00,00,00,\
05,12,00,00,18,c1,0f,00,00,1c,00,ff,01,0f,00,01,0,2,00,00,00,05,\
20,00,00,00,20,02,00,00,00,00,00,00,00,18,00,8d,01,0,0,01,01,00,00,\
00,01,01,00,00,
```

COM+ Settings

```
TPCC.ALLTxns:
  Activation:
    Enable Object Pooling selected
    Minimum Pool Size: 60
    Maximum Pool Size: 60
    Creation timeout: 60,000
    Enable Object Construction
    Enable Just in Time Activation
  Concurrency:
    Concurrency required
```

TPCC Application Registry Parameters

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPC]
"Path"="C:\inetpub\wwwroot\"
"NumberOfServerThreads"=dword:00000040
"MaxConnections"=dword:00009c40
"MaxPendingIo"=dword:00000bb8
"Db.Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="sql"
"DbName"="tpc"
"DbUser"="sa"
"DbPassword"=""
"COM.SignupPool"="YES"
```

Client System Configuration

System Information report written at: 11/03/2003
04:34:28 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 CLIO1
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\System32
System Root C:\WINNT
Boot Device (Device\Harddisk0\Partition1)
Local Disk (D:) C:\
Logical Drives (C:)\
User Name CLIO1\Administrator
Time Zone E. South America
Standard Time
Total Physical Memory 1,048,044 KB
Available Physical Memory 880,168 KB
Total Virtual Memory 3,570,080 KB
Available Virtual Memory 3,329,644 KB
Page File Space 2,522,036 KB
Page File C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource Device
No conflicted/shared resources

[DMA]

Channel Device Status
2 Standard floppy disk controller OK
4 Direct memory access controller OK

[Forced Hardware]

Device PNP Device ID
No Forced Hardware

[I/O]

Address Range Device Status
0x0000-0x00FF PCI bus OK
0x0000-0x00FF Direct memory access controller
0x0000-0x4FFF PCI bus OK
0x1000-0x144F PCI bus OK
0x1000-0x144F ATI Technologies Inc. 3D RAGE IIC
PCI OK
0xF000-0xFFFF PCI bus OK
0x0380-0x038B ATI Technologies Inc. 3D RAGE IIC
PCI OK
0x03C0-0x03DF ATI Technologies Inc. 3D RAGE IIC
PCI OK
0x1400-0x143F Intel 8255x-based PCI Ethernet
AdapteC (C072000) OK
0x0278-0x0279 ISAPNP Read Data Port OK
0x0279-0x0279 ISAPNP Read Data Port OK
0x02E4-0x02E7 ISAPNP Read Data Port OK
0x0070-0x0071 System CMOS/real time clock OK
0x03F8-0x03FE Communications Port (COM1) OK
0x02F8-0x02FF Communications Port (COM2) OK
0x0378-0x037F Printer Port (LPT1) OK
0x03F0-0x03F5 Standard Floppy disk controller OK

0x03F7-0x03F7 Standard floppy disk controller
OK
0x0060-0x0060 Standard 101/102-Key or Microsoft
Natural PS/2 keyboard OK
0x0064-0x0064 Standard 101/102-Key or Microsoft
Natural PS/2 keyboard OK
0x0010-0x001F Direct memory access controller
OK
0x0080-0x008F Direct memory access controller
OK
0x00C0-0x00DF Direct memory access controller
OK
0x0020-0x0021 Programmable interrupt controller
OK
0x0040-0x0041 Programmable interrupt controller
OK
0x0040-0x0043 System timer OK
0x00F0-0x00FF System speaker OK
0x0061-0x0061 Motherboard resources OK
0x002E-0x002F Motherboard resources OK
0x00E8-0x00E9 Motherboard resources OK
0x0400-0x040A Motherboard resources OK
0x040C-0x0417 Motherboard resources OK
0x0418-0x0418 Motherboard resources OK
0x0580-0x058D Motherboard resources OK
0x0092-0x0092 Motherboard resources OK
0x0408-0x0408 Motherboard resources OK
0x0419-0x041B Motherboard resources OK
0x041D-0x041F Motherboard resources OK
0x0406-0x0406 Motherboard resources OK
0x00C0-0x00C1 Motherboard resources OK
0x00C6-0x00C8 Motherboard resources OK
0x0C14-0x0C14 Motherboard resources OK
0x0C49-0x0C4A Motherboard resources OK
0x0C50-0x0C51 Motherboard resources OK
0x0C52-0x0C52 Motherboard resources OK
0x0C6C-0x0C6C Motherboard resources OK
0x0C6F-0x0C6F Motherboard resources OK
0x0CD6-0x0CD7 Motherboard resources OK
0x0F50-0x0F58 Motherboard resources OK
0x0374-0x0375 Motherboard resources OK
0xFE00-0xFE20 Motherboard resources OK
0x0220-0x0220 Motherboard resources OK
0x0223-0x0223 Motherboard resources OK
0x0228-0x0228 Motherboard resources OK
0x022E-0x022E Motherboard resources OK
0x0102-0x0105 Motherboard resources OK
0x0107-0x0107 Motherboard resources OK
0x0CA6-0x0CA6 Microsoft ACPI-Compliant Embedded
Controller OK
0x0CA7-0x0CA7 Microsoft ACPI-Compliant Embedded
Controller OK
0x1440-0x144F Standard dual Channel PCI IDE
OK
Controller
0x01F0-0x01F7 Primary IDE Channel OK
0x03F6-0x03F6 Secondary IDE Channel OK
0x0170-0x0177 Secondary IDE Channel
OK
0x0376-0x0376 PCI bus OK
0x1800-0x20FF Adaptec AIC-7899 Ultra160/m PCI
SCSI Card OK
0x2000-0x20FF Adaptec AIC-7899 Ultra160/m PCI
SCSI Card OK
[IRQs]
IRQ Number Device
31 Microsoft ACPI-Compliant System
19 ATI Technologies Inc. 3D RAGE IIC PCI

18 Intel 8255x-based PCI Ethernet Adapter
(10/100)
8 System CMOS/real time clock
4 Communications Port (COM1)
3 Communications Port (COM2)
6 Standard Floppy disk controller
1 Standard 101/102-key or Microsoft Natural
PS/2 keyboard
12 PS/2 Compatible Mouse
13 Numeric data processor
14 Primary IDE Channel
16 Adaptec AIC-7899 Ultra160/m PCI SCSI Card
17 Adaptec AIC-7899 Ultra160/m PCI SCSI Card
21 Intel(R) PRO/100 T Server Adapter #2
22 Intel(R) PRO/100 T Server Adapter #2
[Memory]
Range Device Status
0xA000-0xBFFF PCI bus OK
0xA000-0xBFFF ATI Technologies Inc. 3D RAGE IIC
PCI OK
0xC000-0xC3FF PCI bus OK
0xC400-0xC7FF PCI bus OK
0xC800-0xCBFF PCI bus OK
0xCC00-0xCFFF PCI bus OK
0xD000-0xD3FF PCI bus OK
0xD400-0xD7FF PCI bus OK
0xD800-0xDBFF PCI bus OK
0xDC00-0xDFFF PCI bus OK
0xE000-0xEFFF Motherboard resources OK
0xF000-0xF3FF PCI bus OK
0xF3000000-0xF33FFFFF ATI Technologies Inc. 3D
RAGE IIC PCI OK
0xF9400000-0xFAFFFFFF PCI bus OK
0xFA000000-0xFAFFFFFF ATI Technologies Inc. 3D
RAGE IIC PCI OK
0xF9001000-0xF901FFFF Intel 8255x-based PCI
Ethernet Adapter (10/100)
0xF9100000-0xF91FFFFF Intel 8255x-based PCI
Ethernet Adapter (10/100)
0xCF800-0xDFFF Motherboard resources OK
0xF8000000-0xFB3DFFFF PCI bus OK
0xF8000000-0xFB3DFFFF Adaptec AIC-7899
Ultra160/m PCI SCSI Card
0xFB3E0000-0xFB0FFFFF PCI bus OK
0xFB3E0000-0xFB0FFFFF Adaptec AIC-7899
Ultra160/m PCI SCSI Card
0xF8020000-0xF803FFFF Intel(R) PRO/1000 T
Server Adapter OK
0xF8010000-0xF801FFFF Intel(R) PRO/1000 T
Server Adapter OK
0xF8060000-0xF807FFFF Intel(R) PRO/1000 T
Server Adapter #2 OK
0xF8040000-0xF804FFFF Intel(R) PRO/1000 T
Server Adapter #2 OK
[Components]
[Following are sub-categories of this main category]
[Multimedia]
[Following are sub-categories of this main category]
[Audio Codescs]
Codec Status Manufacturer Description
Version Size
Creation Date

c:\winnt\system32\iac25_32_ax Intel Corporation
Indeo® audio software OK
c:\winnt\system32\IAC25_32.AX 2.05.53 12/7/1999
5:00:00 PM 195.00 KB (199,680 bytes)
c:\winnt\system32\msg723.acm Microsoft Corporation OK
c:\winnt\system32\MSG723.ACM 4.4.3385 8/20/2003
11:27:00 AM 106.77 KB (109,328 bytes)
c:\winnt\system32\thacm.acm Microsoft Corporation OK
c:\winnt\system32\HACM.ACM 4.4.3385 33.27 KB (34,064 bytes) 8/20/2003 11:27:01 AM
c:\winnt\system32\tsrfe32.acm DSP GROUP, INC. OK
c:\winnt\system32\TSOFT32.ACM 1.01 12/7/1999
5:00:00 PM 9.27 KB (9,488 bytes)
c:\winnt\system32\msg711.acm Microsoft Corporation OK
c:\winnt\system32\MSG711.ACM 5.00.2134.1 10.27 KB (10,512 bytes) 12/7/1999
5:00:00 PM
c:\winnt\system32\msgsm32.acm Microsoft Corporation OK
c:\winnt\system32\MSGSM32.ACM 5.00.2134.1 22.27 KB (22,800 bytes) 12/7/1999
5:00:00 PM
c:\winnt\system32\msadp32.acm Microsoft Corporation OK
c:\winnt\system32\MSADP32.ACM 5.00.2134.1 14.77 KB (15,120 bytes) 12/7/1999
5:00:00 PM
c:\winnt\system32\imaadp32.acm Microsoft Corporation OK
c:\winnt\system32\IMAADP32.ACM 5.00.2134.1 16.27 KB (16,656 bytes) 12/7/1999 5:00:00 PM
[Video Codecs]
Codec Manufacturer Description Status File Version Size
c:\winnt\system32\ir30_32.dll Intel Corporation Indeo® Video 5.32.1R01 32.DLL 8.5.10.15.2.55 12/7/1999 737.50 KB (755,200 bytes)
5:00:00 PM
c:\winnt\system32\msh263.drv Microsoft Corporation OK
c:\winnt\system32\MSH263.DRV 4.4.3385 11:26:37 AM 252.27 KB (258,320 bytes) 8/20/2003
c:\winnt\system32\ir32_32.dll Intel(R) Corporation OK
c:\winnt\system32\IR32_32.DLL Not Available 194.50 KB (199,168 bytes) 12/7/1999
5:00:00 PM
c:\winnt\system32\msvidc32.dll Microsoft Corporation OK
c:\winnt\system32\MSVIDC32.DLL 12/7/1999 5:00:00 PM 27.27 KB (27,920 bytes)
c:\winnt\system32\msrle32.dll Microsoft Corporation OK
c:\winnt\system32\MSRLE32.DLL 5.00.2134.1 10.77 KB (11,024 bytes) 12/7/1999
5:00:00 PM
c:\winnt\system32\iccvd.dll Radius Inc. OK
c:\winnt\system32\ICCDLL.DLL

1-10-0-6 108.00 KB (110,592 bytes) 12/7/1999 5:00:00 PM
c:\winnt\system32\msh261.drv Microsoft Corporation OK
c:\winnt\system32\MSH261.DRV 4.4.3385 163.77 KB (167,696 bytes) 8/20/2003
11:27:00 AM
[CD-ROM]
Item Value
Drive D:
Description CD-ROM Drive
Media Loaded False
Name CD-ROM
Manufacturer CD-ROM LTN526S (Standard CD-ROM drives)
Status OK
Transfer Rate Not Available
SCSI Target ID 0
PNP Device ID IDE\CDROM\LTION_CD-ROM.LTN526S_Y50G_58326853D0&0&0.0.0
[Sound Device]
Item Value
Name ATI Technologies Inc. 3D RAGE IIC PCI
PNP Device ID PCI\VEN_1002&DEV_4756&SUBSYS_00000000&REV_7A\3A267A616A&0&10
Adapter Type ATI 3D RAGE IIC PCI (A21), ATI Technologies Inc. compatible
Adapter Description ATI Technologies Inc. 3D RAGE IIC PCI
Adapter RAM 4.00 MB (4,194,304 bytes)
Installed Drivers atiraged.dll
Driver Version 5.00.2174.1
INF File display.inf (atirage section)
Color Planes 1
Color Table Entries 25536
Resolution 1024 x 768 x 60 hertz
Bits/Pixel 16
[Infrared]
Item Value
No Infrared devices
[Input]
[Following are sub-categories of this main category]
[Keyboard]
Item Value
Description Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
PNP Device ID ACPI\PNP0303\4&23FD4C84&0
NumberOfFunctionKeys 12
[Pointing device]
Item Value
Description Microsoft Mouse
PNP Device ID ACPI\PNP0F13\4&23FD4C84&0
Double Click Threshold 6
Handedness Right Handed operation

[Modem]
Item Value
No modems
[Network]
[Following are sub-categories of this main category]
[Adapter]
Item Value
Name [000000000] Intel 8255x-based PCI Ethernet Adapter (10/100)
Adapter Type Ethernet 802.3
Product Name Intel 8255x-based PCI Ethernet Adapter (10/100)
Installed True
PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_12298086&REV_08\3A267A616A&0&18
Last Reset 11/2/2003 7:01:25 PM
Index 0
Service Name E100B
IP Address
IP Subnet
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server 255.255.255.255
DHCP Lease Expires 1/19/2038 12:14:07 AM
DHCP Lease Obtained 8/20/2003 12:08:11 PM
MAC Address 00:D0:87:C8:78:9D
Service Name E100B
IRQ Number 18
I/O Port 0x1400-0x143F
Driver C:\winnt\system32\drivers\ei100bnt5.sys (35776, 4.02.38.0000)
Name [000000001] RAS Async Adapter
Adapter Type RAS Async Adapter
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 11/2/2003 7:01:25 PM
Index 1
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Not Available
Name [000000002] WAN Miniport (L2TP)
Adapter Type WAN Miniport (L2TP)
Product Name WAN Miniport (L2TP)
Installed True
PNP Device ID ROOT\W5_L2TPMINIPORT\0000

Last Reset	11/2/2003 7:01:25 PM
Index 2	
Service Name	Ras12tp
IP Address	Not Available
IP Subnet	Not Available
Default IP Gateway	Not Available
DHCP Enabled	False
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	Not Available
Service Name	Ras12tp
Driver	C:\winnt\system32\drivers\ras12tp.sys (50800, 5.00.2179.1)
Name	[00000003] WAN Miniport (PPTP)
Adapter Type	Wide Area Network (WAN)
Product Name	WAN Miniport (PPTP)
Installed True	
PNP Device ID	ROOT\MS_PPTP\MiniPORT\0000
Last Reset	11/2/2003 7:01:25 PM
Index 3	
Service Name	Pptpminiport
IP Address	Not Available
IP Subnet	Not Available
Default IP Gateway	Not Available
DHCP Enabled	False
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	Not Available
Service Name	Pptpminiport
Driver	C:\winnt\system32\drivers\raspttp.sys (47856, 5.00.2160.1)
Name	[00000004] Direct Parallel
Adapter Type	Not Available
Product Name	Direct Parallel
Installed True	
PNP Device ID	ROOT\MS_PPTP\MiniPORT\0000
Last Reset	11/2/2003 7:01:25 PM
Index 4	
Service Name	Raspti
IP Address	Not Available
IP Subnet	Not Available
Default IP Gateway	Not Available
DHCP Enabled	False
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	Not Available
Service Name	Raspti
Driver	C:\winnt\system32\drivers\raspti.sys (16880, 5.00.2146.1)
Name	[00000005] WAN Miniport (IP)
Adapter Type	Not Available
Product Name	WAN Miniport (IP)
Installed True	
PNP Device ID	ROOT\MS_NDISWAN\0000
Last Reset	11/2/2003 7:01:25 PM
Index 5	
Service Name	Ndiswan
IP Address	Not Available
IP Subnet	Not Available
Default IP Gateway	Not Available
DHCP Enabled	False
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	Not Available
Service Name	Ndiswan

Driver	C:\winnt\system32\drivers\ndiswan.sys (90096, 5.00.2195.2779)
Name	[00000006] Intel(R) PRO/1000 T Server Adapter
Adapter Type	Ethernet 802.3
Product Name	Intel(R) PRO/1000 T Server Adapter
Installed True	
PNP Device ID	PCI\VEN_8086&DEV_1004&SUBSYS_10048086&REV_02\3813C080C560&60
Last Reset	11/2/2003 7:01:25 PM
Index 6	
Service Name	E1000
IP Address	192.168.10.1
IP Subnet	192.168.10.1
Default IP Gateway	Not Available
DHCP Enabled	False
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	00:07:E9:0C:BD:53
Service Name	E1000
IRQ Number	21
Driver	C:\winnt\system32\drivers\e1000nt5.sys (51376, 2.84.284.0)
Name	[00000007] Intel(R) PRO/1000 T Server Adapter
Adapter Type	Ethernet 802.3
Product Name	Intel(R) PRO/1000 T Server Adapter
Installed True	
PNP Device ID	PCI\VEN_8086&DEV_1004&SUBSYS_10048086&REV_02\3813C080C560&60
Last Reset	11/2/2003 7:01:25 PM
Index 7	
Service Name	E1000
IP Address	192.168.20.1
IP Subnet	192.168.20.1
Default IP Gateway	Not Available
DHCP Enabled	False
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	00:07:E9:10:7F:25
Service Name	E1000
IRQ Number	22
Driver	C:\winnt\system32\drivers\e1000nt5.sys (51376, 2.84.284.0)
[Protocol]	
Item	Value
Name	MSAFD Tcpip [TCP/IP]
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	16 bytes
MaximumMessageSize	0 bytes
MessageOriented	False
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	True
SupportsGracefulClosing	True
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False
Name	MSAFD Tcpip [UDP/IP]

ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	16 bytes
MaximumMessageSize	65467 bytes
MessageOriented	True
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	True
SupportsMulticasting	True
Name	RSVP UDP Service Provider
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	16 bytes
MaximumMessageSize	65467 bytes
MessageOriented	True
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	True
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	True
Name	RSVP TCP Service Provider
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	16 bytes
MaximumMessageSize	0 bytes
MessageOriented	False
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	True
SupportsExpeditedData	True
SupportsGracefulClosing	True
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False
Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{F39A-44A1-B998-354ACE77131F}] SEQUENTIAL_PACKET 4
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{78B9880F-F39A-44A1-899B-354ACE77131F}] DATAGRAM 4			
	ConnectionlessService	True		
	GuaranteedDelivery	False		
	GuaranteedSequencing	False		
	MaximumAddressSize	20 bytes		
	MaximumMessageSize	64000 bytes		
	MessageOriented	True		
	MinimumAddressSize	20 bytes		
	PseudoStreamOriented	False		
	SupportsBroadcasting	True		
	SupportsConnectData	False		
	SupportsDisConnectData	False		
	SupportsEncryption	False		
	SupportsGracefulClosing	False		
	SupportsGuaranteedBandwidth	False		
	SupportsMulticasting	False		
Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{34940D743-366C-43E0-A517-869314A1987E}] SEQPACKET 3			
	ConnectionlessService	True		
	GuaranteedDelivery	True		
	GuaranteedSequencing	True		
	MaximumAddressSize	20 bytes		
	MaximumMessageSize	64000 bytes		
	MessageOriented	True		
	MinimumAddressSize	20 bytes		
	PseudoStreamOriented	False		
	SupportsBroadcasting	False		
	SupportsConnectData	False		
	SupportsDisConnectData	False		
	SupportsEncryption	False		
	SupportsGracefulClosing	False		
	SupportsGuaranteedBandwidth	False		
	SupportsMulticasting	False		
Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{34940D743-366C-43E0-A517-869314A1987E}] DATAGRAM 3			
	ConnectionlessService	True		
	GuaranteedDelivery	False		
	GuaranteedSequencing	False		
	MaximumAddressSize	20 bytes		
	MaximumMessageSize	64000 bytes		
	MessageOriented	True		
	MinimumAddressSize	20 bytes		
	PseudoStreamOriented	False		
	SupportsBroadcasting	False		
	SupportsConnectData	False		
	SupportsDisConnectData	False		
	SupportsEncryption	False		
	SupportsGracefulClosing	False		
	SupportsGuaranteedBandwidth	False		
	SupportsMulticasting	False		
Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{A550D7FE-F320-4FE1-A14B-E5FE08040C70}] SEQPACKET 0			
	ConnectionlessService	True		
	GuaranteedDelivery	True		
	GuaranteedSequencing	True		
	MaximumAddressSize	20 bytes		
	MaximumMessageSize	64000 bytes		
	MessageOriented	True		
	MinimumAddressSize	20 bytes		
	PseudoStreamOriented	False		
	SupportsBroadcasting	False		
	SupportsConnectData	False		
	SupportsDisConnectData	False		
	SupportsEncryption	False		
	SupportsGracefulClosing	False		
	SupportsGuaranteedBandwidth	False		
	SupportsMulticasting	False		

SupportsGracefulClosing	False		
	SupportsGuaranteedBandwidth	False	
	SupportsMulticasting	False	
Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{A550D7FE-F320-4FE1-A14B-E5FE08040C70}] DATAGRAM 0		
	ConnectionlessService	True	
	GuaranteedDelivery	False	
	GuaranteedSequencing	False	
	MaximumAddressSize	20 bytes	
	MaximumMessageSize	64000 bytes	
	MessageOriented	True	
	MinimumAddressSize	20 bytes	
	PseudoStreamOriented	False	
	SupportsBroadcasting	True	
	SupportsConnectData	False	
	SupportsDisConnectData	False	
	SupportsEncryption	False	
	SupportsGracefulClosing	False	
	SupportsGuaranteedBandwidth	False	
	SupportsMulticasting	False	
Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{CED98EC8-90C4-4739-AB9A-DE7CED87FELF}] SEQPACKET 1		
	ConnectionlessService	False	
	GuaranteedDelivery	True	
	GuaranteedSequencing	True	
	MaximumAddressSize	20 bytes	
	MaximumMessageSize	64000 bytes	
	MessageOriented	True	
	PseudoStreamOriented	False	
	SupportsBroadcasting	False	
	SupportsConnectData	False	
	SupportsDisConnectData	False	
	SupportsEncryption	False	
	SupportsGracefulClosing	False	
	SupportsGuaranteedBandwidth	False	
	SupportsMulticasting	False	
Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{CED98EC8-90C4-4739-AB9A-DE7CED87FELF}] DATAGRAM 1		
	ConnectionlessService	True	
	GuaranteedDelivery	False	
	GuaranteedSequencing	False	
	MaximumAddressSize	20 bytes	
	MaximumMessageSize	64000 bytes	
	MessageOriented	True	
	PseudoStreamOriented	False	
	SupportsBroadcasting	True	
	SupportsConnectData	False	
	SupportsDisConnectData	False	
	SupportsEncryption	False	
	SupportsGracefulClosing	False	
	SupportsGuaranteedBandwidth	False	
	SupportsMulticasting	False	
Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{87A8369C-4998-4A9D-9D1E-5ED8D279ACEE}] SEQPACKET 2		
	ConnectionlessService	False	
	GuaranteedDelivery	True	
	GuaranteedSequencing	True	
	MaximumAddressSize	20 bytes	
	MaximumMessageSize	64000 bytes	
	MessageOriented	True	
	MinimumAddressSize	20 bytes	
	PseudoStreamOriented	False	
	SupportsBroadcasting	True	
	SupportsConnectData	False	
	SupportsDisConnectData	False	
	SupportsEncryption	False	
	SupportsGracefulClosing	False	
	SupportsGuaranteedBandwidth	False	
	SupportsMulticasting	False	

SupportsConnectData	False
SupportsDisConnectData	False
SupportsEncryption	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False
Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{87A8369C-4998-4A9D-9D1E-5ED8D279ACEE}] DATAGRAM 2	
ConnectionlessService	True
GuaranteedDelivery	False
GuaranteedSequencing	False
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisConnectData	False
SupportsEncryption	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False
[winsock]	
Item	Value
File	C:\Innt\system32\winsock.dll
Version	3.10
Size	2.80 kb (2,864 bytes)
File	C:\Innt\system32\wsock32.dll
Version	5.00.2195.2871
Size	21.27 KB (21,776 bytes)
[Ports]	
[Following are sub-categories of this main category]	
[Serial]	
Item	Value
Name	COM1
PNP Device ID	ACPI\PNP0501\1
Maximum Input Buffer Size	0
Maximum Output Buffer Size	False
Settable Baud Rate	True
Settable Data Bits	True
Settable Flow Control	True
Settable Parity	True
Settable Stop Bits	True
Settable RLSO	True
Supports RLSO	True
Supports 16 Bit Mode	False
Supports Special Characters	False
Baud Rate	9600
Bits/Byte	8
Parity	None
Busy	0
Abort Read/write on Error	0
Binary Mode Enabled	-1
Continue xmit on Xoff	0
CTS Outflow Control	0
Discard NULL Bytes	0

DSR outFlow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled 0
Event Character 0
Parity Check Enabled 0
RTS Flow Control Type Enable
XOFF Character 19
XOFFxmit Threshold 512
XON Character 17
XONxmit Threshold 2048
XONxmit InFlow Control 0
XONxmit OutFlow Control 0
IRQ Number 4
I/O Port 0x03F8-0x03FF
Driver C:\winnt\system32\drivers\serial.sys (62416,
5.00.2195.2780)

Name COM2
Status OK
PNP Device ID ACPI\PNP0501\2
Maximum Input Buffer Size 0
Maximum Output Buffer Size False
Settable baud Rate True
Settable data Bits True
Settable Flow Control True
Settable Parity True
Settable Parity Check True
Settable Stop Bits True
Settable Parity Check True
Supports RS485 True
Supports 16 Bit Mode True
Supports Special Characters False
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy 0
Abort Read/Write on Error 0
Binary Mode Enabled -1
Continue Xmit on Xoff 0
CTS outFlow Control 0
Discard NULL Bytes 0
DSR outFlow Control 0
DSR Sensitivity 0
DSR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled 0
Event Character 0
Parity Check Enabled 0
RTS Flow Control Type Enable
XOFF Character 19
XOFFxmit Threshold 512
XON Character 17
XONxmit Threshold 2048
XONxmit InFlow Control 0
XONxmit OutFlow Control 0
IRQ Number 3
I/O Port 0x02F8-0x02FF
Driver C:\winnt\system32\drivers\serial.sys (62416,
5.00.2195.2780)

[Parallel1]

Item Value
Name LPT1
PNP Device ID ACPI\PNP0400\4&23FD4C84&0

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item Value
Drive A: 3 1/2 Inch Floppy Drive
Description
Drive C: Local Fixed Disk
Compressed False
File System NTFS
Size 17.08 GB (18,342,338,560 bytes)
Free Space 12.35 GB (13,257,580,544 bytes)
Volume Name BCLC436D
Partition #0
Partition Size 17.08 GB (18,342,342,144 bytes)
Starting Offset 32256 bytes
Drive Description Disk drive
Drive Manufacturer (Standard disk drives)
Drive Model SEAGATE ST318406LC SCSI Disk Device
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIbus 0
Drive ScsiLogicalUnit 0
Drive ScsiTargetId 2
Drive ScsiTargetLun 0
Drive SectorsPerTrack 63
Drive Size 18350599680 bytes
Drive TotalCylinders 35841015
Drive TotalSectors 568905
Drive TotalTracks 255
Drive TracksPerCylinder

[SCSI]

Item Value
Name Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Caption Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Driver adpu160m
PNP Device ID PCI\VEN_9005&DEV_00CF&SUBSYS_00CF8086&REV_01\3&13C0B0C5&0&20
Device ID PCI\VEN_9005&DEV_00CF&SUBSYS_00CF8086&REV_01\3&13C0B0C5&0&20
Device Map
Index Not Available
Max Number Controlled Not Available
IRQ Number 16
I/O Port 0x1800-0x20FF
Driver C:\winnt\system32\drivers\adpu160m.sys (64432, v3.10a)
Name Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Caption Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Status OK
PNP Device ID PCI\VEN_9005&DEV_00CF&SUBSYS_00CF8086&REV_01\3&13C0B0C5&0&21
Device ID PCI\VEN_9005&DEV_00CF&SUBSYS_00CF8086&REV_01\3&13C0B0C5&0&21

Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 17
I/O Port 0x2000-0x20FF
Driver C:\winnt\system32\drivers\adpu160m.sys (64432, v3.10a)

[Printing]

Name Port Name Server Name
No Printing Information
[Problem Devices]
Device PNP Device ID Error Code
No Problem Devices

[USB]

PNP Device ID
No USB Devices
[Software Environment]
[Following are sub-categories of this main category]
[Drivers]

Name Description Start Mode File Type
Status Error Control Accept Pause
abiosdsk Abiosdsk Not Available kernel Driver
False Disabled Stopped OK
Ignore False Stopped
abp480n5 Not Available kernel Driver
False Disabled Stopped OK
Normal False
acpi Microsoft ACPI Driver
C:\winnt\system32\drivers\acpi.sys
kernel Driver True Boot
Running OK Normal False
acpiec Microsoft Embedded Controller Driver
C:\winnt\system32\drivers\acpiec.sys
kernel Driver True Boot
Normal OK Normal False
adpu160m
C:\winnt\system32\drivers\adpu160m.sys
kernel Driver True Boot
Running OK Normal False
afd AFD Networking Support Environment
C:\winnt\system32\drivers\afd.sys
kernel Driver True Auto
Running OK Normal False
ahal54x
Normal Not Available kernel Driver
False Disabled Stopped OK
aic116x
Normal Not Available kernel Driver
False Disabled Stopped OK
aic78u2
Normal Not Available kernel Driver
False Disabled Stopped OK
aic78xx
Normal Not Available kernel Driver
False Disabled Stopped OK
amioht
Normal Not Available kernel Driver
False Disabled Stopped OK

November, 2003

tdspcx c:\winnt\system32\drivers\tdspcx.sys
kernel driver False Manual
Stopped OK Ignore False
tdtcp c:\winnt\system32\drivers\tdtcp.sys
kernel driver False Manual
Stopped OK Ignore False
termdd c:\winnt\system32\drivers\termdd.sys
terminal device driver False Disabled
kernel driver False
tga Not Available Normal Kernel Driver
tga System Stopped OK
False
Ignore
udfs c:\winnt\system32\drivers\udfs.sys
filesystem driver False Disabled
Stopped OK Driver Name False
ultra66 Not Available Normal Kernel Driver
False Disabled Stopped OK
Normal False
update Microcode Update Driver
kernel driver True Manual
Running OK Normal False
vgasave c:\winnt\system32\drivers\vga.sys
kernel driver True System
Running OK Ignore False
wanarp Remote Access IP ARP Driver
c:\winnt\system32\drivers\wanarp.sys
kernel driver True Manual
Running OK Normal False
wdica Not Available Normal Kernel Driver
False Manual Stopped OK
Ignore False

[Environment Variables]
Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
Os2LibPath %SystemRoot%\system32\os2\dll;
<SYSTEM>
Path %SystemRoot%\system32;%SystemRoot%\system32\bin;C:\Program Files\Microsoft SQL Server\80\Tools\BINN
windir %SystemRoot%
OS %SystemRoot%
PROCESSOR_ARCHITECTURE X86 <SYSTEM>
PROCESSOR_IDENTIFIER X86 Family 6 Model 1 8
PROCESSOR_REVISION 0806 <SYSTEM>
NUMBER_OF_PROCESSORS 2 <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 <SYSTEM>
TMP CLIO1\Administrator
CLIO1\Administrator

[Jobs]
[Following are sub-categories of this main category]
[Print]
document Size Owner Notify Status Time
Submitted Start Time Until Time
Elapsed Time Pages Printed Job
ID Priority Parameters Driver Name

Type	Name	Print Processor	Host	Print Queue	Data
No print jobs					
[Network Connections]					
Local Name	Status	Remote Name	Type		
No network connections information					
[Running Tasks]					
Name	Path	Process ID	Priority	Min	
Working Set	Version	Size	Working Set	Start Time	
system idle	process	Not Available	File Date	0	0
Available	Not Available	Unknown	Not Available		Not
system	Not Available	Unknown	Unknown	8	0
1413120	Not Available	Unknown	Unknown	8	Unknown
smss.exe	c:\winnt\system32\smss.exe	164	11		
204800	1413120	11/2/2003 10:01:45 PM			
5:00:2195.2901	44.27 KB (45,328 bytes)				
csrss.exe	Not Available	188	13		Not
Available	Not Available	Unknown	Unknown	10:01:49 PM	Not
winlogon.exe	c:\winnt\system32\winlogon.exe	1413120			
184	13	204800			
11/2/2003 10:01:50 PM					
5:00:00 PM	173.77 KB (177,936 bytes)				
services.exe	c:\winnt\system32\services.exe	248			
236	9	204800			
11/2/2003 10:01:52 PM					
86.77 KB (88,848 bytes)					
5:00:00 PM					
lsass.exe	c:\winnt\system32\lsass.exe	248			
204800	1413120	11/2/2003 10:01:52 PM			
5:00:2195.2964	32.77 KB (33,552 bytes)				
12/7/1999 5:00:00 PM					
svchost.exe	c:\winnt\system32\svchost.exe	428			
204800	1413120	11/2/2003 10:01:55 PM			
5:00:2134.1					
10:01:55 PM	5:00:00 PM				
spoolsv.exe	c:\winnt\system32\spoolsv.exe	460			
8	204800	1413120			
10:01:56 PM	5:00:2161.1				
(44,816 bytes)	8/20/2003 8:10:49 AM				
msdtc.exe	c:\winnt\system32\msdtc.exe	488			
204800	1413120	11/2/2003 10:01:56 PM			
1999.9.3421.3	6.77 KB (6,928 bytes)				
8/20/2003 8:23:06 AM					
svchost.exe	c:\winnt\system32\svchost.exe	620			
8	204800	1413120			
10:01:57 PM	5:00:2134.1				
11ssrv.exe	5:00:00 PM				
12/7/1999 5:00:00 PM					
(117,008 bytes)	5:00:2195.2649				
5/4/2001 12:05:02 PM					
files\symantec\ghost\nsserver.exe	c:\program	684			
204800	1413120	11/2/2003 10:01:58 PM			
7:0:0.245 1016.26 KB (1,040,650 bytes)					
3/22/2001 6:02:16 PM					

regsvcs.exe	8	c:\winnt\system32\regsvcs.exe	768		
10:02:01 PM	5:00:2195.2104	8/20/2003 3:57:08 PM	65.27 KB		
mstask.exe	8	c:\winnt\system32\mstask.exe	784		
10:02:02 PM	4:71.2195.1	11/2/2003 11:27 KB			
winmgmt.exe	820	c:\winnt\system32\wbem\winmgmt.exe			
11/2/2003 10:02:02 PM	8	204800	1413120		
192.08 KB (196,685 bytes)					
mdsvcs.exe	8	c:\winnt\system32\mdsvcs.exe	876		
5:00:0720 13.77 KB (14,096 bytes)					
dfssvc.exe	8	c:\winnt\system32\dfssvc.exe	324		
10:02:07 PM	5:00:2195.2841	88.27 KB			
(90,384 bytes)	8/20/2003 3:56:51 PM				
files\symantec\ghost\bin\dserv.exe	c:\program	292			
204800	1413120	11/2/2003 10:02:40 PM			
7:0:0.245 44.00 KB (45,056 bytes)					
rteng6.exe	3/22/2001 6:01:34 PM				
Files\symantec\ghost\bin\rteng6.exe	c:\program	724			
204800	1413120	11/2/2003 10:02:40 PM			
8/6/1999 3:16:02 PM	39.00 KB (39,936 bytes)				
svchost.exe	8	c:\winnt\system32\svchost.exe	1132		
10:02:40 PM	5:00:2134.1				
explorer.exe	8	c:\winnt\explorer.exe	1312		
10:05:21 PM	5:00:3315.2846	237.27 KB			
(242,960 bytes)	8/20/2003 3:57:14 PM				
dllhost.exe	Not Available	1380			
Not Available	11/2/2003 10:31:36 PM				
Unknown	Unknown				
mmc.exe	8	c:\winnt\system32\mmc.exe	1652		
204800	1413120	11/3/2003 4:33:08 PM			
8:00:2195.2301.57	589.27 KB (603,408 bytes)				
rsrvp.exe	8	c:\winnt\system32\rsrvp.exe	4556		
204800	1413120	11/3/2003 4:33:53 PM			
5:00:2167.1	172.77 KB (176,912 bytes)				
[Loaded Modules]					
Name	Version	Size	File	Date	Manufacturer
Path					
traffic.dll	5.00.2139.1	30.77 KB			
(31,504 bytes)	12/7/1999 5:00:00 PM				
rsrvp.exe	5.00.2167.1	172.77 KB (176,912 bytes)			
12/7/1999 5:00:00 PM					
Corporation	c:\winnt\system32\rsrvp.exe				
Microsoft Corporation	8/20/2003 3:57:17 PM				
440.08 KB					
mlang.dll	5.00.3103.1000	510.77 KB (523,024 bytes)			
8/20/2003 3:56:57 PM					
Corporation	c:\winnt\system32\mlang.dll				
cabinet.dll	5.00.2147.1	54.77 KB			
(56,080 bytes)	12/7/1999 5:00:00 PM				

Microsoft Corporation
C:\winnt\system32\cabinet.dll 312.27 KB
msinfo32.dll 5.00.2177-1 312.27 KB
(319,760 bytes) 8/20/2003 11:26:57 AM C:\program
Microsoft Corporation
files\common files\microsoft shared\msinfo32.dll
mmcmdmgr.dll 5.00.2178-1 815.27 KB
(834,832 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\mmc.dmgrr.dll
mmc.exe 5.00.2195.2301 589.27 KB (603,408 bytes)
8/20/2003 3:56:57 PM Microsoft
Corporation C:\winnt\system32\mmc.exe
5.00.2195.2301 589.27 KB (603,408 bytes)
(332,288 bytes) 8/20/2003 3:57:09 PM
Microsoft Corporation
C:\winnt\system32\shdoclc.dll
urlmon.dll 5.00.3315.1000 441.27 KB
(451,856 bytes) 8/20/2003 3:57:12 PM
Microsoft Corporation
C:\winnt\system32\urlmon.dll
browse1c.dll 5.00.3315.2846 34.50 KB
(35,328 bytes) 8/20/2003 3:56:48 PM
Microsoft Corporation
C:\winnt\system32\browse1c.dll
wininet.dll 5.00.3315.1000 456.77 KB
(467,728 bytes) 8/20/2003 3:57:12 PM
Microsoft Corporation
C:\winnt\system32\wininet.dll
linkinfo.dll 5.00.3315.1000 15.77 KB
(16,144 bytes) 8/20/2003 3:57:12 PM
Microsoft Corporation
C:\winnt\system32\linkinfo.dll
powerprof.dll 5.00.3103.1000 13.27 KB
(13,584 bytes) 8/20/2003 3:57:07 PM
Microsoft Corporation
C:\winnt\system32\powerprof.dll
batmeter.dll 5.00.3103.1000 20.27 KB
(20,752 bytes) 8/20/2003 3:56:48 PM
Microsoft Corporation
C:\winnt\system32\batmeter.dll
stobjact.dll 5.00.2195.2780 79.27 KB
(81,168 bytes) 8/20/2003 3:57:11 PM
Microsoft Corporation
C:\winnt\system32\stobjact.dll
msi.dll 2.0.2600.0 1.90 MB (1,991,168 bytes)
8/20/2003 3:57:00 PM Microsoft
Corporation C:\winnt\system32\msi.dll
webcheck.dll 5.00.3315.1000 231.77 KB
(237,808 bytes) 8/20/2003 3:57:12 PM
Microsoft Corporation
C:\winnt\system32\webcheck.dll
mydocs.dll 5.00.2920.0000 55.77 KB
(57,104 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\mydocs.dll
ntshrui.dll 5.00.2134.1 46.77 KB
(47,888 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\ntshrui.dll
browseui.dll 5.00.3315.2846 788.77 KB
(807,696 bytes) 8/20/2003 3:56:48 PM
Microsoft Corporation
C:\winnt\system32\browseui.dll
shdocvw.dll 5.00.3315.2829 1.05 MB
(1,104,144 bytes) 8/20/2003 3:57:10 PM
Microsoft Corporation
C:\winnt\system32\shdocvw.dll
explorer.exe 5.00.3315.2846 237.27 KB
(242,960 bytes) 8/20/2003 3:57:14 PM
Microsoft Corporation
C:\winnt\explorer.exe

h323.tsp 5.00.2195.2283 248.77 KB (254,736 bytes)
8/20/2003 3:56:53 PM Microsoft
Corporation C:\winnt\system32\h323.tsp
ipconf.tsp 5.00.2143.1 10.77 KB
(11,024 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\ipconf.tsp
ndtpstsp.tsp 5.00.2143.1 38.27 KB
(39,184 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\ndtpstsp.tsp
knddsp.tsp 5.00.2150.1 17.77 KB
(18,192 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\knddsp.tsp
uniplat.dll 5.00.2151.0000 13.77 KB
(14,096 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\uniplat.dll
unimdm.tsp 5.00.2175.1 196.77 KB
(201,488 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\unimdm.tsp
tapisrv.dll 5.00.2195.2955 169.27 KB
(173,328 bytes) 8/20/2003 3:57:11 PM
Microsoft Corporation
C:\winnt\system32\tapisrv.dll
dbctr6.dll 6.0.3.2747 30.00 KB
(30,720 bytes) 8/6/1999 3:13:42 PM Sybase, Inc.
C:\program files\syntec\ghost\bin\dbctr6.dll
dbctr6.dll 6.0.3.2747 2.16 MB
(2,269,184 bytes) 8/6/1999 3:16:00 PM Sybase, Inc.
C:\program files\syntec\ghost\bin\dserv6.dll
rteng6.exe 6.0.3.2747 39.00 KB
(39,936 bytes) 8/6/1999 3:16:02 PM Sybase, Inc.
C:\program files\syntec\ghost\bin\rteng6.exe
dbserv.exe 7.0.0.245 44.00 KB (45,056 bytes)
3/22/2001 6:01:34 PM Symantec New
Zealand Limited C:\program
files\syntec\ghost\bin\dserv.exe
dfssvc.exe 5.00.2195.2841 88.27 KB
(90,384 bytes) 8/20/2003 3:56:51 PM
Microsoft Corporation
C:\winnt\system32\dfssvc.exe
mqdsccli.dll 8/20/2003 3:56:58 PM (75,536 bytes)
Corporation C:\winnt\system32\mqdsccli.dll
mqlogmgr.dll 1999.8.3413.7 85.27 KB
(87,312 bytes) 8/20/2003 8:23:16 AM
Microsoft Corporation
C:\winnt\system32\mqlogmgr.dll
mqutil.dll 5.00.0720.105.27 KB (107,792 bytes)
8/20/2003 3:56:59 PM Microsoft
Corporation C:\winnt\system32\mqutil.dll
security.dll 5.00.2154.1 5.77 KB (5,904
bytes) 12/7/1999 5:00:00 PM Microsoft
Corporation C:\winnt\system32\security.dll
musec.dll 5.00.0720 68.77 KB (70,416 bytes)
8/20/2003 3:56:58 PM Microsoft
Corporation C:\winnt\system32\musec.dll
mqmqm.dll 00.0721 404.58 KB (413,968 bytes)
8/20/2003 3:56:58 PM Microsoft
Corporation C:\winnt\system32\mqmqm.dll
mqsvc.exe 5.00.0720 13.77 KB (14,096 bytes)
8/20/2003 3:56:58 PM Microsoft
Corporation C:\winnt\system32\mqsvc.exe
netui1.dll 5.00.2134.1 210.27 KB
(215,312 bytes) 12/7/1999 5:00:00 PM

Microsoft Corporation
C:\winnt\system32\netui1.dll
netui0.dll 5.00.2134.1 70.27 KB
(71,952 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\netui0.dll
ntlanman.dll 5.00.2157.1 35.27 KB
(36,112 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\ntlanman.dll
wshnetbs.dll 5.00.2134.1 7.77 KB (7,952
bytes) 12/7/1999 5:00:00 PM Microsoft
Corporation C:\winnt\system32\wshnetbs.dll
provnet.dll 5.00.1085.0000 68.07 KB
(69,108 bytes) 8/20/2003 11:26:49 AM
Microsoft Corporation
C:\winnt\system32\provnet.dll
ntevt.dll 1.50.1085.0000 192.06 KB (196,669 bytes)
12/7/1999 5:00:00 PM Microsoft
Corporation C:\winnt\system32\wbem\ntevt.dll
perfos.dll 5.00.2155.1 21.27 KB
(21,776 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\perfos.dll
psapi.dll 5.00.2134.1 28.27 KB (28,944 bytes)
12/7/1999 5:00:00 PM Microsoft
Corporation C:\winnt\system32\psapi.dll
framedyn.dll 1.50.1085.0000 164.05 KB
(167,992 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
C:\winnt\system32\framedyn.dll
cimwin32.dll 5.00.1085.0038 1.02 MB
(1,073,232 bytes) 8/20/2003 3:57:16 PM
Microsoft Corporation
C:\winnt\system32\wbem\cimwin32.dll
wbemsvc.dll 1.50.1085.0007 40.07 KB
(41,036 bytes) 8/20/2003 3:57:17 PM
Microsoft Corporation
C:\winnt\system32\wbem\wbemsvc.dll
wbemess.dll 1.50.1085.0039 364.07 KB
(372,804 bytes) 8/20/2003 3:57:17 PM
Microsoft Corporation
C:\winnt\system32\wbem\wbemess.dll
fastprox.dll 1.50.1085.0037 144.08 KB
(147,536 bytes) 8/20/2003 3:57:16 PM
Microsoft Corporation
C:\winnt\system32\fastprox.dll
wbemcore.dll 8/20/2003 3:57:17 PM
(643,140 bytes) 8/20/2003 3:57:17 PM
Microsoft Corporation
C:\winnt\system32\wbem\wbemcore.dll
wbemcomm.dll 1.50.1085.0021 692.07 KB
(708,675 bytes) 8/20/2003 3:57:17 PM
Microsoft Corporation
C:\winnt\system32\wbem\wbemcomm.dll
winmgmt.exe 5.00.1085.0029 192.08 KB
(196,685 bytes) 8/20/2003 3:57:17 PM
Microsoft Corporation
C:\winnt\system32\wbem\winmgmt.exe
msidle.dll 12/7/1999 5:00:00 PM 6.27 KB (6,416
bytes) 5.00.2920.0000 Microsoft
Corporation C:\winnt\system32\msidle.dll
nttask.exe 8/21.2195.1 115.27 KB
(116,032 bytes) 8/20/2003 3:57:02 PM
Microsoft Corporation
C:\winnt\system32\nttask.exe
regsvc.exe 5.00.2195.2104 65.27 KB
(66,832 bytes) 8/20/2003 3:57:08 PM
Microsoft Corporation
C:\winnt\system32\regsvc.exe
odbc32.dll 3.525.1022.0 ((webdata).030220-
1508) 100.00 KB (102,400 bytes) 8/20/2003

4:55:50 PM Microsoft Corporation
db1gen6.dll c:\winnt\system32\odbccp32.dll 306.50 KB
(313,856 bytes) 8/6/1999 3:03:02 PM Sybase, Inc.
c:\program
files\symantec\ghost\bin\db1gen6.dll 414.50 KB
dbodbc6.dll 6.0.3.2747
(424,448 bytes) 8/6/1999 3:17:44 PM Sybase, Inc.
c:\program
files\symantec\ghost\bin\dbodbc6.dll
ntnarta.dll 5.00.2195.2862 98.77 KB
(101,136 bytes) 8/20/2003 3:57:05 PM
Microsoft Corporation
c:\winnt\system32\ntnarta.dll
odbcint.dll 92.00 KB (94,208 bytes) (Medadata) 030220-
1508 8/20/2003 3:57:05 PM
4:55:50 PM Microsoft Corporation
c:\winnt\system32\odbcint.dll 17.27 KB
snmpapi.dll 5.00.2134.1 12/7/1999 5:00:00 PM
(17,680 bytes) Microsoft Corporation
thread.dll 3/22/2001 6:01:22 PM (20,480 bytes)
c:\winnt\system32\snmpapi.dll Symantec New
Zealand Limited c:\program
files\symantec\ghost\thread.dll
comdlg32.dll 5.00.3103.1000 236.77 KB
(242,448 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\comdlg32.dll
odbc32.dll 216.00 KB (221,184 bytes) (Medadata) 030220-
1508 8/20/2003 3:57:05 PM
4:55:50 PM Microsoft Corporation
c:\winnt\system32\odbc32.dll
ngserver.exe 7.0.0.245 1016.26 KB (1,040,650
bytes) 3/22/2001 6:02:16 PM Symantec New
Zealand Limited c:\program
files\symantec\ghost\ngserver.exe
tlsrcp.dll 5.00.2149.1 45.77 KB
(46,864 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\tlsrcp.dll
tlssrv.exe 5.00.2195.2649 114.27 KB
(117,008 bytes) 5/4/2001 12:05:02 PM
Microsoft Corporation
c:\winnt\system32\tlssrv.exe
wmi.dll 6.0.2.7 KB (6,416 bytes)
12/7/1999 3:00:00 PM Microsoft
Corporation c:\winnt\system32\wmi.dll
netshell.dll 5.00.2195.2779 457.27 KB
(468,240 bytes) 8/20/2003 3:57:04 PM
Microsoft Corporation
c:\winnt\system32\netshell.dll
netman.dll 5.00.2195.2779 89.27 KB
(91,408 bytes) 8/20/2003 3:57:04 PM
Microsoft Corporation
c:\winnt\system32\netman.dll
ntnsdba.dll 5.00.2195.2779 167.27 KB
(171,280 bytes) 8/20/2003 3:57:05 PM
Microsoft Corporation
c:\winnt\system32\ntnsdba.dll
ipbootp.dll 5.00.2168.1 33.77 KB
(34,376 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\ipbootp.dll
rastls.dll 5.00.2195.2671 47.77 KB
(48,912 bytes) 8/20/2003 3:57:08 PM
Microsoft Corporation
c:\winnt\system32\rastls.dll
raschap.dll 5.00.2195.2671 34.77 KB
(35,600 bytes) 8/20/2003 3:57:08 PM

Microsoft Corporation
rasppp.dll c:\winnt\system32\raschap.dll
(197,392 bytes) 5.00.2195.2671 192.77 KB
8/20/2003 3:57:08 PM
Microsoft Corporation
c:\winnt\system32\rasppp.dll
rastapi.dll 5.00.2195.2671 52.77 KB
(54,032 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\rastapi.dll
rasdlg.dll 5.00.2195.2671 514.27 KB
(526,608 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\rasdlg.dll
netcfgx.dll 5.00.2195.2729 534.77 KB
(547,600 bytes) 8/20/2003 3:57:04 PM
Microsoft Corporation
c:\winnt\system32\netcfgx.dll
rasmans.dll 5.00.2195.2728 147.27 KB
(150,800 bytes) 8/20/2003 3:57:08 PM
Microsoft Corporation
c:\winnt\system32\rasmans.dll
sens.dll 5.00.2163.1 36.77 KB (37,648 bytes)
12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\sens.dll
ntmssvc.dll 5.00.2195.2779 391.27 KB
(400,656 bytes) 8/20/2003 3:57:05 PM
Microsoft Corporation
c:\winnt\system32\ntmssvc.dll
es.dll 5.00.2134.1 222.27 KB (227,600 bytes)
8/20/2003 3:56:53 PM
Microsoft Corporation
c:\winnt\system32\es.dll
mxocj.dll 2000.2.3471.1 101.77 KB
(104,208 bytes) 8/20/2003 3:57:04 PM
Microsoft Corporation
c:\winnt\system32\mxocj.dll
resutils.dll 5.00.2195.2787 39.77 KB
(40,720 bytes) 8/20/2003 3:57:08 PM
Microsoft Corporation
c:\winnt\system32\resutils.dll
clusapi.dll 5.00.2195.2104 54.27 KB
(55,568 bytes) 8/20/2003 3:56:49 PM
Microsoft Corporation
c:\winnt\system32\clusapi.dll
msvcps0.dll 12/7/1999 5.00.7051 552.50 KB (565,760 bytes)
Corporation c:\winnt\system32\msvcps0.dll
rgenp.dll 1999.9.3421.3 17.27 KB
(17,680 bytes) 8/20/2003 8:23:06 AM
Microsoft Corporation
c:\winnt\system32\rgenp.dll
msdtclog.dll 1999.9.3421.3 89.77 KB
(91,920 bytes) 8/20/2003 8:23:06 AM
Microsoft Corporation
c:\winnt\system32\msdtclog.dll
mtxclu.dll 2000.2.3471.1 51.27 KB
(52,496 bytes) 8/20/2003 3:57:03 PM
Microsoft Corporation
c:\winnt\system32\mtxclu.dll
msdtcprx.dll 2000.2.3471.1 665.77 KB
(681,744 bytes) 8/20/2003 3:56:59 PM
Microsoft Corporation
c:\winnt\system32\msdtcprx.dll
tyfaux.dll 2000.2.3471.1 374.27 KB
(383,248 bytes) 8/20/2003 3:57:11 PM
Microsoft Corporation
c:\winnt\system32\tyfaux.dll
msdtctm.dll 2000.2.3471.1 1.07 MB
(1,120,528 bytes) 8/20/2003 3:56:59 PM
Microsoft Corporation
c:\winnt\system32\msdtctm.dll

msdtc.exe 1999.9.3421.3 6.77 KB (6,928 bytes)
8/20/2003 8:23:06 AM
Microsoft Corporation
c:\winnt\system32\msdtc.exe
inetpp.dll 5.00.2195.2842 65.27 KB
(66,832 bytes) 8/20/2003 3:56:55 PM
Microsoft Corporation
c:\winnt\system32\inetpp.dll
win32spl.dll 5.00.2195.2780 92.27 KB
(94,480 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\win32spl.dll
usbmon.dll 5.00.2195.2780 11.27 KB
(11,536 bytes) 8/20/2003 3:57:12 PM
Microsoft Corporation
c:\winnt\system32\usbmon.dll
tcpmon.dll 5.00.2195.2780 40.77 KB
(41,744 bytes) 8/20/2003 3:57:11 PM
Microsoft Corporation
c:\winnt\system32\tcpmon.dll
p1mon.dll 5.00.2165.1 12.77 KB
(13,072 bytes) 11/30/1999 8:39:36 PM
Microsoft Corporation
c:\winnt\system32\p1mon.dll
cnbjmon.dll 5.00.2134.1 43.77 KB
(44,816 bytes) 11/30/1999 8:38:48 PM
Microsoft Corporation
c:\winnt\system32\cnbjmon.dll
localspl.dll 5.00.2195.2793 246.77 KB
(252,688 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\localspl.dll
spoolss.dll 5.00.2161.1 61.77 KB
(63,248 bytes) 8/20/2003 8:10:50 AM
Microsoft Corporation
c:\winnt\system32\spoolss.dll
spoolsv.exe 5.00.2161.1 43.77 KB
(44,816 bytes) 8/20/2003 8:10:49 AM
Microsoft Corporation
c:\winnt\system32\spoolsv.exe
rpcss.dll 5.00.2195.2815 231.27 KB (236,816 bytes)
8/20/2003 3:57:09 PM
Microsoft Corporation
c:\winnt\system32\rpcss.dll
svchost.exe 5.00.2134.1 7.77 KB (7,952
bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\svchost.exe
hissuba.dll 5.00.0984 9.77 KB (10,000 bytes)
12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\hissuba.dll
dsenh.dll 5.00.2195.2226 142.77 KB
(146,192 bytes) 8/20/2003 3:57:51 PM
Microsoft Corporation
c:\winnt\system32\dsenh.dll
oakley.dll 5.00.2195.2785 378.77 KB
(387,856 bytes) 8/20/2003 3:57:05 PM
Microsoft Corporation
c:\winnt\system32\oakley.dll
mfc42u.dll 6.00.8665.0 972.05 KB
(995,384 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\mfc42u.dll
polagent.dll 5.00.2183.1 108.27 KB
(110,864 bytes) 12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\polagent.dll
scccli.dll 5.00.2195.2780 105.27 KB
(107,792 bytes) 8/20/2003 3:57:09 PM
Microsoft Corporation
c:\winnt\system32\scccli.dll
atl.dll 3.00.8449 57.56 KB (58,938 bytes)
12/7/1999 5:00:00 PM
Microsoft Corporation
c:\winnt\system32\atl.dll

certcli.dll (133,904 bytes)	5,00,2195,2778 Microsoft Corporation	5,00,2195,2842 Microsoft Corporation	72.27 KB	wincore.dll (74,000 bytes)	8/20/2003 3:57:13 PM	5,00,2195,2842 Microsoft Corporation	72.27 KB	winsta.dll (37,648 bytes)	8/20/2003 3:57:13 PM	VERITAS Software Corp. c:\winnt\system32\dmserver.dll	36.77 KB
ntdsatq.dll (32,016 bytes)	5,00,2195,2878 c:\winnt\system32\certcli.dll	5,00,2168.1 Microsoft Corporation	7.27 KB (7,440 bytes)	rasadhlp.dll (17,680 bytes)	8/20/2003 3:57:05 PM	5,00,2195,2871 Microsoft Corporation	7.27 KB	imhsvc.dll (90,896 bytes)	12/7/1999 5:00:00 PM	Microsoft Corporation c:\winnt\system32\imhsvc.dll	9.77 KB (10,000 bytes)
ntdsapi.dll (141,072 bytes)	5,00,2195,2899 Microsoft Corporation	5,00,2160.1 Microsoft Corporation	18.77 KB	winmr.dll (19,216 bytes)	12/7/1999 5:00:00 PM	5,00,2195,2871 Microsoft Corporation	18.77 KB	dnssrvr.dll (90,896 bytes)	8/20/2003 3:56:52 PM	Microsoft Corporation c:\winnt\system32\dnssrvr.dll	88.77 KB
kdscsvc.dll (137.77 KB)	5,00,2195,2878 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	36.624 bytes)	rmr20.dll (17,680 bytes)	8/20/2003 3:57:08 PM	5,00,2195,2871 Microsoft Corporation	36.624 bytes)	api32.dll (126,224 bytes)	12/7/1999 5:00:00 PM	Microsoft Corporation c:\winnt\system32\api32.dll	123.27 KB
sfmapi.dll (39,696 bytes)	5,00,2195,2899 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	35.77 KB	msafd.dll (109,328 bytes)	8/20/2003 3:56:59 PM	5,00,2195,2871 Microsoft Corporation	35.77 KB	rasman.dll (56,080 bytes)	12/7/1999 5:00:00 PM	Microsoft Corporation c:\winnt\system32\api32.dll	54.77 KB
rasfm.dll (21,776 bytes)	5,00,2195,2671 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	62.77 KB	mswsock.dll (64,272 bytes)	8/20/2003 3:57:03 PM	5,00,2195,2871 Microsoft Corporation	62.77 KB	rasapi32.dll (194,320 bytes)	12/7/1999 5:00:00 PM	Microsoft Corporation c:\winnt\system32\api32.dll	189.77 KB
mpr.dll (54,544 bytes)	5,00,2195,2779 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	34.27 KB	msgsvc.dll (35,088 bytes)	12/7/1999 5:00:00 PM	5,00,2195,2871 Microsoft Corporation	34.27 KB	rtutils.dll (44,816 bytes)	12/7/1999 5:00:00 PM	Microsoft Corporation c:\winnt\system32\api32.dll	43.77 KB
rsabase.dll (131,344 bytes)	5,00,2195,2878 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	48.27 KB	browser.dll (49,424 bytes)	8/20/2003 3:56:48 PM	5,00,2195,2871 Microsoft Corporation	48.27 KB	adslpdc.dll (130,320 bytes)	8/20/2003 3:56:47 PM	Microsoft Corporation c:\winnt\system32\api32.dll	127.27 KB
schannel.dll (141,584 bytes)	5,00,2195,2902 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	17.77 KB	alrsvc.dll (18,192 bytes)	12/7/1999 5:00:00 PM	5,00,2195,2871 Microsoft Corporation	17.77 KB	activeds.dll (178,960 bytes)	8/20/2003 3:56:44 PM	Microsoft Corporation c:\winnt\system32\api32.dll	174.77 KB
netlogon.dll (366,352 bytes)	5,00,2195,2865 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	88.77 KB	trkwns.dll (90,896 bytes)	12/7/1999 5:00:00 PM	5,00,2195,2871 Microsoft Corporation	88.77 KB	mprapi.dll (81,168 bytes)	12/7/1999 5:00:00 PM	Microsoft Corporation c:\winnt\system32\api32.dll	79.27 KB
msv1_0.dll (114,448 bytes)	5,00,2195,2900 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	15.77 KB	seclogon.dll (16,144 bytes)	12/7/1999 5:00:00 PM	5,00,2195,2871 Microsoft Corporation	15.77 KB	iphlpapi.dll (69,392 bytes)	12/7/1999 5:00:00 PM	Microsoft Corporation c:\winnt\system32\api32.dll	67.77 KB
kerberos.dll (203,536 bytes)	5,00,2195,2913 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	111.77 KB	psbase.dll (114,448 bytes)	8/20/2003 3:57:07 PM	5,00,2195,2871 Microsoft Corporation	111.77 KB	icmp.dll (5,00,2195,2871)	8/20/2003 3:57:05 PM	Microsoft Corporation c:\winnt\system32\api32.dll	67.77 KB
msprvs.dll (42,496 bytes)	5,00,2195,2918 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	41.50 KB	cryptsp.dll (63,248 bytes)	12/7/1999 5:00:00 PM	5,00,2195,2871 Microsoft Corporation	41.50 KB	dhcpc.dll (90,896 bytes)	12/7/1999 5:00:00 PM	Microsoft Corporation c:\winnt\system32\api32.dll	88.77 KB
samsrv.dll (378,640 bytes)	5,00,2195,2918 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	369.77 KB	cryptsp.dll (63,248 bytes)	12/7/1999 5:00:00 PM	5,00,2195,2871 Microsoft Corporation	369.77 KB	eventlog.dll (44,816 bytes)	12/7/1999 5:00:00 PM	Microsoft Corporation c:\winnt\system32\api32.dll	43.77 KB
lsasrv.dll (504,592 bytes)	5,00,2195,2964 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	492.77 KB	cryptsp.dll (63,248 bytes)	12/7/1999 5:00:00 PM	5,00,2195,2871 Microsoft Corporation	492.77 KB	ntdsapi.dll (57,104 bytes)	8/20/2003 3:57:05 PM	Microsoft Corporation c:\winnt\system32\api32.dll	55.77 KB
lsass.exe (33,552 bytes)	5,00,2195,2964 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	79.27 KB	wkssvc.dll (97,552 bytes)	12/7/1999 5:00:00 PM	5,00,2195,2871 Microsoft Corporation	79.27 KB	scesrv.dll (231,696 bytes)	8/20/2003 3:57:09 PM	Microsoft Corporation c:\winnt\system32\api32.dll	226.27 KB
essent.dll (6,039,401 bytes)	5,00,2195,2964 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	16.77 KB	srvsrv.dll (81,168 bytes)	8/20/2003 3:56:52 PM	5,00,2195,2871 Microsoft Corporation	16.77 KB	umpphgr.dll (88,336 bytes)	12/7/1999 5:00:00 PM	Microsoft Corporation c:\winnt\system32\api32.dll	86.27 KB
ntdsapi.dll (92,432 bytes)	5,00,2195,2964 Microsoft Corporation	5,00,2195,2871 Microsoft Corporation	11.77 KB	cfmgmr32.dll (17,168 bytes)	12/7/1999 5:00:00 PM	5,00,2195,2871 Microsoft Corporation	11.77 KB	services.exe (88,848 bytes)	12/7/1999 5:00:00 PM	Microsoft Corporation c:\winnt\system32\api32.dll	86.77 KB
				dmserver.dll (12,048 bytes)	8/20/2003 3:56:51 PM	5,00,2195,2871 Microsoft Corporation					

c:\lbcacq.dll 2000.2.3471.1 8/20/2003 3:56:49 PM 496.77 KB
Microsoft Corporation
C:\winnt\system32\lbcacq.dll
oleaut32.dll 2.40.4517.612.27 KB (626,960 bytes)
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\oleaut32.dll
csccut.dll 5.00.2195.2959 228.27 KB (233,744 bytes)
Microsoft Corporation
8/20/2003 3:56:50 PM
C:\winnt\system32\csccut.dll
winspool.drv 5.00.2195.2780 109.77 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\winspool.drv
winscard.dll 5.00.2134.1 77.27 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\winscard.dll
wlnotify.dll 5.00.2195.2780 53.77 KB
Microsoft Corporation
8/20/2003 3:57:13 PM
C:\winnt\system32\wlnotify.dll
csccut1.dll 5.00.2195.2401 98.27 KB
Microsoft Corporation
8/20/2003 3:56:50 PM
C:\winnt\system32\csccut1.dll
1z32.dll 5.00.2134.1 9.77 KB (10,000 bytes)
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\1z32.dll
version.dll 12/7/1999 5:00:00 PM 13.77 KB
Microsoft Corporation
C:\winnt\system32\version.dll
rsaenh.dll 5.00.2195.2228 130.77 KB
Microsoft Corporation
8/20/2003 3:57:52 PM
C:\winnt\system32\rsaenh.dll
mscat32.dll 5.131.2134.1 7.77 KB (7,952 bytes)
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\mscat32.dll
ole32.dll 5.00.2195.2887 969.77 KB (993,040 bytes)
Microsoft Corporation
8/20/2003 3:57:07 PM
C:\winnt\system32\ole32.dll
imagehlp.dll 5.00.2195.2778 125.77 KB
Microsoft Corporation
5/4/2001 12:05:02 PM
C:\winnt\system32\imagehlp.dll
msasn1.dll 5.00.2134.1 51.27 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\msasn1.dll
crypt32.dll 5.131.2195.2833 451.27 KB
Microsoft Corporation
8/20/2003 3:56:50 PM
C:\winnt\system32\crypt32.dll
wintrust.dll 5.131.2195.2779 162.27 KB
Microsoft Corporation
8/20/2003 3:57:13 PM
C:\winnt\system32\wintrust.dll
setupapi.dll 5.00.2195.2663 555.77 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\setupapi.dll
wimm.dll 5.00.2134.1 184.77 KB (189,200 bytes)
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\wimm.dll
conctl32.dll 5.81.537.77 KB (550,672 bytes)
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\conctl32.dll
shlwapi.dll 5.00.3315.1000 282.77 KB
Microsoft Corporation
8/20/2003 3:57:10 PM
C:\winnt\system32\shlwapi.dll

Microsoft Corporation
C:\winnt\system32\shlwapi.dll
shell32.dll 5.00.3315.2902 2.25 MB
Microsoft Corporation
8/20/2003 3:57:10 PM
C:\winnt\system32\shell32.dll
msgina.dll 5.00.2195.2779 324.27 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\msgina.dll
wsoc32.dll 5.00.2195.2871 21.27 KB
Microsoft Corporation
8/20/2003 3:57:14 PM
C:\winnt\system32\wsoc32.dll
dsapi.dll 5.00.2195.2785 130.77 KB
Microsoft Corporation
8/20/2003 3:56:52 PM
C:\winnt\system32\dsapi.dll
wldap32.dll 5.00.2195.2797 125.27 KB
Microsoft Corporation
8/20/2003 3:57:13 PM
C:\winnt\system32\wldap32.dll
ws2help.dll 5.00.2134.1 17.77 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\ws2help.dll
ws2_32.dll 5.00.2195.2780 67.77 KB
Microsoft Corporation
8/20/2003 3:57:13 PM
C:\winnt\system32\ws2_32.dll
samlib.dll 5.00.2195.2780 49.77 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\samlib.dll
netrap.dll 5.00.2134.1 11.27 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\netrap.dll
netapi32.dll 5.00.2195.2808 303.77 KB
Microsoft Corporation
8/20/2003 3:57:04 PM
C:\winnt\system32\netapi32.dll
profmap.dll 5.00.2181.1 29.27 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\profmap.dll
secur32.dll 5.00.2195.2867 46.77 KB
Microsoft Corporation
8/20/2003 3:57:09 PM
C:\winnt\system32\secur32.dll
sfc.dll 5.00.2195.2896 92.11 KB (94,320 bytes)
Microsoft Corporation
8/20/2003 3:57:09 PM
C:\winnt\system32\sfc.dll
nddeapi.dll 5.00.2137.1 15.27 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\nddeapi.dll
userenv.dll 5.00.2195.2780 361.77 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\userenv.dll
user32.dll 5.00.2195.2821 392.77 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\user32.dll
gdi32.dll 5.00.2195.2778 228.77 KB (234,256 bytes)
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\gdi32.dll
rpcrt4.dll 5.00.2195.2832 437.27 KB
Microsoft Corporation
8/20/2003 3:57:08 PM
C:\winnt\system32\rpcrt4.dll

advapi32.dll 5.00.2195.2867 351.77 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\advapi32.dll
kernel32.dll 5.00.2195.2778 714.77 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\kernel32.dll
msvcrt.dll 6.10.8924.0 284.05 KB
Microsoft Corporation
5/4/2001 12:05:02 PM
C:\winnt\system32\msvcrt.dll
winlogon.exe 5.00.2195.2953 173.77 KB
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\winlogon.exe
sfccfiles.dll 5.00.2195.2969 948.27 KB
Microsoft Corporation
8/20/2003 3:57:09 PM
C:\winnt\system32\sfccfiles.dll
ntdll.dll 5.00.2195.2779 478.77 KB (490,256 bytes)
Microsoft Corporation
5/4/2001 12:05:02 PM
C:\winnt\system32\ntdll.dll
smss.exe 5.00.2195.2901 44.27 KB (45,328 bytes)
Microsoft Corporation
12/7/1999 5:00:00 PM
C:\winnt\system32\smss.exe

[Services]

Display Name	Service Name	Path	Start Mode
Alert	Alert	C:\winnt\system32\services.exe	Share Process
Application Management	AppMgmt	C:\winnt\system32\services.exe	Stopped
Computer Browser	Browser	C:\winnt\system32\services.exe	Share
Indexing Service	clsvc	C:\winnt\system32\clsvc.exe	Manual
ClipBook	Clipsrv	C:\winnt\system32\clipsrv.exe	Own Process
Distributed Link Tracking	dfs	C:\winnt\system32\dfs	Running
DHCP Client	Dhcp	C:\winnt\system32\dhcpcsvc.exe	Normal
Logical Disk Manager	dmadmin	C:\winnt\system32\dmadmin.exe	Share Process
Logical Disk Manager	dmserver	C:\winnt\system32\dmserver	Running
DNS Client	Dns	C:\winnt\system32\services.exe	Share
Event Log	Eventlog	C:\winnt\system32\services.exe	Share Process
COM+ Event System	EventSystem	C:\winnt\system32\services.exe	Running
	Manual	C:\winnt\system32\services.exe	Share Process

c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem
Fax Service Fax Stopped Manual Own
Process c:\winnt\system32\faxsvc.exe Normal
LocalSystem
IIS Admin Service IISADMIN Stopped Auto Share
Process c:\winnt\system32\inetrv\inetinfo.exe
Normal LocalSystem
InterSite Messaging Imserv Stopped Disabled Own
Process c:\winnt\system32\imserv.exe Normal
LocalSystem
Kerberos Key Distribution Center kdc
Stopped Disabled Share Process
c:\winnt\system32\kdc.exe Normal
Server Task Scheduler Running Auto Share
Process c:\winnt\system32\services.exe
Normal LocalSystem
Workstation Lanmanworkstation Running Auto
Share Process
c:\winnt\system32\services.exe
Normal LocalSystem
License Logging Service LicenseService
Running Auto Own Process
Process c:\winnt\system32\lssrv.exe Normal
LocalSystem
TCP/IP NetBIOS Helper Service Lmhosts Running Auto
Share Process
c:\winnt\system32\services.exe
Normal LocalSystem
Messenger Messenger Running Auto Share Process
c:\winnt\system32\services.exe
Normal LocalSystem
NetMeeting Remote Desktop Sharing mmsrvc
Stopped Manual Own Process
Process c:\winnt\system32\mmsrvc.exe Normal
LocalSystem
Distributed Transaction Coordinator MSDTC
Running Auto Own Process
Process c:\winnt\system32\msdtc.exe Normal
LocalSystem
Windows Installer MSIServer Stopped Manual Share
Process c:\winnt\system32\instexec.exe /v
Normal LocalSystem
Message Queuing MSMQ Running Auto Own
Process c:\winnt\system32\mqsvc.exe Normal
LocalSystem
Network DDE \winnt\system32\netdde.exe Stopped Manual Share
Process LocalSystem
Network DDE DSDM NetDDEdsdm Stopped Manual
Process c:\winnt\system32\netdde.exe Normal
LocalSystem
Net Logon Netlogon Stopped Manual Share Process
Process c:\winnt\system32\lssass.exe Normal
LocalSystem
Network Connections Netman Running Manual Share
Process c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem
Symantec Ghost Database Service ngdbserv
Running Manual Own Process
c:\program files\symantec\ghost\ngdbserv.exe
Files\symantec\ghost\bin\ngdbserv.exe Normal
Symantec Ghost System Configuration Server NGServer
Running Auto Own Process
c:\program files\symantec\ghost\ngserver.exe
Normal LocalSystem
File Replication NTFrs Stopped Manual Own
Process c:\winnt\system32\ntfrs.exe Ignore
LocalSystem

NT LM Security Support Provider NLMssp
Running Manual Share Process
c:\winnt\system32\lssass.exe Normal
LocalSystem
Removable Storage Ntmsvc Running Auto Share
Process c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem
Plug and Play PlugPlay Running Auto Share
Process c:\winnt\system32\services.exe
Normal LocalSystem
IPSEC Policy Agent PolicyAgent Running Auto
Share Process
c:\winnt\system32\lssass.exe Normal
LocalSystem
Protected Shared ProtectedStorage Running Auto
Share Process
c:\winnt\system32\services.exe
Normal LocalSystem
Remote Access Auto Connection Manager RasAuto
Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem
Remote Access Connection Manager RasMan
Running Manual Share Process
Process c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem
Routing and Remote Access RemoteAccess
Stopped Disabled Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem
Remote Registry Service RemoteRegistry
Running Auto Own Process
Process c:\winnt\system32\regsvc.exe Normal
LocalSystem
Remote Procedure Call (RPC) Locator RpcLocator
Stopped Manual Own Process
Process c:\winnt\system32\locator.exe Normal
LocalSystem
Remote Procedure Call (RPC) Rpcss Running Auto
Share Process c:\winnt\system32\svchost
-k rpcss Normal LocalSystem
QOS RSVP RSVP Running Manual Own Process
Process c:\winnt\system32\rsyp.exe -s Normal
LocalSystem
Security Accounts Manager SamSs Running Auto
Share Process
c:\winnt\system32\lssass.exe Normal
LocalSystem
Smart Card Helper System Scardrv Stopped Manual Share
Process c:\winnt\system32\scardrvr.exe
Normal LocalSystem
Smart Card Ignore Scardrv Stopped Manual Share
Process c:\winnt\system32\scardrvr.exe
Normal LocalSystem
Task Scheduler Scheduler Running Auto Share
Process c:\winnt\system32\ntask.exe Normal
LocalSystem
RunAS Service seclogon Running Auto Share
Process c:\winnt\system32\services.exe
Normal LocalSystem
System Event Notification SENS Running Auto
Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem
Internet Connection Sharing SharedAccess
Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem
Print Spooler Spooler Running Auto Own
Process c:\winnt\system32\spoolsv.exe Normal
LocalSystem

Performance Logs and Alerts SysmonLog Stopped
Manual Own Process
c:\winnt\system32\smlogsvc.exe
Normal LocalSystem
Telephony Tapisrv Running Manual Share Process
Process c:\winnt\system32\svchost.exe -k tapisrv
Normal LocalSystem
Terminal Services TermService Stopped
Disabled Own Process
c:\winnt\system32\termerv.exe Normal
LocalSystem
Telnet Tlntsvr Stopped Manual Own Process
Process c:\winnt\system32\tlntsvr.exe Normal
LocalSystem
Distributed Link Tracking Service TrkSvr
Stopped Manual Share Process
c:\winnt\system32\services.exe
Normal LocalSystem
Distributed Link Tracking Client TrkKws
Running Auto Share Process
c:\winnt\system32\services.exe
Normal LocalSystem
Uninterruptible Power Supply UPS Stopped
Manual Own Process
Process c:\winnt\system32\ups.exe Normal
LocalSystem
Utility Manager UtilMan Stopped Manual Own
Process c:\winnt\system32\utilman.exe Normal
LocalSystem
Windows Time W32Time Stopped Manual Share
Process c:\winnt\system32\services.exe
Normal LocalSystem
World Wide Web Publishing Service W3SVC
Stopped Auto Share Process
c:\winnt\system32\inetrv\inetinfo.exe
Normal LocalSystem
Windows Management Instrumentation WinMgmt
Running Auto Own Process
Process c:\winnt\system32\wbem\winmgmt.exe
Ignore LocalSystem
Windows Management Instrumentation Driver Extensions
Wmi Running Manual Share Process
Process c:\winnt\system32\services.exe
Normal LocalSystem

[Program Groups]

Group Name	Name	User Name
Accessories	Default user	user:Accessories
Accessories\Accessibility	Default	Default User
Accessories\Entertainment	Default	Default User
Accessories\System Tools	Default	Default User
Startup	Default user	Default User
Accessories	All users	Accessories
Accessories\Communications	All	Users
Accessories\Entertainment	All	Users
Accessories\System Tools	All	Users
Administrative Tools	All users	All Users:Administrative
Microsoft SQL Server	All users	All Users:Microsoft SQL
Startup	All users	Startup
Symantec Ghost	All users	Symantec Ghost
Users	All users	Users

[illegible]

rsabase.dll	5.0.2195.2228	128 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
rsaenh.dll	5.0.2195.2228	131 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
rsapi32.dll	<File Missing>	Not Available
Not Available	Not Available	Not Available
rsasig.dll	<File Missing>	Not Available
Not Available	Not Available	Not Available
Available	Not Available	Not Available
channel.dll	5.1.2195.0	138 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
shdoc401.dll	<File Missing>	Not Available
Not Available	Not Available	Not Available
shdocvw.dll	5.0.3315.2879	1078 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
shell32.dll	5.0.3315.2902	2304 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
shlwapi.dll	5.0.3315.1000	283 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
url.dll	5.0.2920.0	82 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
urlmon.dll	5.0.3315.1000	441 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
vbscript.dll	5.0.5907	428 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
webcheck.dll	5.0.3315.1000	252 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
win.com	5.0.2134.1	24 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
5:00:00 PM	12/7/1999	12/7/1999
Corporation	Microsoft	Microsoft
winetnet.dll	C:\WINNT\system32	Microsoft
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
winsock.dll	5.0.3315.1000	457 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
12/7/1999 5:00:00 PM	3 KB	
wintrust.dll	5.0.2195.2779	102 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
wsock.yxd	<File Missing>	Not Available
Available	Not Available	Not Available
wsock32.dll	5.0.2195.2871	21 KB
5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation	
wsock32n.dll	<File Missing>	Not Available
Not Available	Not Available	Not Available
[Connectivity]		
Item	Value	
ConnectionPreference	Never dial	
EnableHttp1.1	1	
ProxyHttp1.1	0	
LAN settings		
AutoConfigProxy	winetnet.dll	
AutoProxyDetectMode	Enabled	
AutoConfigURL		

Proxy Disabled
ProxyServer
ProxyOverride
[Cache]
[Following are sub-categories of this main category]
[Summary]

Item	Value
Page Refresh Type	Automatic
Temporary Internet Files Folder	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
Total Disk Space	17492 MB
Available Disk Space	12643 MB
Maximum Cache Size	546 MB
Available Cache Size	546 MB

[List of Objects]

Program File Status Codebase
No cached object information available

[Content]

[Following are sub-categories of this main category]
[Summary]

Item	Value
Content Advisor	Disabled

[Personal Certificates]

Issued To Issued By Validity Signature Algorithm
Administrator Administrator 8/20/2003 to 7/27/2103 sha1RSA

[Other People Certificates]

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

[Publishers]

Name
No publisher information available

[Security]

Zone	Security Level
Local intranet	Medium-low
Trusted sites	Low
Internet	Medium
Restricted sites	High

Benchcraft Profile

Profile: 4700w_4cli_4rte
File path: C:\Documents and Settings\Administrator\Desktop\4700w_4cli_4rte.pro
Version: 4

Number of Engines: 8

Name: DRIVER1
Description:
Directory: C:\LogFiles\rte01-01.log
Machine: RTE01
Parameter Set: PARAM2
Index: 0
Seed: 4126
Configured Users: 5870
Pipe Name: DRIVER1834887625
Connect Rate: 100
Start Rate: 100
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER2
Description:
Directory: C:\LogFiles\rte01-02.log
Machine: RTE01
Parameter Set: PARAM2
Index: 100000000
Seed: 4126
Configured Users: 5870
Pipe Name: DRIVER2834934843
Connect Rate: 100
Start Rate: 100
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER3
Description:
Directory: C:\LogFiles\rte02-01.log
Machine: RTE02
Parameter Set: PARAM2
Index: 200000000
Seed: 4126
Configured Users: 5870
Pipe Name: DRIVER3834952203
Connect Rate: 100
Start Rate: 100
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER4
Description:
Directory: C:\LogFiles\rte02-02.log
Machine: RTE02
Parameter Set: PARAM2
Index: 300000000
Seed: 4126
Configured Users: 5870
Pipe Name: DRIVER4834993390
Connect Rate: 100
Start Rate: 100
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER5
Description:

163

Directory: c:\LogFiles\rte03-01.log
Machine: RTE03
Parameter Set: PARAM2
Index: 400000000
Seed: 4126
Configured Users: 5870
Pipe Name: DRIVER5835019156
Connect Rate: 100
Start Rate: 100
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER6
Description:
Directory: c:\LogFiles\rte03-02.log
Machine: RTE03
Parameter Set: PARAM2
Index: 500000000
Seed: 4126
Configured Users: 5870
Pipe Name: DRIVER6835035234
Connect Rate: 100
Start Rate: 100
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER7
Description:
Directory: c:\LogFiles\rte04-01.log
Machine: RTE04
Parameter Set: PARAM2
Index: 600000000
Seed: 4126
Configured Users: 5870
Pipe Name: DRIVER7376328
Connect Rate: 100
Start Rate: 100
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER8
Description:
Directory: c:\LogFiles\rte04-02.log
Machine: RTE04
Parameter Set: PARAM2
Index: 700000000
Seed: 4126
Configured Users: 5910
Pipe Name: DRIVER8389250
Connect Rate: 100
Start Rate: 100
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Number of User groups: 8

Driver Engine: DRIVER1
ITS Server: cli01
SQL Server: sql
Database: tpcc

November, 2003

User: sa
Protocol: HTML
w_id Range: 1 - 587
w_id Min warehouse: 1
w_id Max warehouse: 4700
Scale: Normal
User Count: 5870
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: cli01
SQL Server: sql
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 588 - 1174
w_id Min warehouse: 1
w_id Max warehouse: 4700
Scale: Normal
User Count: 5870
District id: 1
Scale Down: No

Driver Engine: DRIVER3
IIS Server: cli02
SQL Server: sql
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1175 - 1761
w_id Min warehouse: 1
w_id Max warehouse: 4700
Scale: Normal
User Count: 5870
District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: cli02
SQL Server: sql
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1762 - 2348
w_id Min warehouse: 1
w_id Max warehouse: 4700
Scale: Normal
User Count: 5870
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: cli03
SQL Server: sql
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2349 - 2935
w_id Min warehouse: 1
w_id Max warehouse: 4700
Scale: Normal
User Count: 5870
District id: 1
Scale Down: No

Driver Engine: DRIVER6
IIS Server: cli03
SQL Server: sql
Database: tpcc
User: sa
Protocol: HTML

w_id Range: 2936 - 3522
w_id Min warehouse: 1
w_id Max warehouse: 4700
Scale: Normal
User Count: 5870
District id: 1
Scale Down: No

Driver Engine: DRIVER7
IIS Server: cli04
SQL Server: sql
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3523 - 4109
w_id Min warehouse: 1
w_id Max warehouse: 4700
Scale: Normal
User Count: 5870
District id: 1
Scale Down: No

Driver Engine: DRIVER8
IIS Server: cli04
SQL Server: sql
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4110 - 4700
w_id Min warehouse: 1
w_id Max warehouse: 4700
Scale: Normal
User Count: 5910
District id: 1
Scale Down: No

Number of Parameter Sets: 2

		~Default		Default Parameter Set		Txn		Think		Key	
RT	Time	RT	Menu	Delay	Fence	Delay	New Order	Weight	Time	Weight	Think
12.05	12.05	18.01	18.01	18.01	18.01	18.01	18.01	10.00	10.00	10.00	10.00
12.05	12.05	3.01	3.01	3.01	3.01	3.01	3.01	10.00	10.00	10.00	10.00
5.05	5.05	2.01	2.01	2.01	2.01	2.01	2.01	1.00	1.00	1.00	1.00
5.05	5.05	2.01	2.01	2.01	2.01	2.01	2.01	20.00	20.00	1.00	1.00
10.05	10.05	2.01	2.01	2.01	2.01	2.01	2.01	5.00	5.00	1.00	0.10

		PARAM2		Txn		Think		Key	
RT	Time	RT	Menu	Delay	Fence	Delay	New Order	Weight	Time
12.07	12.07	18.01	18.01	18.01	18.01	18.01	18.01	44.92	44.92
12.07	12.07	3.01	3.01	3.01	3.01	3.01	3.01	43.02	43.02
5.07	5.07	2.01	2.01	2.01	2.01	2.01	2.01	4.02	4.02
5.07	5.07	2.01	2.01	2.01	2.01	2.01	2.01	4.02	4.02
10.07	10.07	2.01	2.01	2.01	2.01	2.01	2.01	20.00	20.00

Appendix D – 60-Day Space

TPC-C 60 Day Space Requirements						
Warehouses	4700				TpmC	58,161.52
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	4700	512	48	28		588
District	47000	5296	64	268		5628
Customer	141000000	102545456	6399240	5,447,235	114391931	
History	141000000	8417912	32720		814,718	8450632
NewOrder	42300000	753680	1936	37,781		793397
Orders	141000000	4604088	2246000		660,411	6850088
OrderLine	1409999206	93999952	222480		9,083,899	94222432
Item	100000	9528	64	480		10072
Stock	470000000	150400000	318304	7,535,915		158254219
Total		360,736,424	9,220,856	13,021,706	10,559,028	382,978,986
MB						
Dynamic Space	104,514	Sum of Data for Order, Orderline and History				
Static Space	269,489	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Space - (Dynamic + Static Space)				
Daily Growth	20,693	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed				
60 Day Space MB	1,511,090					
60 Day Space GB	1,475.67	GB				
Log Size	244,880.00	MB				
KB Per New Order	4.75	KB				
8 hr log MB	129,520	MB				
8 hr log GB	126.4842	GB				
Space Usage	GB Needed	Disks Measured	GB Priced	Disk Size	Formatted Size	
60 Day Space DB	1,475.67	195	6,822	36GB	34.986	33.92
			0.00	9GB	8.473	
			0.00	4GB	3.999	
Total DB	1,475.67	195.00	6822.27	OK		
8-hr log + mirror	252.9684	14	489.80	36GB	34.986	
OS, Swap	3	1	8.473	9GB		
Total Storage	3,207.32	GB	7,320.55	GB		

Misc_fg	CS_fg
588	
5628	
0	114391931
9265350	
793397	
7510499	
94222432	
10072	
0	158254219
111,807,965	272,646,150
files=	5
size=	6,484,608
Total=	32,423,040
8K blocks	259,384,320
OK	OK
Total needed	111,807,965
	272,646,150

tpmC		58,161.52								
	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	8,417,912	32,720	9,259,703	35,992	841,791	3,272	845,063	0.0292	814,717.78	795.62
Order	4,604,088	2,246,000	5,064,497	2,470,600	460,409	224,600	685,009	0.0237	660,410.78	644.93
Order-Line	93,999,952	222,480	103,399,947	244,728	9,399,995	22,248	9,422,243	0.3254	9,083,899.37	8,871.00
									10,311.55	
	sum(*) Before		sum(*) After		Num New-					
d next o id	141,047,000		170,004,361		28,957,361					
	Before MB		After MB		Grow MB		KB/New-Order	8-Hr Growth MB	8-Hr Growth GB	
Log	1799.90		136143.91		134344.00		4.7507	129,519.84	126.48	
244,880.00		0.73501402	55.596172							
Database tpcc log used (%)										

4,864.7354	bytes
------------	-------